

Contraction and Revision over DL-Lite TBoxes

Zhiqiang Zhuang¹ Zhe Wang¹ Kewen Wang¹ Guilin Qi^{2,3}

¹ School of Information and Communication Technology, Griffith University, Australia

² School of Computer Science and Engineering, Southeast University, China

³ State Key Lab for Novel Software Technology, Nanjing University, Nanjing, China

Abstract

Two essential tasks in managing Description Logic (DL) ontologies are eliminating problematic axioms and incorporating newly formed axioms. Such elimination and incorporation are formalised as the operations of contraction and revision in belief change. In this paper, we deal with contraction and revision for the DL-Lite family through a model-theoretic approach. Standard DL semantics yields infinite numbers of models for DL-Lite TBoxes, thus it is not practical to develop algorithms for contraction and revision that involve DL models. The key to our approach is the introduction of an alternative semantics called type semantics which is more succinct than DL semantics. More importantly, with a finite signature, type semantics always yields finite number of models. We then define model-based contraction and revision for DL-Lite TBoxes under type semantics and provide representation theorems for them. Finally, the succinctness of type semantics allows us to develop tractable algorithms for both operations.

1 Introduction

Ontology, together with its underlying logical formalism, Description Logics (DLs) (Baader et al. 2003), is becoming a prominent knowledge sharing technique in e-Health, bioinformatics and the semantic web. Although DLs are not designed to represent evolving knowledge, the engineering and maintenance of ontologies are a dynamic process. Two essential tasks in managing DL ontologies are the elimination of problematic axioms and the incorporation of newly formed axioms. Such changes are formalised as the operations of contraction and revision in the area of belief change (Gärdenfors 1988).

The dominant approach in belief change is the so called AGM framework (Alchourrón, Gärdenfors, and Makinson 1985; Gärdenfors 1988) which assumes an underlying logic that includes propositional logic. The main strategies for studying contraction and revision are to articulate principles called *rationality postulates* capturing the intuitions behind rational contraction and revision and to specify change mechanisms called *construction methods* for the operations. It is commonly accepted that the AGM framework provides the best set of postulates and well motivated construction

methods. Over the years, many have attempted defining contraction and revision for DL ontologies by using techniques in belief change (Flouris, Plexousakis, and Antoniou 2004; Flouris et al. 2006; Qi and Du 2009; Wang, Wang, and Topor 2010; Ribeiro and Wassermann 2006; Qi et al. 2008; Ribeiro and Wassermann 2009).

In this paper, we will define contraction and revision functions over logically closed DL-Lite_{core} TBoxes. DL-Lite_{core} is the core language of the DL-Lite family (Calvanese et al. 2007) which underlies the OWL 2 QL profile of OWL 2 and gains its popularity through its efficient query answering. In defining such functions we will take a model-based approach similar to (Katsuno and Mendelzon 1992). Instead of DL models the functions are based on models of a newly defined semantics for DL-Lite_{core} called *type semantics*.

Type semantics closely resembles the semantics for propositional logic. Provided that type semantics is equivalent to DL semantics with respect to major DL-Lite_{core} TBox reasoning tasks, models of type semantics (i.e., type models) are more succinct than DL models. More importantly, given a finite signature, any DL-Lite_{core} TBox has a finite number of type models, whereas it usually has infinite many DL models. Hence it is easier to work with type models than with DL models in defining and implementing model-based contraction and revision functions.

We will provide representation theorems for our model-based contraction and revision functions which characterise properties of the functions by a set of rationality postulates. One difficult in proving such theorems is that DL revision has to deal with both inconsistency and incoherence whereas AGM revision only have to deal with inconsistency. As a first step for applying the operation in practice, we also provide tractable algorithms for the contraction and revision functions.

2 DL-Lite

DL-Lite_{core} is the core of the family of DL-Lite languages. It has the following syntax:

$$B \rightarrow A \mid \exists R \quad C \rightarrow B \mid \neg B \quad R \rightarrow P \mid P^-$$

where A denotes an *atomic concept*, P an *atomic role*, P^- the *inverse* of the atomic role P . B denotes a *basic concept*

which can be either an atomic concept or an unqualified existential quantification. C denotes a *general concept* which can be either an basic concept or its negation. We also include \perp denoting the empty set and \top denoting the whole domain. We use \mathcal{B} to represent the universal set of basic concepts and \mathcal{R} as the universal set of atomic roles and their inverses. For an inverse role $R = P^-$, we write R^- to represent P for the convenience of presentation. In this paper, we assume \mathcal{B} and \mathcal{R} are finite.

A DL-Lite_{core} knowledge base consists of a TBox and an ABox. A TBox is a finite set of *concept inclusion axioms* of the form $B \sqsubseteq C$, $B \sqsubseteq \perp$, and $\top \sqsubseteq C$. That is only basic concept or \top can appear on the left-hand side of a concept inclusion. An ABox is a finite set of *assertions* of the form $A(a)$ or $P(a, b)$.

The semantics of DL-Lite_{core} is given in terms of interpretations. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns to each atomic concept A a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, and to each atomic role P a binary relation $P^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$, and to each individual name a an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$. The interpretation function is extended to general concept and special symbols as follows: $\perp^{\mathcal{I}} = \emptyset$, $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $(P^-)^{\mathcal{I}} = \{(o_2, o_1) \mid (o_1, o_2) \in P^{\mathcal{I}}\}$, $(\exists R)^{\mathcal{I}} = \{o \mid \exists o'. (o, o') \in R^{\mathcal{I}}\}$, and $(\neg B)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}}$. An interpretation \mathcal{I} satisfies a concept inclusion $B \sqsubseteq C$ if $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$, a concept assertion $A(a)$ if $a^{\mathcal{I}} \in A^{\mathcal{I}}$, and a role assertion $P(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$. \mathcal{I} satisfies a TBox \mathcal{T} (or ABox \mathcal{A}) if \mathcal{I} satisfies each axiom in \mathcal{T} (resp., each assertion in \mathcal{A}). \mathcal{I} is a model of a TBox \mathcal{T} (a TBox axiom ϕ) if it satisfies \mathcal{T} (resp., ϕ). A TBox or an axiom is *consistent* if it has at least one model. A TBox \mathcal{T} logically implies an axiom ϕ , written $\mathcal{T} \models \phi$, if all models of \mathcal{T} are also models of ϕ . Two TBox axioms ϕ and ψ are logically equivalent, written $\phi \equiv \psi$, if they have identical set of models.

The closure of a TBox \mathcal{T} , denoted as $cl(\mathcal{T})$, is the set of all TBox axioms ϕ such that $\mathcal{T} \models \phi$. The closure of a DL-Lite TBox is finite. In fact, the size of the closure of a TBox \mathcal{T} is quadratic w.r.t. the size of \mathcal{T} (Pan and Thomas 2007). We use $\models \phi$ to denote that ϕ is a tautology such as $A \sqsubseteq A$. We use $\{\top \sqsubseteq \perp\}$ to denote the (unique) inconsistent TBox.

A basic concept B is *satisfiable* with respect to a TBox \mathcal{T} if there is a model \mathcal{I} of \mathcal{T} such that $B^{\mathcal{I}}$ is non-empty, and B is *unsatisfiable* if $B^{\mathcal{I}} = \emptyset$ for every model \mathcal{I} of \mathcal{T} . It is easy to see that B is unsatisfiable with respect to a TBox \mathcal{T} if and only if $B \sqsubseteq \perp \in cl(\mathcal{T})$. A TBox is *coherent* if all basic concepts are satisfiable and *incoherent* otherwise. Notice that, often in DL literatures, coherence comes with the absence of unsatisfiable atomic concepts. Since, in DL-Lite, unsatisfiable non-atomic concepts like $\exists R$ are also unexpected we use the stricter condition for coherence.

In the upcoming sections all TBoxes are assumed to be closed DL-Lite_{core} TBoxes and by DL-Lite we mean DL-Lite_{core}. We will denote TBox axioms by lower case Greek letters (ϕ, ψ, \dots).

3 Type Semantics

Qualified existential and universal quantifications are not permitted in DL-Lite, which makes DL-Lite more similar to

propositional logic than other DLs. In this section, we will take advantage of this similarity and propose an alternative semantics for DL-Lite that is similar to the semantics for propositional logic (i.e., propositional semantics). Clearly, this alternative semantics is more succinct than DL semantics. The succinctness is a significant advantage when DL-Lite TBoxes need to be represented model-theoretically and some computational tasks involve their models. Central to the semantics is the notion of *types* which is first mentioned in (Kontchakov, Wolter, and Zakharyashev 2008). A type is a possibly empty set of basic concepts and we denote the universal set of types as Ω .

If we consider basic concepts as propositional atoms, and concept inclusion $B \sqsubseteq C$ as propositional formula $\neg B \vee C$, then a type is nothing but a propositional interpretation represented by atoms interpreted as true for the formulas. Given a TBox \mathcal{T} , we use $\|\mathcal{T}\|$ to denote the set of propositional models of the corresponding propositional formulas of \mathcal{T} . Many inferences between DL-Lite axioms are propositional in the sense that the inferences also hold when we consider the axioms as propositional formulas, for example the inference from $A \sqsubseteq B$ and $B \sqsubseteq C$ to $A \sqsubseteq C$. There is also a group of inferences that are not propositional, for instance the inference from $\exists R \sqsubseteq \perp$ to $\exists R^- \sqsubseteq \perp$. By giving special treatment for axioms appear in the non-propositional inferences, we define a model under type semantics as follows.

Definition 1. A type model τ of a TBox \mathcal{T} is a type such that $\tau \in \|\mathcal{T}\|$ and if $\tau \models \exists R \sqsubseteq \perp$ then $\exists R^- \notin \tau$.

Firstly, a type model has to be a propositional model for properly handling the propositional inferences. Then the extra condition guarantees the proper handling of the non-propositional inferences. We denote the type models of a TBox \mathcal{T} and an axiom ϕ as $|\mathcal{T}|$ and $|\phi|$ respectively. Type models of the negation of ϕ , denoted by $\neg\phi$, is defined as $\Omega \setminus |\phi|$. The following lemma best captures the non-propositional behaviour of type semantics.

Lemma 1. Let \mathcal{T} be a TBox. Then there is $\tau \in |\mathcal{T}|$ such that $\exists R \in \tau$ iff there is $\tau' \in |\mathcal{T}|$ such that $\exists R^- \in \tau'$.

It says for any TBox \mathcal{T} , there is a type model of \mathcal{T} that contains $\exists R$ if and only if there is one that contains $\exists R^-$.

Under type semantics, the models of stronger axioms are a subset or equal to the models of weaker axioms, as usual. Also a TBox is consistent if and only if it has a type model.

Theorem 1. Let \mathcal{T} be a TBox and ϕ a TBox axiom. Then

1. $\mathcal{T} \models \phi$ iff $|\mathcal{T}| \subseteq |\phi|$.
2. \mathcal{T} is consistent iff $|\mathcal{T}| \neq \emptyset$.

In comparison with DL semantics, type semantics has the clear advantage of being more succinct. While TBox axioms usually have infinite numbers of DL models they have at most 2^N type models, for N the number of basic concepts. Therefore, it is possible to develop algorithms that work directly with type models.

A TBox \mathcal{T} is coherent if and only if it is consistent and $\mathcal{T} \not\models B \sqsubseteq \perp$ for all $B \in \mathcal{B}$. For convenience, we extend the notion of coherence to single TBox axioms and sets of types. An axiom ϕ (a set of types M) is coherent if and only if $\{\phi\} \not\models B \sqsubseteq \perp$ (resp. $M \not\subseteq |B \sqsubseteq \perp|$) for all $B \in \mathcal{B}$.

Most DLs have the inexpressibility problem that some sets of DL models have no syntactic representations. It is no exception for DL-Lite under type semantics. Given a set of types M there may not be a TBox \mathcal{T} whose set of type models is M . In such cases, a corresponding TBox for M is a TBox that has the minimal set of type models that includes M .

Definition 2. A corresponding TBox \mathcal{T} for a set of types M is a TBox such that $M \subseteq |\mathcal{T}|$ and there is no TBox \mathcal{T}' such that $M \subseteq |\mathcal{T}'| \subset |\mathcal{T}|$.

Let $\mathcal{B} = \{\exists R, \exists R^-, A\}$ and a set of types $M = \{\{A\}, \{\emptyset\}, \{\exists R\}\}$. Notice that there is a type in M that contains $\exists R$ but there is no one containing $\exists R^-$. By Lemma 1, any TBox whose set of type models includes M must also have a type model that contains $\exists R^-$. Under the current \mathcal{B} , there are four types containing $\exists R^-$ which are $\{\exists R^-\}$, $\{\exists R^-, A\}$, $\{\exists R^-, \exists R\}$, and $\{\exists R^-, A, \exists R\}$. Thus we have four corresponding TBoxes for M that are $\{A \sqsubseteq \neg \exists R, A \sqsubseteq \neg \exists R^-, \exists R \sqsubseteq \neg \exists R^-\}$, $\{A \sqsubseteq \neg \exists R, \exists R \sqsubseteq \neg \exists R^-, \exists R^- \sqsubseteq A\}$, $\{A \sqsubseteq \neg \exists R, \exists R^- \sqsubseteq \exists R, A \sqsubseteq \neg \exists R^-\}$, and $\{\exists R^- \sqsubseteq \exists R, \exists R^- \sqsubseteq A\}$. Although The corresponding TBoxes are not unique in general, it is straight forward to identify a condition from Lemma 1 that guarantees the uniqueness.

Theorem 2. Let M be a set of types. Then there is an unique corresponding TBox for M iff there is $\tau \in M$ such that $\exists R \in \tau$ implies there is $\tau' \in M$ such that $\exists R^- \in \tau'$, for all $R \in \mathcal{R}$.

Obviously, any coherent set of types satisfies the condition for uniqueness. In fact, the sets of types we will encounter in defining contraction and revision functions for DL-Lite_{core} are always coherent which means there is always a unique corresponding TBox. In the upcoming sections, for any coherent set of types M , we use $\mathcal{T}_{core}(M)$ to denote the unique TBox that corresponds to M .

4 Contraction

In this section, we define contraction functions for DL-Lite TBoxes. The approach is inspired by (Katsuno and Mendelzon 1992). Different from (Katsuno and Mendelzon 1992), we take a more generalised approach such that no explicit ordering over models is assumed. Further, instead of propositional models we work with type models. Since a meaningful TBox is required to be consistent and coherent, we only consider contractions for consistent and coherent TBoxes. Also we have to consider contractions by conjunctions of axioms as such contractions are needed when we find out two or more axioms in a TBox can not hold together but there is no further information on which ones of them do not hold. Let ϕ_1, \dots, ϕ_n be TBox axioms, their conjunction is denoted as $\phi_1 \wedge \dots \wedge \phi_n$. As expected we define $|\phi_1 \wedge \dots \wedge \phi_n| = |\{\phi_1, \dots, \phi_n\}|$.

We first adapt the AGM contraction postulates and some of their alternatives to DL-Lite.

$$(T\dot{-}1) T\dot{-}\phi = cl(T\dot{-}\phi)$$

$$(T\dot{-}2) T\dot{-}\phi \subseteq T$$

$$(T\dot{-}3) \text{ If } T \not\models \phi, \text{ then } T\dot{-}\phi = T$$

$$(T\dot{-}4) \text{ If } \not\models \phi, \text{ then } T\dot{-}\phi \not\models \phi$$

$$(T\dot{-}5) T \subseteq cl((T\dot{-}\phi) \cup \{\phi\})$$

$$(T\dot{-}6) \text{ If } \phi \equiv \psi, \text{ then } T\dot{-}\phi = T\dot{-}\psi$$

$$(T\dot{-}r) \text{ If } \psi \in T \setminus T\dot{-}\phi, \text{ then there is some } T' \text{ such that } T\dot{-}\phi \subseteq T' \subseteq T, T' \not\models \phi, \text{ and } T' \cup \{\psi\} \models \phi$$

$$(T\dot{-}de) \text{ If } \psi \in T \text{ and } |T\dot{-}\phi| \subseteq |\phi| \cup |\psi| \text{ then } \psi \in T\dot{-}\phi$$

($T\dot{-}1$)–($T\dot{-}6$) are adaptations of their AGM origins (i.e., ($K\dot{-}1$)–($K\dot{-}6$)) by considering a belief set as a logically closed TBox and formulas as TBox axioms or conjunctions of them. The principle of minimal change is paramount to all change operations (Gärdenfors 1988). ($K\dot{-}5$), often called Recovery, is the main postulate for formalising the principle for contraction. It requires the information loss during contraction to be minimal such that the original belief set can be recovered by expanding the contracting formula. Recovery has been criticised by many researchers among which Hansson (Hansson 1991) argued that it is an emerging property rather than a fundamental postulate for contraction. One evidence is that other than the contraction itself, its satisfaction relies also on properties of the underlying logic (Ribeiro et al. 2013). In particular most of the DLs including DL-Lite are incompatible with Recovery. Due to the controversy of Recovery, many have proposed alternative postulates. A well known one is Relevance (Hansson 1991) which can replace Recovery in characterising AGM contractions. ($T\dot{-}r$) is DL-Lite_{core} version of Relevance. As noticed in (Fermé, Krevneris, and Reis 2008), Recovery can also be replaced by the following postulate of *Disjunctive Elimination*:

$$\text{If } \psi \in K \text{ and } \phi \vee \psi \in K\dot{-}\phi \text{ then } \psi \in K\dot{-}\phi.$$

Disjunctive Elimination captures the principle of minimal change by stating the condition for retaining a formula during a contraction. That is if a formula is in the original belief set and its disjunction with the contracting formula is retained during the contraction then the formula is retained. Since disjunction of axioms is not permitted in DL-Lite, in adapting the postulate to DL-Lite we describe the disjunction in terms of their type models, thus the postulate ($T\dot{-}de$).

Next we give the intuition behind our contraction functions. Clearly, if the set of models of a TBox contains some counter models of an axiom ϕ (i.e, models of $\neg\phi$) then the TBox does not imply ϕ . Thus, to remove an axiom ϕ from a TBox \mathcal{T} we can first add some counter models of ϕ to those of \mathcal{T} to form an intermediate model set then obtain the corresponding TBox of the model set¹. Therefore a decision has to be made on which counter models to add.

The extralogical information required for making the decision could be provided by a domain expert of the ontology or through some rankings over the models². To study the

¹Since \mathcal{T} is coherent, the intermediate model set which includes models of \mathcal{T} is also coherent. Thus by Theorem 2 there is an unique corresponding TBox for the model set.

²Kalyanpur et al. (Kalyanpur et al. 2006) explored several strategies for ranking axioms in the context of debugging unsatisfiable concepts. Similar ideas could be used here.

theoretical properties we assume there is a *selection function* that plays the role of decision making. A limiting case is when the set of counter models is empty which means the contracting axiom is a tautology. As it is not possible to cease a TBox from implying a tautology, a convenient way is to do nothing and return the original TBox. In line with this intuition a selection function should return the empty set in such cases. Formally, γ is a selection function if and only if for any set of types M , $\gamma(M)$ is a non-empty subset of M unless M is empty. Essentially, the function picks from the set of counter models the “best” ones which are later added to the model set of the TBox for forming the contraction outcome.

A special case is when \mathcal{T} does not imply ϕ which means the model set of \mathcal{T} contains counter models of ϕ . Intuitively, if asked to remove an axiom that is not implied by the TBox then nothing has to be done and the original TBox should be returned as the outcome. In line with this intuition, a selection function is required to be *faithful* such that if the intersection of models of \mathcal{T} and those of $\neg\phi$ is not empty then the selection function picks the intersecting models and no others. Formally, a selection function γ is faithful with respect to a TBox \mathcal{T} if for any set of types M , if $|\mathcal{T} \cap M| \neq \emptyset$ then $\gamma(M) = |\mathcal{T} \cap M|$.

With the above intuitions, model-based contraction function is defined as follows:

Definition 3. A function $\dot{-}$ is a model-based contraction function for a TBox \mathcal{T} iff for all conjunctions of TBox axioms ϕ

$$\mathcal{T} \dot{-} \phi = \mathcal{T}_{core}(|\mathcal{T}| \cup \gamma(|\neg\phi|))$$

where γ is a faithful selection function for \mathcal{T} .

Regarding its behaviour, a model-based contraction functions can be characterised by $(T \dot{-} 1)$ – $(T \dot{-} 4)$, $(T \dot{-} de)$, and $(T \dot{-} 6)$.

Theorem 3. A function $\dot{-}$ is a model-based contraction function iff $\dot{-}$ satisfies $(T \dot{-} 1)$ – $(T \dot{-} 4)$, $(T \dot{-} de)$, and $(T \dot{-} 6)$.

Theorem 3 guarantees that a model-based contraction function satisfies $(T \dot{-} 1)$ – $(T \dot{-} 4)$, $(T \dot{-} de)$, and $(T \dot{-} 6)$ and all functions satisfying these postulates are model-based contraction functions.

In addition to the characterisation we provide a non-deterministic algorithm CONT for computing the contraction outcomes. CONT first checks if the contracting axiom is a tautology or not implied by \mathcal{T} (line 1) in which cases \mathcal{T} is returned (line 2). Otherwise it picks a counter model τ of ϕ (line 3) and check it against each axiom in \mathcal{T} (line 4). If an axiom is not satisfiable under τ (line 5) then the axiom is removed from \mathcal{T} (line 6). Finally, what ever is left of \mathcal{T} is returned (line 7).

It can be verified that, given a TBox \mathcal{T} and an axiom ϕ , CONT returns the outcome of the contraction of \mathcal{T} by ϕ where the contraction carried out through a model-based contraction function.

Proposition 1. Let $\dot{-}$ be a contraction function for a TBox \mathcal{T} such that $\mathcal{T} \dot{-} \phi = CONT(\mathcal{T}, \phi)$ then $\dot{-}$ is a model-based contraction function for \mathcal{T} .

Algorithm 1: CONT

Input: TBox \mathcal{T} and conjunction of TBox axioms ϕ

Output: TBox \mathcal{T}_ϕ^-

```

1 if  $\phi$  is a tautology or  $\mathcal{T} \not\models \phi$  then
2   return  $\mathcal{T}_\phi^- := \mathcal{T}$ ;
3 Let  $\tau \in |\neg\phi|$ ;
4 foreach  $\psi \in \mathcal{T}$  do
5   if  $\tau \notin |\psi|$  then
6      $\mathcal{T} := \mathcal{T} \setminus \{\psi\}$ ;
7 return  $\mathcal{T}_\phi^- := \mathcal{T}$ ;

```

Algorithm CONT runs in polynomial time (if we consider the cardinality of \mathcal{B} linear) with respect to the size of the TBox. In particular, checking if \mathcal{T} entails ϕ takes polynomial time (line 1), obtaining a type model of $\neg\phi$ (line 3) is linear, which can be achieved by simply constructing, e.g., a type containing A but not B for $\phi = A \sqsubseteq B$, and each satisfiability check (line 5) runs in linear time.

5 Revision

In this section, we define revision functions for DL-Lite TBoxes. As for contraction, we only consider revisions for consistent and coherent TBoxes. In the AGM framework, revision can be constructed directly as in (Katsuno and Mendelzon 1992) or indirectly through contraction via the *Levi identity* (Levi 1991). Formally, let $\dot{-}$ be a contraction function for a belief set K , a revision function $*$ for K can be defined as $K * \phi = Cn((K \dot{-} \neg\phi) \cup \{\phi\})$ for all formulas ϕ . Since the syntax of DL-Lite does not permit axiom negation the approach is not applicable for DL-Lite. We will define revision functions directly in a model-theoretic approach. As for contraction the approach which is inspired by (Katsuno and Mendelzon 1992) is based on type models.

We first clarify a fundamental difference between AGM revision and DL revision. AGM revision aims to incorporate a new formula to a belief set while resolving any inconsistency caused. DL revision goes beyond inconsistency resolving. In addition to consistency, meaningful DL TBoxes have to be coherent, thus DL revision has to resolve both the inconsistency and the incoherence caused in incorporating new axioms³. For this reason the revision mechanism for DL is more involved than the AGM one.

Since AGM revision deals with inconsistency, AGM revision postulates are formulated to capture the rationale behind the inconsistency resolving process. DL revision also deals with incoherence, thus the postulates for DL revision have to capture the rationale behind not only inconsistency but also incoherence resolving. By replacing conditions on consistency with coherence, AGM revision postulates are reformulated as follows for revision over DL-Lite TBoxes.

³In fact we can concentrate on incoherence resolving when ABox is not considered. By its definition, a coherent TBox must be consistent. Inconsistency resolving is thus part of incoherence resolving.

($T * 1$) $T * \phi = cl(T * \phi)$

($T * 2$) $\phi \in T * \phi$

($T * 3$) If ϕ is coherent then $T * \phi \subseteq cl(T \cup \{\phi\})$

($T * 4$) If $T \cup \{\phi\}$ is coherent then $cl(T \cup \{\phi\}) \subseteq T * \phi$

($T * 5$) If ϕ is coherent then $T * \phi$ is coherent

($T * 6$) If $\phi \equiv \psi$ then $T * \phi = T * \psi$

($T * f$) If ϕ is incoherent then $T * \phi = \{\top \sqsubseteq \perp\}$

($T * 1$)–($T * 6$) correspond to the six AGM revision postulates. The failure postulate ($T * f$) is dedicated to the limiting case when the revising axiom is incoherent. Since ($T * 2$) requires that the revising axiom is in the revised TBox, if the revising axiom is itself incoherent then the revised TBox must also be incoherent. ($T * f$) requires that in such cases we simply return the inconsistent TBox. Its AGM origin, which states if the revising formula is inconsistent then we return the inconsistent belief set, is deducible from other AGM postulates (i.e., ($K * 2$)) thus is not postulated explicitly.

Next we present the intuitions behind our revision function. If the model set of a TBox \mathcal{T} is the subset of that of an axiom ϕ then \mathcal{T} implies ϕ . Thus to incorporate an axiom ϕ to a TBox \mathcal{T} , we can pick some models of ϕ to form an intermediate model set then obtain its corresponding TBox. Therefore a decision has to be made on which models of ϕ to pick. As for contraction, a selection function is assumed.

Previously, for contraction, a selection function returns empty set if the input is empty which represents the limiting case when the contracting axiom is a tautology (thus its negation is inconsistent and has an empty model set). Now the limiting case is when the revising axiom is incoherent. Since in such cases there is no way to return a coherent TBox that implies the revising axiom, a convenient way is to return the inconsistent TBox. Formally, a function γ is a selection function if $\gamma(M)$ is a non-empty subset of M unless M is incoherent.

The faithfulness condition also has to be modified from the contraction case. A selection function γ is *faithful* with respect to a TBox \mathcal{T} if it satisfies

1. if M is coherent then $|\mathcal{T}| \cap M \subseteq \gamma(M)$, and
2. if $|\mathcal{T}| \cap M$ is coherent then $\gamma(M) = |\mathcal{T}| \cap M$.

In revising \mathcal{T} by ϕ , condition 1 deals with the case when models of \mathcal{T} overlaps with those of ϕ which means $\mathcal{T} \cup \{\phi\}$ is consistent. In line with the principle of minimal change, in this case, the selection function has to pick all the overlapping models to preserve as much as possible the original TBox axioms. Condition 2 deals with the case that not only the overlapping exists but also it is coherent. Since there is no incoherence to resolve, the revision boils down to a set union operation (i.e., $cl(\mathcal{T} \cup \{\phi\})$). The selection function therefore picks all the overlapping models and no others.

Central to the revision, the selection function has to guarantee the type models picked are coherent, thus the following condition. A selection function γ is *coherent preserving* if for all $B \in \mathcal{B}$ there is $\tau \in \gamma(M)$ such that $B \in \tau$.

With the above intuition, a *model-based revision* function is defined as follows.

Definition 4. A function $*$ is a *model-based revision function* for a TBox \mathcal{T} iff for all TBox axioms ϕ

$$\mathcal{T} * \phi = \mathcal{T}_{core}(\gamma(|\phi|))$$

where γ is a selection function that is *coherent preserving* and *faithful* with respect to \mathcal{T} .

Model-based revision functions can be characterised by ($T * 1$)–($T * 6$) and ($T * f$).

Theorem 4. A function $*$ is a *model-based revision function* iff $*$ satisfies ($T * 1$)–($T * 6$) and ($T * f$).

Algorithm 2: REVI

Input: TBox \mathcal{T} and TBox axiom ϕ

Output: TBox \mathcal{T}_ϕ^*

```

1 if  $\phi$  is  $B \sqsubseteq \perp$  for some  $B \in \mathcal{B}$  then
2   return  $\mathcal{T}_\phi^* := \{\top \sqsubseteq \perp\}$ ;
3 foreach  $B \in \mathcal{B}$  do
4   if  $\mathcal{T} \cup \{\phi\} \models B \sqsubseteq \perp$  then
5     Let  $\tau \in |\phi|$  such that  $B \in \tau$ ;
6     foreach  $\psi \in \mathcal{T}$  do
7       if  $\tau \not\models \psi$  then
8          $\mathcal{T} := \mathcal{T} \setminus \{\psi\}$ ;
9 return  $\mathcal{T}_\phi^* := cl(\mathcal{T} \cup \{\phi\})$ ;

```

As for contraction we also provide a non-deterministic algorithm REVI for computing the revision outcomes. REVI starts by checking whether ϕ is incoherent (line 1), and if so it returns the inconsistent TBox (line 2). Otherwise, it checks for each basic concept if it is unsatisfiable under the union of \mathcal{T} with ϕ (line 3–4). For each unsatisfiable concept B , it picks a model τ of ϕ satisfying B (line 5) and check it against each axiom in \mathcal{T} (line 6). If an axiom is not satisfiable under τ (line 7) then the axiom is removed from \mathcal{T} (line 8). Finally, the closure of the union of whatever are left of \mathcal{T} and ϕ is returned (line 9).

It can be verified that, given a TBox \mathcal{T} and an axiom ϕ , REVI returns the outcome of the revision of \mathcal{T} by ϕ where the revision is carried out through a model-based revision function.

Proposition 2. Let $*$ be a revision function for a TBox \mathcal{T} such that $\mathcal{T} * \phi = REVI(\mathcal{T}, \phi)$ then $*$ is a *model-based revision function* for \mathcal{T} .

Algorithm REVI runs in polynomial time (again considering the cardinality of \mathcal{B} linear) in the size of the TBox. In particular, concept satisfiability check (line 4) is in polynomial time for DL-Lite_{core}, and obtaining a type model of ϕ satisfying B (line 5) is linear, which can be achieved by simply constructing, e.g., a type containing A , B and C for $\phi = A \sqsubseteq C$, and each satisfiability check (line 7) runs in linear time.

A group of works usual referred to as ontology debugging (e.g., (Kalyanpur et al. 2006)) also deal with unsatisfiable concepts. The method they used are based on the notion of Minimal Unsatisfiability Preserving Sub-TBoxes

(MUPS). For each unsatisfiable concept C , the MUPS based method first computes all the MUPS for C , then it computes a minimal hitting set for the MUPS. The incoherence is then resolved by removing axioms in the minimal hitting set. REVI deals with the same problem in a more efficient way. Roughly speaking each type model formed in line 5 of REVI corresponds to a minimal hitting set for the MUPS, thus we can avoid the computations of the MUPS and their minimal hitting sets which is a significant saving in computational power.

6 Related Work

In managing changes over DL ontologies, many (Flouris, Plexousakis, and Antoniou 2004; Flouris et al. 2006; Qi and Du 2009; Wang, Wang, and Topor 2010; Ribeiro and Wassermann 2006; Qi et al. 2008; Ribeiro and Wassermann 2009) have taken the same strategy as ours by considering it as a belief change problem.

(Qi and Du 2009; Wang, Wang, and Topor 2010) defined specific revision operators however their postulates are not formulated appropriately to capture the rationales of incoherence resolving. Moreover, the revision operator in (Wang, Wang, and Topor 2010) cannot guarantee coherence in general. (Ribeiro and Wassermann 2006; Qi et al. 2008; Ribeiro and Wassermann 2009) studied contraction and revision over TBoxes and knowledge bases that are not necessarily closed. This means only the axioms explicitly presented in the TBox or knowledge base are considered. The implicit axioms which logically follow from the explicit ones but are not presented are discarded during the operation. Thus the logical contents are not maximally preserved. Axiom negation is not supported by most DLs but is required in defining some change operations. (Flouris et al. 2006) proposed several notions of negated axioms for DLs. They also explored the notions of inconsistent and incoherent TBoxes and emphasised the importance of resolving incoherence in addition to inconsistency.

In a more general setting, (Flouris, Plexousakis, and Antoniou 2004; Ribeiro et al. 2013) identified properties of a monotonic logic under which a contraction function can be defined that satisfies Recovery and Relevance respectively. By their results, it is possible to define contraction functions under DL-Lite that satisfy Relevance. One way to obtain such functions is by properly restricting the selection functions for our model-based contraction functions.

(Grau et al. 2012) studied operations that contract and revise at the same time. A constraint which states the set of axioms to be incorporated and those to be eliminated is first specified. Then the operation maps a knowledge base to another that satisfies the constraint. The operation reduces to a revision and contraction after making empty the eliminating set and the incorporating set respectively. However, they did not identify the postulates that characterise the contraction and revision.

(Giacomo et al. 2009; Calvanese et al. 2010; Kharlamov and Zheleznyakov 2011; Kharlamov, Zheleznyakov, and Calvanese 2013) also dealt with changes over DL-Lite ontologies. Instead of considering it as a belief change prob-

lem, they focus on issues with expressibility of the outcomes for model-based change operations.

7 Conclusion

Due to the diversity of DLs, it is difficult if not impossible to come up with generalised contraction and revision functions that work for all DLs. Each DL is unique that they deserve to be treated individually to make the most out of their uniqueness. A distinguishing feature of DL-Lite is its close resemblance to propositional logic. By taking advantage of this feature, we developed type semantics for DL-Lite_{core} that resembles the underlying semantics for propositional logic. Due to the succinctness and finiteness of type semantics it is easier to work with type models than DL models. We defined and implemented contraction and revision functions for DL-Lite_{core} TBoxes whose outcomes are obtained by manipulating type models of the TBoxes and the contracting and revising axioms. The functions are shown to be sound and complete to sets of commonly accepted postulates. Crucial in obtaining the soundness and completeness result for the revision function is to reformulate AGM revision postulates from inconsistency centred to incoherence centred. As DL revision deals not only with inconsistency but also incoherence, unlike postulates for AGM revision, postulates for DL revision must capture the intuitions of incoherence resolving.

For future work, we plan to study contraction and revision for more expressive DLs. Some preliminary results are obtained for DL-Lite_R, an extension of DL-Lite_{core} with role inclusion axioms. The definition of a type in DL-Lite_R involves not only concepts but also roles. A more challenging task is to extend our results to DLs allowing quantified existential or universal quantifiers. Since concepts of infinite length can be formed in these DLs through unbound nesting of quantifiers, their semantic characterisation through type semantics may not be possible. We need some other techniques that are tailored to these DLs.

Acknowledgments

This work was partially supported by Australian Research Council (ARC) under DP130102302 and DP1093652. Guilin Qi was partially supported by the NSFC grant 61272378.

References

- Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic* 50(2):510–530.
- Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P., eds. 2003. *The Description Logic Handbook*. Cambridge, UK: CUP.
- Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *Journal of Automatic Reasoning* 39(3):385–429.
- Calvanese, D.; Kharlamov, E.; Nutt, W.; and Zheleznyakov, D. 2010. Evolution of dl-lite knowledge bases. In *Proceedings of the 9th International Semantic Web Conference on The Semantic Web (ISWC-2010)*, ISWC’10, 112–128.
- Fermé, E.; Krevneris, M.; and Reis, M. 2008. An axiomatic characterization of enscocement-based contraction. *Journal of Logic and Computation* 18(5):739–753.
- Flouris, G.; Huang, Z.; Pan, J. Z.; Plexousakis, D.; and Wache, H. 2006. Inconsistencies, negations and changes in ontologies. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-2006)*.
- Flouris, G.; Plexousakis, D.; and Antoniou, G. 2004. Generalizing the AGM postulates: preliminary results and applications. In *Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (NMR-2004)*, 171–179.
- Gärdenfors, P. 1988. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. MIT Press.
- Giacomo, G. D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2009. On instance-level update and erasure in description logic ontologies. *Journal of Logic Computation* 19(5):745–770.
- Grau, B. C.; Ruiz, E. J.; Kharlamov, E.; and Zhelenyakov, D. 2012. Ontology evolution under semantic constraints. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR-2012)*, 137–147.
- Hansson, S. O. 1991. Belief Contraction Without Recovery. *Studia Logica* 50(2):251–260.
- Kalyanpur, A.; Parsia, B.; Sirin, E.; and Cuenca-Grau, B. 2006. Repairing unsatisfiable concepts in owl ontologies. In *Proceedings of the 3rd European Conference on The Semantic Web: Research and Applications*, 170–184.
- Katsuno, H., and Mendelzon, A. O. 1992. Propositional knowledge base revision and minimal change. *Artificial Intelligence* 52(3):263–294.
- Kharlamov, E., and Zheleznyakov, D. 2011. Capturing instance level ontology evolution for dl-lite. In *Proceedings of the 10th International Conference on The Semantic Web (ISWC-2011)*, 321–337.
- Kharlamov, E.; Zheleznyakov, D.; and Calvanese, D. 2013. Capturing model-based ontology evolution at the instance level: The case of dl-lite. *Journal of Computer and System Sciences* 79(6):835–872.
- Kontchakov, R.; Wolter, F.; and Zakharyashev, M. 2008. Can you tell the difference between dl-lite ontologies. In *Proc. KR-2008*, 285–295.
- Levi, I. 1991. *The Fixation of Beliefs and its Undoing*. Cambridge University Press.
- Pan, J. Z., and Thomas, E. 2007. Approximating owl-dl ontologies. In *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-2007)*, 1434–1439.
- Qi, G., and Du, J. 2009. Model-based revision operators for terminologies in description logics. In *Proc. IJCAI-2009*, 891–897.
- Qi, G.; Haase, P.; Huang, Z.; Ji, Q.; Pan, J. Z.; and Vlker, J. 2008. A kernel revision operator for terminologies algorithms and evaluation. In *Proceedings of the 7th International Semantic Web Conference, (ISWC-2008)*, 419–434.
- Ribeiro, M. M., and Wassermann, R. 2006. First step towards revising ontologies. In *Proc. WONTO-2006*.
- Ribeiro, M. M., and Wassermann, R. 2009. Base revision for ontology debugging. *Journal of Logic Computation* 19(5):721–743.
- Ribeiro, M. M.; Wassermann, R.; Flouris, G.; and Antoniou, G. 2013. Minimal change: Relevance and recovery revisited. *Artificial Intelligence* 201:59–80.
- Wang, Z.; Wang, K.; and Topor, R. W. 2010. A new approach to knowledge base revision in dl-lite. In *Proc. AAAI-2010*.