Belief Contraction for the Description Logic $\mathcal{EL}$

Zhi Qiang Zhuang and Maurice Pagnucco

ARC Centre of Excellence for Autonomous Systems and National ICT Australia,
School of Computer Science and Engineering, The University of New South Wales,
Sydney, NSW 2052, Australia.
{zqzhuang|morri}@cse.unsw.edu.au

Abstract. In this paper, we propose a service for contracting axioms from ontologies based on the description logic $\mathcal{EL}$. In particular, we devise algorithms that implement the contraction operation of the AGM theory on belief change. We first generalise the AGM contraction postulates to description logics, in which a variant of the recovery postulate is defined by using the notion of logical difference. Plausibility of the contraction operation is demonstrated by showing that it satisfies all the generalised postulates.

1 Introduction

The area of *belief change* in knowledge representation is concerned with how an agent ought to alter its corpus of beliefs in the face of new information. The best known approach in this area is the so-called AGM [1] framework which focuses on two types of belief change:[1] *revision*, in which an agent keeps its beliefs consistent while incorporating new information into it, and *contraction*, in which an agent gives up some of its beliefs in order to avoid drawing unwanted consequences. This paper addresses the contraction operation in the lightweight description logic (DL) $\mathcal{EL}$ which has polynomial time reasoning procedures [2].

The motivation for focusing on this topic is due to its application in repairing DL based ontologies such as the bio-medical ontology SNOMED [3]. The management of DL based ontologies is a complicated process. It is very likely that counterintuitive axioms are accidentally introduced into the ontology. An example of this in SNOMED [3] that has often been pointed out occurs with the concept `AmputationOfFinger` which is classified as being subsumed by the concept `AmputationOfArm`. Repairing the above problem can be seen as performing contraction by the subsumption $\texttt{AmputationOfFinger} \sqsubseteq \texttt{AmputationOfArm}$.

Several attempts [4–8] have been made at developing accounts of belief change for DLs. However, belief change in DLs has proved to be more complicated than the classical case [4], particularly for accounts attempting to adhere to the popular AGM framework [1]. The problem lies in the limited expressiveness of DLs compared to propositional logic. As shown in Flouris *et al.* [4], for some important DLs such as $\mathcal{SHIN}$, it is not possible to define contraction operators which

[1] The AGM also introduces the *expansion* operation but it can be considered a special case of revision.

satisfy the *recovery* postulate of the AGM framework. As this postulate specifies the minimal change property of the contraction operation, these operators can not be justified against this property. In this work we address the problem by generalising the AGM contraction postulates so that their satisfaction no longer relies on the expressiveness of the underlying logic. The main idea is to refine the recovery postulate by using the notion of *logical difference*.

Intuitively, candidate outputs of a contraction operator should at least be logically weaker than the original terminology and not imply the axiom being contracted. These intuitions are captured by the so-called *inclusion* ($K \dot{-} 2$) and *success* ($K \dot{-} 4$) postulates in the AGM framework. Furthermore, among the candidates for retention, the most desirable ones should have the least logical difference from the original terminology compared to the others. In Section 4, we show that for classical logic, contraction satisfies the recovery postulate if and only if it has the above property.

2 The Description Logic $\mathcal{EL}$

Let $\mathsf{N_C}$ and $\mathsf{N_R}$ be disjoint sets of *concept names* and *role names*, respectively. In the description logic $\mathcal{EL}$, *concepts* C are built according to the rule

$$C ::= \top | A | C \sqcap D | \exists R.C$$

where A ranges over $\mathsf{N_C}$, R ranges over $\mathsf{N_R}$, and C, D range over concepts. The semantics of $\mathcal{EL}$ is the standard set theoretic Tarskian semantics. An *interpretations* is a structure $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty set called the *domain*, and $\cdot^{\mathcal{I}}$ is an interpretation function mapping concept names A to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and role name R to binary relations $R^{\mathcal{I}}$ over $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The function $\cdot^{\mathcal{I}}$ is extended to arbitrary concepts as follows:

$$\begin{aligned} \top^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}} &:= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\ (\exists R.C)^{\mathcal{I}} &:= \{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} : (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \end{aligned}$$

Given $\mathcal{EL}$ concepts C, D, then $C \sqsubseteq D$ is a *general concept inclusion axiom* (GCI for short); $C \equiv D$ is an abbreviation for $C \sqsubseteq D$ and $D \sqsubseteq C$. An $\mathcal{EL}$ *terminology* (or TBox for short) is a finite set of GCIs. An interpretation $\mathcal{I}$ *satisfies* an axiom $C \sqsubseteq D$ ($\mathcal{I} \models C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; $\mathcal{I} \models C \equiv D$ iff $C^{\mathcal{I}} = D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of a TBox T ($\mathcal{I} \models T$) if it satisfies every axiom in T. An axiom $C \sqsubseteq D$ is a *consequence* of a TBox T ($T \models C \sqsubseteq D$) iff every model of T satisfies $C \sqsubseteq D$. Given a Tarskian consequence operator Cn, and a TBox T, $Cn(T) = \{C \sqsubseteq D \mid T \models C \sqsubseteq D\}$.

In this paper we assume that a TBox contains only axioms of the form $A \equiv C$ and $A \sqsubseteq C$, where A is a concept name and no concept name occurs more than once on the left-hand side of an axiom. A concept name is *primitive* if it does not occur on the left-hand side of an axiom and is *pseudo-primitive* if it is primitive or occurs on the left-hand side of an axiom of the form $A \sqsubseteq C$. Under these assumptions a useful property regarding $\mathcal{EL}$ TBoxes (Lemma 1) is derived in [9]. In the rest of the paper, unless explicitly stated, all TBoxes are based on $\mathcal{EL}$.

Lemma 1. *[9] Let T be a TBox and $C = F \sqcap \bigsqcap_{(R.D)\in Q} \exists R.D$, where F is a conjunction of concept names and Q is a set of pairs (R, D) in which R is a role and D a concept.*

1. *If $T \models C \sqsubseteq A$ for an A which is pseudo-primitive in T, then $T \models B \sqsubseteq A$, for some conjunct B of F.*
2. *If $T \models C \sqsubseteq \exists S.C_0$, then either*
   - *$T \models B \sqsubseteq \exists S.C_0$, for some conjunct B of F, or*
   - *there exist $(R, D) \in Q$ such that $R = S$ and $T \models D \sqsubseteq C_0$.*

Lemma 1 provides a description of the syntactic form of concepts C such that $T \models C \sqsubseteq A$, where A is pseudo-primitive or of the form $\exists R.B$, from which we can obtain justifications of T entailing $C \sqsubseteq A$. Suppose $T \models C \sqsubseteq A$ for A pseudo-primitive, according to Part 1 of the lemma, if C is of the form $B \sqcap \exists R.C$ then it must be the case $T \models B \sqsubseteq A$ (which implies $B \sqcap \exists R.C \sqsubseteq A$ in T). Similarly if $T \models \exists S.B \sqcap \exists R.D \sqsubseteq \exists R.C$ then according to Part 2 of the lemma, it must be the case $T \models D \sqsubseteq C$ (which implies $\exists S.B \sqcap \exists R.D \sqsubseteq \exists R.C$ in T).

3 Logical Difference Between Terminologies

The notion of *logical difference* between DL terminologies was proposed by Konev *et al.* [9]. Intuitively, TBox T is logically different from T' if there are axioms that T entails but T' does not. In [9] logical difference is defined according to a *signature* Σ which is a subset of $\mathsf{N_C} \cup \mathsf{N_R}$. Only axioms formed by using concepts and roles in Σ are considered when computing the logical difference. However, we are only interested in the logical difference between a terminology and the one resulting from contraction. Since the two are supposed to have the same signature, we use the following definition without referring to signatures.

Definition 1. *Let T and T' be TBoxes, $\mathsf{Diff}(T, T')$ their logical difference, then $C \sqsubseteq D \in \mathsf{Diff}(T, T')$ iff*

1. *$T \models C \sqsubseteq D$*
2. *$T' \not\models C \sqsubseteq D$*
3. *if $T' \models A \sqsubseteq B$ and there is no $A' \sqsubseteq B'$ s.t. $T' \models A' \sqsubseteq B'$ and $\{A' \sqsubseteq B'\} \models A \sqsubseteq B$ then $\{C \sqsubseteq D\} \not\models A \sqsubseteq B$*

For A, A', B, B', C and D concepts.

Our definition of logical difference differs from Konev's (which does not enforce condition 3) in the sense that ours is more strict. It turns out that our definition is more appropriate for the purpose of formalising properties of rational belief change. For example, let $T = \{A \sqsubseteq B \sqcap C\}$ and $T' = \{A \sqsubseteq C\}$ be TBoxes, under Konev's definition $\mathsf{Diff}(T, T') = \{A \sqsubseteq B \sqcap C, A \sqsubseteq B\}$ whereas under ours $\mathsf{Diff}(T, T') = \{A \sqsubseteq B\}$. We will use the notion of logical difference to specify how minimal are the changes that have been made to a TBox to accomplish a contraction. In other words, we want to identify the TBoxes that differ the least from the original one. Let T, T' and T'' be TBoxes, it is natural to conclude

that $T^{'}$ differs less from T than $T^{''}$ does if $\mathsf{Diff}(T, T^{'})$ is logically weaker than $\mathsf{Diff}(T, T^{''})$, that is $\mathsf{Diff}(T, T^{''}) \models \mathsf{Diff}(T, T^{'})$. Continuing with the above example, let $T^{''}$ be empty, intuitively $T^{'}$ differs less from T then $T^{''}$ does. However, under Konev's definition $\mathsf{Diff}(T, T^{'}) \equiv \mathsf{Diff}(T, T^{''})$

Lemma 2. *[9] Let T and $T^{'}$ be TBoxes. If $C \sqsubseteq D \in \mathsf{Diff}(T, T^{'})$, then there exist subconcepts $C^{'}$ and $D^{'}$ of C and D, respectively, such that $C^{'} \sqsubseteq D^{'} \in \mathsf{Diff}(T, T^{'})$ and $C^{'} \sqsubseteq D^{'}$ is of the form $A \sqsubseteq \exists R.B$ or $B \sqsubseteq A$, where A is a concept name.*

Lemma 2 states that if $T^{'}$ does not entail $C \sqsubseteq D$ then it does not entail $A \sqsubseteq \exists R.B$ or $B \sqsubseteq A$. From the proof of Lemma 2 in [9], we can conclude that either $\{A \sqsubseteq \exists R.B\} \models C \sqsubseteq D$ or $\{B \sqsubseteq A\} \models C \sqsubseteq D$. Therefore contraction of $C \sqsubseteq D$ from T is achieved by first removing $C \sqsubseteq D$ from T (if it is present in T) followed by contraction of either $A \sqsubseteq \exists R.B$ or $B \sqsubseteq A$. Although the lemma is obtained under Konev's definition of logical difference, our definition is more strict which means the result can be carried over.

4 Generalising AGM Belief Contraction

A common approach in addressing belief change is to provide a set of *rationality postulates* for belief change operations. The AGM approach provides the best known set of postulates. The aim is to describe belief change at the *knowledge level* without referring to any representational formalism. In the AGM approach belief states are modelled by *belief sets* closed under the Tarskian consequence operator(Cn) for a logic including propositional logic in a language $\mathcal{L}$. The *expansion* of a belief set K by φ, $K + \varphi$, is defined as $Cn(K \cup \{\varphi\})$. *Contraction* represents situations in which an agent has to give up some information from its current beliefs. Formally, a contraction operation is a function from $2^{\mathcal{L}} \times \mathcal{L}$ to $2^{\mathcal{L}}$ satisfying the following set of basic postulates. The contraction of a belief set by a sentence yields a new set, called the contracted belief set.

($K \dot{-} 1$) $K \dot{-} \varphi = Cn(K \dot{-} \varphi)$. ($Closure$)
($K \dot{-} 2$) $K \dot{-} \varphi \subseteq K$ ($Inclusion$)
($K \dot{-} 3$) If $\varphi \notin K$, then $K \dot{-} \varphi = K$. ($Vacuity$)
($K \dot{-} 4$) If $\not\models \varphi$, then $\varphi \notin K \dot{-} A$ ($Success$)
($K \dot{-} 5$) If $\models \varphi \equiv \psi$, then $K \dot{-} \varphi \equiv K \dot{-} \psi$ ($Extensionality$)
($K \dot{-} 6$) $K \subseteq (K \dot{-} \varphi) + \varphi$. ($Recovery$)

Note that AGM presupposes full propositional logic which contains connectives such as negation and disjunction that are not fully supported in DLs. For instance, negations ($\neg(A \sqsubseteq B)$) and disjunctions ($(A \sqsubseteq B) \vee (C \sqsubseteq D)$) of DL axioms are not expressible as DL axioms. As a result, the recovery postulate ($K \dot{-} 6$) cannot be satisfied by contraction operators defined in most DLs [10]. Since recovery captures the minimal change property of belief contraction, which is essential for any rational contraction operator. Our goal is to postulate the property in a way that its satisfaction no longer relies on the expressive power of the underlying logic and to replace recovery with the new postulate.

To specify the minimal difference between the original belief set and the contracted one, we need a notion of difference between belief sets and a way of comparing them. A contraction that always picks the belief set that has the smallest (or weakest) difference from the original belief set as the contracted one, will satisfy the minimal change property. In this paper, we use logical difference as the measure of difference. In Section 3 we defined the notion of logical difference between TBoxes in DLs. This can be easily extended to propositional logic by considering the TBoxes T and $T^{'}$ as belief sets and the GCI $C \sqsubseteq D, A \sqsubseteq B$ as sentences in propositional logic.

The alternative postulate should be as close as possible to Recovery, in the sense that when propositional logic is assumed it coincides with recovery. The following is an equivalent formulation of the Recovery postulate in terms of logical difference.

Lemma 3. *Let K be a belief set, φ a sentence, $\dot{-}$ a contraction operator satisfisfying $(K\dot{-}1) - (K\dot{-}5)$ then $\dot{-}$ satisfies $(K\dot{-}6)$ iff*

$$(K\dot{-}\varphi) \cup \{\varphi\} \models \mathsf{Diff}(K, K\dot{-}\varphi) \qquad (*)$$

Proof.
($\Rightarrow$)
From $K \subseteq (K\dot{-}\varphi) + \varphi$, we have $(K\dot{-}\varphi) + \varphi \models K$. Due to definition of Diff, $K \models \mathsf{Diff}(K, K\dot{-}\varphi)$ which implies $(K\dot{-}\varphi) + \varphi \models \mathsf{Diff}(K, K\dot{-}\varphi)$, thus $(K\dot{-}\varphi) \cup \{\varphi\} \models \mathsf{Diff}(K, K\dot{-}\varphi)$
($\Leftarrow$)
We first prove $K^{'} \cup \mathsf{Diff}(K, K^{'}) \models K$. Assume $K \models \alpha$ and $K^{'} \cup \mathsf{Diff}(K, K^{'}) \not\models \alpha$, then it must be the case $K^{'} \not\models \alpha, \mathsf{Diff}(K, K^{'}) \not\models \alpha$. From the definition of Diff (condition 3), since $K \models \alpha$, $K^{'} \not\models \alpha$ and $\mathsf{Diff}(K, K^{'}) \not\models \alpha$ there exists a β s.t. $K^{'} \models \beta, \{\alpha\} \models \beta$. Since $K \models \alpha$ and $K = Cn(K)$ $(K\dot{-}1)$, $K \models \beta \rightarrow \alpha$. $K^{'} \not\models \beta \rightarrow \alpha$ follows from $K^{'} \not\models \alpha$ and $K^{'} \models \beta$. Now we prove $\mathsf{Diff}(K, K^{'}) \models \beta \rightarrow \alpha$, that is the three conditions in the definition of Diff must be satisfied by $\beta \rightarrow \alpha$. Condition 1 and 2 are already satisfied. Assume condition 3 is not satisfied then it must be the case that $K^{'} \models \psi \vee \beta \rightarrow \alpha$ for some sentence ψ, $(\beta \rightarrow \alpha \models \psi \vee \beta \rightarrow \alpha)$ which is not possible as $K^{'} \models \beta$ and $K^{'} \not\models \alpha$ $(\{\beta, \psi \vee \beta \rightarrow \alpha\} \models \alpha)$. As $\mathsf{Diff}(K, K^{'}) \models \beta \rightarrow \alpha$, $K^{'} \cup \mathsf{Diff}(K, K^{'}) \models \alpha$, contradiction. Finally, from $(K\dot{-}\varphi) \cup \{\varphi\} \models \mathsf{Diff}(K, K\dot{-}\varphi)$ it follows $(K\dot{-}\varphi) \cup \{\varphi\} \models K$. □

However, $(*)$ is not suitable for our purpose, as it still assumes certain expressive power of the underlying logic. As an example, take the TBox $T = Cn(A \sqsubseteq B)$ and let $\varphi = A \sqcap C \sqsubseteq B$. If $\dot{-}$ satisfies the Success postulate then $T\dot{-}\varphi = Cn(\emptyset)$, but $\{A \sqcap C \sqsubseteq B\} \not\models A \sqsubseteq B$. Note that, with a more expressive DL such as $\mathcal{ALC}$, we have $T\dot{-}\varphi = Cn(A \sqsubseteq B \sqcup C)$ then $(*)$ is satisfied. Taking all the above into consideration, we propose the following set of postulates, where $\dot{-}$ is the contraction operator, φ a GCI:

($T\dot{-}1$) $T\dot{-}\varphi = Cn(T\dot{-}\varphi)$
($T\dot{-}2$) $T\dot{-}\varphi \subseteq T$
($T\dot{-}3$) If $\varphi \notin T$, then $T\dot{-}\varphi = T$.
($T\dot{-}4$) If $\not\models \varphi$, then $\varphi \notin T\dot{-}\varphi$
($T\dot{-}5$) If $Cn(\varphi) = Cn(\phi)$, then $T\dot{-}\varphi = T\dot{-}\phi$
($T\dot{-}6$) There exists no $T^{'}$ such that $T \models T^{'}$ and $T^{'} \not\models \varphi$ and $\mathsf{Diff}(T, T\dot{-}\varphi) \models \mathsf{Diff}(T, T^{'})$ and $T^{'} \not\equiv T\dot{-}\varphi$.

Postulates ($T\dot{-}1$) – ($T\dot{-}5$) are analogues of ($K\dot{-}1$) – ($K\dot{-}5$). ($T\dot{-}6$) is the alternative for ($K\dot{-}6$). It selects from all the belief sets which are faithful to ($T\dot{-}1$) – ($T\dot{-}5$) those that have the weakest logical difference from the original one. Most importantly, no expressive power of the underlying logic is assumed in order to satisfy it. ($T\dot{-}6$) is stronger than ($K\dot{-}6$) as there are contraction operators that satisfy ($K\dot{-}6$) but not ($T\dot{-}6$). The reason is that a contraction operator which satisfies ($T\dot{-}6$) must be a *maxichoice contraction* [1], as it is equivalent to the *fullness postulate* in [1] which characterises maxichoice contraction.

Lemma 4. *Let K be belief set, $\dot{-}$ a contraction operator satisfying ($K\dot{-}1$) – ($K\dot{-}5$). Then the following are equivalent:*

1. *$\dot{-}$ satisfies ($T\dot{-}6$).*
2. *if $\beta \in K$ and $\beta \notin K\dot{-}\varphi$, then $\varphi \in Cn((K\dot{-}\varphi) \cup \{\beta\})$*

Proof.
We only sketch the proof from 1 to 2, the proof from 2 to 1 is similar. Assume $\beta \in K$ and $\beta \notin K\dot{-}\varphi$ but $\varphi \notin Cn((K\dot{-}\varphi) \cup \{\beta\})$. We now prove $\beta \rightarrow \varphi \in \mathsf{Diff}(T, T\dot{-}\varphi)$, that is the three conditions in the definition of Diff must be satisfied by $\beta \rightarrow \varphi$. Condition 1 is satisfied as $\varphi \in K$ and $K = Cn(K)$. Condition 2 is satisfied as $\beta \rightarrow \varphi \notin K\dot{-}\varphi$ which follows from $\varphi \notin Cn((K\dot{-}\varphi) \cup \{\beta\})$. Condition 3 is satisfied where the reasoning is the same as in the proof of Lemma 3. Now let $K^{'} = (K\dot{-}\varphi) \cup \{\beta \rightarrow \varphi\}$, thus $\beta \rightarrow \varphi \notin \mathsf{Diff}(K, K^{'})$. $K^{'}$ satisfies ($K\dot{-}1$) – ($K\dot{-}5$) and $K^{'} \not\equiv K\dot{-}\varphi$ however, $\mathsf{Diff}(K, K\dot{-}\varphi) \models \mathsf{Diff}(K, K^{'})$, which violates ($T\dot{-}6$). □

In order for Lemma 4 to hold, the underlying logic needs to have certain expressive power, but as far as $\mathcal{EL}$ is concerned, this is not a problem.

5 $\mathcal{EL}$ Contraction

The previous section proposed an alternative to the recovery postulate which assumes no expressive power of the underlying logic. We also proved that, assuming propositional logic, the proposed postulate characterises maxichoice contraction. In this section we develop algorithms that implement a contraction operator (maxichoice) for the DL $\mathcal{EL}$.

Within the algorithms, let $\mathtt{Sub}_{\mathtt{T}}(A) = \{B \mid T \models B \sqsubseteq A\}$ be the set of concept names that are subsumed by A in a TBox T. Since $\mathcal{EL}$ has a polynomial time subsumption checking procedure [11] $\mathtt{Sub}_{\mathtt{T}}(A)$ can be computed in polynomial time. To simplify the algorithms we assume, without loss of generality, that the input TBox is normalised and thus contains only axioms of the form:

- $A \equiv \exists R.B$ or $A \sqsubseteq \exists R.B$, where B is a concept name;
- $A \equiv F$ or $A \sqsubseteq F$, where F is a conjunction of concept names such that each conjunct B is either pseudo-primitive or $B \equiv \exists R.C$ is in the TBox.

As justified in [9], it takes polynomial time to construct such a normalised TBox that is logically equivalent to the original one. In the explanation of the algorithms we distinguish between two types of GCIs, namely explicit GCIs and implicit GCIs. The explicit GCIs are those that are actually present in the TBox whereas the implicit ones are not present in the TBox but can be deduced from the explicit GCIs.

for A *a subconcept of* C *or* D **do**
  if $A \equiv A^{'} \in T$ **then**
    replace A with $A^{'}$ in C or D
$T^{'} \leftarrow \emptyset$
for $A \in \mathsf{N_C}$ **do**
  if A *is pseudo-primitive in* T **then**
    $T^{'} \leftarrow T^{'} \cup \{B \sqsubseteq A | B \in \mathtt{Sub_T}(A)\} \cup \{A \sqsubseteq C | A \in \mathtt{Sub_T}(C)\}$
  if $A \equiv A_1 \sqcap \cdots \sqcap A_n \in T$ **then**
    $T^{'} \leftarrow T^{'} \cup \{B \sqsubseteq A | B \in \mathtt{Sub_T}(A)\} \cup \{A \sqsubseteq C | A \in \mathtt{Sub_T}(C)\} \cup \{A_1 \sqcap \cdots \sqcap A_n \sqsubseteq A\}$
  if $A \equiv \exists R.A^{'} \in T$ **then**
    $T^{'} \leftarrow T^{'} \cup \{B \sqsubseteq A | B \in \mathtt{Sub_T}(A)\} \cup \{A \sqsubseteq C | A \in \mathtt{Sub_T}(C)\} \cup \{\exists R.A^{'} \sqsubseteq A\} \cup \{A \sqsubseteq \exists R.A^{'}\}$
  if $A \sqsubseteq \exists R.C \in T$ **then**
    $T^{'} \leftarrow T^{'} \cup \{B \sqsubseteq \exists R.C | B \sqsubseteq A \in T\}$
  if $B \sqsubseteq \exists R.A \in T$ **then**
    $T^{'} \leftarrow T^{'} \cup \{B \sqsubseteq \exists R.C | A \sqsubseteq C \in T\}$
$T \leftarrow T^{'}$

Algorithm 1: Algorithm $\mathtt{Simplify}(T, C \sqsubseteq D)$

Before carrying out steps that will result in contraction of a GCI $C \sqsubseteq D$ from a TBox T, we simplify $C \sqsubseteq D$ and reconstruct T in Algorithm 1. The main purpose is to deduce the implicit GCIs that involve a concept name on the left and an existential restriction on the right (such as $A \sqsubseteq \exists R.B$) and to deduce the implicit GCIs that involve only concept names (such as $A \sqsubseteq B$). This will guarantee none of the implicit GCIs are eliminated unnecessarily in the contracted TBox. Upon termination T contains all subsumptions between concept names. Moreover, axioms of the form $A \equiv B$ are replaced with $A \sqsubseteq B$ and $B \sqsubseteq A$. The exception is that if $A \equiv A_1 \sqcap \cdots \sqcap A_n$ is in T, $A \sqsubseteq A_1 \sqcap \cdots \sqcap A_n$ is not included as it can be deduced from the subsumptions between A and $A_1, \ldots, A_n$. As a consequence of the simplification and reconstruction, when we

reduce the contraction of $C \sqsubseteq D$ to contraction of either $A \sqsubseteq \exists R.B$ or $B \sqsubseteq A$ for A, B concept names (as in Section 3) then concept A is guaranteed to be pseudo-primitive. At last, it is easy to check that Algorithm 1 returns a TBox that is equivalent to the input one.

Input: TBox T, GCI $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B$
Output: TBox $T^{'}$
`Simplify`$(T, A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B)$
$T^{'} \leftarrow T$
if *B is of the form $B_1 \sqcap \cdots \sqcap B_m$* **then**
  $B_i \leftarrow \lambda(B_1, \ldots, B_m)$
  $T^{'} \leftarrow$ `Contract`$(T, A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B_i)$
else if *B is pseudo-primitive in T* **then**
  for $i \leftarrow 1$ **to** n **do**
    if $A_i \sqsubseteq B \notin T$ **then**
      continue
    else
      $T^{'} \leftarrow$ `ContractCN`$(T^{'}, A_i, B)$
else if *B is of the form $\exists R.C$* **then**
  for $i \leftarrow 1$ **to** n **do**
    if $A_i \sqsubseteq \exists R.C \notin T$ **then**
      continue
    else if $A_i \in \mathsf{N_C}$ **then**
      $T^{'} \leftarrow$ `ContractER`$(T^{'}, A_i, \exists R.C)$
    else if *A_i is of the form $\exists R.A$* **then**
      $T^{'} \leftarrow$ `ContractCN`$(T^{'}, A, C)$
return $T^{'}$

Algorithm 2: Algorithm `Contract`$(T, A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B)$

Algorithm 2 is our main algorithm. To contract GCIs of the form $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B_1 \sqcap \cdots \sqcap B_m$, it is sufficient to contract one of $A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B_i$ for $1 \leq i \leq m$, as $\{A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B_i\} \models A_1 \sqcap \cdots \sqcap A_n \sqsubseteq B_1 \sqcap \cdots \sqcap B_m$. A selection function λ is used to model the user decision on which GCI to contract. According to Lemma 2, contraction of $C \sqsubseteq D$ can be reduced to contraction of either $A \sqsubseteq \exists R.B$ in which case Algorithm 2 calls Algorithm 4 or $B \sqsubseteq A$ in which case Algorithm 2 calls Algorithm 3. Note that when concept B in the algorithm is pseudo-primitive, then due to Part 1 of Lemma 1, if $T \models A_i \sqsubseteq B$ then A_i must be a concept name. Similarly when B is of the form $\exists R.C$, due to Part 2 of Lemma 1, the possible forms of concepts subsumed by B are fixed.

Algorithm 3 handles contraction of GCIs of the form $A \sqsubseteq B$ where A, B are concept names (B is pseudo-primitive). It first removes $A \sqsubseteq B$ from T as subsumption between concept names are all present in T. $A \sqsubseteq B$ may be implied by other GCIs which have to be eliminated. In detail, for every concept

Input: TBox T, GCI $A \sqsubseteq B$ where $A, B \in \mathsf{N_C}$
Output: TBox T'
$T' \leftarrow T$
$T' \leftarrow T' \setminus \{A \sqsubseteq B\}$
for $C \in \mathsf{N_C}$ *and* $A \sqsubseteq C \in T$ **do**
  if $C \sqsubseteq B$ **then**
    $T' \leftarrow T' \setminus \lambda\{A \sqsubseteq C, C \sqsubseteq B\}$
return T'

Algorithm 3: Algorithm $\texttt{ContractCN}(T, A \sqsubseteq B)$

name C such that $A \sqsubseteq C, C \sqsubseteq B \in T$, either $A \sqsubseteq C$ or $C \sqsubseteq B$ is eliminated. Once again the decision is modelled by the selection function λ. We do not have to consider the case that there are concepts of the form $\exists R.C$ such that $A \sqsubseteq \exists R.C, \exists R.C \sqsubseteq B \in T$, because due to Lemma 1, this will imply B is not pseudo-primitive.

Algorithm 4 handles contraction of GCIs of the form $A \sqsubseteq \exists R.B$ where A, B are concept names (A is pseudo-primitive). If $T \models A \sqsubseteq \exists R.B$ then either there is a concept $\exists R.D$ such that $A \sqsubseteq \exists R.D, D \sqsubseteq \exists B \in T$ or there is a concept name D such that $A \sqsubseteq D, D \sqsubseteq \exists R.B \in T$. For each of the two cases there are two ways of breaking the entailment, the decision is again modelled by the selection function λ. Depending on the decision either Algorithm 4 or Algorithm 3 is called.

Input: TBox T, GCI $A \sqsubseteq \exists R.B$ where $A, B \in \mathsf{N_C}$
Output: TBox T'
$T' \leftarrow T \setminus \{A \sqsubseteq \exists R.B\}$
foreach $D \in \mathsf{N_C}$ *such that* $A \sqsubseteq \exists R.D \in T$ *and* $D \sqsubseteq B \in T$ **do**
  $T' \leftarrow \lambda\{\texttt{ContractER}(T', A \sqsubseteq \exists R.D), \texttt{ContractCN}(T', D \sqsubseteq B)\}$
foreach $D \in \mathsf{N_C}$ *such that* $A \sqsubseteq D \in T$ *and* $D \sqsubseteq \exists R.B \in T$ **do**
  $T' \leftarrow \lambda\{\texttt{ContractER}(T', D \sqsubseteq \exists R.B), \texttt{ContractCN}(T', A \sqsubseteq D)\}$
return T'

Algorithm 4: Algorithm $\texttt{ContractER}(T, A \sqsubseteq \exists R.B)$

The algorithms successfully implement a contraction operator for the DL $\mathcal{EL}$ as justified by the following theorem.

Theorem 1. *Let T be an $\mathcal{EL}$ TBox, $C \sqsubseteq D$ an $\mathcal{EL}$ GCI. If we define $T \dot{-} (C \sqsubseteq D) = \texttt{Contracting}(T, C \sqsubseteq D)$, then $\dot{-}$ satisfies $(T \dot{-} 1) - (T \dot{-} 6)$.*

Proof.
We only sketch the proof for satisfaction of $(T \dot{-} 6)$ as others are straightforward. Due to Lemma 4 $(T \dot{-} 6)$ is satisfied if the following holds: If $\beta \in Cn(T)$ and $\beta \notin Cn(T \dot{-} \varphi)$ then $\varphi \in Cn((T \dot{-} \varphi) \cup \{\beta\})$ for φ, β GCIs. It is clear from the

algorithm, whenever a GCI β is removed, it must be the case that 1) there is a GCI $\psi \in Cn(T)$ such that $\{\beta, \psi\} \models \varphi$, and $\psi \in T \dot{-} \varphi$, and 2) $\beta = \varphi$. In both cases, $\varphi \in Cn((T \dot{-} \varphi) \cup \{\beta\})$ □

The algorithms will run in polynomial time because each for loop in the algorithm iterates a constant number of times and the most time consuming operation is subsumption checking which has a polynomial procedure. In the following, we give an example for contracting a GCI from a TBox. At the beginning Algorithm 2 is called.

Example 1. $T = \{C \sqsubseteq F, F \sqsubseteq A, E \sqsubseteq D \sqcap B, D \sqsubseteq \exists R.A\}$, we want to contact $E \sqcap \exists R.D \sqsubseteq \exists R.A$ from T. Firstly, upon termination of `Simplify`$(T, E \sqcap \exists R.D \sqsubseteq \exists R.A)$, $T = \{C \sqsubseteq F, F \sqsubseteq A, C \sqsubseteq A, E \sqsubseteq B, E \sqsubseteq D, D \sqsubseteq \exists R.A, E \sqsubseteq \exists R.A\}$, that is the implicit GCIs $C \sqsubseteq A, E \sqsubseteq B, E \sqsubseteq D$ and $E \sqsubseteq \exists R.A$ are made explicit. Now we will proceed to the last if statement of Algorithm 2. As $E \sqsubseteq \exists R.A \in T$, `ContractER`$(T, E \sqsubseteq \exists R.A)$ is called. Now in running `ContractER`$(T, E \sqsubseteq \exists R.A)$, $E \sqsubseteq \exists R.A$ is first removed resulting in $T = \{C \sqsubseteq F, F \sqsubseteq A, C \sqsubseteq A, E \sqsubseteq B, E \sqsubseteq D, D \sqsubseteq \exists R.A\}$. As $E \sqsubseteq D$ and $D \sqsubseteq \exists R.A$ are in T, one of them needs to be removed. Assume the selection function selects $E \sqsubseteq D$, which means `ContractCN`(T, E, D) is called which removes $E \sqsubseteq D$ from T resulting in $T = \{C \sqsubseteq F, F \sqsubseteq A, C \sqsubseteq A, E \sqsubseteq B, D \sqsubseteq \exists R.A\}$. Upon termination of Algorithm 2 $T = \{C \sqsubseteq F, F \sqsubseteq A, C \sqsubseteq A, E \sqsubseteq B, D \sqsubseteq \exists R.A\}$. The GCIs removed are $E \sqsubseteq D$ and $E \sqsubseteq \exists R.A$. It is easy to see that $T \cup \{E \sqsubseteq D\} \models E \sqcap \exists R.D \sqsubseteq \exists R.A$ and $T \cup \{E \sqsubseteq \exists R.A\} \models E \sqcap \exists R.D \sqsubseteq \exists R.A$.

Note that if we were to use a belief base approach, where only the explicit GCIs are considered, we will end up with $T = \{C \sqsubseteq F, F \sqsubseteq A, D \sqsubseteq \exists R.A\}$ or $T = \{C \sqsubseteq F, F \sqsubseteq A, E \sqsubseteq D \sqcap B\}$. In either case we have some implicit GCIs removed unnecessarily. For instance, in the former case, $E \sqsubseteq D \sqcap B$ is removed therefore T no longer entails $E \sqsubseteq B$, however $E \sqsubseteq B$ has nothing to do with T entailing $E \sqcap \exists R.D \sqsubseteq \exists R.A$. In this respect our approach, which concerns belief contraction with belief sets (logically closed theory), turns out to be more plausible as it eliminates only what is necessary and preserves the rest.

6 Related Work

The problem of belief revision in DLs has been extensively studied. So far existing works propose different constructions of revision operators that consider only explicitly presented GCIs [6–8]. In these works the resulting TBox is obtained by adding the new GCIs to the original TBox then resolving any inconsistency caused. The latter part is achieved by first using the debugging service in [13, 14] to identify the minimum sets of GCIs responsible for the inconsistency then removing at least one element from each set. As demonstrated in [15] the removal of GCIs responsible for the entailment of some consequences may result in the loss of some implicit GCIs. Since our work adheres to the AGM framework where all the implicit GCIs are taken into account when performing belief change, there

will be no loss of such GCIs. Also, instead of revision, our work is concerned with contraction. In terms of generalising the AGM postulates, the work in [5] is the closest to ours. They generalise the AGM revision postulates model-theoretically. The generalized postulates are applicable to any DLs as expressiveness is no longer an issue.

7 Conclusion and Future Work

In this paper we have developed algorithms for contracting GCIs from an $\mathcal{EL}$ TBox. These algorithms are significant for a number of reasons. Firstly, the operation they realise adheres to all of the basic AGM contraction postulates except the recovery postulate. We argue that the original recovery postulate is not appropriate to DLs as they are less expressive than classical logics. Therefore, we introduced a variant of the recovery postulate that is based on the notion of logical difference. In fact, when the underlying logic is classical, our postulates coincide with the AGM. As such, ours is the closest account of belief change for DLs to the AGM account of those currently proposed in the literature. We argue that it is more important to focus on contraction since revision behavior is simply set union in $\mathcal{EL}$ as it is not possible to have an inconsistent TBox.

It should be noted that as our contraction operation only guarantees minimal change in terms of logical difference, all axioms could change their syntactic form so that it may be the case that the contracted TBox bears no syntactic resemblance to the original TBox. However, for the DL $\mathcal{EL}$, this may not be undesirable. Additionally, the contracted TBox obtained by our contraction operator may contain arbitrary GCIs that violate our assumptions, thus preventing further application of the contraction operator. In other words our algorithm can not handle iterated contraction. In future work, we would like to address these issues. It would also be useful to consider the supplementary AGM postulates and how they or similar postulates would be satisfied.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic **50**(2) (1985) 510–530
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of IJCAI-05. (2005) 364–369
3. Spackman, K.A., Campbell, K.E., Cote, R.A.: SNOMED RT: A reference terminology for health care. In: Journal of the American Medical Informatics Association. (1997) 640–644
4. Flouris, G., Plexousakis, D., Antoniou, G.: Generalizing the AGM postulates: preliminary results and applications. In: Proc. NMR-04. (2004) 171–179
5. Qi, G., Liu, W., Bell, D.A.: Knowledge base revision in description logics. In: Proc. JELIA-06. (2006) 386–398
6. Halaschek-Wiener, C., Katz, Y., Parsia, B.: Belief base revision for expressive description logics. In: Proc. OWL-06. (2006)

7. Ribeiro, M.M., Wassermann, R.: Belief base revision for expressive description logics preliminary results. In: Proc. IWOD-07. (2007)
8. Qi, G., Haase, P., Huang, Z., Z.Pan, J.: A kernel revision operator for terminologies. In: Proc. DL-08. (2008)
9. Konev, B., Walther, D., Wolter, F.: The logical difference problem for description logic terminologies. In: Proc. of IJCAR-08. (2008) 259–274
10. Flouris, G.: On Belief Change and Ontology Evolution. PhD thesis, Department of Computer Science, University of Crete (2006)
11. Baader, F.: Terminological cycles in a description logic with existential restrictions. LTCS-Report LTCS-02-02, Institute for Theoretical Computer Science, Dresden University of Technology (2002)
12. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic $\mathcal{EL}$. In: Proceedings of the 2007 International Workshop on Description Logics (DL2007). (2007)
13. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. Journal of Automated Reasoning **39**(3) (2007) 317–349
14. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of owl dl entailments. In: Proceedings of 6th International Semantic Web Conference(ISWC'07). (2007) 267–280
15. Lam, S.C.J., Pan, J.Z., Sleeman, D., Vasconcelos, W.: A fine-grained approach to resolving unsatisfiable ontologies. In: In Proc. WI-06. (2006) 428–434