

Concept Contraction in the Description Logic \mathcal{EL}

Tjitze Rienstra¹, Claudia Schon¹, Steffen Staab^{2,3}

¹University of Koblenz-Landau, Germany

²University of Stuttgart, Germany

³University of Southampton, UK

{rienstra, schon}@uni-koblenz.de, steffen.staab@ipvs.uni-stuttgart.de

Abstract

In this paper we study the problem of *concept contraction* for the description logic \mathcal{EL} . Concept contraction is concerned with the following question: Given two concepts C and D (with the interesting case being that D subsumes C) how can we find a generalisation of C that is not subsumed by D but is otherwise as similar as possible to C ? We take an AGM-style approach and model this problem using the notion of a *concept contraction* operator. We consider constructive definitions as well as sets of postulates for concept contraction, and link the two by means of representation theorems.

1 Introduction

An agent, when faced with new information, must incorporate this information into its stock of beliefs. Old beliefs may be inconsistent with this new information, which means that this process may involve removal of old beliefs. In the so called *AGM* approach to belief change, a distinction is made between *belief revision* and *belief contraction* (Alchourrón *et al.* 1985). While belief revision deals with consistently adding a new belief, contraction deals with removal of beliefs. Contraction can be seen as a more fundamental type of change, since revision can often be defined in terms of contraction, by first removing beliefs inconsistent with the new belief, after which the new belief can be added without introducing inconsistency.

One area where belief change plays an important role is that of *ontology change* (Flouris *et al.* 2008). Ontologies are a central aspect of knowledge representation and play a foundational role in the Semantic Web, facilitating integration, sharing and reuse of knowledge across different applications and domains. Ontology change is a broad topic and includes *ontology evolution*, or modification due to change in the domain or its conceptualisation; *ontology debugging* and *repair*, where ontologies are modified to restore consistency or eliminate faulty conclusions; and also *ontology mapping*, *matching*, *integration* and *merging*.

In this paper we investigate belief change for ontologies by considering the problem of *concept contraction*: given two concepts C and D , we want to obtain a new concept that is as similar as possible to C but that is not subsumed by D . Obviously, the interesting case here is if D subsumes C , because otherwise C may be left unchanged. We can see this as a form of contraction, since

C must be generalised in a way similar to how a knowledge base is weakened during belief contraction. We take as a basis the light-weight description logic \mathcal{EL} (Brandt 2004), which forms the basis of the OWL 2 EL profile (The OWL Working Group 2012). The most important inference problems in \mathcal{EL} are of polynomial complexity (Baader *et al.* 2005). Despite its limited expressivity, \mathcal{EL} has proven to be sufficient for large-scale ontologies in areas such as medicine and health-care (Rector and Horrocks 1997; Donnelly 2006). Unlike concept contraction, concept revision would be trivial in \mathcal{EL} , since \mathcal{EL} lacks the power to express unsatisfiable concepts, and hence the revision of C by D can be achieved by intersecting C and D , which does not lead to inconsistency.

Concept contraction is related to but different from most existing approaches to belief change for description logics, which focus on change of a T-Box and/or A-Box as a whole. These approaches deal with addition and removal of consequences, typically by adding, removing or changing axioms and assertions in a T-Box or A-Box (see, e.g., (De Giacomo *et al.* 2006; Flouris *et al.* 2005; Liu *et al.* 2006; Qi and Du 2009; Qi *et al.* 2006; Zheleznyakov *et al.* 2019)). By contrast, our approach deals with change of concepts. Of course, axioms and assertions can be changed by changing the concepts to which they refer, so our notion of change may be employed within the context of changing a T-Box or A-Box as well. Consider for example the T-Box $\{Mother \equiv Female \sqcap \exists hasChild. \top\}$ and A-Box $\{Mother(jane)\}$, where contracting $Mother(jane)$ can be done by assigning to $jane$ a new concept not subsumed by $Mother$ but otherwise as similar as possible (e.g. $Female$).

We formalise concept contraction by adapting the AGM approach: our object of study is a *concept contraction operator* \ominus that takes as input two concepts C, D and returns a new concept $C \ominus D$ representing the contraction of C by D . On the one hand, we define operators in the familiar constructive manner, by adapting the notion of AGM partial and full meet contraction (Alchourrón *et al.* 1985). On the other hand, we propose sets of postulates that determine classes of operators that are well-behaved in a precisely stated sense. The constructive definitions are linked with the postulates by representation theorems, which establish a precise correspondence between certain classes of operators and sets of postulates. We furthermore show that our approach can be

applied relative to an acyclic T-Box containing terminological knowledge that has to be taken into consideration during contraction. Despite the restriction to acyclic TBoxes our results are relevant to practice, since EL ontologies with acyclic TBoxes are widely used in practice. For example, SNOMED CT (Donnelly 2006) and Gene Ontology (Ashburner *et al.* 2000) both have an acyclic TBox (Haase and Lutz 2008).

The overview of this paper is as follows. We start with the necessary preliminaries concerning the description logic \mathcal{EL} . We then define the notion of a concept contraction operator. Next, we define the class of *LCS concept contraction operators*. We then discuss sets of postulates that fully characterize LCS concept contraction, as well as the restricted class of operators called *maxi-choice operators*. We then address the matter of dealing with contraction with respect to a background T-Box. This discussion is restricted to T-Boxes that are acyclic, and we also discuss problems that arise in case a T-Box contains cycles. We conclude by discussing related work and some directions for future research.

2 Preliminaries

An \mathcal{EL} signature is a pair $\Sigma = (N_C, N_R)$, where N_C is the set of *atomic concepts* and N_R the set of *atomic roles*. Given a signature $\Sigma = (N_C, N_R)$ we use the letters A, B to range over N_C ; R, S to range over N_R ; and C, D to range over *concepts*, which are formulas inductively generated by the following rule:

$$\top \mid A \mid C \sqcap D \mid \exists R.C.$$

Given a signature Σ , the set of concepts will also be denoted by $\mathcal{C}(\Sigma)$.

The semantics of \mathcal{EL} is defined in terms of *interpretations*. Let $\Sigma = (N_C, N_R)$ be a signature. An *interpretation* I is a pair (Δ^I, \cdot^I) , where Δ^I is a non-empty set called the *domain* and \cdot^I is an *interpretation function* that maps each $A \in N_C$ to a set $A^I \subseteq \Delta^I$ and each $R \in N_R$ to a set $R^I \subseteq \Delta^I \times \Delta^I$. The interpretation function is extended to concepts using the following inductive definition.

$$\top^I = \Delta^I \tag{1}$$

$$(C \sqcap D)^I = C^I \cap D^I \tag{2}$$

$$(\exists R.C)^I = \{a \in \Delta^I \mid (a, b) \in R^I \text{ for some } b \in C^I\} \tag{3}$$

We say that C is subsumed by D (written $C \sqsubseteq D$) if and only if for each interpretation I we have $C^I \subseteq D^I$. We furthermore write $C \equiv D$ (equivalence) as shorthand for $C \sqsubseteq D \wedge D \sqsubseteq C$; and $C \sqsubset D$ (strict subsumption) as shorthand for $C \sqsubseteq D \wedge D \not\sqsubseteq C$. A *T-Box* with signature $\Sigma = (N_C, N_R)$ is a finite set of definitions of the form $A \equiv C$ with $A \in N_C$ and $C \in \mathcal{C}(\Sigma)$ such that no A appears more than once on the left-hand side of a definition in \mathcal{T} . An interpretation I is a model of \mathcal{T} if and only if for all definitions $A \equiv C$ in \mathcal{T} we have $A^I = C^I$. We say that C is subsumed by D with respect to \mathcal{T} (written $C \sqsubseteq^{\mathcal{T}} D$) if, $C^I \subseteq D^I$ holds for every model I of \mathcal{T} and write $C \equiv^{\mathcal{T}} D$ (resp. $C \sqsubset^{\mathcal{T}} D$) as shorthand for $C \sqsubseteq^{\mathcal{T}} D \wedge D \not\sqsubseteq^{\mathcal{T}} C$ (resp. $C \sqsubseteq^{\mathcal{T}} D \wedge D \not\equiv^{\mathcal{T}} C$).

Given a T-Box \mathcal{T} we say that an atomic concept A *depends* on another atomic concept B if B appears on the right-hand side of the definition of A . A T-Box is *acyclic* if no concept depends, directly or indirectly, on itself. Given an acyclic T-Box \mathcal{T} we define the *unfolding* $\hat{\mathcal{T}}(\cdot)$ by $\hat{\mathcal{T}}(A) = A$, if A does not appear on the left-hand side of a definition in \mathcal{T} ; $\hat{\mathcal{T}}(A) = \hat{\mathcal{T}}(C)$, if $A \equiv C$ appears in \mathcal{T} ; $\hat{\mathcal{T}}(C \sqcap D) = \hat{\mathcal{T}}(C) \sqcap \hat{\mathcal{T}}(D)$; and $\hat{\mathcal{T}}(\exists R.C) = \exists R.\hat{\mathcal{T}}(C)$. Note that $\hat{\mathcal{T}}(\cdot)$ is well-defined only if \mathcal{T} is acyclic. If \mathcal{T} is acyclic, subsumption with respect to \mathcal{T} can be reduced to regular subsumption using the rule $C \sqsubseteq^{\mathcal{T}} D$ if and only if $\hat{\mathcal{T}}(C) \sqsubseteq \hat{\mathcal{T}}(D)$ (Nebel 1990).

When we define contraction with respect to a T-Box \mathcal{T} we need to determine the signature of \mathcal{T} consisting only of *base concepts*. Given a T-Box \mathcal{T} with signature $\Sigma = (N_C, N_R)$, we say that an atomic concept $A \in N_C$ is a *base concept* of \mathcal{T} if and only if A does not appear on the left-hand side of a definition in \mathcal{T} . The set of base concepts of \mathcal{T} will be denoted by $\mathcal{B}_{\mathcal{T}}$. The *base signature* of \mathcal{T} is the signature $(\mathcal{B}_{\mathcal{T}}, N_R)$ and will be denoted by $\Sigma_{\downarrow \mathcal{T}}$. Note that, if \mathcal{T} is an acyclic T-Box with signature Σ then for every concept $C \in \mathcal{C}(\Sigma)$, we have $\hat{\mathcal{T}}(C) \in \mathcal{C}(\Sigma_{\downarrow \mathcal{T}})$, i.e. $\hat{\mathcal{T}}(C)$ is expressible in the signature consisting of just the base concepts of \mathcal{T} .

3 Concept Contraction

We model the process of concept contraction by means of a concept contraction operator. Formally, a concept contraction operator \ominus with signature Σ associates each pair of concepts $C, D \in \mathcal{C}(\Sigma)$ with a new concept $C \ominus D \in \mathcal{C}(\Sigma)$. Our aim is to define \ominus in such a way that $C \ominus D$ represents the result of contracting D from C in a rational way. Contraction in this context is understood as finding a weakening or generalisation of C (i.e., some concept C' such that $C \sqsubseteq C'$) that is not subsumed by D . An important requirement is that $C \ominus D$ must somehow be as similar as possible to C .

Our first step is to consider concept contraction without a background T-Box. We look at contraction with respect to a background T-Box in Section 6. In line with the well-known AGM model of contraction we provide two types of definitions, the first based on an explicit construction called *LCS contraction*, and the second is based on sets of postulates. We then establish a correspondence between the two types of definitions by means of representation theorems.

4 LCS Contraction

In this section we provide an explicit construction of a particular kind of contraction operator that we call *LCS contraction operators*. As a first step we define the notion of *remainder*. Formally, a remainder of a concept C with respect to a concept D is a maximally specific generalisation of C not subsumed by D . In the following definitions we assume that we are working with a fixed signature.

Definition 1. A *remainder* of a concept C with respect to a concept D is a concept C' such that:

1. $C \sqsubseteq C'$,
2. $C' \not\sqsubseteq D$,

3. $\nexists C''$ such that $C \sqsubseteq C''$, $C'' \not\sqsubseteq D$, and $C'' \sqsubset C'$.

The following proposition will be useful later on. It states that, given concepts C and D and a generalisation X of C that is not subsumed by D , there exists some intermediate concept between C and X that is a remainder of C with respect to D . This implies (if we set $X = \top$) that a remainder of C with respect to D always exists, provided that $D \neq \top$.

Proposition 1. If $C \sqsubseteq X$ and $X \not\sqsubseteq D$ then there is some Z that is a remainder of C with respect to D and $Z \sqsubseteq X$.

Proof. Suppose $C \sqsubseteq X$ and $X \not\sqsubseteq D$. Suppose there is no remainder Z of C with respect to D such that $Z \sqsubseteq X$. Then for all C' such that $C \sqsubseteq C' \sqsubseteq X$ and $C' \not\sqsubseteq D$ there is a C'' such that $C \sqsubseteq C'' \sqsubset C'$ and $C'' \not\sqsubseteq D$. This implies that there is an infinite sequence of concepts C_0, C_1, C_2, \dots such that $C \sqsubseteq C_0 \sqsubset C_1 \sqsubset C_2 \dots$. It was shown in (Baader and Morawska 2010) that this is not possible. \square

The remainders of C with respect to D can be considered candidate answers for contracting D from C . It is easy to see that sets of remainders are closed under equivalence (i.e., if $C' \equiv C''$ and C' is a remainder of C with respect to D then so is C''). Because we will not concern ourselves with syntactic form it will be convenient to represent sets of remainders as sets of equivalence classes under \equiv . The equivalence class of a concept C is the set $\{D \mid C \equiv D\}$ and will be denoted by $[C]$.

Definition 2. Given concepts C and D we define $C \perp D$ by $C \perp D = \{[C'] \mid C' \text{ is a remainder of } C \text{ with respect to } D\}$.

In general, $C \perp D$ may contain more than one (equivalence class of a) remainder. The question is thus: which ones do we choose? Like in AGM contraction we assume that there is a selection function σ that selects the “most important” remainders. Two limiting cases are (1) selection of single remainders (*maxi-choice*) and (2) selection of *all* remainders (*full meet*).

Definition 3. A selection function σ selects, given every pair of concepts C, D , a set $\sigma(C \perp D)$ such that:

1. If $C \perp D \neq \emptyset$ then $\sigma(C \perp D) \neq \emptyset$.
2. If $C \perp D \neq \emptyset$ then $\sigma(C \perp D) \subseteq C \perp D$.
3. If $C \perp D = \emptyset$ then $\sigma(C \perp D) = \{[C]\}$.

σ is called *maxi-choice* (resp. *full meet*) if and only if σ selects exactly one (resp. all) elements of $C \perp D$.

In general we still need to deal with the fact that the selection function may select more than one remainder. In AGM contraction this is dealt with by intersecting all belief sets chosen by the selection function. This intersection is itself a belief set (i.e., is closed under logical consequence) and contains all beliefs that the chosen belief sets have in common. In our setting this would correspond to the union of the chosen remainders. This union is, however, not expressible in \mathcal{EL} . Instead we use the *least common subsumer*, which captures the best approximation of what a set of concepts have in common (Baader *et al.* 1999).

Definition 4. A concept C is a *least common subsumer* (or *LCS*) of concepts C_1, \dots, C_n if and only if

1. $C_1, \dots, C_n \sqsubseteq C$,
2. For each C' such that $C_1, \dots, C_n \sqsubseteq C'$ we have $C \sqsubseteq C'$.

Clearly, given concepts C_1, \dots, C_n and C'_1, \dots, C'_n such that $C_1 \equiv C'_1, \dots, C_n \equiv C'_n$, we have that C is an LCS of C_1, \dots, C_n if and only if C is an LCS of C'_1, \dots, C'_n . We can thus define LCS's as a function of a set of equivalence classes under \equiv :

Definition 5. Given a set $X = \{[C_1], \dots, [C_n]\}$ we denote by $LCS(X)$ the set of least common subsumers of C_1, \dots, C_n .

For any set of concepts, an LCS always exists and is unique up to equivalence (Baader *et al.* 1999). This implies that $LCS(X)$ coincides with an equivalence class under \equiv . We are now ready to define the class of *LCS contraction operators*.

Definition 6. Let σ be a selection function. A contraction operator \ominus is an *LCS contraction operator* defined by σ if and only if for all concepts C and D ,

$$C \ominus D \in LCS(\sigma(C \perp D)).$$

Furthermore, \ominus is called a *maxi-choice* (resp. *full meet*) contraction operator if it is defined by a *maxi-choice* (resp. *full meet*) selection function.

Clearly, any two LCS contraction operators \ominus and \ominus' defined by the same selection function σ are unique up to equivalence, in the sense that, for all concepts C and D , we have $C \ominus D \equiv C \ominus' D$. For this reason we will also refer to *the* LCS concept contraction operator defined by σ and denote this operator by \ominus^σ .

The picture sketched so far implies that concept contraction, like standard AGM contraction, is in general not uniquely determined. Only full meet contraction yields a uniquely determined operator. Furthermore, the principle of minimal change is best captured by a maxi-choice operator, but this requires one to select single remainders, which may not be feasible.

We now look at some examples. Here it is useful to first introduce the notion of a *most specific generalisation* (MSG) of an \mathcal{EL} concept. Given a concept C , a concept C' is an MSG of C if and only if $C \sqsubset C'$ and there is no C'' such that $C \sqsubset C'' \sqsubset C'$. Finding remainders and finding the LCS can then be done by iteratively generating MSGs of the concept we contract from. Clearly, \top has no MSG and, if C is atomic, then only \top is an MSG of C . If C is a conjunction then replacing one conjunct with its MSG yields an MSG of C . Finally, the MSG of a concept $\exists R.C$ is the concept $\exists R.C_1 \sqcap \dots \sqcap \exists R.C_n$, where C_1, \dots, C_n are the MSGs of C , while the MSG of $\exists R.\top$ is \top . We can construct a lattice of generalisations of a concept by placing the concept at the top, \top at the bottom, and an arrow from C to C' if and only if C' is an MSG of C . Figures 1a and 1b show the lattice of generalisations of the concepts $F \sqcap G \sqcap H$ and $(\exists R.\exists Q.F) \sqcap G$, while figure 1c shows part of the lattice of generalisations of the concept $\exists R.(F \sqcap G \sqcap H)$ (here, F, G and H are atomic concepts). These figures will be used in the following examples. Finally, note that we can sometimes replace concepts in the lattice by logically equivalent

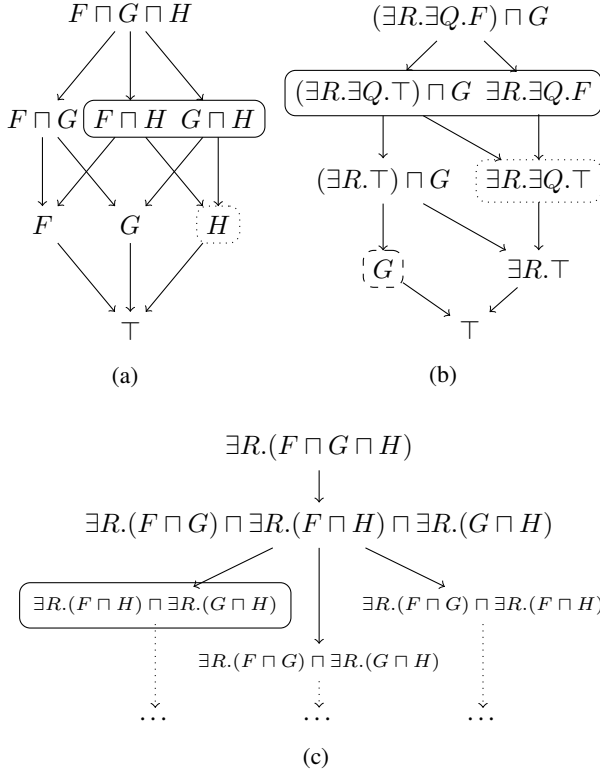


Figure 1: Lattices of generalisations of three concepts.

but more concise ones by the rules outlined above. For example, in figure 1a we write $F \sqcap G$ instead of $F \sqcap G \sqcap \top$, and in figure 1c we write $\exists R. (F \sqcap H) \sqcap \exists R. (G \sqcap H)$ instead of $\exists R. F \sqcap \exists R. G \sqcap \exists R. (F \sqcap H) \sqcap \exists R. (G \sqcap H)$ (since the first two conjuncts are subsumed by the third and fourth).

Example 1. Let σ be a selection function. Suppose we want to determine the contraction

$$F \sqcap G \sqcap H \odot_{\sigma} F \sqcap G. \quad (4)$$

The lattice of generalisations of $F \sqcap G \sqcap H$ is shown in Figure 1a. The remainders of $F \sqcap G \sqcap H$ with respect to $F \sqcap G$ are enclosed in a solid rectangle, while the LCS of the two remainders is enclosed in a dotted rectangle. Thus, if σ is a maxi-choice selection function then (4) equals either $F \sqcap H$ or $H \sqcap G$, and if σ is the full meet selection function, then (4) equals H .

Example 2. Now consider the contraction

$$(\exists R. \exists Q. F) \sqcap G \odot_{\sigma} (\exists R. \exists Q. F) \sqcap G \quad (5)$$

The lattice of generalisations of $(\exists R. \exists Q. F) \sqcap G$ is shown in Figure 1b. The remainders of $(\exists R. \exists Q. F) \sqcap G$ with respect to itself are enclosed in a solid line, while the LCS of the two remainders is enclosed in a dotted rectangle. Thus, if σ is a maxi-choice selection function then (5) equals either $(\exists R. \exists Q. \top) \sqcap G$ or $\exists R. \exists Q. F$, and if σ is the full meet selection function, then (5) equals $\exists R. \exists Q. \top$.

Example 3. Now consider the contraction

$$(\exists R. \exists Q. F) \sqcap G \odot_{\sigma} \exists R. \top. \quad (6)$$

In Figure 1b the one remainder of $(\exists R. \exists Q. F) \sqcap G$ with respect to $\exists R. \top$ is enclosed in a dashed rectangle. Since there is only one remainder, it is also the LCS. Hence (6) equals G for every possible σ .

Example 4. We finally consider the contraction

$$\exists R. (F \sqcap G \sqcap H) \odot_{\sigma} \exists R. (F \sqcap G). \quad (7)$$

Figure 1c shows part of the lattice of generalisations of $\exists R. (F \sqcap G \sqcap H)$. The remainder of $\exists R. (F \sqcap G \sqcap H)$ with respect to $\exists R. (F \sqcap G)$ is enclosed in a solid rectangle. Since there is only one remainder, it is also the LCS. Hence (7) equals $\exists R. (F \sqcap H) \sqcap \exists R. (G \sqcap H)$ for every possible σ .

One may object to the behaviour of full meet contraction demonstrated in some of these examples, which is rather radical. The reason for this lies in the fact that we cannot express unions and are forced to jump to something more general that is expressible. For example, the full meet contraction in Example 1 yields H , while $H \sqcap (F \sqcup G)$ might be considered more reasonable, but not expressible in \mathcal{EL} . It seems however that this is the price we pay for uniquely determined concept contraction in \mathcal{EL} .

5 Postulates for Concept Contraction

In this section we discuss postulates for concept contraction. They are all reformulations of the well-known AGM postulates for contraction. We assume that they are defined with respect to a fixed signature. We reformulate the first four ‘‘basic’’ AGM contraction as follows.

Preservation: If $D \equiv D'$ then $C \odot D \equiv C \odot D'$.

Success: If $D \neq \top$ then $C \odot D \sqsubseteq D$

Inclusion: $C \sqsubseteq C \odot D$

Vacuity: If $C \not\sqsubseteq D$ then $C \odot D \equiv C$

These postulates are rather straightforward and we will omit a detailed discussion. It is easy to see that any LCS contraction operator satisfies them. The fifth AGM postulate is Recovery, which can be translated as follows.

Recovery: $(C \odot D) \sqcap D \sqsubseteq C$.

Note that, like in the AGM setting, the converse of Recovery, i.e., $C \sqsubseteq ((C \odot D) \sqcap D)$, follows from Inclusion, provided that we have $C \sqsubseteq D$. Recovery is not appropriate in the current setting, however. To see why, consider the contraction $\exists R. (X \sqcap Y) \odot \exists R. X$. The lattice of generalisations of $\exists R. (X \sqcap Y)$ is shown in Figure 2. The unique remainder of $\exists R. (X \sqcap Y)$ with respect to $\exists R. X$ is $\exists R. Y$ (see solid rectangle). However $\exists R. Y \sqcap \exists R. X$ is not subsumed by $\exists R. (X \sqcap Y)$. In fact none of the generalisations of $\exists R. (X \sqcap Y)$, which are all shown in Figure 2, are subsumed by $\exists R. (X \sqcap Y)$ when intersected with $\exists R. X$. Recovery is therefore not suitable for our purpose. We now look for a replacement for Recovery. We

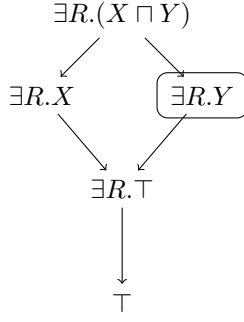


Figure 2: Lattice of generalisations for $\exists R.(X \sqcup Y)$.

first need:

Failure: $(C \ominus \top) \equiv C$.

In the presence of Inclusion, Failure follows from Recovery. However, Failure does not follow from the replacements for Recovery that we consider, and we therefore need Failure as a separate postulate.

We consider two replacements for Recovery, called *Fullness* and *Relevance*. Both are reformulations of postulates considered in (Hansson 1991). Fullness requires that, if we lose X after contracting D from C then adding X to the result gives us D again. Relevance weakens this requirement and states that, if we lose X after contracting D from C then there is some intermediate concept Y between C and $C \ominus D$ such that Y is not subsumed by D but is subsumed by D after adding X back.

Fullness: If $C \sqsubseteq X$ and $(C \ominus D) \not\sqsubseteq X$ then $(C \ominus D) \sqcap X \sqsubseteq D$.

Relevance: If $C \sqsubseteq X$ and $C \ominus D \not\sqsubseteq X$ then there is a Y such that $C \sqsupset Y \sqsubseteq C \ominus D$ and $Y \not\sqsubseteq D$ and $Y \sqcap X \sqsubseteq D$.

Clearly, Relevance implies Fullness. Furthermore, in the presence of Fullness and Inclusion, Vacuity is redundant:

Proposition 2. If \ominus satisfies Inclusion and Fullness then it satisfies Vacuity.

Proof. Suppose \ominus satisfies Inclusion and Fullness and suppose that $C \not\sqsubseteq D$. We need to prove that $(C \ominus D) \equiv C$. Inclusion implies the \supseteq direction. From $C \not\sqsubseteq D$ we get, using Inclusion, $(C \ominus D) \sqcap C \not\sqsubseteq D$. Contraposition using Fullness with $X = C$ now implies $(C \ominus D) \sqsubseteq C$, which completes the proof. \square

Together with the basic postulates, Fullness fully characterises maxi-choice LCS concept contraction:

Theorem 1. Let \ominus be a concept contraction operator. The following are equivalent:

1. \ominus is a maxi-choice LCS concept contraction operator.
2. \ominus satisfies Preservation, Inclusion, Success, Failure and Fullness.

Proof. We first prove that 1 implies 2. Suppose \ominus is a maxi-choice operator. Preservation, Inclusion, Success and Failure follow easily. For Fullness, let C, D, X be concepts and assume:

$$C \sqsubseteq X \quad (8)$$

$$(C \ominus D) \not\sqsubseteq X \quad (9)$$

Inclusion together with (8) implies

$$C \sqsubseteq (C \ominus D) \sqcap X. \quad (10)$$

From (9) it follows that

$$(C \ominus D) \sqcap X \sqsubset (C \ominus D). \quad (11)$$

Since \ominus is a maxi-choice operator, according to Definition 6, $C \ominus D$ is a remainder of C with respect to D . Then (10) and (11) imply $(C \ominus D) \sqcap X \sqsubseteq D$, which proves that \ominus satisfies Fullness.

To prove that 2 implies 1, suppose \ominus satisfies Inclusion, Success, Failure and Fullness. We have to show that \ominus is a maxi-choice contraction operator. According to Definition 3 and 6 and we achieve this by finding a selection function σ such that:

- (i) $\sigma(C \perp D) = \{[C]\}$, if $C \perp D = \emptyset$
- (ii) $\sigma(C \perp D) = \{[X]\}$ for some $[X] \in C \perp D$, if $C \perp D \neq \emptyset$
- (iii) $C \ominus D \in LCS(\sigma(C \perp D))$

We define such a function σ by

$$\sigma(C \perp D) = \{[C \ominus D]\}. \quad (12)$$

Preservation implies that σ is indeed a selection function. Then (i) follows via Failure. For (ii) we have to show that $C \ominus D$ is a remainder of C with respect to D . According to Definition 1, we have to show that:

- (a) $C \sqsubseteq C \ominus D$
- (b) $C \ominus D \not\sqsubseteq D$
- (c) $\nexists C''$ such that $C \sqsubseteq C''$, $C'' \not\sqsubseteq D$ and $C'' \sqsubset (C \ominus D)$

(a) follows from Inclusion and (b) from Success. For (c) assume the contrary, i.e., there is a C'' such that:

$$C \sqsubseteq C'' \quad (13)$$

$$C'' \not\sqsubseteq D \quad (14)$$

$$C'' \sqsubset (C \ominus D) \quad (15)$$

Then (15) implies

$$(C \ominus D) \not\sqsubseteq C''. \quad (16)$$

Fullness together with (13) and (16) implies

$$(C \ominus D) \sqcap C'' \sqsubseteq D. \quad (17)$$

Then (15) and (17) together imply $C'' \sqsubseteq D$. This contradicts (14) and thus proves (c). Finally, (iii) follows directly from (12), which implies that $LCS(\sigma(C \perp D)) = \{[C \ominus D]\}$. \square

It is instructive to look at our earlier examples involving maxi-choice contraction to check that they indeed validate Fullness. Consider Example 4. Suppose we set $C = \exists R.(F \sqcap G \sqcap H)$ and $D = \exists R.(F \sqcap G)$. We then have $(C \odot D) = \exists R.(F \sqcap H) \sqcap \exists R.(G \sqcap H)$. Thus, if $C \sqsubseteq X$ and $(C \odot D) \not\sqsubseteq X$ then either $X = \exists R.(F \sqcap G)$ or $X = \exists R.(F \sqcap G \sqcap H)$. Hence we have $(C \odot D) \sqcap X \sqsubseteq D$, which validates Fullness.

Our examples also demonstrate that Fullness is too strong to characterise non-maxi-choice contraction. Suppose we set $C = D = \exists R.(\exists Q.F) \sqcap G$. In example 2 we saw that, if σ is the full meet selection function, then we have $(C \odot_{\sigma} D) = \exists R.\exists Q.\top$. This violates Fullness because, if we set $X = G$ we get $C \sqsubseteq X$ and $(C \odot D) \not\sqsubseteq X$ but we do not have $(C \odot D) \sqcap X \sqsubseteq D$ and thus Fullness is not satisfied.

We can characterise LCS contraction in general by replacing Fullness with Relevance.

Theorem 2. Let \odot be a concept contraction operator. The following are equivalent:

1. \odot is an LCS concept contraction operator.
2. \odot satisfies Preservation, Inclusion, Success, Failure and Relevance.

Proof. We first prove that 1 implies 2. Let \odot be an LCS contraction operator defined by σ . Preservation, Inclusion, Success, Failure follow easily. To show that \odot satisfies Relevance, let C, D, X be concepts such that

$$C \sqsubseteq X \quad (18)$$

$$(C \odot D) \not\sqsubseteq X. \quad (19)$$

Using the definition of LCS, (19) implies that there is a Y such that $[Y] \in \sigma(C \perp D)$ and

$$Y \not\sqsubseteq X. \quad (20)$$

Because Y is a remainder of C with respect to D we have $C \sqsubseteq Y$ and, using (18) and (20), we get

$$C \sqsubset Y \quad (21)$$

Because Y is a remainder we furthermore have

$$Y \sqsubseteq (C \odot D) \quad (22)$$

$$Y \not\sqsubseteq D. \quad (23)$$

Now (20) implies $Y \sqcap X \sqsubset Y$. Using (18) and (21) it furthermore follows that $C \sqsubseteq Y \sqcap X$. Because Y is a remainder of C with respect to D it then follows that

$$Y \sqcap X \sqsubseteq D. \quad (24)$$

We thus have (21), (22), (23) and (24), which proves that \odot satisfies Relevance.

To prove that 2 implies 1, suppose \odot satisfies Inclusion, Success, Failure and Relevance. We must find a σ such that

- (i) $\sigma(C \perp D) = \{[C]\}$, if $C \perp D = \emptyset$.
- (ii) if $C \perp D \neq \emptyset$ then $\emptyset \subset \sigma(C \perp D) \subseteq C \perp D$.
- (iii) $[C \odot D] \equiv LCS(\sigma(C \perp D))$.

We define σ by

$$\sigma(C \perp D) = \begin{cases} \{[C]\} & \text{if } C \perp D = \emptyset, \\ \{[X] \in C \perp D \\ \mid X \sqsubseteq C \odot D\} & \text{otherwise.} \end{cases} \quad (25)$$

Preservation implies that σ is indeed a selection function. Then (i) holds by definition. For (ii), assume that $C \perp D \neq \emptyset$. We then have $D \not\equiv \top$ and thus Success gives $(C \odot D) \not\sqsubseteq D$ while inclusion gives $C \sqsubseteq (C \odot D)$. Proposition 1 then implies that there is a $[X] \in C \perp D$ such that $X \sqsubseteq C \odot D$ and hence that $\emptyset \subset \sigma(C \perp D)$. Clearly we also have $\sigma(C \perp D) \subseteq C \perp D$ and therefore (ii) holds. For (iii) we consider two cases:

Case 1: $C \not\sqsubseteq D$. Definition 2 then implies that $C \perp D = \{[C]\}$. Inclusion gives $C \sqsubseteq C \odot D$ and thus, using (25), it follows that $\sigma(C \perp D) = \{[C]\}$. Hence

$$LCS(\sigma(C \perp D)) = [C] \quad (26)$$

Together with Vacuity, (26) implies (iii).

Case 2: $C \sqsubseteq D$. If $D \equiv \top$ then (iii) follows via Failure. We now assume $D \not\equiv \top$. The definition of σ implies that, for all $X \in \sigma(C \perp D)$ we have $X \sqsubseteq C \odot D$. The definition of LCS then implies the \sqsupseteq direction.

For the \sqsubseteq direction, consider the following two equations:

$$C \odot D \not\sqsubseteq X \quad (27)$$

$$LCS(\sigma(C \perp D)) \not\sqsubseteq X \quad (28)$$

To prove the \sqsubseteq direction of (iii) we first prove that (27) implies (28). Assume (27). If $C \not\sqsubseteq X$ then clearly (28) holds. We now assume $C \sqsubseteq X$. Then, using Relevance, it follows that there is a Y such that

$$C \sqsubset Y \sqsubseteq C \odot D, \quad (29)$$

$$Y \not\sqsubseteq D, \quad (30)$$

$$Y \sqcap X \sqsubseteq D. \quad (31)$$

Then (29) and (30), using Proposition 1, imply that there is a $[Z] \in C \perp D$ such that $Z \sqsubseteq Y$ and $Z \not\sqsubseteq D$. Using (31) it follows that

$$Z \not\sqsubseteq X. \quad (32)$$

Because $Z \sqsubseteq C \odot D$ it follows that $[Z] \in \sigma(C \perp D)$. Using (32) it then follows that (28) holds. We thus proved that (27) implies (28). The contraposition of this implication, together with reflexivity of \sqsubseteq , implies the \sqsubseteq direction of (iii). \square

6 Contraction Modulo Acyclic T-Boxes

So far we considered concept contraction involving concepts whose semantics do not depend on a T-Box. We now look at the setting where there is an acyclic background T-Box. Let us look at an example. Consider the T-Box $\mathcal{T} = \{W \equiv V \sqcap U, U \equiv H, V \equiv F \sqcap G\}$. To account for \mathcal{T} during contraction we can adapt Inclusion by requiring that $C \sqsubseteq^{\mathcal{T}} C \odot D$, and Success by requiring that, if $D \not\equiv^{\mathcal{T}} \top$, then $C \odot D \not\sqsubseteq^{\mathcal{T}} D$. Examples of results of $W \odot V$ satisfying these requirements include $F \sqcap U$ (or equivalently $F \sqcap H$)

and $G \sqcap U$ (or equivalently $G \sqcap H$). We now adapt all postulates discussed so far in a similar way. That is, we define them relative to a T-Box \mathcal{T} and replace \sqsubseteq with $\sqsubseteq^{\mathcal{T}}$:

\mathcal{T} -Preservation: If $D \equiv^{\mathcal{T}} D'$ then $C \odot D \equiv^{\mathcal{T}} C \odot D'$.

\mathcal{T} -Success: If $D \not\equiv^{\mathcal{T}} \top$ then $C \odot D \not\sqsubseteq^{\mathcal{T}} D$

\mathcal{T} -Inclusion: $C \sqsubseteq^{\mathcal{T}} C \odot D$

\mathcal{T} -Vacuity: If $C \not\sqsubseteq^{\mathcal{T}} D$ then $C \odot D \equiv^{\mathcal{T}} C$

\mathcal{T} -Failure: $(C \odot \top) \equiv^{\mathcal{T}} C$.

\mathcal{T} -Fullness: If $C \sqsubseteq^{\mathcal{T}} X$ and $(C \odot D) \not\sqsubseteq^{\mathcal{T}} X$ then $(C \odot D) \sqcap X \sqsubseteq^{\mathcal{T}} D$.

\mathcal{T} -Relevance: If $C \sqsubseteq^{\mathcal{T}} X$ and $C \odot D \not\sqsubseteq^{\mathcal{T}} X$ then there is a Y such that $C \sqsubseteq^{\mathcal{T}} Y \sqsubseteq^{\mathcal{T}} C \odot D$ and $Y \not\sqsubseteq^{\mathcal{T}} D$ and $Y \sqcap X \sqsubseteq^{\mathcal{T}} D$.

Note that all these postulates are equivalent to the original versions when we assume \mathcal{T} is empty.

Given a T-Box \mathcal{T} we can define a contraction operator for the signature of \mathcal{T} in terms of a contraction operator for the base symbols of \mathcal{T} as follows.

Definition 7. Let \mathcal{T} be a T-Box with signature (N_C, N_R) . Let \odot be a concept contraction operator with signature $(\mathcal{B}_{\mathcal{T}}, N_R)$. We denote by $\odot^{\uparrow \mathcal{T}}$ the contraction operator with signature (N_C, N_R) defined by

$$C \odot^{\uparrow \mathcal{T}} D \equiv^{\mathcal{T}} \hat{\mathcal{T}}(C) \odot \hat{\mathcal{T}}(D).$$

It then holds that:

Proposition 3. \odot satisfies Inclusion (resp. Success, Failure, etc.) if and only if $\odot^{\uparrow \mathcal{T}}$ satisfies \mathcal{T} -Inclusion (resp. \mathcal{T} -Success, \mathcal{T} -Failure, etc.).

Proof. Only for Fullness (other postulates are similar). (Only If) Assume \odot satisfies Fullness and suppose $C \sqsubseteq^{\mathcal{T}} X$ and $(C \odot^{\uparrow \mathcal{T}} D) \not\sqsubseteq^{\mathcal{T}} X$. We then have $\hat{\mathcal{T}}(C) \sqsubseteq \hat{\mathcal{T}}(X)$ and, since $\hat{\mathcal{T}}(C) \odot \hat{\mathcal{T}}(D) \equiv \hat{\mathcal{T}}(\hat{\mathcal{T}}(C) \odot \hat{\mathcal{T}}(D))$ also $\hat{\mathcal{T}}(C) \odot \hat{\mathcal{T}}(D) \not\sqsubseteq \hat{\mathcal{T}}(X)$. Fullness now implies that $(\hat{\mathcal{T}}(C) \odot \hat{\mathcal{T}}(D)) \sqcap \hat{\mathcal{T}}(X) \sqsubseteq \hat{\mathcal{T}}(D)$. It then follows that $(C \odot^{\uparrow \mathcal{T}} D) \sqcap X \sqsubseteq^{\mathcal{T}} D$. Hence $\odot^{\uparrow \mathcal{T}}$ satisfies \mathcal{T} -Fullness. (If) Suppose $\odot^{\uparrow \mathcal{T}}$ satisfies \mathcal{T} -Fullness. and suppose $C \sqsubseteq X$ and $(C \odot D) \not\sqsubseteq X$. We have $\hat{\mathcal{T}}(C) = C$, $\hat{\mathcal{T}}(D) = D$, $\hat{\mathcal{T}}(X) = X$ and $C \odot D = \hat{\mathcal{T}}(C \odot D)$. Hence $C \sqsubseteq^{\mathcal{T}} X$ and $(C \odot^{\uparrow \mathcal{T}} D) \not\sqsubseteq^{\mathcal{T}} X$. \mathcal{T} -Fullness therefore implies $(C \odot^{\uparrow \mathcal{T}} D) \sqcap X \sqsubseteq^{\mathcal{T}} D$. Since $\hat{\mathcal{T}}(C \odot^{\uparrow \mathcal{T}} D) \equiv C \odot D$ it follows that $\hat{\mathcal{T}}(C \odot^{\uparrow \mathcal{T}} D) \sqcap X \sqsubseteq D$. \square

Thus, a contraction operator for just the base symbols of an acyclic T-Box \mathcal{T} determines a contraction operator for \mathcal{T} , and the latter is well-behaved (with respect to \mathcal{T}) if and only if the former is well-behaved. With this result we arrive at the following generalised representation theorem:

Theorem 3. Let \mathcal{T} be a T-Box and \odot' be a contraction operator, both with signature (N_C, N_R) . The following are equivalent:

1. $\odot' = \odot^{\uparrow \mathcal{T}}$ for some maxi-choice LCS contraction operator \odot with signature $(\mathcal{B}_{\mathcal{T}}, N_R)$.
2. \odot' satisfies \mathcal{T} -Inclusion, \mathcal{T} -Success, \mathcal{T} -Failure and \mathcal{T} -Fullness.

The following are also equivalent:

1. $\odot' = \odot^{\uparrow \mathcal{T}}$ for some LCS contraction operator \odot with signature $(\mathcal{B}_{\mathcal{T}}, N_R)$.
2. \odot' satisfies \mathcal{T} -Inclusion, \mathcal{T} -Success, \mathcal{T} -Failure and \mathcal{T} -Relevance.

Proof. From Theorem 1 and 2 with Proposition 3. \square

Example 5. Let \mathcal{T} be the T-Box defined by

$$\mathcal{T} = \{W \equiv V \sqcap U, U \equiv H, V \equiv F \sqcap G\}.$$

Let σ be a selection function, \odot_{σ} the associated contraction operator and let $\odot_{\sigma}^{\uparrow \mathcal{T}}$ be defined as in Definition 7. Contracting V from W with \mathcal{T} as background T-Box yields

$$\begin{aligned} W \odot_{\sigma}^{\uparrow \mathcal{T}} V &\equiv^{\mathcal{T}} \hat{\mathcal{T}}(W) \odot_{\sigma} \hat{\mathcal{T}}(V) \\ &\equiv^{\mathcal{T}} \hat{\mathcal{T}}(V) \sqcap \hat{\mathcal{T}}(U) \odot_{\sigma} \hat{\mathcal{T}}(F) \sqcap \hat{\mathcal{T}}(G) \\ &\equiv^{\mathcal{T}} \hat{\mathcal{T}}(F) \sqcap \hat{\mathcal{T}}(G) \sqcap H \odot_{\sigma} F \sqcap G \\ &\equiv^{\mathcal{T}} F \sqcap G \sqcap H \odot_{\sigma} F \sqcap G \end{aligned} \quad (33)$$

Figure 1a shows the lattice of generalisations of $F \sqcap G \sqcap H$ and in Example 1 we discussed the possible results of this contraction. That is, if σ is a maxi-choice selection function then (33) equals either $F \sqcap H$ or $H \sqcap G$, and if σ is the full meet selection function, then (33) equals H .

7 Contraction modulo cyclical T-Boxes

What about contraction modulo T-Boxes containing cycles? If \mathcal{T} is not acyclic then $\hat{\mathcal{T}}(\cdot)$ is not well-defined, so the approach taken in the previous section does not work. We can show that there is actually a more fundamental problem preventing us from modelling contraction modulo cyclical T-Boxes. Suppose we generalise the notion of remainder as follows.

Definition 8. Let \mathcal{T} be a T-Box. A \mathcal{T} -remainder of C with respect to D is a concept C' such that:

1. $C \sqsubseteq^{\mathcal{T}} C'$,
2. $C' \not\sqsubseteq^{\mathcal{T}} D$,
3. $\nexists C''$ such that $C \sqsubseteq^{\mathcal{T}} C''$, $C'' \not\sqsubseteq^{\mathcal{T}} D$, and $C'' \sqsubseteq^{\mathcal{T}} C'$.

We can now define the result of contracting C with D to be the LCS of one or more \mathcal{T} -remainders of C with respect to D . Unfortunately, this does not work. Recall that a remainder of C with respect to D always exists (unless $D \equiv \top$). This does not hold for \mathcal{T} -remainders if \mathcal{T} contains cycles.

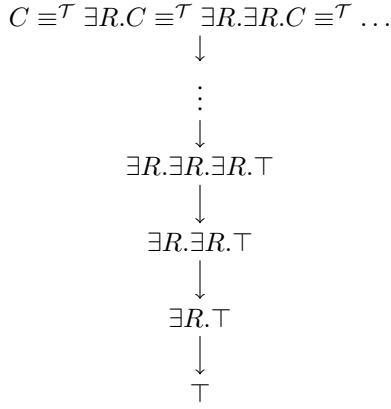


Figure 3: Lattice of generalisations for $\mathcal{T} = \{C \equiv \exists R.C\}$.

Example 6 (No \mathcal{T} -remainder). Consider the cyclical T-Box $\mathcal{T} = \{C \equiv \exists R.C\}$. What is the \mathcal{T} -remainder of C with respect to C ? Define c_i by $c_0 = \top$ and $c_i = \exists R.c_{i-1}$. Note that this remainder must be equivalent to c_i for some $i \geq 0$. But, for any $i \geq 0$, we have $c_{i+1} \sqsubset^{\mathcal{T}} c_i$, so c_i is not the remainder. Thus, a \mathcal{T} -remainder does not exist. The situation is illustrated in Figure 3.

It has furthermore been shown that the LCS may also not exist in the presence of cycles (Baader 2003). Thus, the notion of LCS contraction cannot be applied to cyclical T-Boxes in a straightforward way. A possible way out is to compute approximations of remainders and of the LCS, and using this as a basis for an “approximate LCS contraction” operator. Approximating the LCS in the presence of cycles has already been investigated by Baader (Baader 2003). A postulate-based characterisation of such an operator can perhaps be obtained by adapting the postulates discussed in this paper.

8 Related Work

In the introduction we referred to belief change for ontologies where change applies to a T-Box/A-Box as a whole. The notion of concept contraction is however quite different, so a direct comparison is not straightforward. We did argue, however, that change of T-Box/A-Box can often be achieved by performing contraction of concepts used by axioms and assertions. This is similar to what Baader et al. call *gentle repairs*, i.e., the weakening of axioms rather than complete removal (Baader et al. 2018). One particular kind of weakening that they consider is the weakening of an axiom $C \sqsubseteq D$ by replacing D with a sufficiently general weakening of D . We leave a more detailed study of this aspect, which should clarify possible connections with the approaches mentioned here, to future work.

The original Fullness and Relevance postulates, as well as a further weakening called *Core Retainment*, were proposed by Hansson as possibly more intuitive or reasonable replacements for Recovery (Hansson 1991). He proved, however, that in the presence of the basic AGM postulates, these postulates imply Recovery. However, this is not necessarily

true when the underlying logical formalism that is used is non-classical. For instance, the Relevance postulate is used in (Ribeiro et al. 2009) to characterise partial meet contraction operators for various description logics which are non-classical. This is a form of contraction that applies to sets of axioms and not to concepts.

Several other works study change of concepts in various contexts. Colucci et. al (Colucci et al. 2003) study concept contraction based on satisfiability rather than subsumption: given two concepts C and D such that $C \sqcap D$ is unsatisfiable, they consider the problem of finding a generalisation C' of C so that $C' \sqcap D$ is satisfiable. They consider a specific method to find such a C' that is somehow as similar as possible to C but they do not consider general classes of contraction operators or postulates satisfied by this operation.

Teege (Teege 1994) studies a *difference* operator between concepts. Formally, the difference between C and D (assuming $C \sqsubseteq D$) is defined to be the set $\max_{\sqsubseteq}(\{C' \in \mathcal{L} \mid D \sqcap C' \equiv C\})$, where \mathcal{L} consists of all concepts expressible in a given description logic. It is interesting to note that in \mathcal{EL} this operation may fail to produce a reasonable answer, and the reason is related to the failure of the Recovery postulate that we discussed earlier. More precisely, if we set $C = \exists R.(X \sqcap Y)$ and $D = \exists R.X$ then there is no concept that, intersected with D , is equivalent to C , other than C itself.

In the context of what they call *combinatorial creativity*, Suchanek et al. (Suchanek et al. 2016) study the problem of combining existing concepts in order to produce new concepts. They model addition, removal and modification of (parts of) concepts, focusing on the description logic \mathcal{EL} . While their removal operator superficially resembles our contraction operator, it is different in some crucial aspects. First of all, removal of D from C in their model is a syntactic manipulation of C that depends on a given ordering of the conjuncts of the normal form of C . Secondly, removal does not satisfy Success, since according to their definition, removal of D from C removes *one* conjunct of C that is subsumed by D , but not all. They note that, if Success is desired, then D can be removed repeatedly, until all conjuncts of C subsumed by D are removed.

Wassermann (Wassermann 1998) and Wassermann and Fermé (Wassermann and Fermé 1999) consider concept revision based on Grove’s system of spheres. They apply, like we do, methods from belief revision to the problem of revising concepts (rather than belief sets or knowledge bases). Their model is based on an abstract description language that is quite different from description logics. For example, the language contains no quantification symbols. This makes it hard to compare our approach to theirs.

Finally, the AGM approach to belief change is not only about contraction but also about revision. However, in our setting, revision is a trivial operation. To see why, consider the so called *Levi identity*, which states how a revision operator $*$ may be defined in terms of a contraction operator \ominus (Alchourrón et al. 1985). In our setting we may formulate it as follows: $C * D = (C \ominus \neg D) \sqcap D$, where $\neg D$ is the negation or complement of D . The problem here is

that in \mathcal{EL} we cannot express the complement of a concept. Moreover, every \mathcal{EL} concept is satisfiable, which means that revision of C by D can be achieved simply by intersecting C and D (this could be called expansion of C by D), since this always yields a satisfiable concept. However, none of this affects concept contraction, which is still a non-trivial and interesting operation in \mathcal{EL} .

9 Conclusion and Future Work

We have shown that change of concepts in description logic can be modeled as a form of contraction very similar to AGM contraction. This is different from many existing approaches, which focus on change of T-Box or A-Box. We defined the notion of LCS concept contraction and characterised it with a set of postulates, which are reformulations of the well-known AGM postulates for contraction. However, the Recovery postulate does not apply in our setting and had to be replaced with the Strong or Relevance postulates.

There are several directions for future work. For example one may require, like in the traditional AGM approach (Alchourrón *et al.* 1985) that selection functions are based preference orderings, which can be characterised with extra postulates. In our context an interesting related matter is the representation of such preferences, for example by annotating concepts with strength values. Another direction would be to derive possible selection functions from metaproperties from the OntoClean methodology (Guarino and Welty 2009). Another open question is the complexity of contraction in \mathcal{EL} , which may be tractable since the main inference task in \mathcal{EL} are also tractable. Finally, there is the question of whether our approach can be applied to more expressive description logics, which is a matter that we plan to address in future work. As discussed in the previous section, more expressivity may also mean that concept revision becomes a non-trivial problem, and that concept revision and contraction can perhaps be considered as dual operations, connected by the Levi identity.

References

- Carlos E Alchourrón, Peter Gärdenfors, and David Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic*, 50(2):510 – 530, 1985.
- Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25(1):25, 2000.
- Franz Baader and Barbara Morawska. Unification in the description logic EL. *Logical Methods in Computer Science*, 6(3), 2010.
- Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 96–103, 1999.
- Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the \mathcal{EL} Envelope. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, Scotland, UK*, pages 364 – 369, 2005.
- Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. Making repairs in description logics more gentle. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.*, pages 319–328, 2018.
- Franz Baader. Computing the least common subsumer in the description logic EL w.r.t. terminological cycles with descriptive semantics. In *Proceedings of the 11th International Conference on Conceptual Structures, ICCS 2003 Dresden, Germany, July 21-25, 2003*, pages 117–130, 2003.
- Sebastian Brandt. Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and - What Else? In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'2004), Valencia, Spain*, pages 298 – 302, 2004.
- Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Marina Mongiello. Concept Abduction and Contraction in Description Logics. In *Proceedings of the 16th International Workshop on Description Logics (DL'2003), Rome, Italy*, 2003.
- Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi, and Riccardo Rosati. On the update of description logic ontologies at the instance level. In *Proceedings, The 21st National Conference on AI, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1271–1276, 2006.
- Kevin Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279, 2006.
- Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. On applying the AGM theory to dls and OWL. In *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, pages 216–231, 2005.
- Giorgos Flouris, Dimitris Manakanatas, Haridimos Kondylakis, Dimitris Plexousakis, and Grigoris Antoniou. Ontology Change: Classification and Survey. *The Knowledge Engineering Review*, 23(2):117 – 152, 2008.
- Nicola Guarino and Christopher A. Welty. An overview of ontoclean. In *Handbook on Ontologies*, International Handbooks on Information Systems, pages 201–220. Springer, 2009.
- Christoph Haase and Carsten Lutz. Complexity of subsumption in the \mathcal{EL} family of description logics: Acyclic and cyclic tboxes. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikolaos M. Avouris, editors, *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 25–29. IOS Press, 2008.
- Sven Ove Hansson. Belief Contraction Without Recovery. *Studia Logica*, 50(2):251 – 260, 1991.

Hongkai Liu, Carsten Lutz, Maja Milicic, and Frank Wolter. Updating description logic aboxes. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 46–56, 2006.

Bernhard Nebel. Terminological Reasoning is Inherently Intractable. *Artificial Intelligence*, 43(2):235 – 249, 1990.

Guilin Qi and Jianfeng Du. Model-based revision operators for terminologies in description logics. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 891–897, 2009.

Guilin Qi, Weiru Liu, and David A. Bell. Knowledge base revision in description logics. In *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006, Proceedings*, pages 386–398, 2006.

Alan Rector and Ian Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97), Stanford, CA*, pages 321–325, 1997.

Marcio Ribeiro, Renata Wassermann, Grigoris Antoniou, Giorgos Flouris, and Jeff Pan. Belief contraction in web-ontology languages. In *Proceedings of the 3rd International Workshop on Ontology Dynamics (IWOD-09), Short Paper*, 2009.

Fabian M Suchanek, Colette Menard, Meghyn Bienvenu, and Cyril Chapellier. Can You Imagine... A Language for Combinatorial Creativity? In Paul T. Groth, Elena Simperl, Alasdair J. G. Gray, Marta Sabou, Markus Krötzsch, Freddy Lécué, Fabian Flöck, and Yolanda Gil, editors, *Proceedings of the 15th International Semantic Web Conference (ISWC'2016), Kōbe, Japan*, volume 9981 of LNCS, pages 532 – 548, Cham, Switzerland, 2016. Springer International Publishing.

Gunnar Teege. Making the Difference: A Subtraction Operation for Description Logics. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR'94), Bonn, Germany*, pages 540 – 550, 1994.

The OWL Working Group. OWL 2 Web Ontology Language Profiles (Second Edition). *W3C Recommendation*, World Wide Web Consortium (W3C), 2012. [Online]. Available: <https://www.w3.org/TR/owl2-profiles>, accessed: 2018-01-02.

Renata Wassermann and Eduardo Fermé. *A note on prototype revision*. Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1999.

Renata Wassermann. Revising concepts. In *Proceedings of the 5th Workshop on Logic, Language, Information and Communication (WoLLIC-98)*, 1998.

Dmitriy Zheleznyakov, Evgeny Kharlamov, Werner Nutt, and Diego Calvanese. On expansion and contraction of dl-lite knowledge bases. *Journal of Web Semantics*, 2019.