# A Study of Kernel Contraction in $\mathcal{EL}$

**Amr Dawood** and **James Delgrande** and **Zhiwei Liao**
Simon Fraser University
Burnaby, B.C., Canada
{jim,amr_dawood}@cs.sfu.ca

## Abstract

An intelligent agent must be able to change its beliefs in a rational manner and with a plausible outcome. While the central belief change theories have well-developed formal characterisations, there has been limited work on applying these theories in a practical setting. In this paper we examine belief base contraction, specifically kernel contraction, in the framework of the description logic $\mathcal{EL}$. This is potentially interesting first, because $\mathcal{EL}$ is used in large-scale knowledge bases and second, because the $\mathcal{EL}$ constructs for structuring knowledge allow the specification of heuristics to choose an intuitive, commonsense result for the contraction. We introduce algorithms that perform belief contraction based on hitting sets, and examine different ways of finding a suitable hitting set. Heuristics based on localization and specificity are introduced. As well, our main contraction operators are shown to conform to Hansson's postulates for smooth base contraction.

## Introduction

An agent interacting with some domain will be required to modify its beliefs, perhaps as a result of receiving additional information about the domain, or perhaps in correcting an inaccurate belief. The dominant approach to belief change with respect to a static domain[1] is the *AGM approach*. Two primary belief change operators are considered, *belief revision* in which the agent must consistently (when possible) incorporate a new belief into its belief corpus, and *belief contraction*, in which an agent ceased to believes a given formula.

In this paper we focus on belief contraction: It is seen as the more basic operation; it is applicable even when the underlying system lacks a notion of inconsistency; and as covered in the Background section, revision may be defined in terms of it.

Our goal is to address belief contraction in a Description Logic called $\mathcal{EL}$. A Description Logic (DL) is one of a family of formalisms that are used to represent knowledge. Knowledge is *structured*, in that *concepts* are used to represent classes of individuals and *roles* are used to represent relationships between concepts. Different DLs have different expressive powers and different reasoners with different complexities. $\mathcal{EL}$ is one of the simplest DLs, yet is expressive enough to have found significant application. Consequently it is of interest to consider change within an $\mathcal{EL}$ knowledge base. A further point of interest in addressing change within a DL is that, because of the constructs for structuring knowledge, heuristics can be introduced to select the most "reasonable", or perhaps commonsense, contraction from among a set of candidate contractions.

A key feature of $\mathcal{EL}$ is that an $\mathcal{EL}$ knowledge base (KB) can be translated in polynomial time into a normal form of "small" independent assertions. Then contracting a formula $\phi$ from a KB can be most easily accomplished by, first determining all minimal sets of assertions (called *kernels*) that imply $\phi$ and then removing an element of each kernel from the KB. This means of contraction is called *kernel contraction*, and was first studied by Hansson (1994).

The problem of determining kernels has been previously addressed (as described in the next section), so in this paper we focus on the second part, determining which formulas to remove from each kernel. We look at some basic implementations as well as more advanced ones. Our focus is on investigating the algorithmic aspects of these different approaches. Algorithmic time complexity is taken into consideration as well as the overall suitability of the solutions found. "Suitability" can be measured in different ways, including optimality, but also in terms of what makes more sense as a "reasonable" solution.

We begin by examining simplistic algorithms for removing formulas from a kernel, and move on to examining things from the point of view of the hitting set problem, where the hitting set problem is simply the problem of finding the smallest set of elements that intersect with a given set of sets of elements. However, approaches that only consider the size of the solution do not always make a meaningful choice on which beliefs to remove. A contributions of this study is the subsequent investigation of algorithms that perform kernel contraction based on the structure of a $\mathcal{EL}$ knowledge base. These approaches exploit the structure of $\mathcal{EL}$ knowledge bases to find most reasonable solutions. A first approach is to select beliefs that are somehow related to each other, rather than selecting the beliefs that are not. This idea motivates the heuristic that we call *localization*. A second



[1]The alternative, in which the domain may have changed, is *belief update*, which we do not consider here.

heuristic we introduce is *specificity*, which is based on the idea that removing beliefs about more specific matters is more reasonable than removing beliefs about more general ones. So we can then use these heuristics as a tie-breaker step when we get more than one solution.

The next section covers background material, including the prominent approaches to belief change, and a brief introduction to description logic, focusing on $\mathcal{EL}$. The following section covers basic approaches to belief contraction in $\mathcal{EL}$ while the next section discusses commonsense heuristics, specifically dealing with localization and specificity. Further details and full descriptions of algorithms may be found in (Dawood 2017).

## Background

In this section, we provide some background on the main building blocks of our work. We discuss belief change first with respect to the AGM framework (Alchourrón, Gärdenfors, and Makinson 1985) and second according to Hansson (1999). Following this, Description Logics (DLs) are briefly introduced, focusing on the DL $\mathcal{EL}$.

### Belief Change

**AGM Belief Change** The AGM framework (Alchourrón, Gärdenfors, and Makinson 1985), named after its founders, is the most prominent framework for belief change. Belief change is described on an abstract level, independent of how beliefs are represented and manipulated. Belief states are modeled by sets of sentences, called *belief sets*, closed under the logical consequence operator of a logic that includes classical propositional logic in a language $\mathcal{L}$. Thus a belief set $K$ satisfies the constraint:

$$\text{If } K \text{ logically entails } \phi \text{ then } \phi \in K.$$

Three types of belief change are identified: *expansion*, *revision*, and *contraction*. For expansion, a new belief is simply added to a belief set and the deductive closure taken. That is, for belief set $K$ and formula $\phi$, the *expansion* of $K$ by $\phi$ is defined by

$$K + \phi \doteq Cn(K \cup \{\phi\})$$

where $Cn(\cdot)$ is the deductive closure of its argument. Expansion can be reasonably applied when new information is consistent with a belief set; it is also useful in specifying the postulates for revision and contraction, and in their relation.

*Revision* represents the situation in which new information may be inconsistent with the reasoner's beliefs, and needs to be incorporated in a consistent manner where possible. Consequently, if a new formula is inconsistent with respect to an agent's beliefs, some beliefs have to be dropped before the formula can be consistently incorporated.

*Contraction* represents the situation in which the reasoner loses information. Informally, the contraction of a belief set by a formula is another belief set in which that formula is not believed. We denote the belief set resulting from contracting the belief set $K$ with respect to sentence $\phi$ by $K - \phi$. Since contraction is the subject of this study, we discuss the rationality criteria (or postulates) of such change. In the AGM framework, the following eight postulates are introduced as a basis for ensuring the rationality of contraction functions (more details can be found in (Peppas 2008) and (Gärdenfors 1988)):

**(K-1)** $K - \phi$ is a belief set.

**(K-2)** $K - \phi \subseteq K$.

**(K-3)** If $\phi \notin K$, then $K - \phi = K$.

**(K-4)** If not $\vdash \phi$, then $\phi \notin K - \phi$.

**(K-5)** If $\phi \in K$, then $K \subseteq (K - \phi) + \phi$.

**(K-6)** If $\vdash \phi \leftrightarrow \psi$, then $K - \phi = K - \psi$.

**(K-7)** $K - \phi \cap K - \psi \subseteq K - \phi \wedge \psi$.

**(K-8)** If $\phi \notin K - (\phi \wedge \psi)$, then $K - (\phi \wedge \psi) \subseteq K - \phi$.

The fifth postulate, the so-called *recovery postulate*, is controversial; it asserts that following a contraction, all beliefs are recoverable by an expansion by the formula for contraction.

In (Alchourrón, Gärdenfors, and Makinson 1985; Gärdenfors 1988), various constructions, including *partial meet* and *epistemic entrenchment* are developed, and shown to exactly capture the set of functions that conform to the contraction postulates; we omit details, as they are not directly pertinent here.

Both revision and contraction are formulated to follow a principle of *informational economy*. In the case of contraction, this informally means not giving up a formula if there is no need to do so. Compared to revision, contraction is usually seen as a more basic change function, since a belief set can only decrease. However, revision can be regarded as a combination of contraction and expansion, via the Levi Identity:

$$K * \phi = (K - \neg\phi) + \phi.$$

**Belief Bases** The AGM approach presents belief change at a high level, independent of how a belief is represented, and where the agent's beliefs are described by a deductively-closed *belief set*. Hansson (1999) presents an approach to belief change where the agent's beliefs are represented by an arbitrary set of formulas, called a *belief base*. The intuition is that a belief base may be "closer" to how an agent's beliefs may be represented in a finite computer.

Two constructions for a belief base contraction $B - \phi$ are described:

1. Remainders: Compute all maximal subsets of $B$ that do not entail $\phi$. The contraction is defined as the intersection of some "select" set of these subsets.
2. Kernels: Compute all minimal subsets of $B$ that entail $\phi$. Remove (in some fashion) an element of each such set to obtain a belief base that does not entail $\phi$.

Here, we consider kernel contraction on belief bases. The function in a kernel contraction that removes an element of each subset is called an *incision function*.

For example, consider the following set where we wish to remove the element $b$:

$$\{a, a \rightarrow b, b, c\}$$

It is not enough to simply remove $b$ from the set; if we do so, we get: $\{a, a \rightarrow b, c\}$ which still implies $b$. In this example, there are two kernels, $\{b\}$ and $\{a, a \rightarrow b\}$. An incision function must remove $\{b\}$ and one of $a$, $a \rightarrow b$. The two possible resulting belief bases after contraction are:

$$\{a, c\} \text{ and } \{a \rightarrow b, c\}$$

In the following, $B$ will refer to an arbitrary belief base. We refer to the set of $\phi$-kernels of $B$ by $B \perp \phi$.

**Definition 1** *(Kernel Set (Hansson 1999)) For a belief base $B$ and a formula $\phi$, $B \perp \phi$ is a set such that $X \in B \perp \phi$ if and only if:*

1. $X \subseteq B$,
2. $X \vdash \phi$, *and*
3. *if* $Y \subset X$ *then* $Y \nvdash \phi$.

*$B \perp \phi$ is a kernel set that contains all $\phi$-kernels of $B$.*

Kernels are the smallest subsets of the knowledge base that imply a specific belief. Therefore, in order to give up that belief, every one of those kernels needs to be incised. If the kernels are not minimal, they might contain a belief that does not contribute to the implication of the belief to be contracted. For example, in $\{p, q\}$, $q$ is a belief that does not contribute to make the set imply $p$. In this case, removing $q$ only would not help in giving up $p$.

An incision function is defined in as follows:

**Definition 2** *(Incision function (Hansson 1999)) An incision function $\sigma$ is a function such that for all beliefs $\phi$:*

1. $\sigma(B \perp \phi) \subseteq \bigcup(B \perp \phi)$
2. *If* $\emptyset \neq X \in B \perp \phi$, *then* $X \cap \sigma(B \perp \phi) \neq \emptyset$

Contraction is carried out by removing the beliefs that are selected by the incision function from the original knowledge base. Contraction using the incision function $\sigma$[2] can be defined as:

**Definition 3** *(Hansson 1999) (Contraction)*

$$B - \phi = B \smallsetminus \sigma(B \perp \phi)$$

Hansson provides the following postulates that are shown to exactly capture belief base contraction based on incision functions:

**(B-1)** If $\nvdash \phi$ then $B - \phi \nvdash \phi$ (Success)

**(B-2)** $B - \phi \subseteq B$ (Inclusion)

**(B-3)** If $\psi \in B$ and $\psi \notin B - \phi$ then there is a set $B' \subseteq B$ such that $B' \nvdash \phi$ but $B' \cup \psi \vdash \phi$ (Core retainment)

**(B-4)** If for every $B' \subseteq B$ we have $B' \vdash \phi$ iff $B' \vdash \psi$, then $B - \phi = B - \psi$. (Uniformity)

As well, Hansson provides the following postulate which, along with the preceding, characterises *smooth kernel contraction*:

**(B-5)** $B \cap Cn(B - \phi) \subseteq B - \phi$. (Relative closure)

[2]Technically the contraction function $-$ should be parameterised by $\sigma$. For simplicity and because no confusion arises, we omit it.

See (Hansson 1999) for details.

Although the AGM framework is very popular, we find the postulates introduced by Hansson more suitable to be applied to kernel contraction. First, in our approach we work with a normalized form of an $\mathcal{EL}$ knowledge base, which is to say, a belief base of formulas. Second, as we show at the end of this section, contraction of $\mathcal{EL}$ belief bases do not satisfy the recovery postulate of the AGM framework.

## Description Logic

A *Description Logic* (DL) is one of a class of logics based on describing sets of individuals using *concepts* and describing the relationships between them using *roles*. Two types of knowledge are represented: *Intensional knowledge* is general knowledge about a problem or a domain, while *extensional knowledge* is knowledge about a specific problem instance. For this purpose, a knowledge base[3] is composed of two corresponding components: A *TBox* containing the intensional knowledge and an *ABox* containing the extensional knowledge.

The ABox gives assertions, much like in a relational database, such as *Female(Sally)* and *FatherOf(Adam, Sally)*. Since our focus is on contraction in TBoxes, we don't consider the ABox further. A TBox is composed of a set of assertions, made up of equivalences such as Man $\doteq$ Human $\sqcap$ Male (the individuals satisfying Man are exactly those satisfying Human and Male) and subsumptions such as Man $\sqsubseteq$ Human (every individual that is a Man is a Human). The latter are referred to as General Concept Inclusions (GCIs). Every equivalence can be trivially broken down into two GCIs.

There are many members of the DL family; they vary widely in expressivity power and the complexity of their reasoning algorithms. In the following section we discuss a well-known member of the DL family, $\mathcal{EL}$, which we will use the representation language for the rest of this study.

**The $\mathcal{EL}$ Description Logic** $\mathcal{EL}$ is a description logic that has recently attracted much attention. It is a "light-weight" DL, in that it is limited expressively, although it is used in well-known ontologies such as SNOMED CT (Spackman 2000), a large-scale commercial medical ontology with well over 300000 concepts (Baader 2011). $\mathcal{EL}$ contains the following concept constructors:

- Conjunction ($\sqcap$):
  If $C_1$ and $C_2$ are concepts, then $C_1 \sqcap C_2$ is a concept standing for the conjunction of $C_1, C_2$.
- Existential restriction ($\exists$):
  If $C$ is a concept and $R$ is a role, then $\exists R.C$ is the concept of those individuals that are $R$-related to an item belonging to concept $C$.
- The top concept $\top$, true of all individuals.

Assertions express either equivalences between concepts or a subsumption relation. The following are example assertions in $\mathcal{EL}$.

[3]By *knowledge base* we refer to a belief base – a set of sentences that represent the beliefs of an agent.

1. Woman $\doteq$ Human $\sqcap$ Female.
   A Woman is exactly a Human and a Female.
2. Mother $\doteq$ Female $\sqcap \exists$ParentOf.Human.
   A (human) Mother is a Female and a ParentOf of an individual that is a Human.

$\mathcal{EL}$ is simple and easy to use; as well it has a polynomial-time subsumption algorithm, where the subsumption algorithm determines whether a specific subsumption relation holds or not in a given knowledge base.

The subsumption algorithm is made up of four steps:

1. *Normalize* the TBox.
2. *Translate* the normalized TBox into a graph.
3. *Complete* the graph using completion rules.
4. *Read off* the subsumption relationships from the normalized graph.

We are most interested in the first step, normalization. It is shown (Baader et al. 2007) that any $\mathcal{EL}$ TBox can be translated in polynomial time into a set of GCIs of the following form:

- $A \sqsubseteq B$
- $A_1 \sqcap A_2 \sqsubseteq B$
- $A \sqsubseteq \exists r.B$
- $\exists r.A \sqsubseteq B$

$A$, $A_1$, $A_2$, and $B$ are concept names, and $r$ is a role name. A normalized TBox can be thought of as a knowledge base that has been translated into a canonical form, where each GCI represents an individual item of information. For our contraction functions, we will work with a normalized TBox. Such TBoxes can be thought of as corresponding to a (non-closed) equivalent of a belief set.

## Related work

There has been some related work regarding contraction in light-weight description logics. In (Booth, Meyer, and Varzinczak 2009a) a contraction operator is introduced that contracts $\mathcal{EL}$ belief sets (rather than belief bases) using remainders. A representation result is given relating the first six contraction postulates, adapted to $\mathcal{EL}$, to the construction of *infra-remainder sets* (Booth, Meyer, and Varzinczak 2009b).

Zhuang and Pagnucco (2009) also address belief contraction in $\mathcal{EL}$. They point out that the recovery postulate doesn't hold in $\mathcal{EL}$. Informally the problem, as in other inferentially-weak approaches (see e.g. (Delgrande 2008) regarding Horn clause theories), is that $\mathcal{EL}$ doesn't support negation or disjunction. For a problematic example, consider where the TBox contains just the closure of $A \sqsubseteq B$ and one wishes to contract by $A \sqcap C \sqsubseteq B$; in the contraction, $A \sqsubseteq B$ must be removed, but it is not "recovered" if one expands the result by $A \sqcap C \sqsubseteq B$. The authors propose a suitable alternative to the recovery postulate based on the notion of logical difference, and develop algorithms that implement this approach.

A model-based approach to contraction of DL-Lite belief bases is introduced in (Zhuang et al. 2016). The approach depends on defining an alternative semantics, called *type semantics*, that skirts the problem that a knowledge base with a finite signature may have an infinite number of models.

# Kernel Contraction

For the kernel contraction of a formula $\phi$ from a belief base $B$ there are three steps:

1. Determine the kernels $\mathcal{K}$ of $B$ with respect to $\phi$.
2. Determine a set of incised formulas $\mathcal{I}$ from $\mathcal{K}$.
3. Set $B \leftarrow B \setminus \mathcal{I}$.

The third step is trivial. As well, the first step has been addressed, in the *pinpointing algorithm* introduced in (Baader, Peñaloza, and Suntisrivaraporn 2007).

The pinpointing algorithm computes all kernels using a modified version of the $\mathcal{EL}$ subsumption algorithm. It finds a monotone Boolean formula called a *pinpointing formula*; the propositions of the pinpointing formula are GCIs of the TBox, and a propositional *valuation* represents the kernels. The approach may take exponential time (w.r.t the size of the TBox) to find all kernels, because there may be exponentially many kernels. However, (Baader, Peñaloza, and Suntisrivaraporn 2007) also gives a polynomial-time algorithm that computes a single kernel.

**Example 1** *((Baader, Peñaloza, and Suntisrivaraporn 2007)) Let the TBox, $\mathcal{T}$, be:*

$$ax_1 : A \sqsubseteq \exists r.A, \quad ax_2 : A \sqsubseteq Y,$$
$$ax_3 : \exists r.Y \sqsubseteq B, \quad ax_4 : Y \sqsubseteq B.$$

*We have that $\mathcal{T} \models A \sqsubseteq B$. Let $\phi = A \sqsubseteq B$. We obtain:*

$$\mathcal{T} \perp (A \sqsubseteq B) = \{\{ax_2, ax_4\}, \{ax_1, ax_2, ax_3\}\}$$

Since we can use the pinpointing algorithm to get the set of all kernels, we just need to consider the issue of removing an element from each kernel to perform a contraction. This is the role of the incision function. We next look at some basic incision functions to set the stage for later considerations.

## Incision Functions

We are given an $\mathcal{EL}$ TBox, $\mathcal{T}$, and a formula for contraction $\phi$. We assume without loss of generality that $\mathcal{T}$ has been normalized; this has the important consequence that for $\psi \in \mathcal{T}$ that $\mathcal{T} \setminus \{\psi\} \not\models \psi$.

The main contraction algorithm is shown in Algorithm 1; as discussed, the only issue is specifying the incision function CUT.

**Algorithm 1** Contraction algorithm

```
1: procedure CONTRACT(T, A)
2:     kernelset = PINPOINT(T, A)
3:     giveUpSet = CUT(kernelset)
4:     T = T / giveUpSet
5: end procedure
```

We next consider simple ways of implementing CUT. Due to space considerations, here and elsewhere we omit the pseudocode; see (Dawood 2017; Liao 2014). We first consider non-optimal solutions, followed by optimal solutions.

**Basic Incision Functions** The first incision function is overly simplistic, but serves to provide a base case. In this case, one simply removes a formula from each kernel. The time complexity is $O(m)$, where $m$ is the number of kernels (size of the kernelset), assuming that a formula selection takes constant time. However, this is clearly a poor algorithm. For example, given a kernel set $\{\{a, c\}, \{b, c\}\}$, one might remove $a$ from the first set followed by $c$ from the second set. Then the result of $giveUpSet = \{a, c\}$ is $\{b\}$, which unnecessarily removes $a$. Removing $a$ is unnecessary because removing $c$ alone is enough for contraction. More generally, removing more beliefs than needed is undesired, as it violates the informational economy principle.

The second incision function improves on the previous by selecting a formula from every kernel, as before. However each time a formula is selected, every kernel that contains that formula is excluded from consideration in the following iterations. This algorithm has a worst-case time complexity of $O(m^2)$: There are basically two loops, first scanning through the kernels, and second, once a formula has been selected, scanning the remaining kernels to see whether they contain that formula.[4]

The third function implements a greedy approach: At each step, the algorithm finds the formula that appears in the most kernels and removes it; these kernels are not considered in the following steps. An sketch of the implementation in (Dawood 2017) is given as follows. First, for every formula in a kernel, the number of kernels in which that formula occurs is determined. Then the formula that has the greatest number of occurrences is added to the set of incised formulas; the kernels containing that formula are removed from consideration; the count of formula occurrences in kernels is updated and the process continued. If the number of formulas in kernels is $n$, the number of kernels is $m$, and the size of the largest kernel is $k$, then the time complexity is $O(m^2 \cdot k \cdot n)$. This however could be significantly enhanced by using more clever data structures.

**Hitting Set Approach** According to the principal of information economy (Gärdenfors 1988), in every change to an agent's beliefs, loss of information should be minimized. To satisfy the requirement of minimum change we require an algorithm that removes the fewest GCIs while hitting all the kernels. This however is exactly the *minimal hitting set problem* (Garey and Johnson 1979), which has already received extensive study.

Consequently, another approach to cutting kernels in order to perform contraction is the minimal hitting set algorithm. The minimal hitting set problem is defined as follows (Abreu and van Gemund 2009):

**Definition 4** *Given a set $S = \{s_1, s_2, ..., s_n\}$ of $n$ non-empty sets, a minimal hitting set $d$ is a set such that:*

$$\forall_{s_i \in S}[s_i \cap d \neq \emptyset] \wedge \nexists_{d' \subset d}[\forall_{s_i \in S}(s_i \cap d' \neq \emptyset)]$$

Thus, $d$ is a minimal hitting set if and only if it contains at least one element from every set, and no proper subset of $d$ is a hitting set. In the context of kernel contraction in $\mathcal{EL}$, we can define the minimal hitting set contraction as:

**Definition 5** *(Minimal hitting set contraction) Given a kernel set $K = \{k_1, k_2, ..., k_n\}$ of $n$ kernels, a minimal hitting set giveUpSet is a set such that:*

$$\forall_{k_i \in K}[k_i \cap giveUpSet \neq \emptyset] \quad \wedge$$

$$\nexists_{giveUpSet' \subset giveUpSet}[\forall_{k_i \in K}(k_i \cap giveUpSet' \neq \emptyset)]$$

Since kernels are subsets of the TBox, and our goal to find a $giveUpSet$ that hits all kernels, we can use approaches to the minimal hitting set problem to solve it. The computation of all such hitting sets is known to be NP-hard (Garey and Johnson 1979); there are however known very good approximation algorithms, such as the one introduced in (Abreu and van Gemund 2009), for the minimal hitting set problem, that are feasible in terms of running time.

The following result shows that we obtain a well-behaved contraction function:

**Theorem 1** *Let $\mathcal{T}$ be a normalized TBox, $\phi$ a (normalized) set of GCIs, and with kernel set $K$ and set $giveUpSet$ determined as in Definition 5. Then for $-$ defined by: $\mathcal{T} - \phi = \mathcal{T} \setminus giveUpSet$, we have that $\mathcal{T} - \phi$ satisfies the postulates for smooth kernel contraction (B-1)-(B-5).*

## Heuristics for Contraction

In the last section, we discussed contraction algorithms where the goal was to remove the least number of beliefs. Here we consider heuristics based on the $\mathcal{EL}$ language and subsumption hierarchy. The idea is to bring in commonsense principles that may be expected to yield intuitive or "reasonable" results for a contraction. With *localization* we choose a hitting set that in some fashion affects a minimal region of the TBox; with *specificity* we prefer hitting sets that refer to more specific concepts.

### Localization

The idea here is that change should be as local as possible in a knowledge base. Intuitively, this will involve a lesser overall change to a KB; as well, pragmatically, it would seem that if a belief is incorrect, then it may be expected to arise from a specific part of the KB (much like in diagnosis where pragmatically one would prefer a diagnosis with fewer faults).

Consider following $\mathcal{EL}$ example:

(1) $A \sqsubseteq B$, (2) $B \sqsubseteq D$, (3) $A \sqsubseteq C$, (4) $C \sqsubseteq D$.

These imply that (5) $A \sqsubseteq D$. We see that there are two (5)-kernels: $\{1, 2\}$ and $\{3, 4\}$

So, to contract the knowledge base by (5), we need to remove a belief from each kernel. However, removing (1) and (4) seems unreasonable because it means removing information concerning all four concepts. Removing (1) and (3), on the other hand, removes knowledge concerning only three concepts: $A$, $B$, and $C$. Not only that, but pragmatically, (1) and (3) together imply that perhaps our knowledge of $A$ is wrong; conceivably our knowledge of $B$ and $C$ is not necessarily false. Similar considerations apply to removing (2) and (4), which together would suggest that the source of incorrectness lies with $D$.

[4] This assumes that looking up a formula in a kernel takes constant time, for example if kernels are stored as hash tables.

Localization is implemented in (Liao 2014) using a graph-based approach. The algorithm operates by defining a graph based on a set of GCIs, where an edge connects GCIs sharing a concept or role. Then notions involving graph connectivity and graph clustering can be used to determine localized change.

**Definition 6** *(Connected GCIs) Given a set of GCIs $\mathcal{K}$, define an undirected graph $G = (V, E)$ by: $V$ is the set of GCIs in $\mathcal{K}$ and for $g_1, g_2 \in \mathcal{K}$, $(g_1, g_2) \in E$ just if $g_1$ and $g_2$ share a concept or role name.*

So, for a normalized TBox, given that the set of kernels has been determined and a set of minimal hitting sets $H = \{h_1, \ldots, h_n\}$ has also been found, *localization* can be used to select the preferred hitting set(s), based on the structure of each hitting set:

1. For each $h_i \in H$, determine the underlying graph (Definition 6), and determine the number of strongly connected components, $c_i$, in the graph.
2. Select a hitting set $h_i$ whose number of strongly connected components, $c_i$, is minimum.

In the previous example, we find, the following candidate sets to contract (5):

$$\{1, 3\} \quad \text{and} \quad \{2, 4\}.$$

Thus contraction is given by one of these alternatives, and not the inferior possibilities $\{1, 4\}$ or $\{2, 3\}$.

For time complexity, each hitting set must be considered; and for a hitting set the number of strongly connected components determined. This latter problem can be solved via Tarjan's (1972) algorithm, which is $O(|V|+|E|)$ for a graph $G = (V, E)$. Another pass through the hitting sets finds the minimum, for overall time complexity $O(m \cdot (|V| + |E|))$. Since all we are doing is selecting a particular hitting set, as given in Definition 5 the resulting contraction operator is a smooth kernel contraction, as given in Theorem 1.

Clearly this procedure can be enhanced or modified. A more sophisticated notion of localization could be employed, for example, or the localization heuristic could be applied just to the hitting sets of minimum cardinality (and so help serve as a tie-breaker).

## Specificity

The assumption underlying this heuristic is that in choosing which GCIs to remove, it is preferable to remove those involving more specific concepts over those involving less specific concepts: Arguably more general concepts are more entrenched; as well, removing a more general concept subsumption may be expected to be more disruptive to the TBox as a whole.

The subsumption hierarchy implicit in a normalized $\mathcal{EL}$ TBox categorizes concepts into different levels of generality. Consider the relations:

$$A \sqsubseteq B,\ B \sqsubseteq C,\ C \sqsubseteq D\,.$$

Suppose we would like to contract the TBox by the implied relation $A \sqsubseteq D$. We have three options, corresponding to removing one of the three given subsumptions. Removing any of the three GCIs would guarantee minimum change. However, we can say that $A$ is more specific than $B$ and $B$ is more specific than $C$. Removing $A \sqsubseteq B$, in this case, is more reasonable than removing $C \sqsubseteq D$, because $A \sqsubseteq B$ involves concepts that are more specific than the ones involved in $C \sqsubseteq D$. This also means that removing $C \sqsubseteq D$ results in all individuals that are $Cs$ no longer believed to be $Ds$, while removing $A \sqsubseteq B$ only affects the subset of the $Cs$ which are $As$.

To begin, for a GCI $A \sqsubseteq B$, we say that $A$ is a *child* concept relative to $B$. Similarly, for $A_1 \sqcap A_2 \sqsubseteq B$, we say that $A_1$ and $A_2$ are both *child* concepts relative to $B$. Moreover, we ignore the structure of a concept expression of the form $\exists R.C$, which is to say, we treat a concept expression $\exists R.C$ as if it were a primitive concept name.

The approach assigns weights to every GCI, where a weight reflects its relative generality. Then we select the kernels with minimum overall weight. A GCI $A \sqsubseteq B$ gets weight 0 if $A$ has no children; and a GCI $A \sqsubseteq B$ gets weight $i+1$ if the maximum weight of the GCIs of form $X \sqsubseteq A$ is $i$. Similarly, a GCI $A_1 \sqcap A_2 \sqsubseteq B$ gets weight $i + 1$ if the maximum weight of the GCIs of form $X \sqsubseteq A_j$ for $j \in \{1, 2\}$ is $i$. For simplicity, we assume that the TBox is acyclic – so we avoid the problems that will be caused by loops.

In more detail, let $A$ and $B$ be concept expressions appearing in a normalized TBox.

1. If $A$ has no children, assign $A \sqsubseteq B$ a weight of 0. If $A_1$, $A_2$ have no children, assign $A_1 \sqcap A_2 \sqsubseteq B$ a weight of 0.
2. For GCI $A \sqsubseteq B$ ($A_1 \sqcap A_2 \sqsubseteq B$), assign the GCI a weight of $i + 1$ if the maximum weight of GCIs of the form $X \sqsubseteq A$ (resp. $X \sqsubseteq A_j$ $j \in \{1, 2\}$) is $i$.

Then, given that a set of hitting sets $H = \{h_1, \ldots, h_n\}$ has been found, choose the preferred hitting set $h_i$ where the sum of the weights of the GCIs in $h_i$ is a minimum.

As a simple example, consider the previous example:

- $A \sqsubseteq B$ has weight = 0
- $B \sqsubseteq C$ has weight = 1
- $C \sqsubseteq D$ has weight = 2

In this case, the hitting set consisting of the GCI $A \sqsubseteq B$ is selected.

The time complexity of the underlying algorithm is polynomial. Let $m$ be the size of the normalized TBox $\mathcal{T}$, and $n$ the number of hitting sets. Determining the depth labeling of GCIs in $\mathcal{T}$ takes linear time, and determining the weights of the hitting sets is $O(n \cdot k)$ where $k$ is the maximum size of a hitting set. Hence the overall complexity is $O(m+(n \cdot k))$. As with *localization*, the resulting operator is a smooth kernel contraction.

Again, the basic approach to specificity can be enhanced or modified. For example, one could just look at hitting sets of minimum cardinality, and so the specificity heuristic would be a tie-breaker for equal-sized hitting sets. As well, there are other ways of determining overall specificity of a set of GCIs. One appealing candidate is to choose the hitting set $h_i$ whose maximally-weighted GCI is a minimum. That is: rank the hitting sets by the maximum weight assigned to a contained GCI, and then choose the minimum ranked hitting set in the resulting ranking.

Last, it might be objected that since there may be an exponential number of hitting sets, any approach is bound to be intractable in the worst case. This difficulty can be pragmatically addressed by employing an anytime algorithm for generating an acceptable candidate: simply run through hitting sets, keeping track of the best candidate found so far, until either a "good enough" candidate is found, a time bound is hit, or some other stopping criterion is met.

## Conclusion

In this paper we have studied belief contraction in the description logic $\mathcal{EL}$. The overall goal is to apply the formal theory of belief base change in a practical setting. To this end, kernel contraction seems to most appropriate approach for contraction. We chose to work with $\mathcal{EL}$ first, because it used in large-scale knowledge bases and second, because the $\mathcal{EL}$ constructs for structuring knowledge allow the specification of heuristics to choose intuitive, commonsense outcomes for a contraction. We first examined ways of finding (or approximating) hitting sets. We also proposed two heuristics, based on localization and specificity, for determining suitable hitting sets for contraction. Our main contraction operators are shown to conform to Hansson's postulates for smooth base contraction. All of our algorithms are efficient, being polynomial in (one or both) of the size of the TBox or the number of hitting sets. The bottleneck then lies in the number of hitting sets, which may be exponential in the size of the TBox.

For future work there are two main ways to go. First, the present heuristics can be further explored, elaborated on, and variants developed; as well, these approaches should be implemented to determine what works best in practice. Second, is to explore belief change in a more expressive description logic. To this end, there are enhancements to $\mathcal{EL}$ that preserve the polynomial-time features, but also bring in a notion of consistency (by allowing the concept $\bot$), for which the present approach should apply without change. This would allow a definition of revision in $\mathcal{EL}$, in terms of the Levi Identity. As well, belief change in significantly more expressive DLs, such as in the $\mathcal{ALC}$ family, can also be addressed.

## Acknowledgements

We thank the three reviewers for their very helpful comments. Financial support is gratefully acknowledged from the Natural Sciences and Engineering Research Council of Canada.

## References

Abreu, R., and van Gemund, A. J. C. 2009. A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis. In Bulitko, V., and Beck, J. C., eds., *Eighth Symposium on Abstraction, Reformulation, and Approximation (SARA 2009)*. AAAI Press.

Alchourrón, C.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50(2):510–530.

Baader, F.; Calvanese, D.; McGuiness, D.; Nardi, D.; and Patel-Schneider, P., eds. 2007. *The Description Logic Handbook*. Cambridge: Cambridge University Press, second edition.

Baader, F.; Peñaloza, R.; and Suntisrivaraporn, B. 2007. Pinpointing in the description logic $\mathcal{EL}$. In *Proceedings of the 2007 International Workshop on Description Logics (DL2007)*, volume 250 of *CEUR-WS*, 171–178. Brixen/Bressanone, Italy: Bozen-Bolzano University Press.

Baader, F. 2011. What's new in description logics. *Informatik-Spektrum* 34(5):434–442.

Booth, R.; Meyer, T.; and Varzinczak, I. 2009a. First steps in $\mathcal{EL}$ contraction. In *Workshop on Automated Reasoning about Context and Ontology Evolution (ARCOE-09)*, 16–18.

Booth, R.; Meyer, T.; and Varzinczak, I. 2009b. Next steps in propositional Horn contraction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 702–707.

Dawood, A. 2017. An algorithmic study of kernel contraction in $\mathcal{EL}$. Master's thesis, Simon Fraser University.

Delgrande, J. 2008. Horn clause belief change: Contraction functions. In Brewka, G., and Lang, J., eds., *Proceedings of the Eleventh International Conference on the Principles of Knowledge Representation and Reasoning*, 156–165. Sydney, Australia: AAAI Press.

Gärdenfors, P. 1988. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. Cambridge, MA: The MIT Press.

Garey, M., and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman and Co.

Hansson, S. O. 1994. Kernel contraction. *Journal of Symbolic Logic* 59(3):845–859.

Hansson, S. O. 1999. *A Textbook of Belief Dynamics*. Applied Logic Series. Kluwer Academic Publishers.

Liao, Z. 2014. Kernel contraction in $\mathcal{EL}$. Master's thesis, School of Computing Science, Simon Fraser University.

Peppas, P. 2008. Belief revision. In van Harmelen, F.; Lifschitz, V.; and Porter, B., eds., *Handbook of Knowledge Representation*. Elsevier Science. 317–359.

Spackman, K. 2000. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *J. of the Amer. Med. Inf. Assn.*

Tarjan, R. E. 1972. Depth-first search and linear graph algorithms. *SIAM J. Comput.* 1(2):146–160.

Zhuang, Z., and Pagnucco, M. 2009. Belief contraction in the description logic $\mathcal{EL}$. In *Proceedings of the 22nd International Workshop on Description Logics (DL2009)*, volume 477 of *CEUR-WS*.

Zhuang, Z.; Wang, Z.; Wang, K.; and Qi, G. 2016. DL-Lite contraction and revision. *Journal of Artificial Intelligence Research* 56:328–378.