On the Use of Belief Revision to Merge Description Logic Terminologies

Martín O. Moguillansky Marcelo A. Falappa

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Consejo de Investigaciones Científicas y Técnicas (CONICET)
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
PHONE/FAX: (+54)(291)459-5136
E-MAIL: mom@cs.uns.edu.ar mfalappa@cs.uns.edu.ar

ABSTRACT

Distributed ontologies expressed as description logics may define repeated information. To reason about concepts that these ontologies express, a possible option is to generate unique concept definitions in a different *terminology* or *TBox*. The creation of a new terminology from different ontologies need to be consistent, and expressed with non-monotonic logics to be further updated with new distributed ontologies. The model AGM of theory change seems to be an interesting framework to be studied in conjunction with description logics and generate a new *non-monotonic description logics* model.

1 INTRODUCTION

In order to reason about different ontologies, probably allocated in different places round the web[1], we will consider translated OWL ontologies into description logics (DLs).

In DLs, the concept of Knowledge Base (**KB**) is composed of two main parts, TBoxes or Terminologies and ABoxes or Assertions. In this paper we'll focuss our investigation on how to reason about *terminologies*.

Here many possibilities come through. Just think about two distinct terminologies defining each two different main concepts, but containing a same subset of sub-concepts. Or just two distinct terminologies defining the same main concept, which naturally will define the same subset of sub-concepts.

It might be probably impossible to get two concepts defined by different persons with exactly the same logic intention. Here is where the theory change arises as relevant protagonist in order to join consistently two terminologies redefining or reinforcing sub-concepts.

The next section gives a brief description of the DL formalism, continued by section 3 with the analogous description of the AGM theory change model, section 4 contributes to the formalization of merging DL terminologies, and describes an example operation of two different terminologies, and finally section 5 concludes and explains the future work in the area.

[1]To simplify the complexity of this paper we will consider different ontologies as locals to the host in which the reasoner runs.

2 THE DL BASIC FORMALISM

A Knowledge Representation (KR) system based on Description Logics (DL) provides a formalization to specify the knowledge base (KB) contents, a way to reason about it, and a process to infer implicit knowledge.

A KB is composed by two components. A TBox to manage the *terminology* of the application world and an ABox containing the *assertions* about named individuals in terms of the previous concepts.

A *terminology* is composed by atomic *concepts* which denote sets of individuals and atomic *roles* to manage relationships between individuals. Besides, complex concepts and roles are built from the atomics using given constructors.

Reasoning tasks are dedicated to determine whether a description is satisfiable (*i.e.*, noncontradictory), or whether one description is more general than another one, that is, whether the first subsumes the second.

For an ABox, the problem is to verify the consistence of each set of assertions (*i.e.*, test if there is a model for the set) and find out whether a particular individual is an instance of a concept description in the TBox depending on the assertions in the ABox.

The environment will interact with the KR by querying the KB and finally by adding and retracting concepts, roles and assertions.

2.1 Description Languages

Description Languages are defined by the constructors they provide. In this paper we will consider a subset of the large DL constructors set investigated so far.

The basic Description Language introduced by [Schmidt-Schauß and Smolka, 1991] is the $\mathcal{AL}$ (Attribute Language). Let A be an atomic-concept, R an atomic-role, and C, D complex concepts, the grammar for the $\mathcal{AL}$ language is defined as follows,

$$C, D \rightarrow A|\top|\bot|\neg A|C \sqcap D|\forall R.C|\exists R.\top$$

To define the formal semantics of $\mathcal{AL}$-concepts we use *interpretations* ($\mathfrak{I}$) that consist of a non-empty set $\Delta^{\mathfrak{I}}$ (the domain of the interpretation) and an interpretation function $\imath^{\mathfrak{I}}$, that assigns to every atomic concept A a set $A^{\mathfrak{I}} \subseteq \Delta^{\mathfrak{I}}$ and to every atomic role R a binary relation $R^{\mathfrak{I}} \subseteq \Delta^{\mathfrak{I}} \times \Delta^{\mathfrak{I}}$. The interpretation function $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \imath^{\mathfrak{I}})$ is extended to concept descriptions by the following inductive definitions,

$$\begin{gathered}
\top^{\mathfrak{I}} = \Delta^{\mathfrak{I}} \\
\bot^{\mathfrak{I}} = \emptyset \\
(\neg A)^{\mathfrak{I}} = \Delta^{\mathfrak{I}} \backslash A^{\mathfrak{I}} \\
(C \sqcap D)^{\mathfrak{I}} = C^{\mathfrak{I}} \cap D^{\mathfrak{I}} \\
(\forall R.C)^{\mathfrak{I}} = \{a \in \Delta^{\mathfrak{I}} |\ \forall b.(a,b) \in R^{\mathfrak{I}} \rightarrow b \in C^{\mathfrak{I}}\} \\
(\exists R.\top)^{\mathfrak{I}} = \{a \in \Delta^{\mathfrak{I}} |\ \exists b.(a,b) \in R^{\mathfrak{I}}\}
\end{gathered}$$

Let two concepts C, D be *equivalent*, writing $C \equiv D$, if $C^{\mathfrak{I}} = D^{\mathfrak{I}}$ for all interpretations $\mathfrak{I}$.

Extending $\mathcal{AL}$ by any of the constructors described in Table 1, yields a particular $\mathcal{AL}$-language $\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{N}][\mathcal{C}^2][\mathcal{Q}]^3$.

For example, given a role completedCourse a student is considered advanced if he has approved 15 courses of a total of 25,

$$\geqslant 15\ \mathsf{completedCourse} \sqcap \leqslant 25\ \mathsf{completedCourse}$$

For $\mathcal{Q}$ the number restriction is limited to affect a certain concept. For example, one can also say that an advanced student should approve at least 6 logic courses and 9 computational courses to be considered an advanced student of computer sciences,

$$\begin{gathered}
\geqslant 6\ \mathsf{completedCourse.LogicCourse} \sqcap \\
\geqslant 9\ \mathsf{completedCourse.ComputationalCourse} \sqcap \\
\leqslant 25\ \mathsf{completedCourse}
\end{gathered}$$

2.2 Terminologies

Terminological axioms indicate how concepts or roles are related to each other following the *inclusion* form, $C \sqsubseteq D$ $(R \sqsubseteq S)$, or the *equality* form, $C \equiv D$ $(R \equiv S)$, where C and D are concepts (R and S are roles).

An interpretation $\mathfrak{I}$ satisfies an inclusion $C \sqsubseteq D$ if $C^{\mathfrak{I}} \subseteq D^{\mathfrak{I}}$, and it satisfies an equality $C \equiv D$ if $C^{\mathfrak{I}} = D^{\mathfrak{I}}$. Now given a set of axioms $\mathfrak{T}$, an interpretation $\mathfrak{I}$ satisfies $\mathfrak{T}$ iff $\mathfrak{I}$ satisfies each element of $\mathfrak{T}$. If $\mathfrak{I}$ satisfies an axiom in $\mathfrak{T}$, then we say that it is a *model* of this axiom in $\mathfrak{T}$. Then two axioms or two set of axioms are *equivalent* if they have the same models.

Definitions are used to describe complex concepts and made abstraction of them using a single name. An atomic concept on the left side of an equality *defines* the complex description explained on its right side.

A set of definitions $\mathfrak{T}$ is called a *terminology* or a *TBox* if a symbolic name is defined only once. A terminology $\mathfrak{T}$ contains a *cycle* iff there exists an atomic concept in $\mathfrak{T}$ that uses itself [BN02]; otherwise $\mathfrak{T}$ is called *acyclic*. An *acyclic* terminology $\mathfrak{T}$ can be *expanded* iteratively through each definition in it, replacing each occurrence of a name on the right hand side with the concepts that it stands for. Now, we say that a terminology $\mathfrak{T}$ is *definitorial* if it is *acyclic*, and we call to its semantics *descriptive semantics*.

Those semantics that are motivated by the use of intuitively cyclic definitions are called *fixpoint semantics*. We will not consider fixpoint semantics in this paper.

2.3 Role Constructors

Binary relations between concepts are modeled by roles. If every role name is considered a role description or atomic role, and if R and S are roles descriptions, then $R \sqcap S$ (**intersection**), $R \sqcup S$ (**union**), $\neg R$ (**complement**), $R \circ S$ (**composition**), R^+ (**transitive closure**), and R^- (**inverse**) are also role descriptions. An interpretation $\mathfrak{I}$ is adapted to the inverse role description as follows,

[2] The use of $\mathcal{C}$ is for complement.

[3] For $\mathcal{N}$ and $\mathcal{Q}$, n ranges over the nonnegative integers, and $\|X\|$ stands for the cardinality of the set X.

Constructor	Written	Interpreted
Union ($\mathcal{U}$)	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Negation ($\mathcal{C}$)	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$
Existential Quantification ($\mathcal{E}$)	$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
Number	$\geqslant nR$	$(\geqslant nR)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \mid \left\|\{b \mid (a,b) \in R^{\mathcal{I}}\}\right\| \geq n\right\}$
Restrictions	$\leqslant nR$	$(\leqslant nR)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \mid \left\|\{b \mid (a,b) \in R^{\mathcal{I}}\}\right\| \leq n\right\}$
($\mathcal{N}$)	$= nR$	$(= nR)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \mid \left\|\{b \mid (a,b) \in R^{\mathcal{I}}\}\right\| = n\right\}$
Qualified	$\geqslant nR.C$	$(\geqslant nR.C)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \mid \left\|\{b \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\right\| \geq n\right\}$
Number	$\leqslant nR.C$	$(\leqslant nR.C)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \mid \left\|\{b \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\right\| \leq n\right\}$
Restrictions ($\mathcal{Q}$)	$= nR.C$	$(= nR.C)^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \mid \left\|\{b \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\right\| = n\right\}$

Table 1: Constructors to extend the expressivity of $\mathcal{AL}$-languages.

Inverse ($\mathcal{I}$ or $^{-1}$):
$(R^{-})^{\mathcal{I}} = \{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}$

For instance a hasParent role is obtained applying the inverse role constructor to the given hasChild role.

3 THE AGM CHANGE MODEL

A **belief base** is a knowledge state represented through a set of sentences not necessarily closed under logical consequence. We also know that a **belief set** is a set of sentences of a determined language, closed under logical consequence. In general, a belief set is infinite and that's why it is impossible to deal with them in a computer. Instead, it is possible to characterize the properties that must satisfy each of the change operations on finite representations of a knowledge state.

Expansion operations ("$+$") add a new belief to the epistemic state, without guaranteeing its consistence after the operation.

Contraction operations ("$-$") eliminate a belief α from the epistemic state and those beliefs that make possible its deduction or inference. The sentences to eliminate might represent the minimal change on the epistemic state.

Revision operations ("$*$") consist of the insertion of sentences to the epistemic state, guaranteeing consistence (if it was consistent before the operation). This means that a revision adds a new belief and perhaps it eliminates others to avoid inconsistences.

3.1 Kernel Contractions

The **Kernel Contraction** operator is applicable to belief bases and belief sets. It consist of a contraction operator capable of the selection and elimination of those beliefs in K that contribute to infer α.

Definition 3.1.1 [Han94]: Let K be a set of sentences and α a sentence. The set $K^{\perp}\alpha$, called *set of kernels* is the set of sets K' such that (1) $K' \subseteq K$, (2) $K' \vdash \alpha$, and (3) if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $K^{\perp}\alpha$ is also called *set of α-kernels* and each one of its elements are called *α-kernel.*

For the success of a contraction operation, we need to eliminate, at least, an element of each α-kernel. The elements to be eliminated are selected by an **Incision Function**.

Definition 3.1.2 [Han94]: A function "σ" is an *incision function* for a set K, if for all sentence α it verifies, (1) $\sigma(K^{\perp}\alpha) \subseteq \bigcup(K^{\perp}\alpha)$ and (2) If $K' \in K^{\perp}\alpha$ and $K' \neq \emptyset$ then $K' \cap \sigma(K^{\perp}\alpha) \neq \emptyset$.

Once that the incision function was applied, we must eliminate from K those sentences that the incision function selects, *i.e.* that the new belief base would consist of all those sentences that were not selected by σ.

Definition 3.1.3 [Han94]: Let K be a set of sentences, α a sentence, and $K^{\perp}\alpha$ the set of α-kernels of K. Let "σ" be an incision function for K. The operator "$-_{\sigma}$", called *kernel contraction determined by* "σ", is defined as, $K -_{\sigma} \alpha = K \backslash \sigma(K^{\perp}\alpha)$.

Finally, an operator "$-$" is a kernel contraction operator for K if and only if there exist an incision function "σ" such that $K - \alpha = K -_{\sigma} \alpha$ for all sentence α.

3.2 Consistent Merge of Belief Bases

The union of two different belief bases may be inconsistent. Restoring this property to the resultant union may be thought in terms of a *deductively maximally consistent (d.m.c.)* subset of the union as,

Definition 3.2.1 , Partial Meet Merge [Fuh96]: A *prima facie* candidate for the merge of two sets is any d.m.c. of their union. For each set K, a set X is a d.m.c. subset of K, if (1) $X \subseteq K$, (2) $X \nvdash \perp$, and (3) $\forall Y : X \subset Y \subseteq K$ implies $Y \vdash \perp$. It is easy to see that a set X is a d.m.c. subset of K just in case $X \in K\perp\{\perp\}$[4].

Finally, Fuhrmann defined in [Fuh96] a partial meet merge operation as a union of two bases, not necessarily closed under logic consequence, and a later consistence restoring applying a bottom contraction. Inspired on it we propose a merge operation over two bases, defined by means of the *Kernel Contraction* operator, and determined by an *Incision Function*, as follows,

Definition 3.2.2 : Let " $-$ " be a kernel contraction for a union of two belief bases $K_1 \cup K_2$, determined by an incision function "σ". Then the *Merge for Belief Bases* operator $\circledast$ is defined as, $K_1 \circledast K_2 = (K_1 \cup K_2) -_\sigma \perp$.

4 CONSISTENT UNION OF TERMINOLOGIES

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two terminologies to be unified by a consistent union operation, and let $\star$ be the new proposed *consistent union of terminologies* operator, such that $\mathcal{T}_1 \star \mathcal{T}_2$.

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be composed of n and m distinct definitions named $D_n^{\mathcal{T}_1}$ and $D_m^{\mathcal{T}_2}$ respectively. An operator $\star$ needs to evaluate whether $D_i^{\mathcal{T}_1}, _{1\leq i \leq n}$ specifies the same concept that $D_j^{\mathcal{T}_2}, _{1\leq j \leq m}$ does.

For this we define a mapping $\hbar_{names}$ that identifies two concept names on distinct terminologies, defining both a same logical concept as, $D_i^{\mathcal{T}_1} = \hbar_{names}(D_j^{\mathcal{T}_2})$.

We propose a DL operator $\sqcup_\circledast$ to be the *consistent union of concept definitions* such that,

$$D_k^{\mathcal{T}} = D_i^{\mathcal{T}_1} \sqcup_\circledast D_j^{\mathcal{T}_2}$$

A method to verify the consistent union of two such a concept definitions may intuitively be thought as a belief merge of both concepts generating a new unique and consistent concept $D_k^{\mathcal{T}}, _{1\leq k \leq l}$ where $max(n, m) \leq l < (n + m)$.

So let $K(D_i^{\mathcal{T}_1})$ be the belief base (not necessarily closed under logical consequence) that contains the concept definition $D_i^{\mathcal{T}_1}$ in a terminology $\mathcal{T}_1$, and $K(D_j^{\mathcal{T}_2})$ be the belief base that contains the concept definition $D_j^{\mathcal{T}_2}$ in a terminology $\mathcal{T}_2$. The DL operator $\sqcup_\circledast$ between concept definitions will be translated in the merge operator $\circledast$ between belief bases as defined in the previous section, such that,

$$D_i^{\mathcal{T}_1} \sqcup_\circledast D_j^{\mathcal{T}_2} = K(D_i^{\mathcal{T}_1}) \circledast K(D_j^{\mathcal{T}_2})$$

[4]Fuhrmann's definition of the operator $\perp$ refers to a partial meet contraction defined by the use of a selection function.

4.1 Union of Concepts

Let $D_i^{\mathcal{T}_1}$ and $D_j^{\mathcal{T}_2}$ be two *expanded* concept definitions, and $D_i^{\mathcal{T}_1} \sqcup_\circledast D_j^{\mathcal{T}_2}$ be the *Consistent Union of Concept Definitions* operation determined by a *Merge of Belief Bases* operation $\circledast$, such that

$$D \equiv D_i^{\mathcal{T}_1} \sqcup_\circledast D_j^{\mathcal{T}_2} = D_{1.i}^{\mathcal{T}} \sqcap D_{2.j}^{\mathcal{T}}$$

where exists K_{σ_1} a d.m.c. subset of the incision function $\sigma((K(D_i^{\mathcal{T}_1}) \cup K(D_j^{\mathcal{T}_2}))^{\perp}\perp)$ from now on identified by $\sigma(C)$, and the base $K(D_i^{\mathcal{T}_1})$, such that exists $K_{\sigma_1}^{d.m.c.} \subseteq K(D_i^{\mathcal{T}_1}) \cap \sigma(C)$ and it gives definition to the new sub-concept

$$D_{1.i}^{\mathcal{T}} \equiv D_i^{\mathcal{T}_1} \backslash K_{\sigma_1}$$
$$D_i^{\mathcal{T}_1} \equiv D_{1.i}^{\mathcal{T}} \sqcap K_{\sigma_1}$$

inserted in the resultant terminology $\mathcal{T}$.

Similarly, exists $K_{\sigma_2}^{d.m.c.} \subseteq K(D_j^{\mathcal{T}_2}) \cap \sigma(C)$ and it gives definition to the new sub-concept

$$D_{2.j}^{\mathcal{T}} \equiv D_j^{\mathcal{T}_2} \backslash K_{\sigma_2}$$
$$D_j^{\mathcal{T}_2} \equiv D_{2.j}^{\mathcal{T}} \sqcap K_{\sigma_2}$$

in the same resultant terminology $\mathcal{T}$.

It is important to note that for evaluate and solve this operation, is mandatory to unify first each component of the $\sqcup_\circledast$ operator, *i.e.* $D_i^{\mathcal{T}_1}$ and $D_j^{\mathcal{T}_2}$ with the correspondent mapping (if it exists) $\hbar_{names}(D_i^{\mathcal{T}_1})$ and $\hbar_{names}(D_j^{\mathcal{T}_2})$, respectively.

4.2 Example

The following tables show how two different terminologies might be consistently merged in a new one, following the previous definitions.

$Bird$	$\equiv$	$Animal \sqcap Bipedal \sqcap Oviparous \sqcap$ $hasFeathers \sqcap = 2hasWings$
$Mammal$	$\equiv$	$Animal \sqcap \forall giveBirth.LiveBirth$
$Oviparous$	$\equiv$	$Animal \sqcap \forall giveBirth.Egg$
$Bipedal$	$\equiv$	$= 2hasFoot$
$LiveBirth$	$\equiv$	$\neg Egg \sqcap hasHeartBeat \sqcap$ $hasVoluntaryMovement$

Table 2: A terminology $\mathcal{T}_1$ (TBox) with concepts about animals.

The terminology expressed in table 3 shows among other definitions, some main characteristics of *mammals* and *oviparous* animals, and particularly defines *monotremes* to be a conjunction of both animal classes. When we try to merge terminologies $\mathcal{T}_1$ and $\mathcal{T}_2$ we find that the concept definitions for *mammals* and *oviparous* in $\mathcal{T}_1$ yield the following contradiction for concept *monotreme* in $\mathcal{T}_2$,

$Platypus$	$\equiv$	$Aquatic \sqcap Monotreme$
$Monotreme$	$\equiv$	$Mammal \sqcap Oviparous$
$Mammal$	$\equiv$	$Animal \sqcap$ $\geqslant 2 hasMammaryGlands$
$Oviparous$	$\equiv$	$Animal \sqcap \geqslant 1 layEggs$

Table 3: A terminology $\mathcal{T}_2$ (TBox) with concepts about animals.

$$\forall giveBirth.Egg \sqcap \forall giveBirth.\neg Egg$$

Note that following the proposed method for *Consistent Union of Concept Definitions* operation, we not only eliminate the inconsistence when merging both terminologies, but also keep all information as part of the resultant terminology, by identifying and splitting the problematic concept in two interrelated, revisiting the hierarchy technic of the object oriented paradigm.

$Bird$	$\equiv$	$Animal \sqcap Bipedal \sqcap Oviparous \sqcap$ $hasFeathers \sqcap = 2 hasWings$
$Platypus$	$\equiv$	$Aquatic \sqcap Monotreme$
$Monotreme$	$\equiv$	$Mammal_1 \sqcap Oviparous$
$Mammal_1$	$\equiv$	$Animal \sqcap \geqslant 2 hasMammaryGlands$
$Mammal$	$\equiv$	$Mammal_1 \sqcap \forall giveBirth.LiveBirth$
$Oviparous$	$\equiv$	$Animal \sqcap \forall giveBirth.Egg \sqcap$ $\geqslant 1 layEggs$
$LiveBirth$	$\equiv$	$\neg Egg \sqcap hasHeartBeat \sqcap$ $hasVoluntaryMovement$
$Bipedal$	$\equiv$	$= 2 hasFoot$

Table 4: The resultant terminology $\mathcal{T} = \mathcal{T}_1 \star \mathcal{T}_2$ with concepts about animals.

5 CONCLUSIONS AND FUTURE WORK

A union operation of terminologies probably yields contradictions on concept definitions and further inconsistence in the resultant terminology. The use of a belief revision framework to define terminologies in order to meet a consistent merge operation is proposed and generates a new *Nonmonotonic Description Logics* model as a powerful theory to be applied on future Semantic Web researches.

A deeper investigation on *Epistemic Entrenchment* [5] methods is needed for semi-automate the well functioning of a reasonable incision function (σ) to cut the α-kernels obtained by the use of merge operations. This means that the (σ) function will select those sub-concepts with less epistemic entrenchment to be cut off the resulting definition.

[5] The *Epistemic Entrenchment* method specifies a way to measure the level of importance of a sentence α to belong to the epistemic state.

To achieve this, we'll investigate to incorporate confidence levels on terminologies and definitions, in a way that the origin of a definition will be conditioned to the terminology from which it was "learned", and a terminology confidence level depending of a probabilistic method deduced from its general confidence level.

References

[BN02] F. Baader, W. Nutt. *Basic Description Logics.* In the Description Logic Handbook, Cambridge University Press, 2002, pages 47-100.

[BHS03] F. Baader, Ian Horrocks, U. Sattler. *Description Logics as Ontology Languages for the Semantic Web*, Lecture Notes in Artificial Intelligence. Springer, 2003.

[AGM85] Carlos Alchourrón, Peter Gärdenfors, David Makinson. *On the logic of theory change: Partial Meet Contraction and Revision Functions*, The Journal of Symbolic Logic, 50:510-530, 1985. 1985.

[Han94] Sven Ove Hansson. *Kernel Contraction*, The Journal of Symbolic Logic, 59:845-859, 1994.

[MF03] M. Moguillansky, M. Falappa. *Aplicacion de Operaciones de Cambio en Sistemas Basados en Conocimiento*, IX Congreso Argentino de Ciencias de la Computación, CACIC'2003: 1490-1501, Universidad Nacional de La Plata, Octubre de 2003.

[Fuh96] André Fuhrmann. *An Essay on Contraction*, The European Association for Logic, Language and Information (FOLLI), ISBN (Paperback): 1575860546, ISBN (Cloth): 1575860554, 1996.