

On the Implementation of the Belief Change Operators

Marcelo A. Falappa
Dep. de Ciencias de la Computación
Universidad Nacional del Sur
Argentina
mfalapp@criba.edu.ar

Eduardo L. Fermé
Departamento de Computación
Fac. de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Argentina.
ferme@dc.uba.ar

Key Words: Theory Change, AGM Model, Knowledge Representation, Epistemic States, Epistemic Attitude, Change Operations.

Abstract

In this article we will introduce certain considerations that must be taken into account while implementing the change operations of the belief change theory. The abstract models of belief change make no considerations in respect to the internal structures of the beliefs that compound the knowledge of an agent, and much less take into account the computational complexity of the proposed algorithms. In this article we will explicitly introduce some of the inconveniences and the possible solutions that can be found while implementing the change operations.

Firstly, we will introduce the different change operations of the AGM model. These operations are characterized by certain *rationality postulates* that determine their behavior on the *knowledge level*. This means that the change operations are specified entirely in terms of the knowledge content (beliefs), making no distinction between the explicit information and the implicit information.

In second term, we will introduce the change operations applicable to knowledge bases (that is, finite sets not necessarily closed under logical consequence) in which the explicit beliefs of the knowledge base are clearly distinguished from the beliefs deduced from them. In this case the change operations are defined on a *symbolic level*, which contains explicit references about the syntax of the sentences that compound the base. We will see how certain postulates defined on the knowledge level must be transformed into “less restrictive” postulates with the objective of respecting the basic principles of the AGM model: minimum change, syntax irrelevancy and adequacy of knowledge representation. We will also introduce a *meta-symbolic level* where we preach about the behavior of the sentences that compound a knowledge base, making no reference to its syntactic form.

Lastly, we will analyze the change operations on the *implementation level*. On this level, it is necessary to make certain decisions that are not considered not only on the knowledge level but also on the symbolic level. This analysis is of vital importance because the decisions that are made on the implementation level can have a strong impact on the knowledge level.

1 Introduction

The change of belief theory seeks to model the knowledge dynamic, that is, how beliefs are updated on the arrival of new information. For example, if an agent believes that birds fly, and that tweety is a bird, it can infer that tweety flies. Notwithstanding, if external information arrives saying that birds that are penguins do not fly and that tweety is a penguin then if the agent gives greater priority to the new information, it will think that tweety does not fly. This type of knowledge updating is very common in human reasoning.

Among the different existing models that formalize the belief dynamics, the most popular one is the AGM model, developed by Alchourrón, Gärdenfors and Makinson in the 80s [AGM85]. This model characterizes certain change operations (expansions, contractions and revisions) that cover a large part of the changes that an agent can perform on his knowledge. Each one of these operations is characterized by a set rationality postulates that determine the behavior of the operations, independent of the way they are implemented. These postulates make it possible to see the change operations as “black boxes”, *i.e.*, determining the result of a change operation independent of how that operation is “built”.

The postulates characterize the change operations on knowledge bases (that is, sets of any sentences) or on belief sets (that is, sets of sentences closed under certain operation of logical consequence). When implementing change operations, it is necessary to represent the knowledge with belief bases because if we use a language (at least propositional) declarative enough, it is implausible to treat closed sets. The logic closure of the belief base will represent the epistemic state of the agent. Notwithstanding, the changes in the epistemic state will be reflected on the changes in the knowledge base that generates it. That is, the

changes on the knowledge level will be reflected on the changes on the symbolic level in which an epistemic state is represented.

In the present article we will show the different levels in which a dynamic belief model can be defined and we will analyze some concrete implementation problems.

2 Preliminaries

2.1 Different change operations

In the AGM model [AGM85] it is assumed that an agent's knowledge is represented by a belief set, *i.e.*, a set of sentences closed under logical consequence on a language \mathcal{L} . The consequence operator Cn^1 is a mapping from sets of propositions to sets of propositions, $Cn : 2^{\mathcal{L}} \Rightarrow 2^{\mathcal{L}}$ that satisfies, at least, the following properties:²

- i) **Inclusion:** $A \subseteq Cn(A)$.
- ii) **Monotony:** If $A \subseteq B$ then $Cn(A) \subseteq Cn(B)$.
- iii) **Iteration:** $Cn(A) = Cn(Cn(A))$.

We will assume that the notion of logical consequence includes the classical truth-functional consequence and that it satisfies the properties of deduction and compactness.

- iv) **Supraclassicality:** If α can be derived from A in classical logic, then $\alpha \in Cn(A)$.
- v) **Deduction:** $\beta \in Cn(A \cup \{\alpha\})$ if and only if $(\alpha \rightarrow \beta) \in Cn(A)$.
- vi) **Compactness:** If $\alpha \in Cn(A)$ then $\alpha \in Cn(B)$ for some finite subset $B \subseteq A$.

Given a state of knowledge (represented by a belief base or a belief set), three basic epistemic attitudes are assumed: acceptance, rejection and undetermination. That is, if K is a consistent set of sentences³, and α a sentence of the language, then we can have:

- α is *accepted* in K : in the case that $\alpha \in Cn(K)$.⁴
- α is *rejected* in K : in the case that $\neg\alpha \in Cn(K)$.⁵
- α is *undetermined* in K : in the case that $\alpha \notin Cn(K)$ and $\neg\alpha \notin Cn(K)$.

The set of epistemic states adopted determines the corresponding model's representation power. Other sets of epistemic states could be adopted. For example, probabilities represented by numerical values between 0 and 1, but this will not be treated here. In the AGM model we have three possible states and the different change operations allow the change from one to another.

Let K be a set of sentences and α a sentence of the language. The three basic operations of the AGM model are the following:

- **Expansion:** This operation is in charge of incorporating sentences to the original set, without eliminating any sentence from it. It allows the passage from an epistemic state in which a belief is undetermined to another epistemic state in which the belief is accepted or rejected.
- **Contraction:** This operation eliminates sentences from the original set without incorporating any new ones. It allows the passage from an epistemic state in which a belief is accepted or rejected to another epistemic state in which the belief is undetermined.

¹If X is a set of sentences of the language and α a sentence of the language, then $X \vdash \alpha$ is equivalent to noting that $\alpha \in Cn(X)$.

² 2^X refers to the set of parts of any set X .

³A set K is consistent if there does not exist any sentence α of the language \mathcal{L} such that $\alpha \in Cn(K)$ and $\neg\alpha \in Cn(K)$. In particular, if K is a belief set, *i.e.*, $K = Cn(K)$, and K is inconsistent then K is the whole language, *i.e.*, $K = \mathcal{L}$. This causes the set of inconsistent beliefs to have no epistemic value whatsoever, although this does not happen with belief bases, because the inconsistencies can be produced by a particular sentence and can be "locally" eliminated.

⁴In this case, it is also said that $\neg\alpha$ is rejected in K .

⁵In this case, it is also said that $\neg\alpha$ is accepted in K .

- *Revision*: This operation incorporates a sentence to the original set, but it can eliminate some beliefs in order to preserve consistency in the revised set. It allows the passage from an epistemic state in which a belief is accepted (rejected) to another state in which the belief is rejected (accepted).

The revision operator can be defined starting from the contraction operator and Levi's identity [AGM85, Gär88, Han96]. In an analogous way, the contraction operation can be defined starting from Harper's identity [AGM85, Gär88, Han96]. Of the three, the simplest is the expansion operation because to make it effective it is enough to define union operations on sets and (optionally) closure operations, since it is a monotonic operation. Formally, if K is a belief base, the expansion of K by α , noted by $K+\alpha$ is defined as $K \cup \{\alpha\}$. If \mathbf{K} is a belief set, the expansion of \mathbf{K} by α , noted by $\mathbf{K}+\alpha$ is defined as $Cn(\mathbf{K} \cup \{\alpha\})$.

On the other hand, the contraction and revision operations require, besides operations on sets and consequence operations, a *selection mechanism* to be able to effect the changes in the epistemic state. This makes it necessary, at the moment of implementing these operations, to use constructive methods (for example: *epistemic entrenchment*), lexicographic orders (on the language or part of it) or ad-hoc orders, with the objective of being able to make the corresponding selection possible. The contraction and revision operations are characterized by a series of rationality postulates. These postulates determine the behavior of such change operations on the knowledge level (they are defined on belief sets) and were initially introduced by Gärdenfors [Gär88].

2.1.1 Postulates for Contractions

Let \mathbf{K} be a belief set, *i.e.*, $\mathbf{K} = Cn(\mathbf{K})$, α and β sentences of \mathcal{L} , $+$ an expansion operator and $-$ a contraction operator. The *rationality postulates for contractions* are the following:

- (K⁻1) **Closure**: If $\mathbf{K} = Cn(\mathbf{K})$ then $\mathbf{K}-\alpha = Cn(\mathbf{K}-\alpha)$.
- (K⁻2) **Inclusion**: $\mathbf{K}-\alpha \subseteq \mathbf{K}$.
- (K⁻3) **Vacuity**: If $\alpha \notin \mathbf{K}$ then $\mathbf{K}-\alpha = \mathbf{K}$.
- (K⁻4) **Success**: If $\not\vdash \alpha$ then $\alpha \notin \mathbf{K}-\alpha$.
- (K⁻5) **Recovery**: $\mathbf{K} \subseteq (\mathbf{K}-\alpha)+\alpha$.
- (K⁻6) **Extensionality/Preservation**: If $\vdash \alpha \leftrightarrow \beta$ then $\mathbf{K}-\alpha = \mathbf{K}-\beta$.
- (K⁻7) **Conjunctive Overlap**: $\mathbf{K}-\alpha \cap \mathbf{K}-\beta \subseteq \mathbf{K}-(\alpha \wedge \beta)$.
- (K⁻8) **Conjunctive Inclusion**: If $\alpha \notin \mathbf{K}-(\alpha \wedge \beta)$ then $\mathbf{K}-(\alpha \wedge \beta) \subseteq \mathbf{K}-\alpha$.

2.1.2 Postulates for Revisions

Let \mathbf{K} be a belief set, α and β sentences of \mathcal{L} , $+$ an expansion operator and $*$ a revision operator. The *rationality postulates for revisions* are the following:

- (K^{*}1) **Closure**: If $\mathbf{K} = Cn(\mathbf{K})$ then $\mathbf{K}*\alpha = Cn(\mathbf{K}*\alpha)$.
- (K^{*}2) **Success**: $\alpha \in \mathbf{K}*\alpha$.
- (K^{*}3) **Inclusion**: $\mathbf{K}*\alpha \subseteq \mathbf{K}+\alpha$.
- (K^{*}4) **Vacuity**: If $\mathbf{K} \not\vdash \neg\alpha$ then $\mathbf{K}*\alpha = \mathbf{K}+\alpha$.
- (K^{*}5) **Consistency**: $\not\vdash \neg\alpha$ then $\mathbf{K}*\alpha \neq \mathbf{K}_\perp$.
- (K^{*}6) **Extensionality/Preservation**: If $\vdash \alpha \leftrightarrow \beta$ then $\mathbf{K}*\alpha = \mathbf{K}*\beta$.
- (K^{*}7) **Superexpansion**: $\mathbf{K}*(\alpha \wedge \beta) \subseteq (\mathbf{K}*\alpha)+\beta$.
- (K^{*}8) **Subexpansion**: If $\neg\beta \notin \mathbf{K}*\alpha$ then $(\mathbf{K}*\alpha)+\beta \subseteq \mathbf{K}*(\alpha \wedge \beta)$.

The postulates (K⁻1)–(K⁻6) [(K^{*}1)–(K^{*}6)] are called *basic postulates* for contractions (revisions) while the postulates (K⁻7)–(K⁻8) [(K^{*}7)–(K^{*}8)] are called *supplementary postulates* for contractions (revisions). Some of these postulates have been questioned for being considered implausible among the change operations. For example, the *recovery* postulate (K⁻5) in contractions, and the *success* postulate (K^{*}2) in revisions. This questioning is made on the knowledge level without mentioning the symbolic level. In this article, we may question some of the postulates not only on the knowledge level, but on the symbolic level.

2.1.3 Relations between Contractions and Revisions

There is a relationship between the postulates for contractions and for revisions. Let K be a set of sentences, α a sentence of the language and $+$ an expansion operator.

The contractions and revisions are interdefined in the following way:

- *Levi's Identity*: $K*\alpha = (K-\neg\alpha)+\alpha$.
- *Harper's Identity*: $K-\alpha = K \cap K*\neg\alpha$.

Given these two identities, it can be proven that if a contraction operator satisfies the postulates (K⁻1)–(K⁻6) [an optionally (K⁻7)–(K⁻8)] then the revision operator obtained applying Levi's identity satisfies the postulates (K^{*}1)–(K^{*}6) [and optionally (K^{*}7)–(K^{*}8)]. In an analogous way, if a revision operator satisfies the postulates (K^{*}1)–(K^{*}6) (and optionally (K^{*}7)–(K^{*}8)) then the contraction operator obtained applying Harper's identity satisfies the postulates (K⁻1)–(K⁻6) (and optionally (K⁻7)–(K⁻8)). Notwithstanding, there exists certain known contraction operators such as *withdrawals* [Mak87], that satisfy the postulates for contractions (K⁻1)–(K⁻4) and (K⁻6)–(K⁻8) except *recovery* (K⁻5) that generate revision operators that satisfy the postulates (K^{*}1)–(K^{*}8) upon applying Levi's identity.

2.2 Belief Bases vs. Belief Sets

The *belief sets* are sets of sentences closed under logical consequence. That is, if Cn is a logical consequence operator, then \mathbf{K} is a belief set if and only if $\mathbf{K} = Cn(\mathbf{K})$. This type of representation for an epistemic state is adequate to understand the behavior of the operations but it is inadequate for their implementation. The reason for its inadequacy is simple: if we have a language with a finite set of propositions, it is possible to obtain infinite sentences deducible from any finite set. For this reason, in case of seeking the implementation of the change operations (of the AGM model) it is necessary to represent the epistemic states with belief bases. In [GM88], Gärdenfors and Makinson stated the implausibility of defining change operations on belief sets:

“... a knowledge set is supposed to be closed under logical consequence will cause problems when it comes to implementing a system, since there are in general infinitely many logical consequences to take care of.”

The *belief bases* are any sets of sentences not necessarily closed under logical consequence. Notwithstanding, from this definition, infinite belief bases can be obtained. For this reason, we should restrain even more the representation of epistemic states to finite belief bases, also known as *knowledge bases*. The problem is that the epistemic state associated to a belief set (on the knowledge level) can be generated by a variety of different knowledge bases (on the symbolic level). For example, the epistemic state reflected by $Cn(\{a, b\})$ can be generated, among others, by the following knowledge bases: $\{a, b\}$, $\{a, a \leftrightarrow b\}$, $\{b, a \leftrightarrow b\}$, $\{a, a \rightarrow b\}$, $\{a, \neg b \rightarrow \neg a\}$. Because of this, it is necessary to work with knowledge bases in a way that the changes in the associated belief set be as close as possible to the ideal state, *i.e.*, as if we had has performed the change operations directly on belief sets.

Another reason to choose the finite belief base implementation of the change operators is that, intuitively, the process of changing beliefs is applied on finite sets and not on closed sets. That is, we prefer the foundational theory to represent the epistemic state of an agent, instead of the coherency theory adopted in the AGM model [Gär90]. The foundational theory assumes that the reasons for which a belief is accepted in an epistemic state should be kept track of. On the other hand, the coherency theory assumes that beliefs require no justification because they justify themselves. According to Gärdenfors [Doy92, Gär90], to a coherentist the body of knowledge is like a raft in which a board helps directly or indirectly to keep the rest afloat, and no board could stay stable without the help of the rest.

3 Knowledge Level vs. Symbolic Level

In a famous article from the 80s, Allen Newell [New82] postulates the existence of the “knowledge level” situated immediately above the “symbolic level”. The knowledge level is entirely specified in terms of the knowledge (belief) content, making no distinction between explicit and implicit information.

If we apply these concepts to belief revision, the postulates and the constructions should make no reference to the sentences of the initial set. This criterion is compatible with the AGM postulates for belief sets. In belief bases, this criterion is defended by Dalal [Dal88], who argues:

“The revised knowledge base should not depend of the syntax (or representation) of either the old knowledge or the new information.”

Then Dalal proposes a function for belief bases based on the distances between the models that satisfy the original set and the models that satisfy the new sentence. This idea is later taken up again by, among others, Katsuno and Mendelzon in [KM92] and Lin [Lin]. These functions, correct from a theoretical point of view, cannot be used in implementations, for they violate a basic principle in the construction of programs: declarativity.

However, if we wish to formalize change functions, we should not use a particular syntax. This does not mean that there is no difference between explicit and implicit sentences in a base, but that nothing will be said about their particular syntactic form, although their behavior will be commented. We can distinguish three well differenced levels:

- The *knowledge level*, where we do not distinguish between explicit or implicit sentences in a base. That is to say, we only preach about the result of the contraction operation and not about the behavior of the sentences in the base. This is reflected in the following postulate:

Strong Extensionality: If $Cn(K) = Cn(H)$ then $Cn(K-\alpha) = Cn(H-\alpha)$.

Besides the mentioned functions, we can cite the contractions proposed by Fuhrmann [Fuh91] and Nebel [Neb89]. Notwithstanding, they both violate the functions proposed by Hansson [Han92, Han94], given that they do not satisfy their inclusion postulate for bases: $K-\alpha \subseteq K$. The functions before cited can also produce the result $K-\alpha = \{\}$ in the case in which K is written as an only sentence.⁶

- A *meta-symbolic level* where it is preached about the behavior of sentences that compound the belief base, but making no reference to its syntactic form.

We can cite as an example the postulates proposed by Hansson [Han92, Han94] of *core-retainment*, *uniformity* and *relevance* among others.

- Un *symbolic level* with explicit references about the syntax of the sentences that compound the base.

Fact Preference: If $\beta \in K$ and $(\beta \rightarrow \alpha) \in K$. then $(\beta \rightarrow \alpha) \notin K-\alpha$.

On the first level the axiomatic from the postulates is situated, and on the meta-symbolic level we can cite the *epistemic entrenchment*. Notwithstanding, one may ask oneself, given the existence of a representation theorem that integrates epistemic entrenchment with the AGM axiomatic framework, whether these two levels do not mix. Clearly they do not since the relation between both models given by the rule:

$$(C \leq) \alpha \leq_{\mathbf{K}} \beta \text{ if and only if } \alpha \notin \mathbf{K}-(\alpha \wedge \beta) \text{ or } \vdash (\alpha \wedge \beta)$$

is situated on the meta-symbolic level.

4 Implementation of the Change Operations

On an implementation level, it is necessary to make certain decisions that are not considered not only on the knowledge level but also on the symbolic level. In the following subsections we will analyze the effect that certain decisions made at the moment of implementing the change operations can have.

4.1 Expansions

The expansion operation is the one that requires a more simple definition although this is not extensive to the complexity of its implementation. In the case of representing the epistemic state of an agent using belief

⁶Let us note that the belief base $K = \{a, b, a \rightarrow \neg c, d \rightarrow \neg b\}$ is equivalent (on the knowledge level) to $K = \{a \wedge b \wedge (a \rightarrow \neg c) \wedge (d \rightarrow \neg b)\}$.

sets, the expansion operation is defined in the following way. Let \mathbf{K} be a belief set and α a sentence of the language. Then the expansion of \mathbf{K} by α , noted by the expression $\mathbf{K}+\alpha$, is defined as follows:

$$\mathbf{K}+\alpha = Cn(\mathbf{K} \cup \{\alpha\})$$

In the case that the consequence operator is supraclassical, this operation turns implausible because of its computational cost. For example, if \mathcal{L} is a propositional language containing the propositional letters a, b, \dots, z , $\mathbf{K} = Cn(\{a, b \vee c\})$ and Cn is a supraclassical consequence operator, then the expansion of \mathbf{K} by d contains the sentences $d \vee a, d \vee b, \dots, d \vee z, \dots, \neg d \rightarrow a, \dots, d \wedge a, \dots$, and infinite sentence more. The infinitude of the expanded set (actually, the belief set $\mathbf{K} = Cn(\{a, b \wedge c\})$ is already extensionally infinite) makes it impossible to implement expansion operations starting with a classical consequence operator. For this reason, it is necessary to implement the change operations on knowledge bases, that is, finite belief bases.

The definition of expansion on belief bases does not need to use a consequence operator. Let K be a finite knowledge base and α any sentence. Then the expansion of K by α is defined as follows:

$$K+\alpha = K \cup \{\alpha\}$$

This type of expansion is the most common among systems based on knowledge in which beliefs are represented by finite sets. Notwithstanding, in the case that $\alpha \in K$, that is, there exists at least one proof or syntactic derivation of α in K : Is it necessary to add a new evidence for α ? For example, if an agent thinks that birdy flies because it is a bird and that birds fly, and then it is informed that birdy flies: is it necessary or convenient to introduce the fact that tweety flies? Let us formalize the former knowledge in a first order language. $K = \{bird(tweety), \forall(x)[bird(x) \rightarrow flies(x)]\}$ and let $\alpha = vuela(tweety)$ be a belief. If we use the former definition, the expanded set would be $K = \{bird(tweety), flies(tweety), \forall(x)[bird(x) \rightarrow flies(x)]\}$. Then, there are two proofs or derivations that tweety flies from the expanded knowledge base. If in the future the agent stops believing that tweety is a bird: should it keep believing that it flies or not? The answer to this question is yes, only if the fact that tweety flies was incorporated into the belief base (in some moment of the past) without depending on the “proper evidences” that existed about that fact.

In the case that the former conditions are not satisfied (independence of proofs to incorporate a belief) it is necessary to use an alternative expansion definition. Let K be a finite knowledge base and α any sentence. Then the expansion of K by α is defined as follows:

$$K+\alpha = \begin{cases} K & \text{si } \alpha \in Cn(K) \\ K \cup \{\alpha\} & \text{in another case} \end{cases}$$

This definition of expansion is generally implemented in *truth maintenance systems* [Doy79] in which circularity or redundancy in the demonstrations of basic facts are avoided.

4.2 Contractions

The contraction operations are defined in various ways. In the following sections we will introduce certain alternative models for contraction.

Definition 4.1: Let K be set of sentences and α a sentence. The *remainder set* of K with respect to α is the set of subsets X of K such that:

1. $X \subseteq K$.
2. $X \not\vdash \alpha$.
3. There does not exist W such that $X \subset W \subseteq K$ and $W \not\vdash \alpha$.

The remainder set is the set of maximal subsets of K that fail to imply α . □

The operation of contraction of the AGM model [AGM85] is based on the selection of the “best elements” of the remainder set.

Definition 4.2: Let K be set of sentences, α a sentence and $K \perp A$ the remainder set of K with respect to α . We say that γ is a *selection function* if and only if:

1. If $K \perp \alpha = \emptyset$ then $\gamma(K \perp \alpha) = \{K\}$.
2. If $K \perp \alpha \neq \emptyset$ then $\gamma(K \perp \alpha) = X$ such that $\emptyset \subset X \subseteq (K \perp \alpha)$.

□

Definition 4.3: Let \mathbf{K} be set of beliefs, α a sentence and $\mathbf{K} \perp \alpha$ the remainder set of \mathbf{K} with respect to α . Let γ be a selection function for $\mathbf{K} \perp \alpha$. The operator $\div \gamma$, called *partial meet contraction determined by γ* is defined in the following way:

$$\text{(Def Partial Meet Contraction)} \quad \mathbf{K} \div \gamma \alpha = \cap \gamma(\mathbf{K} \perp \alpha)$$

□

That is, the operation of *partial meet contraction* [AGM85] of \mathbf{K} with respect to α is defined as the intersection of the “best” maximal subsets of α (selected by γ) that fail to imply α .

An alternative for defining contraction operators is using kernel contractions [Han96]. Differing with the partial meet contraction operations, the kernel contraction operations are defined starting from the α -kernels (minimal sets of a set that imply the sentence α), performing a “cut” on each of the elements of that set through an incision function.

Definition 4.4: Let K be a set of sentences and let α be a sentence. The set $K \perp\!\!\!\perp \alpha$, called *set of kernels* is the set of sets X such that:

1. $X \subseteq K$.
2. $X \vdash \alpha$.
3. There does not exist W such that $W \subset X \subseteq K$ and $W \vdash \alpha$.

The set $K \perp\!\!\!\perp \alpha$ is also called *set of α -kernels* and each of its elements is called a α -kernel. Each α -kernel is a minimal subset of K that implies α . □

Definition 4.5: An *incision function* σ for a set K if for every sentence α it is true that:

1. $\sigma(K \perp\!\!\!\perp \alpha) \subseteq \cup(K \perp\!\!\!\perp \alpha)$.
2. If $X \in K \perp\!\!\!\perp \alpha$ and $X \neq \emptyset$ then $X \cap \sigma(K \perp\!\!\!\perp \alpha) \neq \emptyset$.

□

The incision function selects the sentences that will be discarded. The result of contracting a set K with respect to a belief α should consist of all the elements of the original set K that are not removed by the incision function.

Definition 4.6: Let K be a set of sentences, α any sentence and $K \perp\!\!\!\perp \alpha$ the set of α -kernels of K . Let σ be an incision function for K . The operator $\sim \sigma$, called *kernel contraction determined by σ* is defined in the following way:

$$\text{(Def Kernel Contraction)} \quad K \sim \sigma \alpha = K \setminus \sigma(K \perp\!\!\!\perp \alpha)$$

An operator $-$ is a kernel contraction operator for K if and only if there exists some incision function σ such that $K - \alpha = K \sim \sigma \alpha$ for every sentence α . □

The kernel contraction operations are equivalent to the partial meet operations when applied to belief sets. Notwithstanding, this does not happen when they are applied to belief bases. Actually, every partial meet contraction can be generated through a kernel contraction operation although there exist kernel contractions that cannot be generated by partial meet contractions.

5 Contraction Properties

As we said before, the contraction operations are characterized by certain rationality postulates. These postulates allow the characterization of the changes on the knowledge level but they must be appropriately reformulated when the problem is treated on the symbolic level. The postulate of *extensionality* seeks to capture the notion of irrelevancy of the syntax, that is, independent of the “syntax” form with which certain belief is represented, the changes in the epistemic state should be the same. This postulate (formulated to characterize the contractions on the knowledge level), is not generally satisfied in the change operations constructible on the symbolic level. In such sense, there exist different alternative postulates that seek to capture (in greater or lesser magnitude) the notion of irrelevancy of the syntax. For example:

- (a) **Uniformity:** If for every subset H of K it holds that $\alpha \in Cn(H)$ if and only if $\beta \in Cn(H)$ then $K-\alpha = K-\beta$.
- (b) If $\vdash \alpha \leftrightarrow \beta$ then $Cn(K-\alpha) = Cn(K-\beta)$.
- (c) If $Cn(K) = Cn(H)$ then $K-\alpha = H-\alpha$.
- (d) **Strong Extensionality:** If $Cn(K) = Cn(H)$ then $Cn(K-\alpha) = Cn(H-\alpha)$.

Every contraction operation of the AGM model satisfies the alternate postulates introduced formerly. Notwithstanding, the contraction operations defined on the symbolic level satisfy only some of them.

Example 5.1: Let $K = \{p, p \rightarrow q\}$ be a belief base. We know that q is logically equivalent to $(q \wedge \top) \equiv (q \wedge (p \vee \neg p))$. The results of contracting K with respect to q and $(q \wedge (p \vee \neg p))$ could be:

$$K-q = \{p \rightarrow q\}.$$

$$K-(q \wedge (p \vee \neg p)) = \{p\}.$$

In this case, it is clear that $K-p \neq K-(p \wedge (q \vee \neg q))$, and $Cn(K-p) \neq Cn(K-(p \wedge (q \vee \neg q)))$ □

Another example about the properties that are not satisfied on the symbolic level can be found among *inclusion* and the postulates (c) and (d).

Example 5.2: Let $K_1 = \{p, q, r, s\}$ and $K_2 = \{p \wedge q \wedge r, p \wedge q \rightarrow s\}$. Lets suppose that we wish to contract K_1 and K_2 with respect to q . The results of contracting K_1 and K_2 with respect to the belief q could be the following:

$$K_1-q = \{p, r, s\}.$$

$$K_2-q = \{p \wedge q \rightarrow s\}.$$

It is clear that in this case, the postulates of irrelevancy of the syntax (c) and (d) are not satisfied. □

According to Hansson [Han96], a logic is extensional if it permits logically equivalent sentences to be freely interchangeable among each other. In the case of belief change theory, we will say that a contraction (or revision) operator is extensional if the contractions (revisions) with respect to the logically independent sentences produce the “same” result (on the knowledge level or on the symbolic level).

The *inclusion* postulate can also clash with the *recovery* postulate working on the symbolic level. Let us consider the following example.

Example 5.3: Let $K = \{a, b, a \wedge b \rightarrow c, d\}$ be a belief base. Lets suppose that we wish to contract K with respect to c . The possible results of the contraction (that satisfy at least the success postulate) are the following:

$$K_1 = \{a, b, d\}.$$

$$K_2 = \{a, a \wedge b \rightarrow c, d\}.$$

$$K_3 = \{b, a \wedge b \rightarrow c, d\}.$$

$$K_4 = \{a, d\}.$$

$$K_5 = \{b, d\}.$$

$$K_6 = \{a \wedge b \rightarrow c, d\}.$$

In any of the of the resulting bases K_i , $1 \leq i \leq 6$ the beliefs that allowed to deduce c but cannot be deduced from c cannot be recovered. If K was closed under the operator Cn , for each sentence x in K then $c \rightarrow x \in Cn(K)$. Therefore, if c was reincorporated then the eliminated sentences would be recovered. This would imply that, instead of K_2 (to cite a particular case) we would have the following belief base:

$$K'_2 = \{a, a \wedge b \rightarrow c, d, c \rightarrow b\}$$

In this case, the *recovery* postulate is satisfied but the *inclusion* postulate is violated. \square

An aspect to take into the account in the former example is the following. Lets suppose that we wish to contract a set K with respect to a sentence α . Then, to be able to satisfy *recovery* in the contradiction, for each sentence β such that $\beta \in K$ and $\beta \notin K-\alpha$ then $\alpha \rightarrow \beta \in K-\alpha$. This property assures *recovery* but eliminates the *inclusion* property.

An issue that rises at this point, and that is generally ignored in the knowledge level, is to determine which is the intuitive meaning of having this type of sentences. Lets consider the following example.

Example 5.4: Let $K = \{b(t), os(t), \forall(x)(b(x) \wedge os(x) \rightarrow \neg f(x)), k(o)\}$, where t refers to the object tweety, o to the object opus, $b(x)$ is un predicate that is true when x is a bird, $os(x)$ is true when x is an ostrich, $f(x)$ is true when x flies and $k(x)$ is true when x is a kangaroo. Lets suppose that we wish to contract K with respect to the fact $\neg f(t)$. Then, one of the possible contractions would be the following set:

$$K' = \{b(t), \forall(x)(b(x) \wedge os(x) \rightarrow \neg f(x)), k(o)\}$$

In order to satisfy *recovery* we need the sentence $\neg f(t) \rightarrow os(t)$. Although it is true that in the original set it can be inferred that things that fly, are not birds or are not ostriches, and if we abstract from the effect of the logical consequence operators, the former knowledge does not seem to express that if tweety does not fly then it is an ostrich. \square

This example seeks to show that, trying to satisfy recovery, the sought meaning of the knowledge can be altered. Furthermore, while implementing (and applying) contraction operators, the iteration (*i.e.*, repeated application) of changes should be allowed. If the former policy is followed in situations in which the contractions are applied with relative frequency, a knowledge base containing false sentences that totally escape the sought meaning could be obtained (introduced with the sole motive of guaranteeing *recovery*). Maybe the former problem presents itself because of the inclusion of the sentences that seek to guarantee recovery on the "same level" than the sentences that constitute the knowledge base. In the former example, we can obtain the following belief base:

$$K' = \{b(t), \forall(x)(b(x) \wedge os(x) \rightarrow \neg f(x)), k(o)\}$$

an a set **Rec** that contains the tuple $(\neg f(t), os(t))$ that indicates that if the following change of K' is an expansion by $\neg f(t)$ then $os(t)$ must be incorporated. Then, the expansion is defined in the following way:

$$K+\alpha = K \cup \{\alpha\} \cup \{\beta : (\alpha, \beta) \in \mathbf{Rec}\}$$

$$\mathbf{Rec}' = \mathbf{Rec} \setminus \{(\alpha, \beta) : \beta \in K+\alpha\}$$

This mechanism guarantees the satisfaction of recovery but can make some of the postulates that rationally characterize expansions void. In particular, the following inconsistencies can be generated in $K+\alpha$ in spite of $\neg\alpha \notin Cn(K)$. This case would be present if the expansion is not performed immediately after a contraction. Therefore, the former definition must be reformulated in the following way:

$$K+\alpha = K \cup \{\alpha\} \cup \{\beta : (\alpha, \beta) \in \mathbf{Rec} \text{ y } \neg\beta \notin Cn(K)\}$$

$$\mathbf{Rec}' = \mathbf{Rec} \setminus \{(\alpha, \beta) : \beta \in K+\alpha\}$$

The elimination of tuples in the set of recoverable sentences is with the purpose of eliminating tuples that no longer make sense when expansions are performed. For example, if $(\alpha, \beta) \in \mathbf{Rec}$ means that β was eliminated from the knowledge base K in a (former) contraction by α . If we expand by α , we recover (if it is possible) the sentence eliminated in the past, and we eliminate it from the set of sentences to be recovered.

6 Implementing Contractions

In this section we will analyze the possible implementations of the contraction operations. Expansions are defined in an almost direct way, on the base of operations on sets. Revisions are much more complex but can be obtained from expansions and contractions applying Levi's identity. It is for this reason that we centralize our analysis of the implementation of the contraction operations.

One of the possible contraction operator constructions is using *kernel contractions*. These operations are defined in the following way: let K be a knowledge base and α the sentence to be contracted. The minimal subsets of K that imply α (α -kernels) are obtained and at least one sentence of each of these subsets is eliminated. The elimination is performed through an incision function. This incision function determines the sentences to be eliminated from each α -kernel (because of this, they are also eliminated from K), generating in this way, a "cut" in each of the α -kernels.

How can the α -kernels be obtained? One way of doing it is through a logic programming language such as PROLOG. This language works with an object language that is a subset of first order logic. This subset is formed by Horn clauses [StSh86], which are (implicitly) universally quantified, do not contain existential quantifiers or functional symbols. Because free variable clauses are not allowed, the property of deduction assumed in the consequence operator is valid.

One of Prolog's greatest potentialities is the easy way of writing *metainterpreters* [StSh86]. Prolog's metainterpreters are Prolog interpreters written in Prolog. For example, the following metainterpreter [StSh86] is an interpreter for a pure Prolog that obtains the proof tree of the query A :

```
% solve(<A>,<Proof>): <Proof> is the proof tree of the query <A>.
```

```
solve(true,true).
solve((A,B),(ProofA,ProofB)):-solve(A,ProofA),solve(B,ProofB).
solve(A,A <- ProofB):-clause(A,B),solve(B,ProofB).
```

Generalizing this metainterpreter through second order predicates of the language Prolog, (like `findall` or `setof`) all the proofs of the query A can be obtained. Afterwards, through some selection criterion (determined by the lexicographic order of the sentences that compound the proof or by an ad-hoc order) operations that perform the contraction of a knowledge base K with respect to a sentence α can be generated.

One of the characteristics of this process is that it builds directly on Prolog's inference motor. This method selects the leftmost subgoal and uses as a search rule the first clause that has a head that unifies with the subgoal to resolve. This process is formally known as *SLD-resolution* [StSh86] and it is complete if the search tree is generated *breadth first* and not *depth first*. Unfortunately, Prolog uses depth search trees for reasons of space in the computation of the search trees.

Example 6.1: Lets consider the knowledge base $K = \{p \leftrightarrow q\}$ that would be represented in Prolog with the following defined program [StSh86]:

```
p :- q.
q :- p.
p.
```

in this case, the goal p is not successful unless we guarantee a breadth-first search of the SLD tree. Because Prolog generates the trees depth-first, the query for p would not have a positive answer. Notwithstanding, if a contraction of K by q is performed and preference is given to the facts, then the following logical program is obtained:

```
p :- q.
p.
```

After performing the contraction of K by q , the query for p is successful. Therefore, it would seem that in the contraction of K by q beliefs were added because the query p is successful in $K-q$ (*i.e.*, $q \in Cn(K-q)$) but is not successful in K (*i.e.*, apparently $q \notin Cn(K)$). This situation can be generated by the not completeness of the inference motor used. \square

The former metainterpreter works with a very reduced language as is with Horn clauses, that is, clauses with, at most, one positive literal⁷ If A, B_1, \dots, B_n are atoms then the valid sentences in this language are the following:

1. Facts: this type of sentences are composed of only one positive literal (A) and is represented as $A \leftarrow$.
2. Rules: this type of sentences are composed of only one positive literal and at least one negative literal ($A \vee \neg B_1 \vee \dots \vee \neg B_n$) and are represented as $A \leftarrow B_1 \wedge \dots \wedge B_n$.
3. Queries: this type of sentences are composed of, at least, a negative literal ($\neg B_1 \vee \dots \vee \neg B_n$) and is represented as $\leftarrow B_1 \wedge \dots \wedge B_n$.
4. Empty Clause: this sentence represented by the symbol \perp or \square is the contradiction and is a clause without atoms.

Clauses are, generally, of the form $P \leftarrow Q$. P is called the *head* of the clause while Q is called the *body*. Even though this language has a great expressive power and is widely used in logic programming languages, it is very specific and does not allow the inference of negative information.⁸

Therefore, inconsistencies cannot be produced and the revision operations (defined by Levi's identity) would not have greater sense. Notwithstanding, the contractions can be defined and be applied in the proposed language.

A more general language must allow the usage of clauses with more than one positive literal. In this way, rules with negative clauses in the body and/or the head could be expressed. That is, sentences of the form $\neg A$ or $\neg A \leftarrow B_1 \wedge \dots \wedge \neg B_i \wedge \dots \wedge B_n$ with $A, B_1, \dots, B_i, \dots, B_n$ being atoms of the language. Notwithstanding, in these languages, other inference mechanisms must be used to obtain the sets of kernels and, with an incision function, determine which sentences will be eliminated from the knowledge base. Among the inference mechanisms used, we can choose to represent the sentence in clausal form (that is, as a conjunction of a disjunction of literals) and use elimination of complementary literals [GN88]. Among these methods, we can use the *unitary, input, linear, ordered, or set of support* resolution methods [GN88]. The ordered resolution method is extremely efficient but can result to be not complete with languages composed of clauses that are not Horn clauses. The support set method can be initiated taking as a support the negation of the sentence to be proven, working "backwards" from the goal (negation of the sentence to be proven) until it arrives to an empty clause. One way of making the change operations more efficient consists in maintaining the possible demonstrations of the sentences that compose a knowledge base (in the symbolic level and the knowledge level) with a *connection graph* that unites them. Every computation made while performing a query to the system is stored in this graph. This demands a greater storage space but speeds up the proofs, avoiding computations (which include, among others, executions of unification algorithms) and performing operations directly on the connection graph.

7 Conclusions

In this article we introduced some of the inconveniences and the possible solutions that can be found while implementing the change operations. To do this, we introduced the different change operations of the AGM model analyzing their behavior on the knowledge level, in terms of the content of the knowledge (beliefs) making no distinction between explicit and implicit information.

Then, we analyzed the change operations on knowledge bases (that is, finite sets not necessarily closed under classical logical consequence) in which the explicit beliefs are clearly distinguished. We also saw how certain postulates defined on the knowledge level must be transformed into "less restrictive" postulates with the object of respecting the basic principles of the AGM model.

We have introduced a meta-symbolic level where the behavior of the sentences in a knowledge base is described, but making no reference to its syntactic form. Lastly, we analyzed the change operations on the

⁷A literal is an atom or the negation of an atom. A positive literal is an atom while a negative literal is the negation of an atom. An atom is a well formed formula of the object language of the form $p(t_1, \dots, t_n)$ where p is a predicate symbol of arity n and the t_i 's are terms (constants, variables, composite functions) of the language.

⁸The extension of this language that uses *negation by failure* allows the inference of negative information. Notwithstanding, this type of language is not pure Horn because it admits a type of negation (by failure, not classical) in the body of the clause.

implementation level, giving different alternatives for implementing change operations (expansions as well as contractions).

Acknowledgements

We would like to thank Guillermo R. Simari for his support in the elaboration of this paper.

References

- [AGM85] Alchourrón C., Gärdenfors P., Makinson D.: *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*. **Journal of Symbolic Logic** **50**: 510-530, 1985.
- [Dal88] Dalal, M.: *Investigation into a Theory of Knowledge Base Revision: Preliminary Report*. **Seventh National Conference on Artificial Intelligence, (AAAI-88)**: 475-479, 1988.
- [Doy79] Doyle, J.: *A Truth Maintenance Systems*. **Artificial Intelligence** **12**: 231-272, 1979.
- [Doy92] Doyle, J.: *Reason Maintenance and Belief Revision: Foundations versus Coherence Theories*. En Gärdenfors (ed.), **Beilef Revision**, Cambridge University Press, Cambridge, 1992.
- [Fuh91] Fuhrmann, André: *Theory Contraction through Base Contraction*. **Journal of Philosophical Logic** **20**: 175-203, 1991.
- [Gär88] Gärdenfors P.: *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. **Cambridge, MA: The MIT Press, Bradford Books**, 1988.
- [Gär90] Gärdenfors Peter: *The Dynamics of Belief Systems: Foundations vs. Coherence Theories*. **Revue Internationale de Philosophie**: 24-46, 1990.
- [GM88] Gärdenfors P., Makinson D.: *Revisions of Knowledge Systems using Epistemic Entrenchment*. **Second Conference on Theoretical Aspects of Reasoning About Knowledge**: 83-95, 1988.
- [GN88] Genesereth M., Nilsson N.: *Logical Foundations of Artificial Intelligence*. **Stanford University**. 1988.
- [Han92] Hansson, S.: *In Defense of Base Contraction*. **Synthese** **91**: 239-245, 1992.
- [Han94] Hansson, S.: *Taking Belief Bases Seriously*. **D. Prawitz and, D. Westertähl (eds.), Logic and Philosophy in Uppsala** **236**: 13-28 , 1994.
- [Han96] Hansson Sven Ove: *A Textbook of Belief Dynamics: Theory Change and Database Updating*. Uppsala University, Sweden, 1996.
- [KM92] Katsuno, H. & Mendelzon, A.: *Propositional Knowledge Base Revision and Minimal Change*. **Artificial Intelligence** **52**: 263-294, 1991.
- [Lin] Lin, Jinxin. *Integration of Weighted Knowledge Bases*. To appear in **Artificial Intelligence**.
- [Mak87] Makinson, David: *On The Status of the Postulate of Recovery in the Logic of Theory Change*. **Journal of Philosophical Logic** **16**: 383-394, 1987.
- [Neb89] Nebel, B.: *A Knowledge Level Analysis of Belief Revision*. **First International Conference of Principles of Knowledge Representation**: 301-311, 1989.
- [New82] Newell Allen: *The Knowledge Level*. **Artificial Intelligence** **18**: 87-127, 1982.
- [StSh86] Sterling Leon, Shapiro Ehud: *The Art of Prolog*. **The MIT Press**, Cambridge, London, England, 1986.