Some approaches to Belief Bases Merge*

Marcelo A. Falappa

and

Guillermo R. Simari

Instituto de Investigación en Ciencia y Tecnología Informática (IICyTI)
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina
PHONE/FAX: (+54)(291)459-5136
E-MAIL: [mfalappa,grs]@cs.uns.edu.ar

Abstract

In this work, we define some non-prioritized merge operators, that is, operators for the consistent union of belief bases. We define some postulates for several kinds of merge operator and we give different constructions: trivial merge, partial meet merge and kernel merge. For some constructions we provide representation theorems linking construction with a set of postulates. Finally, we propose that the formulated operators can be used in some multi-agent systems.

Key words: Belief Revision, AGM Model, Knowledge Dynamics, Multi-agent Systems.

1 INTRODUCTION

Belief Revision has as its main objective to model the dynamics of knowledge, that is, the way in which an agent's knowledge must be updated when it finds new information. That is, how the agents modify their beliefs when they receive new information. The main problem arises when that information is inconsistent with the beliefs that represent her/his epistemic state.

Revisions are the most commonly used change operators because they allow a sentence α to be included into a set K, generating a new set K', preserving consistency in the new set. The traditional revision models [1, 13] are prioritized, that is, they give priority to new information over the information that is already part of their knowledge. This property does not seem plausible in the real world, because in many cases it is not reasonable to give priority to information just because it is new. In non prioritized models, it is possible for new information not to be totally accepted. Such new information can be rejected, partially accepted, or fully accepted only after a debate process. In this sense, there exists a variety of different non prioritized belief revision models, among which are screened revision [23], semirevision operators [18], merge operations [10], credibility limited revisions [20], revisions by sets of sentences [8], etc.

*Partially supported by CONICET (PIP 5050), Universidad Nacional del Sur (PGI 24/ZN11), and Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 13096).

We adopt a fixed finite language $\mathcal{L}$ with a complete set of boolean connectives. Formulae in $\mathcal{L}$ will be denoted by lower case Greek characters $\alpha, \beta, \delta, \ldots$, while sets of formulae from $\mathcal{L}$ will be denoted by upper case letters $A, B, C, \ldots$. We identify the underlying logic with its consequence operator $Cn : \mathbf{2}^{\mathcal{L}} \Rightarrow \mathbf{2}^{\mathcal{L}}$. The underlying logic is assumed to be *supraclassical* (*i.e.*, include classical propositional calculus), *compact* ($Cn(A) = Cn(B)$ for some finite subset B of A) and satisfy the *deduction theorem* ($\alpha \rightarrow \beta \in Cn(A)$ if and only if $\beta \in Cn(A \cup \{\alpha\})$). Sometimes, we will use the relation $\vdash$ as an alternative notation of the consequence operator: $A \vdash \alpha$ if and only if $\alpha \in Cn(A)$.

The paper is organized as follows. Section 2 presents some limitations of the most popular framework for belief dynamics: AGM. We present the difference of representing epistemic states by means of arbitrary sets of sentences or by means of logically closed sets of sentences. We also compare logical frameworks in which the new information is always accepted with another formalisms in which the new information could be rejected or partially accepted. Section 3 presents different postulates for belief base merging, giving different constructions and representation theorems. Section 4 shows an application of the new operators in multi-agent systems and Section 5 presents the conclusions and future work.

2 AGM MODEL AND ITS LIMITATIONS

AGM model [1] is the most popular framework for belief change and most others rely on the foundations of AGM. This model represents the epistemic states through belief sets, that is, set of sentences closed under logical consequence. We will use bold upper case letters to represent belief sets, for instance, $\mathbf{K}$. Three types of belief change operators are considered: *belief expansion* (incorporation of an epistemic input α into $\mathbf{K}$ without retraction of existing beliefs); *belief contraction* (removal of an existing belief α from $\mathbf{K}$); and *belief revision* (incorporation of a new belief into $\mathbf{K}$ with possible retraction of existing beliefs in order to preserve consistency).

Belief expansion is trivially defined using consequence operator and set operations. Given a belief set $\mathbf{K}$ and a sentence α, the expansion of $\mathbf{K}$ with respect to α, noted by $\mathbf{K}+\alpha$, is equal to $Cn(\mathbf{K} \cup \{\alpha\})$. However, belief contraction and belief revision (noted by "$-$" and "$*$" respectively) can not be defined in that way: they need to use selection functions to determine which beliefs will be erased from the epistemic state. There are two approach to construct contractions: *partial meet contractions* [1] based on a selection among subsets of the original set that do not imply the information to be retracted; *kernel contractions* [17] based on a selection among the sentences that imply the information to be retracted. On the other hand, revision operators can be defined through Levi identity; in order to revise an epistemic state with respect to a sentence α, we contract with respect to $\neg\alpha$ and then expand the new epistemic state with respect to α.

AGM Model of theory revision has several controversial points. For instance, in AGM belief revision the input sentence is always accepted. That property in know as *success*:

$$\alpha \in \mathbf{K}*\alpha$$

Success specifies that the new information has primacy over the beliefs without discussion or debate. Another controversial property is know and *consistency*:

$$\text{If } \nvdash \neg\alpha \text{ then } \mathbf{K}*\alpha \nvdash \bot$$

which establishes that the revised epistemic state is consistent only if the input sentence is consistent. In non-prioritized belief revision, these requirement are relaxed and we will define new revision operators modifying success and consistency postulates.

2.1 Belief Bases vs. Belief Sets

When talking about change we have been assuming that there is something that changes. This something, the object of change, is the belief state [19]. We can develop different constructions to be used as models of the belief state. Among different alternatives, the basic representation of an epistemic state is by mean of a belief set (set of sentences closed under logical consequence) or a belief base (set of sentences not necessarily closed).

Clearly, the representation of belief states with belief sets has many disadvantages from a computational point of view. A belief set is a very large entity. For any two sentences α, β in a belief set $\mathbf{K}$ then so are $\alpha \vee \beta$, $\alpha \vee \neg\beta$, $\alpha \vee \beta \vee \delta, \ldots$ because they are logical consequences of $\mathbf{K}$. If the language is sufficiently rich, the belief set will contain innumerable sentences that the believer has never thought of [19].

It is more natural to think of the belief state as represented by a limited number of sentences that may (roughly speaking) correspond to the explicit beliefs. Changes can operate on this smaller set, rather than directly on the belief set. Such a model is much closer to computational applications such as knowledge representation systems, deductive databases, multi-agent systems, etc.

The distinction between belief sets and belief bases is similar to the distinction between the coherence approach and the foundational approach to belief revision. The *coherence approach* focuses on logical relations among beliefs rather than on inferential relations, that is, no belief is more fundamental than another [5]. In the coherentist approach, beliefs provide each other with mutual support; therefore, a belief set represents the limit case of this approach. On the other hand, the *foundational approach* divides beliefs into two classes: explicit beliefs and those beliefs justified by the explicit beliefs. The explicit beliefs can be seen as "self-justified beliefs" whereas the other beliefs are considered as derived, justified or supported beliefs. The foundational approach provides explanation of beliefs by requiring that each belief be supportable by means of non-circular arguments from explicit or basic beliefs [5]. Another feature of foundational approach is that a belief α may be justified or derived by several independent beliefs, so that even is some of the justifications for α are removed, the belief α may be retained because it is supported by other beliefs. A belief base is considered a good example of the foundational approach.

We may found several advantages and disadvantages for these approaches. While a definitive conclusion about whether either of these approaches is better than the other awaits answers, we believe that the foundational approach by means of belief bases is better in computational environments such as argumentative systems [12, 25], truth maintenance systems [4], and multi-agent systems [7, 26].

2.2 Prioritized and Non-prioritized Changes

Every belief change framework defines an epistemic model (the formalism in which the beliefs will be represented) and then defines different kinds of operators. Each operator may be presented in two ways: by giving an explicit construction (algorithm) for the operator, or by giving a set of rationality postulates to be satisfied. Rationality postulates determine constraints that the operators should satisfy. They treat the operators as black boxes; after receiving certain inputs (of new information) we know what the response will be, but not the internal mechanisms used.

In conventional, AGM-style belief revision [1] the epistemic model has two components: an epistemic state represented by a belief set, and an input sentence represented by a formulae of the language. The main property of AGM model is that the input sentence is always accepted. Non-prioritized belief revisions relax this requirement and they operate on belief bases [10, 18, 15], on belief sets (theories) [9, 23, 20], and on entrenchment-based and sphere-based systems [20].

The principle of primacy of new information is often criticized, since it can not be accepted in all circumstances. Sometimes we have more confidence in our current beliefs than in the new information. Here we will represent belief states by means of belief bases. It is more natural to think of the belief state as represented by a limited number of sentences that may (roughly) correspond to the explicit beliefs. Changes can operate on this smaller set, rather than directly on the belief set. Such a model is much closer to the workings of actual human minds and actual computers.

Merge operators open the possibility that the new evidence is partially or even completely ignored if old information is more entrenched or more plausible. The merge operator joins old and new information to a consistent whole without giving undue precedence to the one or the other [10].

3 MERGING BELIEF BASES

3.1 Postulates

Now we will define different postulates for belief base merging. Let A, B, C, D, E be belief bases and "$\circledast$" a belief merge operator. We will notate the merge of A and B as $A \circledast B$ and we propose the following postulates for a merge operator.

Inclusion $A \circledast B \subseteq A \cup B$.

This postulate establishes that, if we merge two belief bases A and B, then the new stock of beliefs will be contained in the union of A and B.

Symmetry $A \circledast B = B \circledast A$.

This postulate stablishes the two belief base to be merged are equality considered.

Consistency Preservation If $A \nvdash \bot$ and $B \nvdash \bot$ then $A \circledast B \nvdash \bot$.

This postulate ensures that the merged belief base is consistent whenever the original belief bases are consistent.

Strong Consistency $A \circledast B \nvdash \bot$.

This postulate ensures consistency in the merged belief base without consider the consistency of each belief base.

Vacuity If $A \cup B \nvdash \bot$ then $A \circledast B = A \cup B$.

This postulate establishes that A and B are jointly consistent then the merge is equal to the union of them.

Core Retainment If $\alpha \in (A \cup B) \setminus (A \circledast B)$ then there is a set E such that $E \subseteq (A \cup B)$, E is consistent but $E \cup \{\alpha\}$ is inconsistent.

This postulate expresses the intuition that nothing is removed from the union of the original belief bases unless its removal in some way contributes to making the new belief base consistent.

Relevance If $\alpha \in (A \cup B) \setminus (A \circledast B)$ then there is a set E such that $A \circledast B \subseteq E \subseteq (A \cup B)$, E is consistent but $E \cup \{\alpha\}$ is inconsistent.

This postulate is a stronger version of core retainment and we will use it to characterize some kinds of revision operators. Together with core retainment, this postulate tries to capture the notion of minimal change.

Reversion If $A \cup B$ and $C \cup D$ have the same minimally inconsistent subsets then $(A \cup B) \setminus (A \circledast B) = (C \cup D) \setminus (C \circledast D)$.

This postulate establishes that, if $A \cup B$ and $C \cup D$ contain the same minimally inconsistent subsets then the sentences erased in the respective merges are the same.

Congruence If $A \cup B = C \cup D$ then $A \circledast B = C \circledast D$.

This postulate expresses that A joined with B is equal to C joined with D then the merge of A and B is equal to the merge of C and D.

Relevance and Core Retainment have been adapted from Hansson's work [16, 19] and they was used in [8] for multiple change operators. Congruence was presented by Fuhrmann's work [10] for his merge operator and Reversion was presented by Falappa *et al.* [8].

In the next subsections we will propose different constructions for merge operators. In each case, we will show which postulates are satisfied and, in some special cases, we will present representation theorems of defined operators.

3.2 Trivial Merge

Definition 1: Let K and H be two belief bases. The *trivial merge operator* "$\wedge$" for K and H is defined as $K \wedge H = K \cap H$.

Observation 1: Let K and H be two belief bases and "$\wedge$" be a trivial merge operator for K and H. Then "$\wedge$" satisfies *inclusion*, *symmetry* and *consistency preservation*. ■

The trivial merge operator produces a very drastic change and it violates the principle of *minimal change* [13]. However, it can be useful from a theoretical point of view because it can be seen as a lower bound of every plausible merge operator. With some examples, we will show why the trivial merge operator does not satisfy the principle of minimal change.

Example 1: Let p, q, r, s logically independent propositional letters. Suppose that $K = \{p, q, r\}$ and $H = \{p, q, \neg s\}$. The trivial merge of K and H is equal to $\{p, q\}$. We may look that r and $\neg s$ are erased from $K \wedge H$ without necessity because their are both consistent with the $K \wedge H$. From this example, we may verify that "$\wedge$" does not satisfy vacuity, core retainment and relevance.

Example 2: Let p, q, r logically independent propositional letters. Suppose that $K = \{p, p \rightarrow q, r\}$ and $H = \{p, q, r\}$. The trivial merge of K and H is equal to $\{p, r\}$ even though q are logically implied by K and H. Again, we may verify that "$\wedge$" does not satisfy vacuity, core retainment and relevance.

3.3 Kernel Merge

Definition 2: Let K be a belief base and α a sentence. Then $K^{\perp\!\!\perp}\alpha$ is the set of all K' such that $K' \in K^{\perp\!\!\perp}\alpha$ if and only if $K' \subseteq K$, $K' \vdash \alpha$, and if $K'' \subset K'$ then $K'' \nvdash \alpha$. The set $K^{\perp\!\!\perp}\alpha$ is called the *kernel set*, and its elements are called the *α-kernels of K*.

In order to define the operator of kernel merge we need to use an incision function. This function selects sentences to be removed from $K \cup H$ and it is called incision function because it makes an incision in every $\perp$-kernel.

Definition 3: Let K and H be two belief bases. A *general incision function* for K and H is a function "σ" ($\sigma : \mathbf{2^{2^{\mathcal{L}}}} \Rightarrow \mathbf{2^{\mathcal{L}}}$) such that for any sets $K, H \subseteq \mathcal{L}$, the following hold:
1) $\sigma((K \cup H)^{\perp\!\!\perp}\perp) \subseteq \cup((K \cup H)^{\perp\!\!\perp}\perp)$.
2) If $X \in (K \cup H)^{\perp\!\!\perp}\perp$ and $X \neq \varnothing$ then $(X \cap \sigma((K \cup H)^{\perp\!\!\perp}\perp)) \neq \varnothing$.
The limit case in which $(K \cup H)^{\perp\!\!\perp}\perp = \varnothing$ then $\sigma((K \cup H)^{\perp\!\!\perp}\perp) = \varnothing$.

Definition 4: Let K and H be belief bases and "σ" a general incision function. The operator "$\circledast$" of *kernel merge* ($\circledast : \mathbf{2^{\mathcal{L}}} \times \mathbf{2^{\mathcal{L}}} \Rightarrow \mathbf{2^{\mathcal{L}}}$) is defined as $K \circledast H = (K \cup H) \setminus \sigma((K \cup H)^{\perp\!\!\perp}\perp)$.

The mechanism of this operator is to join K and H and then eliminate from the result all possible inconsistency by means of an incision function that makes a "cut" over each minimally inconsistent subset of $K \cup H$. Since this operator uses an incision function and the set of $\perp$-kernels, we call it kernel merge.

Theorem 1: Let K and H be two belief bases. The operator "$\circledast$" is a *kernel merge* of K and H if and only if it satisfies *inclusion*, *strong consistency*, *core retainment* and *reversion*.

Proof.

[Construction to Postulates] Let "$\circledast$" be a kernel revision by a set of sentences for K. We must show that "$\circledast$" satisfies the postulates enumerated in the theorem. Let $K \circledast H = (K \cup H) \setminus (\sigma((K \cup H)^{\perp\!\!\perp}\perp))$.

Inclusion: Straightforward from the definition.

Strong Consistency: Since all sets in $(K \cup H)^{\perp\!\!\perp}\perp$ are minimally inconsistent, and σ cuts every set in it, then $(K \cup H) \setminus \sigma((K \cup H)^{\perp\!\!\perp}\perp)$ is consistent.

Core Retainment: Suppose that $\alpha \in (K \cup H) \setminus (K \circledast H)$. That is, $\alpha \in K \cup H$ and $\alpha \notin K \circledast H$. Then $\alpha \in \sigma((K \cup H)^{\perp\!\!\perp}\perp)$. Since $\sigma((K \cup H)^{\perp\!\!\perp}\perp) \subseteq \cup((K \cup H)^{\perp\!\!\perp}\perp)$ there is some X such that $\alpha \in X$ and $X \in (K \cup H)^{\perp\!\!\perp}\perp$. Let $Y = X \setminus \{\alpha\}$. Then there is some Y such that $Y \subseteq (K \cup H)$, $Y \nvdash \perp$ but $Y \cup \{\alpha\} \vdash \perp$. Therefore, core retainment is satisfied.

Reversion: Suppose that $K \cup H$ and $K' \cup H'$ have the same minimally inconsistent subsets. That means that $(K \cup H)^{\perp\!\!\perp}\perp = (K' \cup H')^{\perp\!\!\perp}\perp$. Since σ is a well defined function then $\sigma((K \cup H)^{\perp\!\!\perp}\perp) = \sigma((K' \cup H')^{\perp\!\!\perp}\perp)$. We need to show that $(K \cup H) \setminus (K \circledast H) = (K' \cup H') \setminus (K' \circledast H')$.

$\subseteq$) If $\alpha \in (K \cup H) \setminus (K \circledast H)$ then, by definition of "$\circledast$", $\alpha \in \sigma((K \cup H)^{\perp\!\!\perp}\perp)$. Since $\sigma((K \cup H)^{\perp\!\!\perp}\perp) = \sigma((K' \cup H')^{\perp\!\!\perp}\perp)$ then $\alpha \in K' \cup H'$ and $\alpha \notin K' \circledast H'$. Therefore, $(K \cup H) \setminus (K \circledast H) \subseteq (K' \cup H') \setminus (K' \circledast H')$.

$\supseteq$) If $\alpha \in (K' \cup H') \setminus (K' \circledast H')$ then, by definition of "$\circledast$", $\alpha \in \sigma((K' \cup H')^{\perp\!\!\perp}\perp)$. Since $\sigma((K' \cup H')^{\perp\!\!\perp}\perp) = \sigma((K \cup H)^{\perp\!\!\perp}\perp)$ then $\alpha \in K \cup H$ and $\alpha \notin K \circledast H$. Therefore, $(K' \cup H') \setminus (K' \circledast H') \subseteq (K \cup H) \setminus (K \circledast A)$.

[Postulates to Construction] We need to show that if an operator satisfies the enumerated postulates then it is possible to build an operator in the way specified in the theorem. Let "σ" be a function such that, for every pair of sets K and H, it holds that:

$$\sigma((K \cup H)^{\perp\!\!\perp}\perp) = \{\alpha : \alpha \in (K \cup H) \setminus (K \circledast H)\}$$

We must show:

Part A.

1. "σ" is a well defined function.
   That is, if K' and H' are sets of sentences such that $(K \cup H)^{\perp\!\!\perp}\perp = (K' \cup H')^{\perp\!\!\perp}\perp$, we must show that $\sigma((K \cup H)^{\perp\!\!\perp}\perp) = \sigma((K' \cup H')^{\perp\!\!\perp}\perp)$. From the hypothesis we have that $K \cup H$ and $K' \cup H'$ have the same minimally inconsistent subsets. It follows from **reversion** that $(K \cup H) \setminus (K \circledast H) = (K' \cup H') \setminus (K' \circledast H')$. Therefore:
   $$\begin{aligned}\sigma((K \cup H)^{\perp\!\!\perp}\perp) &= \{\alpha : \alpha \in (K \cup H) \setminus (K \circledast H)\}\\ &= \{\alpha : \alpha \in (K' \cup H') \setminus (K' \circledast H')\}\\ &= \sigma((K' \cup H')^{\perp\!\!\perp}\perp)\end{aligned}$$
   Therefore, "σ" is well defined.
2. $\sigma((K \cup H)^{\perp\!\!\perp}\perp) \subseteq \cup((K \cup H)^{\perp\!\!\perp}\perp)$.
   Let $\alpha \in \sigma((K \cup H)^{\perp\!\!\perp}\perp)$. Then $\alpha \in (K \cup H) \setminus (K \circledast H)$. Due to **core retainment** there is some E such that $E \subseteq (K \cup H)$, $E \nvdash \perp$ but $E \cup \{\alpha\} \vdash \perp$. Since $\alpha \in K \cup H$ then there is a $\perp$-kernel X in $(K \cup H)$ (*i.e.*, there is a minimally inconsistent subset of $K \cup H$) such that $X \subseteq E \cup \{\alpha\}$ and $\alpha \in X$. Therefore, $\alpha \in \cup((K \cup H)^{\perp\!\!\perp}\perp)$.
3. If $X \in (K \cup H)^{\perp\!\!\perp}\perp$ then $(X \cap \sigma((K \cup H)^{\perp\!\!\perp}\perp)) \neq \varnothing$.
   Let $X \in ((K \cup H)^{\perp\!\!\perp}\perp)$. We need to show that $X \cap \sigma((K \cup H)^{\perp\!\!\perp}\perp) \neq \varnothing$. Due to **strong consistency** $K \circledast H \nvdash \perp$. Since $X \vdash \perp$ we may conclude that $X \nsubseteq K \circledast H$. This means that there is some β such that $\beta \in X$ and $\beta \notin K \circledast H$. Since $X \subseteq (K \cup H)$ then $\beta \in (K \cup H) \setminus (K \circledast H)$, *i.e.*, $\beta \in \sigma((K \cup H)^{\perp\!\!\perp}\perp)$. So $\beta \in (X \cap \sigma((K \cup H)^{\perp\!\!\perp}\perp))$. Therefore, $(X \cap \sigma((K \cup H)^{\perp\!\!\perp}\perp)) \neq \varnothing$.

Part B: "$\circledast_\sigma$" is equal to "$\circledast$".

Due to **inclusion** and from the definition of $\sigma((K \cup H)^{\perp\!\!\perp}\perp)$ we conclude that $K \circledast H = K \circledast_\sigma H$.

3.4 Partial Meet Merge

Definition 5: Let K be a set of sentences and α a sentence. Then $K^{\perp}\alpha$ is the set of all X such that $X \in K^{\perp}\alpha$ if and only if $X \subseteq K$, $X \nvdash \alpha$ and if $X \subset X' \subseteq K$ then $X' \vdash \alpha$. The set $H^{\perp}\alpha$ is called the *remainder set* of K with respect to α, and its elements are called the *α-remainders of K*.

In order to define the partial meet version of this operator, we need a general selection function.

Definition 6: Let K and H be two belief bases. A *general selection function* for K and H is a function "γ" ($\gamma : \mathbf{2}^{\mathbf{2}^{\mathcal{L}}} \Rightarrow \mathbf{2}^{\mathbf{2}^{\mathcal{L}}}$) such that for any $K, H \subseteq \mathcal{L}$, it holds that:
1) $\gamma((K \cup H)^{\perp}\perp) \subseteq (K \cup H)^{\perp}\perp$.
2) $\gamma((K \cup H)^{\perp}\perp) \neq \varnothing$.

Since every set $X \subseteq \mathcal{L}$ contains a consistent subset then $X^{\perp}\perp$ is always non-empty.

Definition 7: Let K, H, K', H' be belief bases and "γ" a general selection function. Then γ is an *equitable selection function* if $(K \cup H)^{\perp\!\!\perp}\perp = (K' \cup H')^{\perp\!\!\perp}\perp$ implies that $(K \cup H) \setminus \cap\gamma((H \cup H)^{\perp}\perp) = (K' \cup H') \setminus \cap\gamma((K' \cup H')^{\perp}\perp)$.

The intuition behind this definition is that, if the set of minimally inconsistent subsets of $K \cup H$ is equal to the set of minimally inconsistent subsets of $K \cup B$ then α is erased in the selection of $\perp$-remainders of $K \cup A$ if and only if it is erased in the selection of $\perp$-remainders of $K' \cup H'$ [8].

Definition 8: Let K and H be two belief bases and "γ" an equitable selection function for K and H. The operator "$\circledast$" of *partial meet merge* ($\circledast : \mathbf{2}^{\mathcal{L}} \times \mathbf{2}^{\mathcal{L}} \Rightarrow \mathbf{2}^{\mathcal{L}}$) is defined as $K \circledast H = \cap\gamma((K \cup A)^{\perp}\perp)$.

The mechanism of this operator is to join K and H and then eliminate from the result all possible inconsistencies by means of an equitable selection function that makes a choice among the maximally consistent subsets of $K \cup H$ and intersect them. Since this operator uses a selection function and the remainder set, we call it partial meet revision merge.

Lemma 1: The following results are useful to prove the representation theorem of partial meet merge.

a) If $A^{\perp}\perp = B^{\perp}\perp$ then $A = B$.

b) If "$\circledast$" satisfies *reversion* then it satisfies *congruence*.

Proof. The item a) is proven in [8]. We will prove the item b). Let $A \cup B = C \cup D$. Then $A \cup B$ and $C \cup D$ have the same minimallity inconsistent subsets. Due to reversion, $(A \cup B) \setminus (A \circledast B) = (C \cup D) \setminus (C \circledast D)$. Therefore, $A \circledast B = C \circledast D$.

Theorem 2: Let K and H be two belief bases. The operator "$\circledast$" is a *partial meet merge* of K and H if and only if it satisfies *inclusion*, *strong consistency*, *relevance* and *reversion*.

The proof of this theorem is left to the reader. Fuhrmann [10] found another representation theorem for this kind of operator that can be showed using the part b) of the Lemma 1.

Theorem 3: Let K and H be two belief bases. The operator "$\circledast$" is a *partial meet merge* of K and H if and only if it satisfies *inclusion*, *strong consistency*, *relevance* and *congruence*.

Corolary 1: Let K and H be two belief bases:

a) If "$\circledast$" is a *partial meet merge* for K and H then "$\circledast$" is a *kernel merge* for K and H.

b) If "$\circledast$" is a partial meet merge for K and H then it satifies *vacuity* and *symmetry*.

c) If "$\wedge$" is a *trivial merge* for K and H and "$\circledast$" is a *kernel merge* or *partial meet merge* for K and H. Then $K \wedge H \subseteq K \circledast H$.

The proof of this corollary is trivial. A partial meet merge is a kernel merge because relevance implies core retainment.[1] Looking the constructions, is easy to show that every partial meet (or kernel) merge satisfies vacuity and symmetry. The latest observation is trivial by definition of partial meet merge.

4 APPLICATION ON MULTI-AGENT SYSTEMS

It is in the field of Cognitive Robotics where belief revision finds its most appropriate application. An intelligent agent is a physical or virtual entity in which certain general characteristics are recognized. It should be capable of acting on its environment in a flexible, autonomous manner, including the ability to communicate with similar entities. Furthermore, its behavior should be controlled by a set of tendencies. In designing agents with these characteristics, we need to devise an architecture in which the components of the agent are described and the interactions among these components are defined.

[1]More relations between partial meet and kernel operator can be found in [6].

The *Belief-Desire-Intention* (BDI) model has its roots in the philosophical tradition of understanding practical reasoning. In the BDI model an agent has a set of *Beliefs*, a set of *Desires* and a set of *Intentions* [2]. Intentions play a crucial role in the practical reasoning process. Perhaps the most obvious property of intentions is that they tend to lead to action. Intentions drive means-ends reasoning, constrain future deliberation, persist, and influence beliefs upon which future practical reasoning is based.

In order to design agents that have these (and other) desirable properties, a given model must be followed. The model's characteristics will greatly depend on the environment that the agent occupies, and on the type of behavior that is expected from it. These models carry the name of *architectures*, and they define a set of components that interrelate in order to generate the agent's behavior.

The basic BDI model needs to be complemented with two mechanisms: one for reasoning about intentions, and one for revising beliefs upon perception. In this work, we propose the use of Defeasible Logic Programming for knowledge representation and reasoning about beliefs and intentions, and we introduce a *non-prioritized* belief merge function that changes the agent's beliefs.

4.1 DeLP Framework

Defeasible Logic Programming (abbreviated DeLP) will provide a representation language and a reasoning mechanism. Consequently, the agent's beliefs will be represented as a defeasible logic program. Here, we will introduce DeLP in an intuitive manner. The reader is referred to [11, 12] for a complete presentation of DeLP.

In DeLP, a program $\mathcal{P}$ is a pair $[K, \Delta]$ where K is a set of *facts* and *strict rules* (undefeasible beliefs) and Δ is a set of *defeasible rules*. Facts are represented by literals (ground atoms or negated ground atoms that use strong negation "$\neg$"), strict rules are denoted "$L_0 \leftarrow L_1, \ldots, L_n$", and defeasible rules are denoted"$L_0 \prec L_1, \ldots, L_n$". In both types of rules, the head L_0 is a literal, and the body $L_1, \ldots, L_n$ is a finite non-empty conjunction of literals. Defeasible rules are used to represent tentative information that may be used if nothing can be posed against it, whereas strict rules and facts represent non-defeasible knowledge. Thus, a defeasible rule represents a weak connection between the body and the head, and should be read as "reasons to believe in $L_1, \ldots, L_n$ provide reasons to believe in L_0". These rules, by representing weak connections, equip the representation language with a natural device to characterize a link between information that could be invalidated when more information comes into play [24, 25]. However, since defeasible rules represent tentative information, defeasible derivations for contradictory literals are allowed from $K \cup \Delta$. When this happens, a *defeasible argumentation* formalism is used for deciding which literal prevails as warranted. In DeLP a literal L is *warranted* if there exists a non-defeated argument $\mathcal{A}$ supporting L. A set of defeasible rules $\mathcal{A} \subseteq \Delta$ is an *argument* for a literal L if $K \cup \mathcal{A}$ is a minimal consistent set that entails L.

When we have two agents deliberating or cooperating, it is necessary a way to establish the *common knowledge* for arguments construction. We propose to use merge operators in order to get the facts and strict rules that can be used in their arguments. We will assume that two agents have associated two defeasible logic programs $[K_{\mathbf{O}}, \Delta_{\mathbf{O}}]$ and $[K_{\mathbf{P}}, \Delta_{\mathbf{P}}]$ respectively. We propose that these agents may obtain arguments using their respective defeasible rules sets and a subset of their merged undefeasible beliefs. Let "$\circledast$" be a kernel merge operator for $K_{\mathbf{O}}$ and $K_{\mathbf{P}}$. Then, two agents could deliberate obtaining arguments from $[(K_{\mathbf{O}} \circledast K_{\mathbf{P}}) \cap K_{\mathbf{O}}, \Delta_{\mathbf{O}}]$ and $[(K_{\mathbf{O}} \circledast K_{\mathbf{P}}) \cap K_{\mathbf{P}}, \Delta_{\mathbf{P}}]$ respectively.

Example 3: Suppose that we have two agents **O** and **P** with these defeasible logic programs associated:[2]

$$K_{\mathbf{O}} = \left\{ \begin{array}{l} peng(tweety) \\ bird(X) \leftarrow peng(X) \\ animal(X) \leftarrow bird(X) \end{array} \right\} \text{ and } \Delta_{\mathbf{O}} = \{\neg flies(X) \prec peng(X), bird(X)\}$$

$$K_{\mathbf{P}} = \left\{ \begin{array}{l} \neg peng(tweety) \\ bird(tweety) \\ animal(X) \leftarrow bird(X) \end{array} \right\} \text{ and } \Delta_{\mathbf{P}} = \{flies(X) \prec bird(X)\}$$

Suppose that **O** and **P** are deliberating about properties of Tweety. **O** may obtain the argument $\mathcal{A}_1 = \{\neg flies(tweety) \prec peng(tweety), bird(tweety)\}$ for $\neg flies(tweety)$, whereas **P** may obtain the argument $\mathcal{A}_2 = \{flies(tweety) \prec bird(tweety)\}$ for $flies(tweety)$. The problem is that the arguments can not be compared because they are supported by inconsistent (undefeasible) beliefs. The agent **P** can not accept the argument $\mathcal{A}_1$ because it is supported by the fact $peng(tweety)$ which is inconsistent with its undefeasible beliefs $K_{\mathbf{P}}$. Our approach is that the two agents may support their arguments with a *consistent* subset of their respective undefeasible beliefs; this consistent subset will be a subset of the merged undefeasible sets.

Now, suppose that we want to obtain a kernel merge of $K_{\mathbf{O}}$ and $K_{\mathbf{P}}$. We have to join $K_{\mathbf{O}}$ and $K_{\mathbf{P}}$ and then erase all possible inconsistencies from the resulting set.

$$K_{\mathbf{O}} \cup K_{\mathbf{P}} = \left\{ \begin{array}{l} peng(tweety) \\ \neg peng(tweety) \\ bird(tweety) \\ bird(X) \leftarrow peng(X) \\ animal(X) \leftarrow bird(X) \end{array} \right\}$$

Clearly, the minimal inconsistent subset of $K_{\mathbf{O}} \cup K_{\mathbf{P}}$ is $\{peng(tweety), \neg peng(tweety)\}$ and an incision function will cut this $\perp$-kernel. Then, we have several cases:

1. The incision function cut the sentence $\neg peng(tweety)$.

$$K_{\mathbf{O}} \circledast K_{\mathbf{P}} = \left\{ \begin{array}{l} peng(tweety) \\ bird(tweety) \\ bird(X) \leftarrow peng(X) \\ animal(X) \leftarrow bird(X) \end{array} \right\}$$

   The agent **O** may obtain the argument $\mathcal{A}_1$ for $\neg flies(tweety)$ based on $[(K_{\mathbf{O}} \circledast K_{\mathbf{P}}) \cap K_{\mathbf{O}}, \Delta_{\mathbf{O}}]$ and the agent **P** may obtain the argument $\mathcal{A}_2$ for $flies(tweety)$ based on $[(K_{\mathbf{O}} \circledast K_{\mathbf{P}}) \cap K_{\mathbf{P}}, \Delta_{\mathbf{P}}]$ Now, both arguments are comparable because they are supported by a consistent set of undefeasible beliefs. Using specificity criteria, the literal $\neg flies(tweety)$ is warranted because is supported by the argument $\mathcal{A}_1$ which defeat to the argument $\mathcal{A}_2$. Therefore, the two agents agree with $\neg flies(tweety)$.

2. The incision function cut the sentence $peng(tweety)$.

$$K_{\mathbf{O}} \circledast K_{\mathbf{P}} = \left\{ \begin{array}{l} \neg peng(tweety) \\ bird(tweety) \\ bird(X) \leftarrow peng(X) \\ animal(X) \leftarrow bird(X) \end{array} \right\}$$

[2]Strict and defeasible rules are **ground**. However, following the usual conventions, this example uses “schematic rules” with variables.

in which case the agent **P** may obtain the argument $\mathcal{A}_2$ for $flies(tweety)$ but the agent **O** can not obtain the argument $\mathcal{A}_1$ any more. Therefore, the literal $flies(tweety)$ is warranted and the two agents agree with $flies(tweety)$.

3. The incision function cut both sentences. In this case, the result is the same that the case 2.

5 CONCLUSION AND FUTURE WORK

In this work, we define several non-prioritized merge operators. In a constructive approach, a mechanism for change was explicitly constructed. In a black box approach, we specified the postulates that an operator should have (irrespective of how it is constructed). Representation theorems join the two approaches, improving our understanding both of the constructions and of the postulates.

The new merge operators can be used for changing the beliefs of two agents. This operator has the desirable property of conserving as much information as possible since the knowledge is represented using the language of Defeasible Logic Programming. The combination of both frameworks results in a formalism for knowledge representation and reasoning about beliefs, allowing that two agents can deliberate or negotiate using arguments supported by consistent subsets of their own beliefs.

As a future work, we will explore more properties of this operator and we will develop multi-agent applications. The intention is to generalize the merging operator to more than two belief bases, allowing the use of comparison methods such as specificity [3], explanations [8], and distances between models [21, 22].

REFERENCES

[1] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic*, 50:510–530, 1985.

[2] Michael Bratman. *Intentions, Plans, and Practical Reasoning*. Harvard University Press (Reprinted by CSLI Publications), Cambridge, Massachusetts, 1987.

[3] Mukesh Dalal. Investigations into a Theory of Knowledge Base Revision: Preliminary Report. *Seventh National Conference on Artificial Intelligence*, (AAAI–88):475–479, 1988.

[4] Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[5] Jon Doyle. Reason Maintenance and Belief Revision: Foundations versus Coherence Theories. In **Belief Revision** [14], pages 29–51.

[6] Marcelo Alejandro Falappa, Eduardo Leopoldo Fermé, and Gabriele Kern-Isberner. On the Logic of Theory Change: Relations between Incision and Selection Functions. In *Proceedings of ECAI-2006*, page in press, 2006.

[7] Marcelo Alejandro Falappa, Alejandro Javier García, and Guillermo Ricardo Simari. Belief Dynamics and Defeasible Argumentation in Rational Agents. In *Proceedings of NMR-2004*, pages 164–170, 2004.

[8] Marcelo Alejandro Falappa, Gabriele Kern-Isberner, and Guillermo Ricardo Simari. Belief Revision, Explanations and Defeasible Reasoning. *Artificial Intelligence Journal*, 141:1–28, 2002.

[9] Eduardo L. Fermé and Sven Ove Hansson. Selective Revision. *Studia Logica*, 63:331–342, 1998.

[10] André Fuhrmann. *An Essay on Contraction*. Studies in Logic, Language and Information, CSLI Publications, Stanford, California, 1997.

[11] Alejandro Javier García and Guillermo Ricardo Simari. Parallel Defeasible Argumentation. *Journal of Computer Science and Technology Special Issue: Artificial Intelligence and Evolutive Computation*, 1(2):45–57, 1999.

[12] Alejandro Javier García and Guillermo Ricardo Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.

[13] Peter Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. The MIT Press, Bradford Books, Cambridge, Massachusetts, 1988.

[14] Peter Gärdenfors. *Belief Revision*. Gärdenfors, Cambridge University Press, 1992.

[15] Sven Ove Hansson. *Belief Base Dynamics*. PhD thesis, Uppsala University, Department of Philosophy, Uppsala, Sweden, 1991.

[16] Sven Ove Hansson. Belief Contraction Without Recovery. *Studia Logica*, 50:251–260, 1991.

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

[18] Sven Ove Hansson. Semi-Revision. *Journal of Applied Non-Classical Logic*, 7:151–175, 1997.

[19] Sven Ove Hansson. *A Textbook of Belief Dymanics: Theory Change and Database Updating*. Kluwer Academic Publishers, 1999.

[20] Sven Ove Hansson, Eduardo Fermé, John Cantwell, and Marcelo Falappa. Credibility Limited Revision. *The Journal of Symbolic Logic*, 66(4):1581–1596, 2001.

[21] Hirofumi Katsuno and Alberto Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

[22] Hirofumi Katsuno and Alberto Mendelzon. On the Difference Between Updating a Knowledge Database and Revising It. In **Belief Revision** [14], pages 183–203.

[23] David Makinson. Screened Revision. *Theoria: Special Issue on Non-Prioritized Belief Revision*, 63:14–23, 1997.

[24] Donald Nute. Defeasible Logic. In D.M. Gabbay, C.J. Hogger, and J.A.Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol 3*, pages 355–395. Oxford University Press, 1994.

[25] Guillermo R. Simari and Ron P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.

[26] Gerhard Weiss, editor. *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachussetts, 1999.