# Weakening Conflicting Information for Iterated Revision and Knowledge Integration

**Salem Benferhat** and **Souhila Kaci**
I.R.I.T.–C.N.R.S. Université Paul Sabatier
118 route de Narbonne
31062 Toulouse Cedex 4, France
{benferhat,kaci}@irit.fr

**Daniel Le Berre** and **Mary-Anne Williams**
Business & Technology Research Laboratory
The University of Newcastle
Newcastle, NSW 2308, Australia
{daniel,maryanne}@cafe.newcastle.edu.au

## Abstract

The ability to handle exceptions, to perform iterated belief revision and to integrate information from multiple sources are essential skills for an intelligent agent. These important skills are related in the sense that they all rely on resolving inconsistent information. We develop a novel and useful strategy for conflict resolution, and compare and contrast it with existing strategies. Ideally the process of conflict resolution should conform with the principle of Minimal Change and should result in the *minimal loss* of information. Our approach to minimizing the loss of information is to weaken information involved in conflicts rather than completely removing it. We implemented and tested the relative performance of our new strategy in three different ways. We show that it retains more information than the existing Maxi-Adjustment strategy at no extra computational cost. Surprisingly, we are able to demonstrate that it provides a computationally effective compilation of the lexicographical strategy, a strategy which is known to have desirable theoretical properties.

## 1 Introduction

Information modeling and management is a fundamental activity of intelligent systems. Intelligent systems require robust and sophisticated information management capabilities such as exception handling, iterated revision and the integration of information. In this paper we develop a novel and useful strategy for conflict resolution which can be applied to exception handling, iterated revision, and information integration. Throughout we assume that the available information is given as ordered knowledge bases, i.e. a ranking of information as logical sentences. Solving conflicts in our context means computing a consistent knowledge base. One well known system that can deal with conflicts in knowledge bases is the so-called *Adjustment* procedure [Williams, 1994]. In essence, Adjustment propagates as many highly ranked formulas as possible, and ignores information at and below the highest rank where an inconsistency is found. The main advantage of this system is its computational efficiency. For example, it only needs at most $Log_2 n$ calls to a SAT solver to build the consistent knowledge base where $n$ is the number of ranks in the knowledge base. The obvious disadvantage of Adjustment, however, is that it can remove more formulae than is necessary to restore the consistency of the knowledge base if the independence of information is not made explicit. In order to overcome this shortcoming another strategy called *Maxi-Adjustment* was introduced [Williams, 1996] and implemented [Williams and Sims, 2000]. Maxi-Adjustment has proved to be a useful strategy for real world applications e.g. software engineering [Williams, 1998], information filtering [Lau *et al.*, 2000] and intelligent payment systems [Wong and Lau, 2000]. The main idea of Maxi-Adjustment is to solve conflicts at each rank of priority in the knowledge base. This is done, incremently, starting from the information with highest rank. When inconsistency is encountered in the knowledge base, then all formulas in the rank responsible for the conflicts are removed. The other formulas are kept, and the process continues to the next rank.

Clearly Maxi-Adjustment keeps more information than Adjustment, since it does not stop at the first rank where inconsistency is met. Even though Maxi-Adjustment propagates more information than Adjustment, one can still argue that Maxi-Adjustment removes too much information because it adopts a sceptical approach to the way it removes the conflict sets at each rank.

The purpose of this paper is to describe a significant improvement to Maxi-Adjustment. We call this system *Disjunctive Maxi-Adjustment*, and denote it by DMA. The idea is similar to Maxi-Adjustment, except that information is weakened instead of being removed when conflicts are detected. So instead of removing all formulas involved in conflicts, as it is done in Maxi-Adjustment, DMA takes their disjunctions pairwise. If the result is consistent, then we move to the next rank. If the result is still inconsistent, then we replace the formulas in conflicts by all possible disjunctions involving $3$ formulas in the conflict sets and again if the result is consistent we move to the next layer, and if it is inconsistent we consider disjunctions of size $4, 5$, etc. The only case where all formulas responsible for conflicts are removed is when the disjunction of all these formulas is inconsistent with the higher priority information.

This paper focuses on the DMA strategy from the theoretical and experimental perspectives. In particular,

- We show that DMA is equivalent to the well known lexicographical strategy [Benferhat *et al.*, 1993; Lehmann, 1995]. More precisely, we show that for an inconsistent base $K$ if $\delta_{DMA}(K)$ is the classical base obtained using DMA, and $\delta_{Lex}(K)$ is the set of all lexicographically maximal consistent subbases of $K$, then:

$$\forall \psi, \delta_{DMA}(K) \vdash \psi \text{ iff } \forall A \in \delta_{Lex}(K), A \vdash \psi.$$

In other words, we obtain the surprising and computationally useful result that DMA provides a "compilation" of lexicographical systems.

- It is well known that computing conflicts is a hard task, and we are able to show that DMA works even if the conflicts are not explicitly computed. For this, we propose an alternative, but equivalent, approach to DMA called whole-DMA where disjunctions are built on the whole stratum when we meet inconsistency instead of only on the conflicts.
- We also propose another equivalent alternative to DMA called iterative-DMA where instead of considering disjunctions of size (3,4, etc) on the initial set of conflicts, we only compute disjunctions of size 2 but on new sets of conflicts.
- Lastly, we compare these different implementations of DMA experimently, and contrast their applicability.

# 2 Ordered information in Spohn's OCF framework

We consider a finite propositional language denoted by $\mathcal{L}$. Let $\Omega$ be the set of interpretations. $\vdash$ denotes the classical consequence relation, Greek letters $\phi, \psi, ...$ represent formulas.

We use Spohn's ordinal conditional function [Spohn, 1988] framework, which is also known as the Kappa function framework.

At the semantic level, the basic notion of Spohn's ordinal conditional function framework is a distribution called an OCF, denoted by $\kappa$, which is a mapping from $\Omega$ to $\mathcal{N}$, such that $\exists \omega, \kappa(\omega) = 0$. $\mathcal{N}$ is the set of natural numbers. $\kappa(\omega)$ can be viewed as the degree of impossibility of $\omega$.

By convention, $\kappa(\omega) = 0$ means that nothing prevents $\omega$ from being the real world, and $\kappa(\omega) = +\infty$ means that $\omega$ is certainly not the real world[1]. The lower $\kappa(\omega)$ is, the more expected it is, i.e. if $\kappa(\omega) < \kappa(\omega')$ then $\omega$ is said to be more plausible than $\omega'$.

In practice, OCF distributions over all possible worlds are not available, however a ranked knowledge base provides a compact representation of an OCF distribution [Williams, 1994]. Since we will be working with ranked knowledge bases throughout, we define a knowledge base to be ranked. In particular, a knowledge base is a set of weighted formulas of the form $K = \{(\phi_i, k_i) : i = 1, \ldots, n\}$ where $\phi_i$ is a classical formula and $k_i$ is a positive number representing the level of priority of $\phi_i$. The higher $k_i$, the more important the formula $\phi_i$.

Given $K$, we can generate a unique OCF distribution, denoted by $\kappa_K$, such that all the interpretations satisfying all the formulae in $K$ will have the lowest value, namely $0$, and the other interpretations will be ranked with respect to the highest formulae that they falsify. Namely:

**Definition 1** $\forall \omega \in \Omega$,

$$\kappa_K(\omega) = \begin{cases} 0 & \forall (\phi_i, k_i) \in K, \omega \models \phi_i \\ max\{k_i : (\phi_i, k_i) \in K \; and \; \omega \not\models \phi_i\} & otherwise. \end{cases}$$

Then, given $\kappa_K$ associated with a knowledge base $K$, the models of $K$ are the interpretations $\omega$ s.t. $\kappa_K(\omega) = 0$.

[1] Note that the notion of impossible worlds ($+\infty$) does not exist in original works of Spohn.

# 3 Adjustment and Maxi-Adjustment

## 3.1 Stratified vs ranked knowledge base

We have seen that ranked information is represented by means of knowledge bases of the form $K = \{(\phi_i, k_i) : i = 1, \ldots, n\}$. We sometimes also represent this base $K$ in a stratified form as follows: $K = \{S_1, \ldots, S_n\}$ where $S_i$ $(i = 1, \ldots, n)$ contains classical formulas of $K$ having the same rank and which are more reliable than formulas of $S_j$ for $j > i$. So the lower the stratum, the higher the rank.

In this representation, subbases are also stratified. That is, if $A$ is a subbase of $K = \{S_1, \ldots, S_n\}$, then $A = \{A_1, \ldots, A_n\}$ such that $A_j \subseteq S_j$, $j = 1, \ldots, n$. ($A_j$ may be empty).

Conversely, we can represent a stratified base $K = \{S_1, \ldots, S_n\}$ using a weighted knowledge base by associating formulas of each strata $S_i$ to the same rank $k_i$. These ranks should be such that $k_1 > \ldots > k_n$.

Let us now introduce the notion of conflicts and kernel which will prove useful in the subsequent discussion:

**Definition 2** *Let $K = \{S_1, \ldots, S_n\}$ be a stratified base. A conflict in $K$, denoted by $C$, is a subbase of $K$ such that:*

- *$C \vdash \perp$ (inconsistency),*
- *$\forall \phi, \phi \in C$, $C - \{\phi\} \nvdash \perp$ (minimality).*

**Definition 3** *Let $\mathcal{C}$ be the set of all possible conflicts in $K$. We define the kernel of $K$, denoted by $kernel(K)$, as the set of formulas of $K$ which are involved in at least one conflict in $\mathcal{C}$ i.e., $kernel(K)$ is the union of all conflicts in $K$.*

Formulas in $K$ which are not involved in any conflict in $K$ are called *free* formulas.

## 3.2 The problem

Our aim in this paper is to address the problem of identifying conflicts for the purposes of drawing plausible inferences from inconsistent knowledge bases, iterated revision and information integration. Our technique for resolving conflicts can be used: (i) to build a transmutation for iterated belief revision [Williams, 1994] where the new information can be incorporated into any rank, and (ii) for theory extraction [Williams and Sims, 2000] which provides a natural and puissant mechanism for merging conflicting information. Without loss of generality we focus on a particular case of revision where some new information $\varphi$ is added to some ranked knowledge base $K$. Namely, given a knowledge base $K$, and a new formula $\varphi$ we compute $\delta(K \cup \{(\varphi, +\infty)\})$, the classical (not stratified) consistent subbase of $K \cup \{(\varphi, +\infty)\}$. Then, $\psi$ is said to be a plausible consequence of $K \cup \{(\varphi, +\infty)\}$ iff $\delta(K \cup \{(\varphi, +\infty)\}) \vdash \psi$. In the rest of this paper we simply write $K_\varphi$ instead of $K \cup \{(\varphi, +\infty)\}$. In a stratified form we write $\{S_0, S_1, \ldots, S_n\}$ where $S_0 = \{\varphi\}$. We briefly recall two important methods to compute $\delta(K \cup \{(\varphi, +\infty)\})$: *Adjustment* and *Maxi-Adjustment*. We will illustrate them using a simple example. We point the reader to [Williams, 1994; 1996] for more details.

### 3.3 Adjustment

From a syntactical point of view, the idea of Adjustment is to start with formulas having the highest rank in $K_\varphi$ and to add as many prioritized formulas as possible while maintaining consistency. We stop at the highest rank (or the lowest stratum) where we meet inconsistency called the inconsistency rank of $K_\varphi$, denoted by $Inc(K_\varphi)$.

Note that a more efficient binary search based algorithm which only needs $Log_2 n$ consistency checks has been developed and implemented[2] [Williams and Sims, 2000]. The selected base will be denoted by $\delta_A(K_\varphi)$. Note that the process of selecting the consistent base using the Adjustment for new pieces of information placed in the highest rank is identical to that used in possibilistic logic [Dubois *et al.*, 1994].

One can easily see that this is not a completely satisfactory way to deal with the inconsistency since formulas with rank lower than $Inc(K_\varphi)$ are ignored even if they are consistent with the selected base.

A formula $\psi$ is said to be an Adjustment consequence of $K_\varphi$, denoted by $K_\varphi \vdash_A \psi$, if $\delta_A(K_\varphi) \vdash \psi$. One important property of Adjustment is that it is semantically well defined. More precisely, we have the following soundness and completeness result: $K_\varphi \vdash_A \psi$ iff $\forall \omega \in Pref(\kappa_{K_\varphi}), \omega \models \psi$, where $Pref(\kappa_{K_\varphi})$ is the set of interpretations which satisfy $\varphi$ and have minimal rank in the OCF distribution $\kappa_{K_\varphi}$ given by Definition 1.

**Example 1** *Let $K = \{S_1, S_2, S_3\}$ be such that*
$S_1 = \{(\neg a \vee \neg b \vee c, 3), (\neg d \vee c, 3), (\neg e \vee c, 3)\}$,
$S_2 = \{(d, 2), (e, 2), (f, 2), (\neg f \vee \neg g \vee c, 2)\}$ *and*
$S_3 = \{(a, 1), (b, 1), (g, 1), (h, 1)\}$. *Let* $\varphi = \neg c$.
*First, we have* $\delta_A(K_{\neg c}) = \{\neg c\}$.
*There is no conflict in* $\delta_A(K_{\neg c}) \cup S_1$ *then*
$\delta_A(K_{\neg c}) \leftarrow \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c\}$.
*Now,* $S_2$ *contradicts* $\delta_A(K_{\neg c})$ *due to the conflicts* $\{d, \neg d \vee c, \neg c\}$ *and* $\{e, \neg e \vee c, \neg c\}$. *Then, we do not add the stratum* $S_2$ *and the computation of* $\delta_A(K_{\neg c})$ *is achieved, and we get* $\delta_A(K_{\neg c}) = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c\}$.
*Note that* $\delta_A(K_{\neg c}) \not\vdash h$, *even if* $h$ *is not involved in any conflict in* $K_{\neg c}$.

### 3.4 Maxi-Adjustment

*Maxi-Adjustment* [Williams, 1996] was developed to address the problem of discarding too much information for applications like software engineering [Williams, 1998] and information filtering [Lau *et al.*, 2000].

[2] http://cafe.newcastle.edu.au/systems/saten.html

The idea in Maxi-Adjustment also involves selecting one consistent subbase from $K$ denoted by $\delta_{MA}(K_\varphi)$. The difference is that it does not stop at the first rank where it meets inconsistency. Moreover, conflicts are solved rank by rank. We start from the first rank and take the formulas of $S_1$ which do not belong to any conflict in $\{\varphi\} \cup S_1$. Let $S_1'$ be the set of these formulas. Then, we move to the next rank and add all formulas which are not involved in any conflict in $S_1' \cup S_2$, and so on. It is clear that Maxi-Adjustment keeps more formulas than the Adjustment.

**Example 1**(using Maxi-Adjustment)
First, we have $\delta_{MA}(K_{\neg c}) = \{\neg c\}$.
There is no conflict in $\delta_{MA}(K_{\neg c}) \cup S_1$ then $\delta_{MA}(K_{\neg c}) \leftarrow \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c\}$.
Now, $S_2$ contradicts $\delta_{MA}(K_{\neg c})$ due to the conflicts $\{d, \neg d \vee c, \neg c\}$ and $\{e, \neg e \vee c, \neg c\}$. Then, we do not add the clauses from $S_2$ involved in these conflicts: $\delta_{MA}(K_{\neg c}) \leftarrow \delta_{MA}(K_{\neg c}) \cup \{f, \neg f \vee \neg g \vee c\}$.
Now, $S_3$ contradicts $\delta_{MA}(K_{\neg c})$ due to the conflicts $\{a, b, \neg a \vee \neg b \vee c, \neg c\}$ and $\{f, g, \neg f \vee \neg g \vee c, \neg c\}$. Since all the clauses, except $h$, from the stratum $S_3$ are involved in one conflict, we only add $h$ to $\delta_{MA}(K_{\neg c})$. Finally, we get: $\delta_{MA}(K_{\neg c}) = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c, f, \neg f \vee \neg g \vee c, h\}$.
Note that $\delta_{MA}(K_{\neg c}) \vdash h$.

## 4 Disjunctive Maxi-Adjustment

Although Maxi-Adjustment retains more information than Adjustment, it can still be argued that it is too cavalier in the way it solves the conflicts.

In this section, we propose a new strategy which is a significant improvement of Maxi-Adjustment. The computation of the consistent base is essentially the same as in Maxi-Adjustment, the only difference is when we meet an inconsistency at some rank, instead of removing all the formulas involved in the conflicts at this rank we weaken them, by replacing them by their pairwise disjunctions. If the result is consistent then we move to the next rank, else we replace these formulas by their possible disjunctions of size 3. If the result is consistent then we move to the next rank, else we add the disjunctions of size 4 of these formulas, and so on. We summarize this process in Algorithm 1:

**Notation:** $d_k(C)$ is the set of all possible disjunctions of size $k$ between formulas of $C$. If $k > |C|$ then $d_k(C) = \emptyset$.

**Example 1** (using DMA)
First, we have $KB = \{\neg c\}$.
There is no conflict in $KB \cup S_1$ then
$KB \leftarrow \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c\}$.
Now, $S_2$ contradicts $KB$ due to the conflicts $\{d, \neg d \vee c, \neg c\}$ and $\{e, \neg e \vee c, \neg c\}$. We do not add the clauses from $S_2$ involved in these conflicts: $KB \leftarrow KB \cup \{f, \neg f \vee \neg g \vee c\}$.
Now we create all the possible disjunctions of size 2 with $C = \{d, e\}$: $d_2(C) = \{d \vee e\}$. Since $KB \cup d_2(C)$ is inconsistent, and we cannot create larger disjunctions, we do not add anything from $S_2$ to $KB$.

**Algorithm 1:** DMA($K$, $\varphi$)

Data: a stratified knowledge base $K = \{S_1, \ldots, S_n\}$ ;
a new sure formula: $\varphi$ ;

Result: a consistent subbase $\delta_{DMA}(K_\varphi)$

**begin**
  $KB \leftarrow \{\varphi\}$ ;
  **for** $i \leftarrow 1$ *to* $n$ **do**
    **if** $KB \cup S_i$ *is consistent* **then** $KB \leftarrow KB \cup S_i$
    **else**
      Let $C$ be the subset of $S_i$ in kernel of $KB \cup S_i$;
      $KB \leftarrow KB \cup \{\phi : \phi \in S_i \ and\ \phi \notin C\}$;
      $k \leftarrow 2$ ;
      **while** $k \leq |C|$ *and* $KB \cup d_k(C)$ *is inconsistent* **do**
        $k \leftarrow k + 1$;
      **if** $k \leq |C|$ **then** $KB \leftarrow KB \cup d_k(C)$ ;
  **return** $KB$
**end**

Please note at this rank, we do not add more information than Maxi-Adjustment.
Now, $S_3$ contradicts $KB$ due to the conflicts $\{a, b, \neg a \vee \neg b \vee c, \neg c\}$ and $\{f, g, \neg f \vee \neg g \vee c, \neg c\}$. $h$ is not involved in any conflict. Then, $KB \leftarrow KB \cup \{h\}$.
We now create all the possible pairwise disjunctions with $C = \{a, b, g\}$: $d_2(C) = \{a \vee b, a \vee g, b \vee g\}$. Since $KB \cup d_2(C)$ is inconsistent, we create $d_3(C) = \{a \vee b \vee g\}$. Since $KB \cup d_3(C)$ is consistent, we add $d_3(C)$ to $KB$ and the algorithm stops.
Then $\delta_{DMA}(K_\varphi) = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c, f, \neg f \vee \neg g \vee c, h, a \vee b \vee g\}$
which is equivalent to $\{\neg c, \neg a \vee \neg b, \neg d, \neg e, f, \neg g, h, a \vee b\}$.
DMA keeps more information from the last stratum than Maxi-Adjustment does.

**Definition 4** *A formula $\psi$ is said to be a DMA consequence of $K$ and $\varphi$, denoted by $K_\varphi \vdash_{DMA} \psi$, if it is inferred from $\delta_{DMA}(K_\varphi)$. Namely, $K_\varphi \vdash_{DMA} \psi$ iff $\delta_{DMA}(K_\varphi) \vdash \psi$.*

## 5 Two other implementations of DMA

In the previous section we have shown a way to compute $\delta_{DMA}(K_\varphi)$ using the computation of the kernel. In this section, we propose two alternative ways to compute $\delta_{DMA}(K_\varphi)$. The first approach, called whole-DMA(K,$\varphi$), does not compute the kernel. The main motivation for this alternative is that computing the kernel is in general hard. For the second approach, called iterative-DMA(K,$\varphi$), when inconsistency is (again) met after weakening the kernel, then rather than weakening the original kernel by considering its disjunctions of size 3, we only weaken the newly computed kernel obtained by considering disjunctions of size 2. The motivation of this approach is to reduce the size of added (disjunctions) formulas.

### 5.1 Whole Disjunctive Maxi-Adjustment

We propose a slightly modified version of the DMA algorithm. The idea is that when $KB \cup S_i$ is inconsistent, instead of considering all possible disjunctions of size $j$ of elements of $S_i$ which are in $kernel(KB \cup S_i)$, we consider all possible disjunctions of size $j$ of $S_i$ without computing a kernel. This is justified by the following proposition:

**Proposition 1** *Let $KB \cup S$ be inconsistent. Let $C$ be the subset of $S$ in $kernel(KB \cup S)$, and $F = S - C$ be the set of remaining formulas. Let $d_j(C)$ (resp. $d_j(S)$) be the set of all possible disjunctions of size $j$ from $C$ (resp. $S$). Then,*

$$KB \cup d_j(C) \cup F \equiv KB \cup d_j(S).$$

**Proof** (sketch)
Let us assume that $d_{j-1}(S) \cup KB$ is inconsistent, and show that $d_j(S) \cup KB \equiv d_j(C) \cup KB \cup F$.
It is clear that $d_j(C) \subseteq d_j(S)$. Hence it is enough to show that $KB \cup d_j(S) \vdash F$.
Let $A$ be a conflict of $KB \cup d_{j-1}(S)$, and $\{\psi_1, \ldots, \psi_n\}$ be a subset of $A$ in $d_{j-1}(S)$. Let $\varphi \in F$.
Then $\{\varphi \vee \psi_1, \ldots, \varphi \vee \psi_n\} \subseteq d_j(S)$, with $\psi_i \neq \varphi$ since $\varphi \notin A$ (because $\varphi$ is free).
Now since $KB \cup A$ is inconsistent then $KB \vdash \neg\psi_1 \vee \cdots \vee \neg\psi_n$.
Applying successive resolutions between $\{\varphi \vee \psi_1, \ldots, \varphi \vee \psi_n\}$ and $\neg\psi_1 \vee \cdots \vee \neg\psi_n$ leads to entail $\varphi$.
Hence there is no need to consider disjunctions containing free formulas since they will be subsumed. □

With the help of this proposition, the "else" block in the DMA algorithm is replaced by
**else**
  $k \leftarrow 2$
  **while** $KB \cup d_k(S_i)$ is inconsistent and $k \leq | S_i |$ **do**
    $k \leftarrow k + 1$
  **if** $k \leq| S_i |$ **then** $KB \leftarrow KB \cup d_k(S_i)$
to obtain the whole DMA algorithm.

**Example 1** (using whole DMA)
First, we have $KB = \{\neg c\}$.
$S_1$ is consistent with $KB$. Then, $KB \leftarrow KB \cup S_1$.
Now, $S_2$ contradicts $KB$. We compute all possible pairwise disjunctions with $S_2$. $d_2(S_2) = \{d \vee e, d \vee f, d \vee \neg f \vee \neg g \vee c, e \vee f, e \vee \neg f \vee \neg g \vee c\}$.
Since, $KB \cup S_2$ is inconsistent, we compute all possible disjunctions of size 3 between formulas of $S_2$. We get $d_3(S_2) = \{d \vee e \vee f, d \vee e \vee \neg f \vee \neg g \vee c\}$ which is consistent with $KB$. Then, $KB \leftarrow KB \cup d_3(S_2)$.
Now, $S_3$ is inconsistent with $KB$. We compute all possible pairwise disjunctions with $S_3$. $d_2(S_3) = \{a \vee b, a \vee g, a \vee h, b \vee g, b \vee h, g \vee h\}$ which is still inconsistent with $KB$. We have $d_3(S_3) = \{a \vee b \vee g, a \vee b \vee h, b \vee g \vee h, a \vee g \vee h\}$ which is consistent with $KB$, then $KB \leftarrow KB \cup d_3(S_3)$.
Hence, $\delta_{WDMA}(K_{\neg c}) = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c, d \vee e \vee f, d \vee e \vee \neg f \vee \neg g \vee c, a \vee b \vee g, a \vee b \vee h, b \vee g \vee h, a \vee g \vee h\}$ which is equivalent to $\{\neg c, \neg a \vee \neg b, \neg d, \neg e, f, \neg g, a \vee b, h\}$. Then, it is equivalent to $\delta_{DMA}(K_{\neg c})$.

### 5.2 Iterative Disjunctive Maxi-Adjustment

The idea of this alternative implementation of DMA is as follows: let $S_i$ be inconsistent with $KB$. Let $C$ and $F$ be the kernel and the remaining formulas of $S_i$.
Now assume that $KB \cup F \cup d_2(C)$ is still inconsistent. Then

rather than weakening $C$ again by considering disjunctions of size 3, we only weaken those formulas in $d_2(C)$ which are still responsible for conflicts. Namely, we split $d_2(C)$ into $C'$ and $F'$ which respectively represent the kernel and remaining formulas of $d_2(C)$. Then instead of taking $KB \cup F \cup d_3(C)$ as in DMA, we take $KB \cup F \cup F' \cup d_2(C')$. The algorithm becomes:

**Algorithm 2:** IDMA($K, \varphi$)

Data: a stratified knowledge base $K = \{S_1, \ldots, S_n\}$ ;
 a new sure formula: $\varphi$

Result: a consistent subbase $\delta_{IDMA}(K_\varphi)$

**begin**
- $KB \leftarrow \{\varphi\}, i \leftarrow 1$;
- **while** $i \leq n$ **do**
  - **if** $KB \cup S_i$ *is consistent* **then**
    - $KB \leftarrow KB \cup S_i$ ; $i \leftarrow i + 1$ ;
  - **else**
    - Let $C \subseteq S_i$ be in $kernel(KB \cup S_i)$ ;
    - $S_i \leftarrow \{\phi : \phi \in S_i \; and \; \phi \notin C\}$ ;
    - **if** $\mid C \mid = 1$ **then** $i \leftarrow i + 1$ **else** $S_i \leftarrow S_i \cup d_2(C)$ ;
- **return** *KB*

**end**

**Proposition 2** *Let $KB \cup F \cup d_i(C)$ be inconsistent. Then,*
$$KB \cup F \cup d_i(C) \equiv KB \cup F \cup F' \cup d_2(C'),$$
*where $F'$ and $C'$ are kernels from $d_i(C)$.*

The proof is a corollary of Prop. 1 and the following lemma:

**Lemma 1** *Let $A$ be a set of formulas. Let $B = d_i(A)$ and $C = d_{i+1}(A)$ be the set of all possible disjunctions of $A$ of size $i$ and $i + 1$ respectively. Then, $C = d_2(B)$.*

This lemma means that taking all disjunctions of size $i$, then reconsidering all disjunctions of size 2 again on the result is the same as considering all disjunctions of size $i + 1$.

**Example 1** (using iterative DMA)
First, we have $KB = \{\neg c\}$.
There is no conflict in $KB \cup S_1$. Then, $KB \leftarrow KB \cup S_1$.
$S_2$ is inconsistent with $KB$ due to the conflicts $\{\neg c, \neg d \vee c, d\}$ and $\{\neg c, \neg e \vee c, e\}$. We add $\{f, \neg f \vee \neg g \vee c\}$ to $KB$. The disjunction $d \vee e$ is still inconsistent with $KB$, then we move to $S_3$.
$S_3$ contradicts $KB$ due to the conflicts $\{a, b, \neg a \vee \neg b \vee c, \neg c\}$ and $\{f, g, \neg f \vee \neg g \vee c, \neg c\}$. $h$ is not involved in any conflict. Then, $KB \leftarrow KB \cup \{h\}$.
We now create all the possible pairwise disjunctions with $C = \{a, b, g\}$: $d_2(C) = \{a \vee b, a \vee g, b \vee g\}$.
$KB \cup d_2(C)$ is inconsistent due to the conflict $\{\neg c, \neg a \vee \neg b \vee c, f, \neg f \vee \neg g \vee c, a \vee g, b \vee g\}$. $a \vee b$ in $d_2(C)$ is not involved in the conflict, then $KB \leftarrow KB \cup \{a \vee b\}$.
Now, we take the pairwise disjunctions with $C = \{a \vee g, b \vee g\}$. $d_2(C) = \{a \vee b \vee g\}$. $KB \cup d_2(C)$ is consistent. However, there is no need to add $a \vee b \vee g$ to $KB$ since $a \vee b$ already belongs to $KB$. Hence, $\delta_{IDMA}(K_{\neg c}) = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c, f, \neg f \vee \neg g \vee c, a \vee b, h\}$ which is equivalent to $\{\neg c, \neg a \vee \neg b, \neg d, \neg e, f, \neg g, a \vee b, h\}$.
Hence, it is equivalent to $\delta_{DMA}(K_{\neg c})$.

# 6 DMA: Compilation of lexicographical inferences

The aim of this section is to show that $DMA$ is a compilation of the lexicographical system, hence it satisfies the AGM postulates [Alchourrón *et al.*, 1985]. First let us recall the lexicographical inference.

## 6.1 Lexicographical inference

The lexicographical system [Benferhat *et al.*, 1993; Lehmann, 1995] is a coherence-based approach where an inconsistent knowledge base is replaced by a set of maximally preferred consistent subbases. The preference relation between subbases is defined as follows:

**Definition 5** *Let $A = \{A_1, \ldots, A_n\}$ and $B = \{B_1, \ldots, B_n\}$ be two consistent subbases of $K$.*
*$A$ is said to be lexicographically preferred to $B$, denoted by $A >_{Lex} B$, iff*
$$\exists k \text{ s.t. } \mid A_k \mid > \mid B_k \mid \text{ and } \forall j < k, \mid A_j \mid = \mid B_j \mid.$$

Let $\delta_{Lex}(K_\varphi)$ denotes the set of all lexicographically preferred subbases of $K_\varphi$, those which are maximal w.r.t. $>_{Lex}$. Then, the lexicographical inference is defined by:

**Definition 6** *A formula $\psi$ is said to be a lexicographical consequence of $K_\varphi$, denoted by $K_\varphi \vdash_{Lex} \psi$, if it is a classical consequence of all the elements of $\delta_{Lex}(K_\varphi)$, namely*
$$\forall A \in \delta_{Lex}(K_\varphi), A \vdash \psi.$$

**Example 1** (continued)
We have $\delta_{Lex}(K_{\neg c}) = \{A, B\}$ where $A = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c, f, \neg f \vee \neg g \vee c, a, h\}$ and $B = \{\neg c, \neg a \vee \neg b \vee c, \neg d \vee c, \neg e \vee c, f, \neg f \vee \neg g \vee c, b, h\}$.
For example, we have
$K_{\neg c} \vdash_{Lex} a \vee b$ since $A \vdash a \vee b$ and $B \vdash a \vee b$.

## 6.2 Basic steps of the compilation

The aim of this section is to show that DMA is equivalent to the lexicographical system. DMA offers a clear advantage over the lexicographical system because it obviates the need to explicitly compute $\delta_{Lex}(K_\varphi)$ which may be exponential in size. Formally, we will show the following equivalence:
$$K_\varphi \vdash_{Lex} \psi \Leftrightarrow K_\varphi \vdash_{DMA} \psi \quad (1)$$
Note that $\delta_{DMA}(K_\varphi)$ is a classical consistent base.

**Example 1** (continued)
Let us first show that applying the lexicographical system on $K_{\neg c}$ gives the same results as applying DMA on $K_{\neg c}$.
Indeed, $K_{\neg c} \vdash_{Lex} \psi$ iff $A \vdash \psi$ and $B \vdash \psi$
iff $A \vee B \vdash \psi$ iff $\{\neg c, \neg a \vee \neg b, \neg d, \neg e, f, \neg g, a \vee b, h\} \vdash \psi$
(after removing subsumed formulas in $A \vee B$)
iff $\delta_{DMA}(K_{\neg c}) \vdash \psi$ iff $K_{\neg c} \vdash_{DMA} \psi$.

To show (1) we follow the following steps:
*Step 1:* we construct a new base $K'$ from $K$ s.t.
$$K_\varphi \vdash_{Lex} \psi \Leftrightarrow K'_\varphi \vdash_A \psi \quad (2)$$
Namely, applying lexicographical system on $K_\varphi$ is equivalent to applying Adjustment to $K'_\varphi$.

*Step 2:* in the second step we show that

$$K'_\varphi \vdash_A \psi \Leftrightarrow K_\varphi \vdash_{DMA} \psi \tag{3}$$

Namely, applying Adjustment to $K'_\varphi$ is equivalent to applying DMA to $K_\varphi$.

### Step 1: Constructing $K'$

In order to show the proof of (2), we need to rewrite the lexicographical system at the semantic level, which is immediate:

**Definition 7** *Let $K = \{S_1, \dots, S_n\}$. Let $\omega$ and $\omega'$ be two interpretations, and $A_\omega$, $A_{\omega'}$ be the subbases composed of all formulas of $K$ satisfied by $\omega$ and $\omega'$ respectively.*
*Then, $\omega$ is said to be lexicographically preferred to $\omega'$ w.r.t. $K$, denoted by $\omega >_{Lex,K} \omega'$, iff $A_\omega >_{Lex} A_{\omega'}$ (using Definition 5).*

**Proposition 3** *Let $\delta_{Lex}(K)$ be the set of lexicographical preferred consistent subbases of $K$.*
*Let $A_\omega$ be the set of formulas in $K$ satisfied by $\omega$. Then,*
*$i$. If $\omega$ is minimal w.r.t. $>_{Lex,K}$ then $A_\omega \in \delta_{Lex}(K)$*
*$ii$. $\forall A \in \delta_{Lex}(K), \exists \omega \models A$ s.t. $\omega$ is minimal w.r.t. $>_{Lex,K}$.*

Using Prop. 3, at the semantic level, (2) is equivalent to:

$$\kappa_{K'_\varphi}(\omega) < \kappa_{K'_\varphi}(\omega') \text{ iff } \omega >_{Lex,K_\varphi} \omega' \tag{4}$$

where $\kappa_{K'_\varphi}$ is the OCF associated to $K'_\varphi$ obtained from Definition 1.
Let us now show how to construct $K'$ from $K$ such that it satisfies (4). For this, we use two intuitive ideas.

The first idea is that Adjustment is insensitive to the number of equally reliable formulas falsified while lexicographical system is not (i.e. cardinality of conflict sets). Assume that we have a base $K = \{(\phi, i), (\psi, i)\}$ which contains two formulas with a same rank. Then, the rank (using Def. 1) associated with an interpretation $\omega$ falsifying one formula has a same rank as an interpretation falsifying two formulas. However, if we use the lexicographical system, an interpretation falsifying one formula is preferred to an interpretation falsifying two formulas. Now one can check that if we construct a knowledge base $K' = \{(\phi, i), (\psi, i), (\phi \vee \psi, 2i)\}$ from $K$ by adding the disjunction $\phi \vee \psi$ with a higher rank, then equation (4) is satisfied. So the first idea is to add disjunctions with the rank equal to the sum of ranks of formulas composing the disjunctions.

The second idea is related to the notion of compensation. To illustrate this idea, let us now consider $K = \{S_1, S_2\}$ such that $S_1 = \{\phi_1\}$ and $S_2 = \{\phi_2, \phi_3, \phi_4\}$. The intuition behind this example is to show that ranks associated with the formulas should satisfy some constraints in order to recover the lexicographical inference. Indeed, let us for instance associate the rank 2 with $\phi_1$, and the rank 1 with $\phi_2, \phi_3, \phi_4$. Let $\omega$ and $\omega'$ be two interpretations such that $A_\omega = \{\{\phi_1\}, \{\}\}$ and $A_{\omega'} = \{\{\}, \{\phi_2, \phi_3, \phi_4\}\}$. $A_\omega$ means that $\omega$ satisfies all formulas of $S_1$ but falsifies all formulas of $S_2$. $A_{\omega'}$ means that $\omega'$ satisfies all the formulas of $S_2$ but falsifies all the formulas of $S_1$. Following the suggestion of the first idea, let us add all possible disjunctions. We obtain:
$K' = \{\{(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4, 5)\}; \{(\phi_1 \vee \phi_3 \vee \phi_4, 4); (\phi_1 \vee \phi_2 \vee \phi_4, 4); (\phi_1 \vee \phi_2 \vee \phi_3, 4)\}; \{(\phi_1 \vee \phi_2, 3); (\phi_1 \vee \phi_3, 3); (\phi_1 \vee \phi_4, 3); (\phi_2 \vee \phi_3 \vee \phi_4, 3)\}; \{(\phi_1, 2); (\phi_2 \vee \phi_3, 2); (\phi_2 \vee \phi_4, 2); (\phi_3 \vee \phi_4, 2)\}; \{(\phi_2, 1); (\phi_3, 1); (\phi_4, 1)\}\}$.
We can easily check that $\kappa_{K'}(\omega) = 3$ and $\kappa_{K'}(\omega') = 2$ while $A_\omega >_{Lex,K} A_{\omega'}$. This is due to the fact that the disjunction $\phi_2 \vee \phi_3 \vee \phi_4$ has a rank higher than $\phi_1$. Hence, there is a compensation effect. So, in order to recover the lexicographical order, $\phi_1$ must have a rank strictly greater than the rank of $\phi_2 \vee \phi_3 \vee \phi_4$. A way to do this is to significantly differentiate the different ranks associated with strata. For this, we associate to each formula $(\phi_{ij}, k_i) \in S_i$ the rank $N^{k_i}$ where $N$ is very large. $N$ should be s.t. $\forall i$, $N^{k_i} > \Sigma_{j>i} N^{k_j}$. Such an $N$ always exists. It means that the rank given to a stratum must be greater than the sum of all the ranks of the less reliable strata.
Following these two ideas, $K'$ is formally constructed as follows:
Let $K = \{S_1, \dots, S_n\}$, and $\varphi$ a new sure information:

1. We define a new base $B$:
   $B = \{(\phi_{ij}, N^{k_i}) : i = 1, n \; and \; \phi_{ij} \in S_i\}$.
2. $K' = \{(D_j(B), a_j)\}$ where $D_j(B)$ is the set of all possible disjunctions of size $j$ between formulas of $B$, and $a_j$ is the sum of ranks of formulas in $D_j(B)$.

Then we have:

**Proposition 4** $K'_\varphi \vdash_A \psi$ *iff* $K_\varphi \vdash_{Lex} \psi$.

### Step 2: Adjustment on $K'_\varphi \equiv$ DMA on $K_\varphi$

The following proposition shows that the base $K'$ constructed in Step 1 allows us to recover the lexicographical system.

**Proposition 5** *Let $K = \{S_1, \dots, S_n\}$ be a stratified base, and $\varphi$ be a sure formula. Let $K'$ be a base constructed in Step 1. Then,*

$$K'_\varphi \vdash_A \psi \Leftrightarrow K_\varphi \vdash_{DMA} \psi.$$

Due to the lack of space, we skip the proof of Prop. 5 and illustrate its main ideas by an example. The idea is to simplify the computation of $\delta_A(K'_\varphi)$ until recovering $\delta_{DMA}(K_\varphi)$.

**Example 2** *Let $K = \{S_1, S_2\}$ where $S_1 = \{\neg a \vee \neg b \vee c\}$ and $S_2 = \{a, b, g\}$. Let $\varphi = \neg c$.*
*First it can be checked that*
*$\delta_{DMA}(K_{\neg c}) = \{\neg c, \neg a \vee \neg b \vee c, a \vee b, g\}$.*
*Let $N$ be a large number. Using Step 1, we have:*
*$B = \{(\neg a \vee \neg b \vee c, N^2), (a, N), (b, N), (g, N)\}$.*
*The base $K'$ obtained from Step 1 (after removing tautologies): $K' = \{(\neg a \vee \neg b \vee c \vee g, N^2 + N), (\neg a \vee \neg b \vee c, N^2), (a \vee b \vee g, 3N), (a \vee b, 2N), (a \vee g, 2N), (b \vee g, 2N), (a, N), (b, N), (g, N)\}$.*
*Since we apply Adjustment on $K'_{\neg c}$, the first idea is to ignore formulas in $K'_{\neg c}$ under the inconsistency level (see Section 3.3). We can check that $Inc(K'_{\neg c}) = N$. Then, $\delta_A(K'_{\neg c})$ is the classical base (obtained by ignoring the ranks) associated with $\{(\neg c, +\infty), (\neg a \vee \neg b \vee c \vee g, N^2 + N), (\neg a \vee \neg b \vee c, N^2), (a \vee b \vee g, 3N), (a \vee b, 2N), (a \vee g, 2N), (b \vee g, 2N)\}$.*
*The second idea is that subsumed disjunctions are not added. In this example, since $\neg a \vee \neg b \vee c$ and $a \vee b, a \vee g, b \vee g$ will*

*belong to $\delta_A(K'_{\neg c})$ then there is no need to keep the disjunctions $\neg a \vee \neg b \vee c \vee g$ and $a \vee b \vee g$.*
*Lastly, the other disjunctions can be refined. Since $C = \{\neg c, \neg a \vee \neg b \vee c, a, b\}$ is inconsistent, then all disjunctions constructed from $g$ and this conflict $C$ are reduced to $g$.*
*Therefore, we have $\delta_A(K'_{\neg c}) \equiv \{\neg c, \neg a \vee \neg b \vee c, a \vee b, g\}$ which is equivalent to $\delta_{DMA}(K_{\neg c})$.*

## 7 Experimental results

We now present some experimental results which illustrate the different behaviour of each strategy. We used a propositional logic implementation of the strategies[3]. We chose 8 inconsistent bases at random from the DIMACS challenge (aim-50-no) containing 50 variables each and 80 clauses for the first 4, 100 clauses for the others. Then we stratified the bases with 20 clauses per strata, keeping the clauses in their original order. It appeared that each time the conflicts were discovered and weaken in the second strata, no more appeared in the remaining strata. The following table gives the number of clauses in the second strata after applying a given strategy. WDMA (resp. IDMA) stands for whole-DMA (resp. iterative DMA).

| #clauses | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 |
|---|---|---|---|---|---|---|---|---|
| Adj. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MA | 17 | 7 | 8 | 18 | 13 | 7 | 10 | 17 |
| DMA | 17 | 54 | 49 | 18 | 21 | 60 | 35 | 18 |
| WDMA | 168 | 149 | 153 | 161 | 161 | 155 | 152 | 160 |
| IDMA | 17 | 54 | 49 | 18 | 21 | 60 | 35 | 18 |

There are no differences between DMA and IDMA because on these examples consistency was either restored using $d_2(C)$ (t2,t3,t5,t6,t7) or all the clauses involved in a conflict have to were removed. Whole-DMA clearly hides the information contained in the knowledge base by generating a large number of clauses but timewise its fast. Let us now take a look at the time spent computing each strategy.

| time (s) | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 |
|---|---|---|---|---|---|---|---|---|
| Adj. | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| MA | 137 | 0.6 | 2.0 | 332 | 6.1 | 0.3 | 1.2 | 304 |
| DMA | 136 | 0.6 | 2.1 | 329 | 6.2 | 0.3 | 1.2 | 302 |
| WDMA | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| IDMA | 139 | 0.6 | 2.1 | 329 | 6.0 | 0.3 | 1.2 | 306 |

These results can be interpreted as follows: computing the set of clauses involved in conflicts (kernel) is costly, so all methods relying on this information will require small KB's to revise. This can be achieved for instance using modular KB's, a common practice in knowledge engineering.
Interestingly, since the three DMA approaches we introduced are logically equivalent, we can propose one way to efficiently compute the DMA policy: whole DMA, only based on satisfiability testing. This method can be used for instance if the knowledge base is hidden to the final user, and that only the queries are important. On the other hand, if the knowledge base itself is important for the user, such that the revised base must be as "close" as possible to the original one, an IDMA approach should be used (only necessary information will be weakened), but a computational cost must be paid. DMA is a tradeoff between these two policies.

[3] ADS: http://cafe.newcastle.edu.au/daniel/ADS/

## 8 Conclusion

We introduced a new family of computationally effective strategies for conflict resolution which can be used for exception handling, iterated belief revision and merging information from multiple sources. The most important feature of our strategy is that it relies on weakening conflicting information rather than removing conflicts completely, and hence is retains at least as much, and in most cases more, information than all other known strategies. Furthermore, it achieves this higher retention of information at no extra computational cost. We compared and contrasted three implementations of our new strategy with existing ones from a theoretical standpoint and by measuring their relative performance. We were also able to show the surprising result that the DMA policy provides a compilation of the lexicographical system which is known to have desirable theoretical properties. DMA offers the clear advantage of obviating the need to explicitly compute the set of all preferred subbases which can be hard. Another pleasing result is that the DMA strategy can be implemented as whole-DMA where the need to explicitly compute the culprits responsible for the conflicts is not required.

## References


[Alchourrón *et al.*, 1985] C. E. Alchourrón, P. Gärdenfors and D. Makinson *On the logic of theory change: Partial meet contraction and revision functions*. Journal of symbolic logic, 50:510-530, 1985.

[Benferhat *et al.*, 1993] S. Benferhat, C. Cayrol, D. Dubois, J. Lang, H. Prade. *Inconsistency management and prioritized syntax-based entailment*. Proc. IJCAI, France, 640-645.

[Dubois *et al.*, 1994] D. Dubois, J. Lang, H. Prade. *Possibilistic logic*. Handbook of Logic in AI and Log. Prog., (3), 439-513.

[Lau *et al.*, 2000] R. Lau, A.H.M. ter Hofstede, P.D. Bruza and K.F. Wong. *Belief Revision and Possibilistic Logic for Adaptive Information Filtering Agents*. Proc. IEEE (ICTAI), Canada, 19-26.

[Lehmann, 1995] D. Lehmann. Another perspective on default reasoning. Annals of Mathematics and Artif. Intel., 15, 61-82.

[Spohn, 1988] W. Spohn. Ordinal conditional functions: A dynamic theory of epistemic states. In: Causation in Decision, Belief Change and Statistic,(2), Reidel, Dordrecht, 105-134.

[Williams, 1994] M. A. Williams. Transmutations of knowledge systems. Proc. KR'94, 619-629.

[Williams, 1996] M. A. Williams. A practical approach to belief revision: reason-based change. Proc. KR'96, 412-421.

[Williams, 1998] M. A. Williams. Applications of Belief Revision. in Transactions and Change in Logic Databases, LNAI 1472, 285-314, Springer Verlag.

[Williams and Sims, 2000] M. A. Williams and A. Sims *SATEN: An object-oriented web-based revision and extraction engine*. Proc. of the Int. Workshop on Nonmonotonic Reasoning (NMR' 2000). Online Computer Science Abstract http://arxiv.org/abs/cs.AI/0003059/

[Wong and Lau, 2000] O. Wong and R. Lau *Possibilistic Reasoning for Intelligent Payment Agents*. Proc. of the $2^{nd}$ Workshop on AI in Electronic Commerce (AIEC' 2000), 1-13.