Proving theorems in a multi-source environment*

Laurence Cholvy
ONERA-CERT
2 avenue Edouard Belin,
31055 Toulouse, France
email : cholvy@tls-cs.cert.fr

Abstract

This paper describes a logic for reasoning in a multi-source environment and a theorem prover for this logic. We assume the existence of several sources of information (data/knowledge bases), each of them providing information. The main problem dealt with here is the problem of the consistency of the information : even if each separate source is consistent, the global set of information may be inconsistent. In our approach, we assume that the different sources are totally ordered, according to their reliability. This order is then used in order to avoid inconsistency. The logic we define for reasoning in this case is based on a classical logic augmented with pseudo-modalities. Its semantic is first detailed. Then a sound and complete axiomatic is given. Finally, a theorem prover is specified at the meta-level. We prove that it is correct with regard to the logic. We then implement it in a PROLOG-like language.

1 Introduction

More and more, computer science applications need to use information which is not provided by a single source of information but by several.

This is the case, for instance, when one wants to use several expert systems, each of them dealing with a particular part of the global problem to be solved. In this case, the knowledge coming from each expert system must be combined. This combination is not necessarily physical and the knowledge may remain distributed among the different systems. However, the set of knowledge necessary to solve the global problem is obtained by virtually grouping the knowledge of the different systems. It is also the case of distributed databases. Each database stores information concerning a particular application domain. When considering a larger domain, one has to federate different databases. Here again, the grouping is virtual since, very often, the different-databases are locally managed by other people. Another case of collection of information provided by different sources can be found in the grouping of beliefs. The classic example is that of a police inspector [C Baral and Subrahmanian, 1992) who questions different witnesses. Each witness has his own beliefs concerning the crime and the inspector has to collect all of them in order to find the clue.

Several problems arise when merging different information sources [Cholvy, 1992a]. First of all, there are problems of language : the sources do not necessarily share a common language for describing information. In databases for instance, this happens when the different databases to be merged do not have the same schema. In such a case, translations are necessary [Elmagarmid and Pu,]. There also are problems of redundancy : a source may provide information which subsumes other information provided elsewhere. In this case, pruning is necessary. Finally, there are problems of consistency. Even if each source is consistent, the global set of information may be contradictory. This is augmented by the delocalised management of the different information sources. People who need to gather information are not those who manage the different bases and, due to the distribution, the different sources are managed by different people. So the notion of global consistency does not exist, since the sources are independently developed. However, when grouping all of them, the problem of consistency arises.

Our work concerns the last problem : we are interested in defining a logical framework which allows us to reason with information provided by difTerent sources and which does not collapse in case of inconsistency. Classical logic cannot be used directly in this case since everything is deducible from a contradiction (one can say it collapses in case of inconsistency). In the next section, we will present different studies which dealt with this problem. We will also present our approach. In section 3, we will present the logic we have defined in order to reason in a multi-source environment. In section 4, we define a theorem prover which implements our approach. It is described as a meta-program and we run it under a PROLOG-like language. Correctness of the theorem prover is proved.

*This work has been financed by the CNRS and the French Ministry of Research

2 Some related work and our proposals

Work dealing with the problem of reasoning in the presence of inconsistency can be divided into two groups.

- In the first group, we find studies which use classical logics but which eliminate inconsistency in order not to collapse. The main way to discard inconsistency is to manage maximal consistent subsets. Work by [C Baral and Subrahmanian, 1992] belongs to this group. They define several types of theory combinations, mainly based on the maximal consistent subsets of the global theory. Our past work [Cholvy, 1990], [Bauval and Cholvy, 1991], also belongs to this group. In the related domain of database updates, work by [Fagin *et* al., 1983] and [Kupper *et* al., 1984] also privilege maximal consistent subsets, as do [Gardenfors, 1988], [Nebel, 1989] in belief revision area, and [Ginsberg and Smith, 1988] in the problem of reasoning about actions.
- In the second group, we find studies which do not use classical logics. For instance, paraconsistent logics are defined in such a way that some classical theorems (which allow us to derive everything from a contradiction) are not deducible in these logics [Besnard, 1990]. For example, $A \wedge \neg A \rightarrow B$ is not a theorem in paraconsistent. logics. So a contradiction, say $A \wedge \neg A$, cannot be used to derive anything, say B. Another example of non-classical logic which can manage with inconsistency is the modal logic for reasoning about updates defined in [Cerro and Herzig, 1986] and [Farinas and Herzig, 1992]. Even if the update is contradictory with the current database, the logic does not collapse.

Besides this classification, another classification could be made : indeed, in each previous group, most of researchers have studied two cases. In a first step, they consider that all the data are equal with regard to the problem of inconsistency. In a second step, they studied the case where data are associated with extra-information (tags, labels, degrees of priority, actions to be performed ...) which are used to restore consistency. For instance, [Fagin *et al,* 1983], [C Baral and Subrahmanian, 1992], [Besnard, 1990], [D. Dubois and Prades, 1992], [Gabbay and Hunter, 1991] present different types of extra-information.

Finally, the previous works could be grouped according to another classification : most of them adopt a constructive approach in the sense that avoiding inconsistency leads to construct a new consistent base. Only work by [Cerro and Herzig, 1986] and [Farinas and Herzig, 1992] adopts a hypothetical approach : the logic they define, called assume, allows us to reason with hypothetical updates. The user can then deduce theorems of the form : such a formula will be true if I assume such other formula. The state after the update is never constructed.

As for us, we have defined two logics for reasoning in a multi-source environment which do not collapse under inconsistency, [Cholvy, 1992a] [Cholvy, 1992b]. Both of them allow the user to assume that the different sources of information are totally ordered according to their reliability. Our approach is then a hypothetical one. The two logics differ on the attitude they modelize :

- The first logic, called FUSION-S, modelizes a suspicious attitude : it consists in suspecting all the information provided by a source if this source contradicts a more reliable source.
- The second logic, called FUSION-T, modelizes a trusting attitude : if an information source contradicts a more reliable source, only the smallest set of contradictory information is suspected.

Let us take the example of a police inspector who questions witnesses. Assume that a first witness, Bill, said that he saw a black car, while a second, John, said that he saw two men in a white car. An order may reflect the fact that the inspector himself has some conviction (which cannot be denied) or possesses some information which is true : for instance, he went to the meteorological station and he is sure that the crime was committed on a foggy day. So, he can assume that John is less reliable than Bill, since John was standing too far away from the scene of the crime and, because of the fog, he could not see well. So, the inspector may trust him less. In this case, the inspector will consider that :
inspector $> Bill > John$.

If the inspector adopts the suspiscious attitude, he will conclude only that the car was black, i.e., he suspects all the information provided by John because John contradicts a more reliable witness. If he adopts the trusting attitude, he will conclude that they were two men in a black car. In fact, concerning the colour of the car, he trusts Bill more than John, so he can assume that the colour is black. Concerning the number of persons, John provides new information that does not contradict Bill's account.

In the rest of this paper, we focus on the trusting attitude. The next section describes the logic which modelizes it.

Remark 1. As said previously, the trusting attitude suspects the smallest set of information which contradicts more reliable information. In this paper, smallest contradictory sets are pairs of literals : 1 and - 1.

We can imagine a more "application-adapted" approach in which the notion of topics of information [Cazalens, 1992] [Cazalens and Demolombe, 1992], is taken into account in order to characterize the minimality. For example, assume a meeting in which a person listens to two people. The first is a teacher of logic, the second is a student. The conversation is technical. Implicitely, the person who listens trusts the teacher more than the student. Assume that during the conversation, the student affirms that "the first order logic is decidable". Immediately, the teacher denies it and reminds the student of the main result of undecidability of the first order logic. For the person, it is clear that the student is not reliable concerning the first order logic. If, later on, the student makes another affirmation about the first order logic, it is quite sure that the listener will not trust him. Indeed, because of the contradiction about the decidability, he rejects any affirmation of the student which "is about" the first order logic. In this case, works previously cited which try to formalize the notion of "is-about", could be used in order to reject a smallest set of information, defined by the domain of the first order logic. An attempt to use the notion of topics of information for a syntactical characterization of updates can be found in [Cholvy, 1993].

Remark 2. In this paper, we assume the existence of a unique order on the different sources of information. Again, we can imagine a more "application-adapted" approach in which the sources are ordered according to several orders which are topic-dependent.

Let us take the previous example and consider that the teacher of logic (who is an old-fashioned man) and the student are now speaking about pop music. It is quite sure that, regarding "pop music", he will trust the student more than the teacher, i.e., the student will be considered as more reliable than the teacher. So there will be two orders, one for the "logics" topic and one for the "pop music" topic.

An attempt at a formalization of this idea can be found in [Cholvy, 1993].

3 A logic for reasoning in a multi-source environment

In this section, we present a logic which implements the trusting attitude previously introduced.

3.1 The language

We assume that our language L is a finite set of propositional variables : p_1, p_2, ... PL- We note 1,2.. n the n information sources we reason with. We will also say the databases to be merged. We assume that the sources (or databases) are finite, satisfiable but not necessarily complete, sets of literals of L.

The logic we define, called FUSION-T(l..n), is obtained from the propositional logic, augmented with pseudo-modalities i.e., marks on formulas. These pseudo-modalities are :

$[i_1 i_2 .. i_m]$, where $m \geq 1$ and $i_j \in \{1..n\}$ and $(j \neq k \Longrightarrow i_j \neq i_k)$

$[i_1 i_2 .. i_m]$ F will mean that, when considering the total order on $\{i_1..i_m\}$: $i_1 > i_2, i_2 > i_3, .. i_{m-1} > i_m$, F is true in the database obtained by virtually merging database i_1 and .. database i_m.

Notice that the general form of these pseudo-modalities allows us to represent the particular case : [i] F, i=l..n, which means that F is true in database i.

3.2 Semantics

The semantics of FUSION-T(l.n) is the following : a model of FUSION-T(L.n) is a pair : $I = (W, r)$, where :

- W is the finite set of all the interpretations of the underlying propositional language L
- r is a finite set of equivalence relations between interpretations in W.

Each pseudo-modality is associated with a relation. So, if [O] is a pseudo-modality, we note R(O) the associated equivalence relation and $\overline{R}(O)$ its equivalence class. The equivalence classes are recursively defined by :

- $\overline{R}(i)$ is a non empty subset of W, i = 1..n
- $\overline{R}(i_1 i_2 .. i_m)) = f_{i_m}(..(f_{i_2}(\overline{R}(i_1))..))$ where :

$f_{i_j}(E) = \{ w : w \in E \text{ and } w \models L_{i_j E} \}$ and
$L_{i_j E} = \{ l : l \text{ literal of L such that } (\forall v \in \overline{R}(i_j) \Longrightarrow v \models l) \text{ and } (\exists u \in E \text{ and } u \models l) \}$

Definition. (Satisfaction of formulas).
Let F be a formula of L. Let F1 and F2 be formulas of L'. Let O be a total order on a subset of {1..n}.
Let M = (W,r) be an interpretation of FUSION-T(1..n) and let w ∈ W.

FUSION-T(1..n),r,w ⊨ F ⟺
w ⊨ F
FUSION-T(1..n),r,w ⊨ [O] F ⟺
∀ w' w' ∈ $\overline{R}$(O) ⟹ w' ⊨ F
FUSION-T(1..n),r,w ⊨ ¬ F1 ⟺
non(FUSION-T(1..n),r,w ⊨ F)
FUSION-T(1..n),r,w ⊨ F1 ∧ F2 ⟺
(FUSION-T(1..n),r,w ⊨ F1) and
(FUSION-T(1..n),r,w ⊨ F2)

Definition. (Valid formulas in FUSION-T(1..n)).
Let F be a formula of L'.
F is a valid formula (in FUSION-T(1..n)) ⟺
∀ M = (W,r) an interpretation of FUSION-T(1..n),
∀ w ∈ W, FUSION-T(1..n),r,w ⊨ F

We note FUSION-T(1..n) ⊨ F, the valid formulas.

3.3 Axiomatics

Let us write O an order $i_1..i_m$, $m \geq 1$. (i.e., $i_1 > i_2, .., i_{m-1} > i_m$). By convention, $O \cup \{i_{m+1}\}$ is the order $i_1..i_m i_{m+1}$.

Axioms of FUSION-T(1..n) are :

- (A0) Axioms of the propositional logic
- (A1) $[O] \neg F \rightarrow \neg [O] F$
- (A2) $[O] F \wedge [O](F \rightarrow G) \rightarrow [O] G$
- (A3) $[O \cup \{i\}] l \longleftrightarrow [O] l \vee ([i] l \wedge \neg [O] \neg l)$ if l is a literal

Inferences rules of FUSION-T(1..n) are :

- (Nec) $\vdash F \Longrightarrow \vdash [O] F$ (if F is a propositional formula)
- (MP) $\vdash F$ and $\vdash (F \rightarrow G) \Longrightarrow \vdash G$

Axiom (A3) expresses the trusting attitude. Indeed, we could decompose it in three axioms :

- (A3.1) $[O] l \rightarrow [O \cup \{i\}] l$
- (A3.2) $[i] l \wedge \neg [O] \neg l \rightarrow [O \cup \{i\}] l$

- (A3.3) $\neg [O] l \wedge \neg [i] l \rightarrow \neg [O \cup \{i\}] l$

This means that :

- if a literal is true under order O, then it is still true under order $O \cup \{i\}$, for any i, since, by convention, $O \cup \{i\}$ means that i is the least reliable source.
- if it is the case that a literal is true in database i, and if it is not the case that its negation is true under order O, then it is the case that it is true under $O \cup \{i\}$.
- if it is not the case that a literal is true under order 0 and if it is not the case that it is true in database i then it is not the case that it is true under order $O \cup \{i\}$

Remark. This logic will be used to modelize the merging of n databases db1 ... dbn, in the case where they are finite consistent sets of L literals and according to the trusting attitude.

Let us note $\psi = \bigwedge_{i=1}^{n} (\bigwedge_{l \in bdi} [i]l \wedge \bigwedge_{bdi \not\vdash c} \neg[i]c)$

(where l is a literal of L and c a clause of L).

We will be interested in finding valid formulas of the form : $(\psi \rightarrow [O]\ F)$, i.e., finding formulas F which are true in the database obtained by merging db1 ... dbn, when the order is O.

We have proved the following facts :

Proposition 1. Let ψ be the formula previously defined. Let F be a formula of L. Let O be a total order on a subset of $\{1..n\}$. Then :
FUSION-T(1..n) $\models (\psi \rightarrow [O]F) \iff$
FUSION-T(1..n) $\vdash (\psi \rightarrow [O]F)$

Proposition 2. With the same assumptions as in proposition 3 :
FUSION-T(1..n) $\vdash (\psi \rightarrow [O]\ F)$ or
FUSION-T(1..n) $\vdash (\psi \rightarrow \neg [O]\ F)$

3.4 A quick comparison with the belief revision problem

Let us first recall some definitions [Katsuno and Mendelzon, 1991] :

Definition. Let m and m1 be two interpretations. We define the distance between m and m1 by :
$d(m,m1) = \{p \in L : (p \in m \text{ and } p \notin m1) \text{ or } (p \notin m \text{ and } p \in m1)\}$

Definition. Let m be an interpretation. We define a partial order on a set of interpretations $\leq_m$ by :
$m1 \leq_m m2 \iff d(m,m1) \subseteq d(m,m2)$

Then we can prove that the functions f_{i_j} defined in section 3.2 are such that : $f_{i_j}(E) = \bigcup_{m \in \overline{R}(i_j)} Min(E, \leq_m)$

So, according to [Katsuno and Mendelzon, 1991], fj(E) is the set of models of belief base DBj updated by E. In other terms, the result of merging two databases according to the trusting attitude and such that i >j, is equivalent to updating database j with database i.

Generalisation. Merging n databases 1,2 ..n, according to the trusting attitude and given an order : $i_1 > i_2, .., i_{n-1} > i_n$ conies down to updating DBi_n with the result of the update of DBi_{n-1} with the result of the update .... of DBi2 with DBii.

4 A theorem prover for this logic

In this section, we deal with the implementation aspects. Our aim is to define a theorem prover for this logic which allows us to answer questions of the form : is formula F true in the merged information sources if the order is 0 ? i.e., to prove theorem [0]F. We suggest using a PROLOG-like language form implementing such a theorem prover, in order to reuse its facilities (unification, negation as failure, strategy ...)

In subsection 4.1, we describe a first meta-program which describes a prover for proving formula [O]l, if 1 is a literal. This meta-program is a set of definite Horn-clauses (where negation is explicitely managed with positive literals) and which can easily be run on a PROLOG-like interpreter. In subsection 4.2, we give an optimisation where the ncgation-as-failure of PROLOG is used to manage the negation. Finally, in section 4.3, we extend this prover in order to prove formula [0] F, where F is any propositional formula (written under the conjunctive normal form).

4.1 The theorem prover as a meta-program : first version

Let us consider a meta-language ML, based on language L, defined by :

- constants of ML are literals of L, plus a constant rioted nil
- a function rioted $\cup$. By convention, $\{i_1..i_m\}$ represents the term : $i_1 \cup (i_2 ... \cup (i_m \cup nil)...)$.
- predicat symbols of ML are : LFUSION, nonLFUSION, NIL and nonNIL.

The intuitive semantics of the predicates is the following :

- LFUS10N(0,1) means that it is the case that literal 1 is true in the merged databases if the order is O.
- nonLFUSION(0,l) means that it is not the case that literal 1 is true in the merged database if the order is 0
- NIL(O) is true only for nil
- nonNIL(O) is true except if O is nil.

4.1.1 The meta-program

Let us consider META1, the following set of the ML formulas :

(1) LFUSION({i}, l), for any literal l in database i
(2) LFUSION(O,l) ∧ nonNIL(O) → LFUSION(O ∪ {i}, l)
(3) LFUSION({i},l) ∧ nonLFUSION(O,¬ l) ∧ nonNIL(O)→ LFUSION(O ∪ {i}, l)
(4) nonLFUSION({i}, l), for any literal l not in database i
(5) LFUSION(O,l) → nonLFUSION(O, ¬ l)
(6) nonLFUSION(O,l) ∧ nonLFUSION({i},l) ∧ nonNIL(O) → nonLFUSION(O ∪ {i},l)
(7) nonNIL(O) , for any O except nil

Proposition 3. (soundness and completeness)
Let l be a literal of L, let O be a total order on a subset of $\{1..n\}$.
META1 $\vdash$ LFUSION(O,l) $\Longleftrightarrow$
FUSION-T(1..n) $\vdash$ ($\psi \rightarrow$ [O]l)
and
META1 $\vdash$ nonLFUSION(O,l) $\Longleftrightarrow$
FUSION-T(1..n) $\vdash$ ($\psi \rightarrow \neg$ [O] l)

(To prove the soundness, we consider the fixed point operator FP, associated to META1 which is such that META1 $\vdash$ LFUSION(O,l) $\Longrightarrow \exists$ k LFUSION(O,l) $\in$ $FP^k(\emptyset)$ (resp nonFUSION(O,l)). We then prove the two points by induction on k. We prove the completeness, jointly for the two points, by induction on the length of the prefixes).

Proposition 4. (Corollary)
Let l be a literal of L and O a total order on a subset of $\{1..n\}$. Then,
META1 $\vdash$ LFUSION(O,l) or
META1 $\vdash$ nonLFUSION(O,l)

4.1.2 Implementation

In this first version, PROLOG is used in this meta-program META1. The following proposition ensures that, although this meta-program is recursive, PROLOG does not loop when running it.

Proposition 5.
Let l be a literal of L, let O be a total order on a subset of $\{1..n\}$. The tree developped by PROLOG in order to prove the goal : LFUSION(O,l) (resp nonLFUSION(O,l)) in META1 is finite.

(We prove it by induction on the length of the prefix O)

4.2 Optimisation

The main remark about the previous meta-program concerns the explicit management of the predicate nonFUSION. An optimisation can be realised by managing the predicates nonFUSION and nonNIL with negation-as-failure of PROLOG. This leads to the meta-program, META2 :

(1) LFUSION($\{i\}$, l) . for any literal l in database i
(2) LFUSION(O,l) $\wedge \neg$ NIL(O) $\rightarrow$ LFUSION(O $\cup$ $\{i\}$, l)
(3) LFUSION($\{i\}$,l) $\wedge \neg$ LFUSION(O,$\neg$ l) $\wedge \neg$ NIL(O) $\rightarrow$ LFUSION(O $\cup$ $\{i\}$, l)
(4) NIL(nil)

Proposition 6. Let l be a literal, let O be a total order on a subset of $\{1..n\}$.
PROLOG fails to prove LFUSION(O,l) in META2 using the negation-as-failure iff PROLOG proves nonLFUSION(O,l) in META1 (without using negation as failure).
(The proof is immediate)

Proposition 6 and proposition 3 allow us to give:

Proposition 7. Let l be a literal, let O be a total order on a subset of $\{1..n\}$.
Using negation-as-failure on the meta-progran META2, PROLOG succeeds in proving LFUSION(O,l) iff
FUSION-T(1..n) $\vdash$ ($\psi \rightarrow$ [O]l) ; it fails iff
FUSION-T(1..n) $\vdash$ ($\psi \rightarrow \neg$ [O] l)

4.3 Extension to any propositional formula

We consider the following meta-program, META3, obtained from META2 by adding three axioms for the management of conjunctions and disjunctions :

(1) LFUSION($\{i\}$, l) for any literal l in database i
(2) LFUSION(O,l) $\wedge \neg$ NIL(O) $\rightarrow$ LFUSION(O $\cup$ $\{i\}$, l)
(3) LFUSION($\{i\}$,l) $\wedge \neg$ LFUSION(O,$\neg$ l) $\wedge \neg$ NIL(O) $\rightarrow$ LFUSION(O $\cup$ $\{i\}$, l)
(4) NIL(nil)
(5) CFUSION(O,nil)
(6) DFUSION(O,d) $\wedge$ CFUSION(O,c) $\rightarrow$ CFUSION(O,d $\cup$ c)
(7) LFUSION(O,l) $\rightarrow$ DFUSION(O,l $\cup$ d)
(8) DFUSION(O,d) $\rightarrow$ DFUSION(O, l $\cup$ d)

Proposition 8.
Let F be a formula under its conjunctive normal form. Let O be a total order on a subset of $\{1..m\}$.
Using negation-as-failure on the meta-progran META3, PROLOG proves the goal CFUSION(O,F) iff
FUSION-T(1..n) $\vdash$ ($\psi \rightarrow$ [O]F) ; it fails iff
FUSION-T(1..) $\vdash$ ($\psi \rightarrow \neg$ [O] F)

(We prove the if condition by using the fixed point operator FP and show that : if PROLOG proves DFUSION(O,D), then FUSION-T(1..n) $\vdash$ ($\psi \rightarrow$ [O]D), and if PROLOG proves CFUSION(O,C), then FUSION-T(1..n) $\vdash$ ($\psi \rightarrow$ [O]C).
We prove the only-if condition by decomposing F in a conjunction of disjunctions of literals. Proposition 2 is then used for the failing case)

5 Conclusion

In this paper, we have dealt with the problem of reasoning in a multi-source environment by focusing on the global consistency of information. We have shown that this problem is a particular case of reasoning with inconsistency. For discarding inconsistency, we have suggested considering the relative reliability of the different information sources. This comes down to considering a total order on the sources. There are different ways to use this order to avoid the inconsistency but in this paper we have focused on one attitude, called trusting : when two sources are contradictory, it consists in rejecting the minimal contradictory information which is provided by the less reliable source.

Our aim was to define a logic that would allow the user to reason in a multi-source environment according to a hypothetical approach : the merging of the different sources is never done, i.e., the database obtained by merging the information coming from the different sources is never construted. The user only assumes the order of the sources and tries to derive true formula in this assumed database. As far as we know, only the work cited in section 2 for database updates adopts such a hypothetical approach.

The semantics we have attached to this attitude is in terms of possible models and is an extension of the semantics defined for belief base updates. The notion of "nearest" models is defined in terms of complementary literals. The semantics is appropriate even if information is extended to clauses [Cholvy, 1993]. Unfortunately, the axiomatics we have given, and the theorem prover we have defined, are only adequate for literal information.

In addition, one can wonder what happens if the order on the sources is not total. In this case, we find again the problem of reasoning with inconsistency with no extra-information to restore consistency. Solutions briefly described in section 2 could be adapted here leading to a formalism which mixes our logics and a mechanism to avoid inconsistency (management of maximal consistent subsets for instance). But this needs to be studied more carefully.

Finally, in section 2, we have shown that another semantics could be attached to the reasoning in a multi-source environment. It consists in taking into account notions like topics of information. This application-oriented notion could be used to give a new definition of minimality as well as an extension to several orders on sources of information.

References

[Bauval and Cholvy, 1991] A Bauval and L. Cholvy. Automated reasoning in case of inconsistency. In Proceedings of the first World conference on fundamentals of AI, 1991.

[Besnard, 1990] P Besnard. Logics for automated reasoning in the presence of contradictions. In North Holland, editor, Proc of Artificial Intelligence Methodology, systems and applications, 1990.

[C Baral and Subrahmanian, 1992] J Minker C Baral, S. Kraus and VS. Subrahmanian. Combining knowledge bases consisting of first order theories. Computational intelligence, 8(1), 1992.

[Cazalens and Demolombe, 1992] S. Cazalens and R. Demolombe. Intelligent access to data and knowledge bases via users' topics of interest. In Proceedings of IFIP conference, 1992.

[Cazalens, 1992] S. Cazalens. Formalisation en logique non-standard de certaines methodes de raisonnement pour forunir des reponses cooperatives, dans des systemes de bases de donnees et de connaissances. PhD thesis, Universite Paul Sabatier, Toulouse, 1992.

[Cerro and Herzig, 1986] L. Farinas Del Cerro and A. Herzig. Reasoning about database updates. In Jack Minker, editor, Workshop of Foundations of deductive databases and logic programming, 1986.

[Cholvy, 1990] L. Cholvy. Querying an inconsistent database. In North Holland, editor, Proc of Artificial Intelligence : Methodology, systems and applications, 1990.

[Cholvy, 1992a] L. Cholvy. Consistency of merged databases. In Proceedings of the Workshop on Co-operationg systems, Keele University (GB), 1992.

[Cholvy, 1992b] L. Cholvy. A logical approach to multi-sources reasoning. In Proceedings of the Applied Logic Conference, University of Amsterdam, 1992.

[Cholvy, 1993] L. Cholvy. Mises a jour et fusion de bases de connaissances : extension aux clauses et prise en compte de la notion de themes d'informations. Technical report, CERT-DERI, 1993.

[D. Dubois .and Prades, 1992] J. Lang D. Dubois and II. Prades. Dealing with multi-source information in possibilistic logic In Proceedings of ECAI, 1992.

[Elmagarmid and Pu,] A. Elmagarmid and C. Pu. The special issue on heterogeneous databases. ACM Computing Surveys, 22(3).

[Fagin et al, 1983] R. Fagin, J.D. Ullman, and M. Vardi. On the semantics of updates in databases. In ACM TODS, 1983.

[Farinas and Herzig, 1992] L. Farinas and A. Herzig. Constructive minimal changes. In Report I RIT, 1992.

[Gabbay and Hunter, 1991] D. Gabbay and A. Hunter. Making inconsistency respectable. In International workshop on fundamentals of Artificial Intelligence, 1991.

[Gardenfors, 1988] P. Gardenfors. Knowledge in flux : modeling the dynamics of epistemic states. The MIT Press, 1988.

[Ginsberg and Smith, 1988] M. L. Ginsberg and D. E. Smith Reasoning about action i : a possible worlds approach Artificial Intelligence, 35:165 195, 1988.

[Katsuno and Mendelzon, 1991] H. Katsuno and A Mendelzon. Propositional knowledge base revision and nimmal change. Artificial Intelligence, 52, 1991.

[Kupper et ai, 1984] G.M. Kupper, J.D. Ullman, and M. Vardi. On the equivalence of logical databases. In Proc of ACM-PODS, 1984.

[Nebel, 1989] B. Nebel. A knowledge level analysis of belief revision. In First conference on Principles ok knowledge representation and reasoning, 1989.