Modeling Knowledge Merging Systems

Frédéric Koriche
Département Acquisition et Représentation de Connaissances
LIRMM, UMR 9928, Université Montpellier II CNRS,
161 rue Ada, 34392 Montpellier Cedex 5, France

Abstract

With the great richness and diversity of knowledge bases available today, a central difficulty is to combine knowledge coming from several sources in order to solve complex problems that could not be handled in isolation. In this study, we focus on two important characteristics about the knowledge merging problem: inconsistency and incompleteness. The purpose of this work is to present a logical framework that enables to formally specify knowledge merging systems and which tolerates inconsistency and incompleteness. This framework is based on a functional approach which is used in this paper to guide our formal development. The major features of this approach are dual reasoning*, which includes an uniform way to deal with both aspects of partial and conflicting knowledge, and* incrementality*, which enables to gradually build pools of knowledge and to improve incomplete answers.*

1 Introduction

During this past decade, the world of knowledge based systems has spread considerably, providing a great variety of information on different domains of expertise. In presence of this increasing community of distributed, autonomous, and heterogeneous sources of knowledge, the problem of *cooperation* between such systems is becoming more and more critical (see e.g. [1, 3, 7, 14]). In particular, a central issue is to enable several knowledge bases to work interactively, grouping their items of knowledge in order to solve complex problems that could not be handled in isolation.

Several difficulties arise when we attempt to merge different knowledge bases. First, we need to consider the possible heterogeneity of representation languages. Knowledge based systems are generally developed to stand alone, and so they do not necessarily share the same representation language. In order to allow cooperation, we must break away from this monolithic conception of knowledge engineering (see e.g. [7]). Second, a problem of consistency arises when a reasoning system obtains its knowledge from a variety of sources. Even if each source is consistent and trustworthy, the global pool of knowledge may be contradictory. Moreover, none of these sources can be assumed to be infallible, a universal truth-teller. Therefore, inconsistency is sometimes unavoidable. Third and finally, combined knowledge may be incomplete. Each local source has to deal with a partial knowledge, and the whole information gathered from a set of sources is not necessarily complete. This is augmented when network partitioning occurs: some knowledge bases may be inaccessible to others, and the problem of completeness is exacerbated still further. To sum up, specifying a knowledge merging system requires the consideration of combining knowledge in an “open environment”, and the possibility of deriving approximate answers to the original query.

Our present work concerns the last two problems: the purpose of this paper is to define a logical framework which allows us to specify knowledge merging systems, and which tolerates incompleteness and inconsistency. In section 2, we begin to examine the problem of combining knowledge and we present several studies which have attempted to solve this problem. In section 3, we present a conceptual model for knowledge merging systems. This model will be used to guide the development of our logical framework. We proceed to the formal treatment in the next sections. We present a representation language in section 4, and we develop a formal semantics of this language in section 5. A brief discussion of possible extensions of the framework in section 6 concludes the paper.

2 The Knowledge Merging Problem

In this work, the problem of combining knowledge is analyzed at the *knowledge level*, which is precisely the domain of logic [9]. From this viewpoint, we define the characteristics of knowledge merging systems in terms of requirements that the logical framework must satisfy. We distinguish four requirements.

First and topmost, the logical framework must handle *inconsistency*. We recall that classical logic collapses in presence of inconsistency. Given a contradictory pool of knowledge and a classical logic for deducing information from it, everything can be entailed. So we need a formal tool that is able to handle conflicts and which deduces non-degenerate conclusions.

Second, the logical framework must handle *incompleteness*. Here also, representation formalisms derived from classical logic are often based on the "Closed World Assumption" which rejects incompleteness. For example, in conventional logic programming, every sentence that is not deduced from a knowledge base is assumed to be false. However, an "open environment" is characterized by the incompleteness of its domain knowledge. Consequently, we need a formal tool which is able to infer approximate conclusions.

Third, we would like a framework which enables *dual reasoning* about combined knowledge. As suggested by Pinkas and Loui [10], an agent may express two dual attitudes relative to conflicting knowledge. On the one hand, he may insist on a consensus or "skeptical" attitude. When he is faced with contradictory answers, he says that he has no information. On the other hand, the agent may treat his neighbors as trustfully sources, and so when faced with two conflicting answers, he expresses a speculative or "credulous" attitude: he keeps both answers and records the existence of an inconsistency. The key idea in dual reasoning is that both skeptical *and* credulous attitudes can be used to answer a query.

Fourth and finally, the logical framework should be *incremental*. That is, we should be able to identify a new item of knowledge by the minimal change it would induce if it were added to the pool of combined knowledge. Moreover, only this minimal change should be analyzed, not the global set of knowledge. In this way, approximate answers may be improved.

Past research on knowledge merging has almost exclusively focused on the problem of consistency. The representation formalisms used to handle conflicting knowledge can broadly be classified in two groups. On the one hand, we have extensions to classical logic which use "extra-logical" features in order to restore consistency. The methods suggested in [1, 3, 14] are based on a similar strategy. Essentially, this consists of finding all maximal consistent subsets from a pool of knowledge, and concluding only the sentences that are in the intersection of all these subsets. This strategy is also used in belief update [13], and belief revision [5]. Paraconsistent logics, on the other hand, adopt a radically different strategy. Such logics tolerate inconsistency: they do not collapse in presence of contradictory knowledge. The most known representative of this family is the Belnap's four valued logic [2], which has been exploited in different domains, such as knowledge representation [8] and non-monotonic reasoning [6]. In particular, this logic was used in [4] to combine knowledge from distributed logic programs.

In the setting of combining knowledge, there are several advantages to use Belnap's four valued logic. First at all, this logic is "robust": it tolerates inconsistency. From an algorithmic perspective, this is a crucial point since the problem of consistency check is known to be either undecidable in the first order case, or intractable in the propositional case. Consequently, contradiction may be undetectable, and so extra-logical features used in the first group may be inefficient in presence of a large pool of knowledge. Furthermore, this logic allows us to represent both inconsistency and incompleteness. In a four valued setting, a particular sentence can be valued to be true, false, both or neither. At a result, a state of knowledge may contain no information about some issues and contradictory information about others.

We take the four valued logic as a starting point for specifying the process of combining knowledge. Notice however that this logic alone does not provide any insight concerning the last two requirements (i.e. dual reasoning and incrementality). We need to develop a conceptual model that guides the formalization in such a way that all desiderata will be fulfilled.

3 A conceptual model

A knowledge merging system is modelized as an *agent* that interacts with its environment through *epistemic inputs* in order to gather information from its multi-source environment, and expresses *epistemic attitudes* relative to queries issued by the user. The agent is viewed as a dynamic system which consists of two parts: a static part, represented by *epistemic states*, and a dynamic part defined by *state transformers*. The agent interacts with its environment through *epistemic inputs* in order to gather information from its multi-source environment, and expresses *epistemic attitudes* relative to queries issued by the user. The specificity of the knowledge merging problem is identified by the characteristics of each components, which are detailed as follows.

An epistemic state represents the set of sentences which are deductible given what the agent is told. In our setting, we insist on two key points. First, an epistemic state may carry an incomplete or contradictory information. Second, for each epistemic state corresponds a "dual epistemic state". It represents the set of sentences whose negation cannot be concluded from the original epistemic state. We notice that if the original state is complete and consistent then the dual epistemic state produces the same set of sentences as the original one. However, if the agent is told an incomplete and/or contradictory information, we have two distinct sets of sentences, and then two separate epistemic states.

Epistemic attitudes are used to access knowledge. An epistemic attitude corresponds to an interpretation function of a certain sentence (i.e. the query) in a certain epistemic state. In a classical setting (complete and consistent), an epistemic state and its dual capture the same information. Therefore, only one state of knowledge has to be considered, and a sentence is simply "accepted" or "refuted" in this state. However, in a more general setting, we must take into account a richer typology of epistemic attitudes. Namely, a sentence may be (1) *accepted*: the agent has an evidence for the sentence, (2) *refuted*: the agent has a counter-evidence against the sentence, (3) *acceptable*: the sentence is not refuted, and (4) *refutable*: the sentence is not accepted.

The purpose of epistemic inputs is to modify what is known. In this study, we concentrate on two kinds of epistemic inputs that will be of use in the formalization of knowledge merging problems. First, are *facts* which are used to augment what is known. A fact may be seen as an "extensional" information provided by some logical database (for instance, a set of tuples in a relational database). Second are *rules* which are intended to "conditionally" augment what is known. For example, a rule may be a clause in a logical database. It provides additional information about the truth value of some sentence.

State transformers represent the "epistemic transitions" an agent makes in its space of epistemic states in order to acquire information conveyed by its epistemic inputs. In our functional viewpoint, the key idea is to associate the meaning of a certain kind of epistemic input with a certain function on epistemic states. In this work, we are interested by an important class of state transformers, called *expansions* which have been notably studied in belief revision [5] and dynamic semantics [12]. They represent some increase of information, and they are used here to define the formal semantics of facts and rules.

Based on this conceptual model, the goal of our logical framework is thus (1) to provide a language for describing epistemic inputs, (2) to represent in a multi-valued setting epistemic states and epistemic attitudes, and (3) to define systematic connections between state transformers and epistemic inputs. We now proceed to the formal treatment.

4 A Representation Language

We consider a first-order language $\mathcal{L}_W$ built up from a *workspace* $W = (C, R)$, where C is a finite collection of constant symbols and R is a finite set of relation symbols. Additionally, the language contains a countable set of variable symbols. We use a vector notation $\vec{x}$ to stand for a list of variables $(x_1, x_2, \ldots, x_k)$. Similarly, $\vec{d}$ is used to stand of a list of constants $(d_1, d_2, \ldots, d_k)$. Given the usual definitions of terms and atoms, a *ground atom* is an atom where all terms are constants. The set of all ground atoms generated from W is denoted $\mathcal{G}_W$. The *formulas* of $\mathcal{L}_W$ are built up in the classical way from the atoms, the connectives $\neg, \vee, \wedge$ and the quantifiers $\forall, \exists$. An *open formula* is a formula which contains free variables. We use the notation $A(\vec{x})$ to stand for an open formula where $\vec{x}$ is the list of free variables. A *sentence* is a closed formula, that is a formula without free variable.

Definition 4.1 The representation language $\mathcal{R}_W$ is the smallest set for which the following conditions hold:

- *if A is a sentence of $\mathcal{L}_W$, then $\rightarrow A \in \mathcal{R}_W$,*
- *if $A(\vec{x}), B(\vec{x}) \in \mathcal{L}_W$, then $A(\vec{x}) \rightarrow B(\vec{x}) \in \mathcal{R}_W$.*

Intuitively, an expression $\rightarrow A$ denotes a fact. An expression $A(\vec{x}) \rightarrow B(\vec{x})$ denotes a rule where $A(\vec{x})$ is the antecedent and $B(\vec{x})$ the consequent of the rule. A *knowledge source* is a structure $K = (F, R)$ where F is a finite set of facts and R is a finite set of rules. A *multi-source environment* is a structure $E = (W, K_1, \ldots, K_n)$ where W is a workspace, and $K_1, \ldots, K_n$ is a finite collection of knowledge sources.

5 The Formal Semantics

This section constitutes the core part of our logical framework. In the first subsection, we turn to the general notion of bilattice and we present the Belnap's bilattice, a mathematical structure used to define spaces of four valued states. In the second subsection, we define valuations in order to assign truth values to statements. In the three last subsections we respectively concentrate on the notions of epistemic state, epistemic attitude and state transformer.

5.1 The Belnap's bilattice

The idea to use bilattices in computer science is not new (see for instance [4, 6]). We first examine general characteristics about bilattices.

We define a bilattice as a structure $(B, \leq_t, \leq_i)$ where B is a nonempty set and $\leq_t, \leq_i$ are each partial orderings giving B the structure of a lattice. Meet and join under $\leq_t$ (respectively $\leq_i$) are denoted $\wedge$ and $\vee$

(respectively $\sqcap$ and $\sqcup$). We distinguish three important properties about bilattices.

A bilattice $(B, \leq_t, \leq_i)$ is *complete* if its partial orderings give the structure of complete lattices. In a complete bilattice, infinitary meet and join under $\leq_t$ (respectively $\leq_i$) are denoted $\bigwedge$ and $\bigvee$ (respectively $\bigsqcap$ and $\bigsqcup$). A bilattice $(B, \leq_t, \leq_i)$ is *distributive* if all distributive laws connecting $\wedge, \vee, \sqcap$ and $\sqcup$ hold. It is *infinitary distributive* is all finitary and infinitary distributive laws hold. A bilattice $(B, \leq_t, \leq_i)$ is *complemented* if the following conditions hold: (1) $(B, \leq_t, \leq_i)$ is distributive, (2) it has an idempotent *negation* operation $\neg$ which reverses the ordering $\leq_t$ and leaves unchanged the ordering $\leq_i$, (3) it has an idempotent *dual* operation $-$ which reverses the ordering $\leq_i$ and leaves unchanged the ordering $\leq_t$, and (4) the negation and the dual operations commute: for any $x \in B$, $\neg - x = -\neg x$. Finally, the bilattice is *infinitary complemented* if it is both infinitary distributive and complemented.

We now turn to the smallest nontrivial bilattice, called the Belnap's bilattice, and defined as follows:

Definition 5.1 The Belnap's bilattice is an infinitary complemented bilattice $(\mathcal{F}, \leq_t, \leq_i)$ where $\mathcal{F}$ is the powerset $\mathcal{P}(\{false, true\})$. By convention, the elements $\emptyset$, $\{false\}$, $\{true\}$ and $\{false, true\}$ are respectively denoted $\bot$, 0, 1 and $\top$.

Intuitively, the partial ordering $\leq_t$ may be regarded as an "approximation of truth". Meet and join under this ordering directly provide a four valued interpretation for conjunction and disjunction. The partial ordering $\leq_i$ expresses an "approximation of information". For two elements x and y of the Belnap's bilattice, the relationship $x \leq_i y$ may be interpreted to mean that y is at least as informative, or carries at least as knowledge, as x.

5.2 Valuations

Taking the Belnap's bilattice as the basic construct, we may define valuations in a simple and intuitive way.

Definition 5.2 A valuation *in $\mathcal{F}$ is a function from the set of ground atoms $\mathcal{G}_W$ to $\mathcal{F}$. The space of valuations is the set $\mathcal{V} = [\mathcal{G}_W \rightarrow \mathcal{F}]$. It is given two pointwise orderings $\leq_t$ and $\leq_i$ respectively defined by the following conditions:*

$$v \leq_t v' \quad \text{iff} \quad \text{for any } A \in \mathcal{G}_W,\ v(A) \leq_t v'(A),$$
$$v \leq_i v' \quad \text{iff} \quad \text{for any } A \in \mathcal{G}_W,\ v(A) \leq_i v'(A).$$

We remark that a valuation may also be represented by a pair of sets $v = (v^+, v^-)$, where v^+ is the collection of ground atoms which are at least true in v, while v^- is the set of ground atoms which are at least false in v. It should also be noticed that, in practice, most of ground atoms are valued to $\bot$. In the following definition, valuations are extended in order to assign truth values to arbitrary statements of the representation language.

Definition 5.3 Given a valuation v, an extended valuation is a function, also denoted v, from the statements of $\mathcal{L}_W$ to $\mathcal{F}$, inductively defined by the following conditions:

$$\begin{aligned} v(\neg A) &= \neg v(A), \\ v(A \vee B) &= v(A) \vee v(B), \\ v(A \wedge B) &= v(A) \wedge v(B), \\ v((\exists x)A(x)) &= \bigvee \{v(A(c)) : c \in C\}, \\ v((\forall x)A(x)) &= \bigwedge \{v(A(c)) : c \in C\}. \end{aligned}$$

5.3 Epistemic states

In our study, an epistemic state explicitly represents what an agent is told. We consider two key points about epistemic states. First, given the expressivity of our language, every state of knowledge cannot be represented by a single valuation. As suggested for instance in [8], we need to define epistemic states as *sets of valuations*. Second, dual epistemic states must be represented. From this viewpoint, if we examine a valuation v as a pair of sets (v^+, v^-), then it is easy to see that the dual of v is the pair of sets $(\mathcal{G}_W - v^-, \mathcal{G}_W - v^+)$. Notice in particular that the left part of this pair corresponds to the grounds atoms which are not (at least) false under v. Since dual epistemic states are intended to represent the statements which are not refuted given what the agent is told, they may be characterized as *sets of dual valuations*.

Definition 5.4 An epistemic state *σ is a nonempty subset of valuations, i.e. $\sigma \in \mathcal{P}(\mathcal{V})^*$. Given an epistemic state σ, the* dual *of σ is also an epistemic state, denoted σ^-, and defined as follows:*

$$\sigma^- = \{-v : v \in \sigma\}.$$

5.4 Epistemic attitudes

As mentioned in section 3, epistemic attitudes must specify accurately how we want a query answered by defining the interpretation of a statement in an epistemic state. Starting from the preceding definition, an epistemic state may be seen as the set of all statements which are (at least) true in each valuation. Therefore, an attitude expressed in a certain epistemic state relative to a certain statement can be determined by taking the meet under the information ordering $\leq_i$ of all its truth values captured in each valuation. In a symmetrical way, a dual epistemic state may be viewed as the set of all statements which are not (at least) false in each valuation. The notion of "dual attitude" may thus be defined by taking the join under the ordering $\leq_i$ of all its truth values.

Definition 5.5 An epistemic attitude $\mathcal{A}$, *and a* dual attitude $\mathcal{A}^-$ *are functions from statements of* $\mathcal{L}_W$ *and epistemic states of* $\mathcal{P}(\mathcal{V})^*$ *to truth values of* $\mathcal{F}$, *respectively defined by the following conditions:*

$$\begin{aligned}\mathcal{A}(A,\sigma) &= \bigsqcap\{v(A) : v \in \sigma\},\\ \mathcal{A}^-(A,\sigma) &= \bigsqcup\{-v(A) : v \in \sigma\}.\end{aligned}$$

Since the space of valuations is an infinitary complemented bilattice, it is easy to check that $\mathcal{A}(A,\sigma) = -\mathcal{A}^-(A,\sigma)$ for any statement A and any epistemic state σ. Now we turn to the typology of attitudes mentioned in section 3. We suppose that an agent in a given epistemic state σ is asked the query represented by the statement A. Therefore, the four kinds of attitudes may be specified as follows.

$$\begin{array}{rll} A \text{ is } \textit{accepted} \text{ in } \sigma & \text{iff} & \mathcal{A}(A,\sigma) \geq_i 1,\\ A \text{ is } \textit{refuted} \text{ in } \sigma & \text{iff} & \mathcal{A}^-(A,\sigma) \leq_i 0,\\ A \text{ is } \textit{acceptable} \text{ in } \sigma & \text{iff} & \mathcal{A}^-(A,\sigma) \geq_i 1,\\ A \text{ is } \textit{refutable} \text{ in } \sigma & \text{iff} & \mathcal{A}(A,\sigma) \leq_i 0.\end{array}$$

5.5 State Transformers

We now turn to the central question of our functional approach: how is the agent to interpret a new epistemic input ? The goal is to provide systematic connections between epistemic inputs and state transformers. In the setting suggested by our approach, we consider two important points concerning state transformers. First, since the notion of epistemic state is defined in terms of sets of valuations, "state transformers" can be characterized in terms of relations on valuations, or *sets of (binary) transitions.* Second, since we allow dual reasoning in our framework, we need to represent changes of dual epistemic states. In this purpose, we present "dual state transformers" characterized by *sets of dual transitions.* These considerations motivate the following definition:

Definition 5.6 A transition *is a pair* (v, v') *where* $v, v' \in \mathcal{V}$. *A* state transformer π *is a binary relation on* $\mathcal{V}$, *i.e.* $\pi \in \mathcal{P}(\mathcal{V} \times \mathcal{V})$. *Given a state transformer* π, *the* dual of π *is also a state transformer, denoted* π^-, *and defined as follows:*

$$\pi^- = \{(-v, -v') : (v, v') \in \pi\}.$$

It should be noticed here that there is no conflict between relational and functional structures. Every binary relation π on $\mathcal{V}$ induces a function on $\mathcal{P}(\mathcal{V})$ by setting $\pi(\sigma) = \{v' : \exists v \in \mathcal{V} \text{ such that } (v, v') \in \pi\}$. Hence, the epistemic state resulting from π and σ is defined by $\pi(\sigma)$, and its dual state is given by $\pi^-(\sigma^-)$.

A fact is characterized by the expansion it induces on epistemic states. As mentioned in section 3, this transformation should be as minimal as possible in order to respect the criterion of informational economy. The key idea in defining such state transformers comes from the structure of infinitary complemented bilattice.

Definition 5.7 A fact interpretation $[\![\ ]\!]_f$ *maps any fact of* $\mathcal{R}_W$ *to a state transformer of* $\mathcal{P}(\mathcal{V} \times \mathcal{V})$ *such that the following condition holds:*

$$[\![\to A]\!]_f = \{(v, v \sqcup w) : w(A) \geq_i 1\}.$$

Definition 5.8 Let $e = (W, K_1, \ldots, K_n)$ *be a multi-source environment where each knowledge base contains only facts, i.e.* $K_i = (F_i, \emptyset)$ *for* $i \in \{1, n\}$. *The state transformer generated from the environment is denoted* $\pi_{f,e}$ *and defined by the following condition:*

$$\pi_{f,e} = \bigcap_{i=1}^{n} \left(\bigcap \{[\![\to A]\!]_f :\to A \in F_i\}\right).$$

The epistemic state generated from the environment is denoted $\sigma_{f,e}$ *and defined as follows:*

$$\sigma_{f,e} = \pi_{f,e}(\sigma_\perp).$$

Finally, we extend our study to the case where we have rules in knowledge bases. Despite their similarities, rules differ from facts in two points. First, in a four valued setting, a rule $A \to B$ cannot be identified by the fact $\neg A \vee B$. This aspect is notably studied in [11], and the authors have shown in particular that a material implication of the form $\neg A \vee B$ has counterintuitive properties in open environments. In our functional point of view, we characterize a rule by a conditional expansion: if the antecedent of the rule is accepted in a certain epistemic state then the state must be expanded to accept the consequent. Second, we must take into account the free variables of the rule. Intuitively, for each rule instantiation corresponds a particular conditional expansion. The interpretation of the complete rule is thus determined by taking the join (under the lattice $\mathcal{P}(\mathcal{V} \times \mathcal{V})$) of every possible conditional expansions.

Definition 5.9 A rule interpretation $[\![\ ]\!]_r$ *maps any rule of* $\mathcal{R}_W$ *to a state transformer of* $\mathcal{P}(\mathcal{V} \times \mathcal{V})$ *such that the following condition holds:*

$$[\![A(\vec{x}) \to B(\vec{x})]\!]_r = \bigcup_{\vec{d} \in \mathbf{D^k}} \{(v, v \sqcup w) : v(A(\vec{d})) \geq_i 1 \text{ and } w(B(\vec{d})) \geq_i 1\}$$

Definition 5.10 Let $e = (W, K_1, \ldots, K_n)$ *be a multi-source environment where each knowledge base may contain facts and rules, that is,* $K_i = (F_i, R_i)$ *for* $i \in \{1, n\}$*. The state transformer generated from the environment is denoted* $\pi_{r,e}$ *and defined as follows:*

$$\pi_{r,e} = \bigcap_{i=1}^{n} \Big(\bigcap_{A(\vec{x}) \to B(\vec{x}) \in R_i} [\![A(\vec{x}) \to B(\vec{x})]\!]_r \Big).$$

Theorem 5.1 The state transformer $\pi_{r,e} \cap \pi_{f,e}$ *has a least fixpoint and a greatest fixpoint by under the* $\subseteq$ *ordering.*

6 Conclusion

In this paper, we have studied the knowledge merging problem by focusing on the characteristics of incompleteness and inconsistency. This problem was considered in terms of requirements that our logical framework should meet. In particular, we have introduced the concept of dual reasoning which is both able to deal with incompleteness and inconsistency. Furthermore, we have insisted on the criterion of incrementality which enables to build pools of combined knowledge and to improve approximate answers.

From our approach, knowledge merging systems were defined in terms of epistemic states and state transformers. Our first concern was to formally specify such systems as a function of the range of questions they can answer (i.e. the epistemic attitudes) and assertions they can accept (i.e. the epistemic inputs). We have then developed a logical framework based on this functional approach. Dual and incremental reasoning mechanisms have been integrated in order to investigate both incompleteness and inconsistency.

Much work remains to be done before we have a complete understanding of knowledge merging problems. First at all, our typology of epistemic inputs should be extended in order to represent *integrity constraints* which are used in a wide variety of knowledge based systems. Furthermore, we have not tackled the question of an adequate proof theory. In particular, we need to develop such a calculus and prove its soundness and completeness. These two points constitute the main aspect of our future work.

References

[1] C. Baral, S. Kraus, and J. Minker. Combining multiple knowledge bases. *IEEE Trans. on Knowledge and Data Eng.*, 3(2):200–220, 1991.

[2] N. Belnap. A useful four-valued logic. In J.Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 8–37. Dordrecht: D. Reidel, 1977.

[3] L. Cholvy. Proving theorems in a multi-source environment. In *Proc. of IJCAI-93*, pages 66–71, Chambéry. France, 1993. Morgan Kaufmann.

[4] M. Fitting. Bilattices and the semantics of logic programming. *Journal of Logic Programming*, 11:91–116, 1991.

[5] P. Gärdenfors. *Knowledge in Flux : Modeling the Dynamics of Epistemic States.* MIT Press, Cambridge MA, 1988.

[6] M.L. Ginsberg. Multivalued logics: A uniform approach to inference in artificial intelligence. *Computational Intelligence*, 4(3):265–316, 1988.

[7] F. Koriche. A method to model cooperative information systems (in french). *Ingénieurie des Systèmes d'Information (Editions Hermès)*, 4(2):195–218, 1996.

[8] H. Levesque. A logic of implicit and explicit belief. In *Proc. of AAAI 84*, pages 198–202. Morgan Kaufman, 1984.

[9] A. Newell. The Knowledge Level. *Artificial Intelligence*, 18(1):87–127, 1982.

[10] G. Pinkas and R.P. Loui. Reasoning from inconsistency: A taxonomy of principles for resolving conflict. In *Proc. of KR'92*, pages 709–719, 1992.

[11] M.Y. Vardi R. Fagin, J.Y. Halpern. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79:203–240, 1995.

[12] J. van Benthem. *Language in Action. Categories, Lambdas and Dynamic Logic.* North Holland, Amsterdam, 1991.

[13] M. Winslett. *Updating Logical Databases.* Cambridge Tracts in Theorical Computer Science. Cambridge University Press, 1990.

[14] S.T.C. Wong. Preference-based decision making for cooperative knowledge based systems. *ACM Transactions on Information Systems*, 12(4):407–435, October 1994.