MERGING RULE-BASED BELIEF DATABASES

Ricardo Wehbe
Institut für Informatik und angewandte Mathematik, Universität Bern
Neubrückstrasse 10, CH-3012 Bern, Switzerland. E-mail: wehbe@iam.unibe.ch

ABSTRACT
The problem of revising a belief database is treated in many classical works. We will consider here the problem of merging two belief databases (BDBs for short) Ψ_1 and Ψ_2, operation that will be denoted by $\Psi_1 \odot \Psi_2$, and whose result will be a new BDB. Since belief not necessarily reflects the actual state of the world (as opposed to *knowledge*), both BDBs could be incompatible. The goal is to construct a new BDB trying to retain as much as possible of the original beliefs of Ψ_1 and Ψ_2.

KEY WORDS
Logic, Knowledge Representation, Belief Revision.

1 Introduction

The problem of revising a belief database is treated in [1] and many other works. We will consider here the problem of merging two belief databases (BDBs for short) Ψ_1 and Ψ_2, operation that will be denoted by $\Psi_1 \odot \Psi_2$, and whose result will be a new BDB. Since belief not necessarily reflects the actual state of the world (as opposed to *knowledge*), both BDBs could be incompatible. The goal is to construct a new BDB trying to retain as much as possible of the original beliefs of Ψ_1 and Ψ_2.

The paper is organised as follows. In section 2 we will consider the problem in an abstract way, without considering the actual implementation of the BDBs and we will establish some postulates that such an operation should verify, in the spirit of the AGM or the KM postulates [1, 2]. In section 3 we propose a simple language for representing beliefs, similar to Prolog [3] but allowing classical negation and restricted to propositional clauses. Then we will consider the operation of merging two rule-based BDBs. In section 4 we propose a sequent system to obtain the invariant of such a set of clauses and in section 5 we provide a way to merge such systems of clauses. In section 6 we consider the behaviour of such a process with respect to the postulates and section 7 contains the conclusions and some lines of future research.

2 Postulates for Merging Belief Databases

We will assume the existence of some BDBs, which will be collection of formulæ. We assume further that there is a set of propositions Π, whose elements will be denoted by p or q. An *atom* will be either an element of Π or its negation. The set of all atoms that can be formed with the propositions of Π is denoted by $\mathsf{AT}(\Pi)$. The set Π will induce a *universe* $\mathcal{U}_\Pi$, which is a set of *possible worlds*, i.e. maximal consistent subsets of $\mathsf{AT}(\Pi)$ with respect to Π. The syntax of the formulæ will be given by the following grammar:

$$\varphi ::= \mathsf{p} \mid \overline{\mathsf{p}} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

The semantics of formulæ is defined inductively: a world $W \in \mathcal{U}_\Pi$ is a model of p iff $\mathsf{p} \in W$, it is a model of $\overline{\mathsf{p}}$ iff $\mathsf{p} \notin W$, and as usual for more complex formulæ. In the same way, a world $W \in \mathcal{U}_\Pi$ is a model of a BDB Ψ iff it is a model of all its formulæ. Given a BDB Ψ, we will also use $\Psi \models \varphi$ with the usual meaning. The set of logical consequences of a BDB Ψ is the set $\mathcal{C}(\Psi) = \{\varphi \mid \Psi \models \varphi\}$.

There are some conceptual differences between belief merging and belief revision, since when we merge two BDBs Ψ_1 and Ψ_2, we are dealing with two entities of the same status: if there is a conflict, there is no reason why we should prefer the beliefs of Ψ_1 to those of Ψ_2 or vive-versa.

To manage belief-merging we introduce the idea of *conflict sets*. Given two BDBs Ψ_1 and Ψ_2 we define their *conflict set* as the set $\Psi_1 \downarrow \Psi_2 = \{\mathsf{p}, \overline{\mathsf{p}} \mid \{\mathsf{p}, \overline{\mathsf{p}}\} \subseteq \mathcal{C}(\Psi_1) \cup \mathcal{C}(\Psi_2)\}$. The set $\Psi_1 \downarrow \Psi_2$ contains the complementary atoms of the union of the logical consequences of Ψ_1 and Ψ_2.

We will further define the *generating set* of a given subset of logical consequences. If $A \subseteq \mathcal{C}(\Psi)$, then the *generating set* of A in Ψ, denoted by $\mathcal{GD}_\Psi(A)$, is the minimal subset of formulæ of Ψ such that $\mathcal{GD}_\Psi(A) \models A$.

After these prolegomena, we may define the postulates. We assume two BDBs Ψ_1 and Ψ_2.

(M1) If $\Psi_1 \models \varphi$, and (1) $\varphi \notin \mathcal{GD}_{\Psi_2}(\mathcal{C}(\Psi_2) \cap \Psi_1 \downarrow \Psi_2)$, and (2) $\mathcal{GD}_{\Psi_1}(\mathcal{C}_1 \cap \Psi_1 \downarrow \Psi_2) \not\models \varphi$, then $\Psi_1 \odot \Psi_2 \models \varphi$.

(M2) If for all $\mathcal{C}(\Psi_2) \subseteq \mathcal{C}(\Psi_1)$, then $\Psi_1 \odot \Psi_2 \equiv \Psi_1$.

(M3) If both Ψ_1 and Ψ_2 are consistent, then $\Psi_1 \odot \Psi_2$ is consistent.

(M4) If $\Psi_1 \models \varphi \wedge \mu$ and $\Psi_2 \models \overline{\mu}$ and (1) $\varphi \notin \mathcal{GD}_{\Psi_2}(\mathcal{C}(\Psi_2) \cap \Psi_1 \downarrow \Psi_2)$, and (2) $\mathcal{GD}_{\Psi_1}(\mathcal{C}_1 \cap \Psi_1 \downarrow \Psi_2) \not\models \varphi$, then $\Psi_1 \odot \Psi_2 \models \varphi$.

(M5) If $\Psi_1 \models \varphi \vee \mu$ and $\Psi_2 \models \overline{\mu}$ and (1) $\varphi \notin \mathcal{GD}_{\Psi_2}(\mathcal{C}(\Psi_2) \cap \Psi_1 \downarrow \Psi_2)$, and (2) $\mathcal{GD}_{\Psi_1}(\mathcal{C}_1 \cap \Psi_1 \downarrow \Psi_2) \not\models \varphi$, then $\Psi_1 \odot \Psi_2 \models \varphi$.

(M6) If $\Psi_1 \models \varphi$ and $\Psi_2 \models \overline{\varphi}$ then either $\Psi_1 \odot \Psi_2 \models \varphi$ or $\Psi_1 \odot \Psi_2 \models \overline{\varphi}$.

Postulate M1 states that no belief is lost unless it is inconsistent with other beliefs being merged or it is a logical consequence of conflicting beliefs. M2 says that if Ψ_2 is already contained in Ψ_1, then the merging should have no effect. Postulate M3 says that we cannot produce an inconsistent database by merging two consistent ones; any suitable merging procedure should detect and eliminate inconsistencies. M4 states that if Ψ_1 believes $\varphi \wedge \mu$ and Ψ_2 believes $\overline{\mu}$, there is no reason to cease believing φ unless it is in the intersection of $\mathcal{C}(\Psi_1)$ and $\Psi_1 \downarrow \Psi_2$. Postulate M5 is quite straightforward: if Ψ_1 believes $\varphi \vee \mu$ and μ does not hold, then Ψ_1 believes φ. M6 implies the condition that if there are conflicting formulæ, at one of them will be retained.

Postulates M1, M4, M5, and M6 aim at characterising necessary conditions for minimal change. The conditions are not sufficient, since not all formulæ in the generating sets of the conflict set must be eliminated. Next we will analyse the behaviour of a representation of belief [4] in view of these postulates.

3 Syntax and Semantics

We will also start with a set of propositional symbols Π. The syntax of the language is given by the following grammar:

$$\mathsf{clause} ::= \varepsilon \mid \mathsf{atom}[,\mathsf{atom}] \rightarrow \mathsf{atom}$$

A *belief database* (BDB) is a set of clauses K. As usual, the intended meaning of a clause is that if the antecedent is true, so is the consequent. If Π contains all the propositional symbols occurring in K we will say that K is *based* on Π.

If K is a set of clauses based on Π, a *model* of K is a world $W \in \mathcal{U}_\Pi$ such that for all clauses $\Gamma \rightarrow \mathsf{p} \in K$, if $\Gamma \subseteq W$ then $\mathsf{p} \in W$. A set of clauses is *consistent* iff it has a model. The *invariant* of the set of clauses K, denoted by $\mathcal{J}(K)$, is the set of atoms that belong to all models of it.

We will proposed in the next section a sequent system that may be used to construct the invariants and the extensions of a BDB.

4 A Sequent System for Rule-Based Belief Databases

The system B_0 proves properties of tuples $\langle K, S\rangle$, where K is a set of clauses and S is a set of atoms. A tuple is *consistent* iff there is some model W of K such that $S \subseteq W$.

$$\mathsf{id}\ \frac{}{\langle K, \emptyset\rangle}$$

$$\mathsf{inf}\ \frac{\langle K \cup \{\rightarrow \mathsf{p}\}, S\rangle}{\langle K, S \cup \{\mathsf{p}\}\rangle}$$

$$\mathsf{te1}\ \frac{\langle K \cup \{\Gamma, \mathsf{p} \rightarrow \mathsf{q}\}, S\rangle \quad \mathsf{p} \in S}{\langle K \cup \{\Gamma \rightarrow \mathsf{q}\}, S\rangle}$$

$$\mathsf{te2}\ \frac{\langle K \cup \{\Gamma \rightarrow \mathsf{p}\}, S\rangle \quad \mathsf{p} \in S}{\langle K, S\rangle}$$

$$\mathsf{ce1}\ \frac{\langle K \cup \{\Gamma, \overline{\mathsf{q}} \rightarrow \overline{\mathsf{p}}\}, S\rangle \quad \mathsf{p} \in S}{\langle K \cup \{\Gamma \rightarrow \mathsf{q}\}, S\rangle}$$

$$\mathsf{ce2}\ \frac{\langle K \cup \{\Gamma, \overline{\mathsf{p}} \rightarrow \mathsf{q}\}, S\rangle \quad \mathsf{p} \in S}{\langle K, S\rangle}$$

$$\bot\ \frac{\langle K \cup \{\rightarrow \overline{\mathsf{p}}\}, S\rangle \quad \mathsf{p} \in S}{\langle \emptyset, \mathsf{AT}(\Pi)\rangle}$$

$$\mathsf{rec}\ \frac{\begin{array}{ccc} & \langle K, S \cup \{\mathsf{p}\}\rangle & \langle K, S \cup \{\overline{\mathsf{p}}\}\rangle \\ & \vdots & \vdots \\ \langle K, S\rangle & \langle K_1, S_1\rangle & \langle K_2, S_2\rangle \end{array}}{\langle K, S_1 \cap S_2\rangle}$$

Figure 1: The system B_0.

Rule rec may be applied only once for each pair of complementary atoms. This is to avoid infinite sequences, as the following example shows.

Example. Let $C = \{\mathsf{p} \rightarrow \mathsf{q}, \overline{\mathsf{p}} \rightarrow \mathsf{r}\}$. Then rule rec could be indefinitely applied with no changes if we would not have the above proviso.

We will also assume that in rule rec no rules are applicable in the tuples resulting from the subproofs ($\langle K_1, S_1\rangle$ and $\langle K_2, S_2\rangle$.) If rule $\bot$ is applicable, no other rule may be applied; otherwise, if rule inf is applicable, no other rule may be applied.

If K is a set of clauses, then we define a *proof* in system B_0 starting from K as a finite sequence of tuples $\langle K_0, S_0\rangle, \ldots, \langle K_n, S_n\rangle$ such that: (1) $K_0 = K$, (2) $S_0 = \emptyset$, and (3) for all $0 \leq j < n$, tuple $j+1$ is obtained by application of some allowed rule of B_0 on tuple j.

Informally, we begin with a tuple $\langle C, S\rangle$ where $S = \emptyset$ and "collect" all atoms that are bound in S. Once we have collected them, we may eliminate redundancies and contradictions. Rule inf states that if we have the fact $\rightarrow \mathsf{p}$, then all models must include p, and thus p is bound. So, it is added to set S and eliminated from the set of clauses. Rule te1 (te stays for "tautology elimination") state that we may eliminate every occurrence of a bound atom from the consequent of an clause, and te2 says that we need no longer a clause whose consequent is already known to be bound. Rule ce1 (ce stays for "contradiction elimination") says that if we have a clause whose consequent appears negatively in S, then we must rewrite this clause by taking an atom of the antecedent and putting its complement in the consequent. If there are no atoms in the consequent, then rule $\bot$ is applied. Rule ce2 states that we may eliminate

any clause whose antecedent contains the complement of an atom known to be bound, this clause is false and we do not need it anymore. Rule $\bot$ says that if we have an atom p which must appear in all models and a fact $\rightarrow \overline{\mathsf{p}}$, which may only be satisfied with $\overline{\mathsf{p}}$, then the whole set of clauses is unsatisfiable (inconsistent.) Rule rec (the name is for "recursion") is rather more complex. It means that when we have two complementary atoms p and $\overline{\mathsf{p}}$ in a tuple $\langle C, S\rangle$, we may start two subproofs, one beginning with $\langle C \cup \{\rightarrow \mathsf{p}\}, S\rangle$ and the other one beginning with $\langle C \cup \{\rightarrow \overline{\mathsf{p}}\}, S\rangle$. If these subproofs yield $\langle C_1, S_1\rangle$ and $\langle C_2, S_2\rangle$ respectively, then we may go on with $\langle C, S_1 \cap S_2\rangle$. The proof of the correctness of this rule, which is not evident, is in [4].

It also is proved in [4] that this system is sound (if K is a consistent set of clauses, then in any tuple $\langle K_i, S_i\rangle$ belonging to a proof starting with $\langle \Delta, \emptyset\rangle$ all atoms in S_i are bound in K) and complete (if $\langle K_n, S_n\rangle$ is the final tuple of a proof starting from $\langle K, \emptyset\rangle$, then all atoms that are bound in K are in S_n.

A proof is linear except in the case of the rule rec, the only case where the proof branches out into two subproofs. In both cases, all rules must be applied until no one is allowed. Thus, any other occurrence of rule rec must occur within the sub-proof of another application of the same rule. This is shown in the figure next.

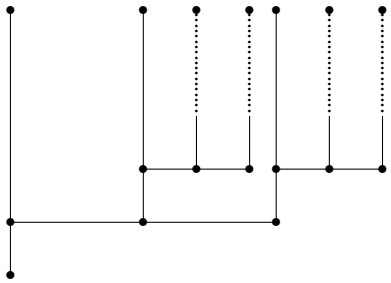

Figure 2: The general structure of a proof.

We will establish some ordering among the atoms of an invariant, which will be used to merge sets of clauses. To do this, we will add the proviso that rule $\bot$ has the highest precedence and rule inf has the second highest precedence over all other rules (if $\bot$ is applicable, no other rule may be applied; if $\bot$ is not applicable and inf is applicable, then no other rule may be applied.) Hence we may inductively assign a pair (x, y) to each step in the proof:

1. The value of (x, y) is $(0, 0)$ at the beginning.
2. Applications of rules inf, ce2, and te2 have no effect on (x, y).
3. If either rule te1 or ce1 are applied, (x, y) is modified to $(x, y + 1)$.
4. If rule rec is applied, (x, y) is modified to $(x + 1, y)$.

The *weight* of an atom in the invariant is the value of the pair (x, y) corresponding to the rule in which it was incorporated to the invariant. Given two atoms p and q, with weights (x_1, y_1) and (x_2, y_2) respectively, we say that p is *lighter* than q, denoted by $\mathsf{p} < \mathsf{q}$, iff (1) $x_1 < x_2$, or (2) $x_1 = x_2$ and $y_1 < y_2$.

5 Merging Sets of Clauses

Now we will consider the merging of two BDBs K_1 and K_2. The interesting case is $K_1 \downarrow K_2 \neq \emptyset$, since otherwise we may just append the clauses modulo some elimination of redundancies. We will assume that a weight $w(\mathsf{p})$ is assigned to each atom p belonging to $\mathcal{J}(K_1)$ and $\mathcal{J}(K_2)$. Notice that the invariant of a set of clauses K should not be confused with its set of logical consequences. In general, $\mathcal{J}(K) \subseteq \mathcal{C}(K)$.

The procedure we propose to merge two sets of clauses does not directly use the conflict sets. Instead, we give preference to lighter atoms. Given two sets of clauses K_1 and K_2, we take the union of them and then we apply the B_1 system on it. This is almost the same as the B_0 system, with two differences: first, the rule $\bot$ is replaced with the following rule:

$$\textsf{ce3}\ \frac{\langle K \cup \{\rightarrow \overline{\mathsf{p}}\}, S\rangle \quad \mathsf{p} \in S}{\langle K, S\rangle}$$

This rule eliminates any clause that causes an inconsistency.

We have to specify what to do when this rule occurs in a subproof within an application of the rec rule. There are two possibilities: either the clause is eliminated in both subproofs and then it must be eliminated from the set of clauses, or it is eliminated in one of the subproofs only and then we discard the subproof. Since rules change as the proofs advances, we assign each rule a unique identifier to keep track of eliminated rules. The notation $k : \Gamma \rightarrow \mathsf{p}$ indicates that the clause has identifier k. The second difference between the B_1 system and the B_0 system is that rule rec rules is replaced by rules rec1, rec2a, and rec2b. In the following rules $K \cup \{k\}$ will be an abbreviation for $K \cup \{k : \Gamma \rightarrow \mathsf{q}\}$, and $S_1 = S \cup \{\mathsf{p}\}$, $S_2 = S \cup \{\overline{\mathsf{p}}\}$.

$$\textsf{rec1}\ \frac{\begin{array}{ccc} & \langle K \cup \{k\}, S_1\rangle & \langle K \cup \{k\}, S_2\rangle \\ & \vdots & \vdots \\ \langle K \cup \{k\}, S\rangle & \langle K_1, S_1'\rangle & \langle K_2, S_2'\rangle \end{array}}{\langle K, S_1' \cap S_2'\rangle}$$

Rule rec1 assumes that rule ce3 was applied on clause k in both subproofs.

$$\textsf{rec2a}\ \frac{\begin{array}{ccc} & \langle K \cup \{k\}, S_1\rangle & \langle K \cup \{k\}, S_2\rangle \\ & \vdots & \vdots \\ \langle K \cup \{k\}, S\rangle & \langle K_1, S_1'\rangle & \langle K_2, S_2'\rangle \end{array}}{\langle K \cup \{k\}, S_2'\rangle}$$

Rule rec2a assumes that rule ce3 was applied on clause k only in the left-side subproof; thus we discard the subproof and use only the right-side one.

Rule rec2b is symmetric to rule rec2a. The idea underlying this system is that when we decide to eliminate one atom of a set of clauses to preserve consistency, it is better to differ the elimination as long as possible, so as to preserve lighter atoms. The following example shows why we give preference to lighter clauses.

Example. Let $K = \{1 :\rightarrow \mathsf{p}, 2 : \mathsf{p} \rightarrow \mathsf{q}, 3 : \mathsf{q} \rightarrow \mathsf{r}, 4 : \mathsf{r} \rightarrow \overline{\mathsf{p}}\}$. We can eliminate any clause to avoid inconsisteny. We have: $\mathcal{J}(K\backslash\{1\}) = \emptyset$; $\mathcal{J}(K\backslash\{2\}) = \{\mathsf{p}\}$, $\mathcal{J}(K\backslash\{3\}) = \{\mathsf{p}, \mathsf{q}\}$, and $\mathcal{J}(K\backslash\{4\}) = \{\mathsf{p}, \mathsf{q}, \mathsf{r}\}$. Minimal change is accomplished in the latter (weightiest) case, since otherwise we eliminate either q or r.

6 Sets of Clauses and the Postulates for Belief Merging

In this section we examine how the system shown previously behaves with respect to the postulates of section 2. We will write k instead of $k : \Gamma \rightarrow \mathsf{p}$. This will allow us to keep track of a clause through its "metamorphoses." We assume now that we have two sets of clauses, K_1 and K_2.

The first postulate says that if $K_1 \models \varphi$, and (1) $\varphi \notin \mathcal{GD}_{K_2}(\mathcal{C}(K_2) \cap K_1 \downarrow K_2)$, and (2) φ is not a logical consequence of $\mathcal{GD}_{K_1}(\mathcal{C}(K_1) \cap K_1 \downarrow K_2)$, then $K_1 \odot K_2 \models \varphi$. Suppose that we have a proof on system B_1 beginning with $K_1 \cup K_2$. Since $k \notin (\mathcal{C}(K_1) \cap K_1 \downarrow K_2$ and $k \notin (\mathcal{C}(K_2) \cap K_1 \downarrow K_2$, then rule ce3 will not be applied on k. If either rule ce2 or te2 is applied on k, then the original clause is still a logical consequence of $K_1 \cup K_2$, since in the former case the antecedent is false and in the latter case the consequent is true. If rule te1 is applied, then the original clause is also a logical of $K_1 \cup K_2$, since we have retracted from the antecedent an atom which is bound in the set of clauses. Finally, if we apply rule ce1, we have a clause $\Gamma, \overline{\mathsf{q}} \rightarrow \overline{\mathsf{p}}$ and the consequent is false. So, for this clause is a logical consequence of $K_1 \cup K_2$, iff $K_1 \cup K_2 \models \overline{\Gamma}$ or $K_1 \cup K_2 \models \mathsf{q}$ iff $K_1 \cup K_2 \models \Gamma \rightarrow \mathsf{q}$.

Postulate M2 says that if for all φ such that $K_2 \models \varphi$ it is the case that $K_1 \models \varphi$, then $K_1 \odot K_2 \equiv K_1$. This postulate is fulfilled, since the set of clauses will have some redundancies but no new logical consequences.

Postulate M3 says that if both K_1 and K_2 are consistent, then $K_1 \odot K_2$ is consistent. By the properties of the system, if $K_1 \odot K_2$ is inconsistent, rule ce3 will be applied on the "offending" clause k. The same reasoning can be iterated with $K_1 \odot K_2\backslash\{k\}$.

The fourth postulate says that if $K_1 \models \varphi \wedge \mu$ and $K_2 \models \overline{\mu}$ and (1) $\varphi \notin GD_{K_2}(\mathcal{C}_2 \cap K_1 \downarrow K_2)$, and (2) φ is not a logical consequence of $GD_{K_1}(\mathcal{C}_1 \cap K_1 \downarrow K_2)$, then $K_1 \odot K_2 \models \varphi$. In the case of sets of clauses this is a consequence of the first postulate, since the conjunction of φ and μ is originated in the union of the sets of clauses that entail φ and the set of clauses that entail μ. Thus we can separate both sets, and then apply the first postulate to clause φ.

We let M5 aside, since our sets of clauses do not model disjunctions.

Postulate M6 is also fulfilled, since to eliminate one conflicting atom its complement must be already in the invariant (rule ce3.)

7 Conclusions and Future Work

We have proposed some postulates that may provide a good basis for the behaviour of a system in which belief merging is performed. Besides, we have proposed a belief representation based on rules, similar to the ones presented in [5, 6]. This has the advantage of being simple enough to throw some light on the problem. Belief merging is similar to belief revision [1, 7, 2]. The aim is quite the same: to avoid inconsistencies while trying to retain as much as possible of older beliefs.

We are working on extended this approach to nonmonotonic logic, especially *default logic* [8, 9]. This implies several interesting problems, as a definition of several degrees of belief (depending a formula is in the invariant, in all extensions, or in some extensions) and it seems to be a promising approach to work with *compromise belief revision* [10]. We are also investigating the possibility of expressing introspection [11] within this framework.

Besides, the postulates may be refined, since they could be too strong in its current state.

References

[1] Alchourrón, C.; Gärdenfors, P.; Makinson, D., On the Logic of Theory Change: Partial Meet Contraction and Revision Functions, *The Journal of Symbolic Logic* 50 (2), 1985, pp. 510–530.

[2] Katsuno, H.; Mendelzon, A., On the Difference Between Updating a Knowledge Base and Revising it, *Proc. of KR-1991*, Morgan-Kaufmann, pp. 387–394.

[3] Lloyd, J.W.: *Foundations of Logic Programming*, Springer Verlag, 1987.

[4] Wehbe, R., A Sequent System for Sets of Clauses, Tech. Report IAM-06-007, Institut für Informatik und angewandte Mathematik, University of Bern, www.iam.unibe.ch/publikationen/tech-reports/2006/

[5] Alechina, N.; Jago, M.; Logan, B., Modal Logics for Communicating Rule-Based Agents, *Proc. of the 17th European Conf. on Artificial Intelligence (ECAI'06)*, IOS Press, 2006, pp. 322–326.

[6] Sadri, F.; Toni, F., Interleaving Belief Updating and Reasoning in Abductive Logic Programming, *Proc. of ECAI'06*, IOS Press, pp. 442–446.

[7] Herzig, A.; Rifi, O., Propositional Belief Base Update and Minimal Change, *Artificial Intelligence* 115 (1), 1999, pp. 107–138.

[8] Besnard, Ph.: *An Introduction to Default Logic*, Springer Verlag, 1989.

[9] Marek, W.; Truszczyński, M.: *Nonmonotonic Logic*, Springer-Verlag, 1993.

[10] Gabbay, D.M., Compromise Update and Revision: A Position Paper, in R. Pareschi and B. Frönhofer (eds.), *Dynamic Worlds*, Kluwer Academic Publishers, 1999, pp. 111–148.

[11] Fagin, Ronald; Halpern, Joseph; Moses, Yoram; Vardi, Moshe: *Reasoning About Knowledge*, The MIT Press, Cambridge, MA, 1996.