Autoepistemic Belief-revision for Integration of Mutually Inconsistent Knowledge

Zoran Majkić/ Copyright IICAI

Applied Mathematics Department
ETF, University of Belgrade, Serbia
majkic@etf.bg.ac.yu
http://$www.geocities.com/zoran\_it$/

Abstract. It is well known that the standard 3-valued logic programs with constraints can be inconsistent. Because of that we can not use it for a data integration where mutually inconsistent information comes from different data sources. We argue that a natural way to answer to this challenge, without collapsing all sentences into inconsistency, is by passing to 4-valued bilattice-based logic (with logic values: true, false, unknown and possible), and by interpreting the inconsistent information with a logic value "possible". Differently from the paraconsistent approach we adopt the belief-revision approach, but in such many-valued repairing of inconsistent information we do not eliminate mutually inconsistent information as in the case of a 2-valued database repairing.
The original contribution of this paper is an Autoepistemic Many-valued Logic with intuitionistic implication, epistemic negation and Moore's modal operator: the inference reasoning of this logic is able to change its belief in the truth value of ground facts which come from external sources, and to preserve its internal consistency. We show that each Autoepistemic Logic Program is consistent and we define its minimal many-valued Herbrand models.

1 Introduction

As classical logic semantics decrees that inconsistent theories have no models, classical logic is not the appropriate formalism for reasoning about inconsistent databases: some inconsistences should not be allowed to significantly alter the intended meaning of such databases. Once one has made the transition from classical 2-valued logic, with stable models, [1,2], to partial models [3,4], or three-valued logic models [5,6,7,8], allowing *incomplete* information, it is a small step towards allowing models admitting *inconsistent* information. Doing so provides a natural framework for the semantic understanding of logic programs that are distributed over several sites, with possible conflicting information coming from different places.
So far, research in many-valued logic programming has proceeded along different directions: *Signed* logics [9,10] and *Annotated* logic programming [11,12,13] which can be embedded into the first, *Bilattice-based* logics, [14,15], and *Quantitative rule-sets*, [16]. Earlier studies of these approaches quickly identified various distinctions between these frameworks. For example, one of the key insights behind bilattices was the interplay between the truth values assigned to sentences and the (non classic) notion of *implication* in the language under considerations. Thus, rules (implications) had

weights (or truth values) associated with them as a whole. The problem was to study how truth values should be propagated "across" implications. Annotated logics, on the other hand, appeared to associate truth values with each component of an implication rather than the implication as a whole. Roughly, based on the way in which uncertainty is associated with facts and rules of a program, these frameworks can be classified into *implication based* (IB) and *annotation based* (AB).
In the IB approach a rule is of the form $A \leftarrow^{\alpha} B_1, .., B_n$, which says that the certainty associated with the implication is α. Computationally, given an assignment I of logical values to the B_is, the logical value of A is computed by taking the "conjunction" of logical values $I(B_i)$ and then somehow "propagating" it to the rule head A.
In the AB approach a rule is of the form $A : f(\beta_1, .., \beta_n) \leftarrow B_1 : \beta_1, ..., B_n : \beta_n$, which asserts "the certainty of the atom A is least (or is in) $f(\beta_1, .., \beta_n)$, whenever the certainty of the atom B_i is at least (or is in) β_i, $1 \leq i \leq n$", where f is an n-ary computable function and β_i is either constant or a variable ranging over many-valued logic values.
The comparison in [17] shows:
1- while the way implication is treated on the AB approach is closer to the classical logic, the way rules are fired in the IB approach has definite intuitive appeal.
2- the AB approach is strictly more expressive than IB. The down side is that query processing in the AB approach is more complicated, e.g. the fixpoint operator is not continuous in general, while it is in the IB approaches.
From the above points discussed in [17], it is believed that IB approach is easier to use and is more amenable for efficient implementations.
The other problem is that the Fitting fixpoint semantics for IB logic programs, based exclusively on a bilattice-algebra operators, suffer two drawbacks: the lack of the notion of tautology (bilattice negation operator is an *epistemic* negation) leads to difficulties in defining proof procedures and to the need for additional complex truth-related notions as "formula closure"; there is an unpleasant asymmetry in the semantics of implication (which is strictly 2-valued) w.r.t. all other bilattice operators (which produce any truth value from the bilattice) - it is a sign that strict bilattice language is not enough expressive for logic programming, and we need some reacher (different) syntax for logical programming.
What we need is a right definition for the many-valued implication, not present in a bilattice algebra operators, and the more flexible consideration of the truth of the given ground facts. We can consider a Data integration system (DIS) as an intelligent agent with a knowledge base and a proper reasoning capability, which integrates the (external) extensions of source databases represented by the set of facts (ground atoms). Thus, instead of considering such external ground facts rigidly true, such intelligent DIS can create its own *many-valued belief* for such facts in order to preserve the internal consistency. For instance, if two facts from two different sources are mutually inconsistent (when they are considered as *true* information) w.r.t. the DIS knowledge system, they can be believed as *possible* (4-th logic value) information.
The purpose of this paper is to show that the Modal autoepistemic Moore's operator defined for the 4-valued Belnap's bilattice, together with the intuitionistic semantics for many-valued logic implication, can be unified into a simple autoepistemic many-valued

logic programming language, which is capable to reason with inconsistency.
The approach taken in this paper is minimalistic: we make a minimal extension of the 3-valued logic programming into the 4-valued (Belnap's bilattice) logic, and the extension of the standard logic programming by modal formulae for ground facts only. Such effort is justified by the very intuitive and natural approach to resolve inconsistency and by the possibility to obtain the simple monotonic w.r.t. the knowledge ordering "immediate consequence operator" for the least-fixpoint calculation of the semantics of such autoepistemic logic programs.
Other important point is that we use the 4-valued Belnap's bilattice, which is used also in the paraconsistent logic programming, but differently from such approach we use the *belief-revision* where we change the truth of some ground facts (which make inconsistent logic theory) from their initial value t (true) into value $\top$ (possible).
We argue that such logic will be good framework for supporting the data integration systems with key and foreign key integrity constraints with incomplete and inconsistent source databases, with a minor computational complexity for query-answering [18].
Such issue will not be considered here because of space limitations, but intuitively comes from the fact that the unique Herbrand many-valued canonical model, which represents the semantics for the family of Autoepistemic logic programs defined in this paper, is more simple framework for a query answering than the 2-valued framework of data-repairing where the number of models (minimal repairs) can be very high. We remark that in our framework we can have not only certain answers to user queries, but also answers with a possible logic value, so that in query-answering we do not lose any information in data sources.
The plan of this paper is the following: After brief introduction to Belnap-Ginsberg bilattices and modal operators for bilattices (nondeductive Moor's modal operator M), in Section 3 we discuss the semantic issue for bilattice-based repairs of inconsistent information: i.e., derivation of true, false and *possible* information also. In Section 4 we define the 4-valued bilattice inference requirements and the many-valued *intuitionistic* implication for rules of a 4-valued logic programming, which is used for modal formulae over source-data facts. Finally, in Section 5 we present the syntax of the autoepistemic programming logic, their semantics based on the set of minimal (w.r.t. the belief revision) 4-valued Herbrand models and their *canonical* 4-valued Herbrand model semantics.

2 Preliminaries: Introduction to Belnap/Ginsberg's bilattice

In [19], Belnap introduced a logic intended to deal in a useful way with inconsistent or incomplete information. It is the simplest example of a non-trivial bilattice and it illustrates many of the basic ideas concerning them. We denote the four values as $\mathcal{B} = \{t, f, \top, \bot\}$, where t is *true*, f is *false*, $\top$ is inconsistent (both true and false) or *possible* , and $\bot$ is *unknown*.
As Belnap observed, these values can be given two natural orders: *truth* order, $\leq_t$, and *knowledge* order, $\leq_k$, such that $f \leq_t \top \leq_t t$, $f \leq_t \bot \leq_t t$, and $\bot \leq_k f \leq_k \top$, $\bot \leq_k t \leq_k \top$. This two orderings define corresponding equivalences $=_t$ and $=_k$. Thus any two members α, β in a bilattice are equal, $\alpha = \beta$, if and only if (shortly 'iff')

$\alpha =_t \beta$ and $\alpha =_k \beta$.
Meet and join operators under $\leq_t$ are denoted $\wedge$ and $\vee$; they are natural generalizations of the usual conjunction and disjunction notions. Meet and join under $\leq_k$ are denoted $\otimes$ (*consensus*, because it produces the most information that two truth values can agree on) and $\oplus$ (*gullibility*, it accepts anything it's told). We have that:
$f \otimes t = \bot$, $f \oplus t = \top$, $\top \wedge \bot = f$ and $\top \vee \bot = t$.
There is a natural notion of truth negation, denoted $\sim$, (reverses the $\leq_t$ ordering, while preserving the $\leq_k$ ordering): switching f and t, leaving $\bot$ and $\top$, and corresponding knowledge negation, denoted $-$, (reverses the $\leq_k$ ordering, while preserving the $\leq_t$ ordering):switching $\bot$ and $\top$, leaving f and t. These two kind of negation commute: $- \sim x = \sim -x$ for every member x of a bilattice.
It turns out that the operations $\wedge, \vee$ and $\sim$, restricted to $\mathcal{B}_3^{\bot} = \{f, t, \bot\}$ are exactly those of Kleene's strong 3-valued logic. A more general information about bilattice may be found in [15]. The Belnap's 4-valued bilattice is infinitary distributive. In the rest of this paper we denote by $\mathcal{B}_4$ (or simply $\mathcal{B}$) this 4-valued Belnap's bilattice.
A (ordinary) Herbrand interpretation is a many-valued mapping $I : H_P \rightarrow \mathcal{B}$. If P is a many-valued logic program with the Herbrand base H_P , then the ordering relations and operations in a bilattice $\mathcal{B}_4$ are propagated to the function space $\mathcal{B}_4^{H_P}$, that is the set of all Herbrand interpretations (functions), $I = v_B : H_P \rightarrow \mathcal{B}_4$.

Definition 1. *Ordering relations are defined on the Function space $\mathcal{B}_4^{H_P}$ pointwise, as follows: for any two Herbrand interpretations $v_B, w_B \in \mathcal{B}_4^{H_P}$*
1. $v_B \leq_t w_B$ if $v_B(A) \leq_t w_B(A)$ for all $A \in H_P$.
2. $v_B \leq_k w_B$ if $v_B(A) \leq_k w_B(A)$ for all $A \in H_P$.
3. $\sim v_B$, such that $(\sim v_B)(A) = \sim (v_B(A))$: $-v_B$, such that $(-v_B)(A) = -(v_B(A))$.

It is straightforward [15] that this makes a function space $\mathcal{B}_4^{H_P}$ itself a complete infinitary distributive bilattice.

Definition 2. *[15] Let P be a logic program, with P^* the set of all ground instances of members of P, and a $\mathcal{B}$-valuation $I : H_P \rightarrow \mathcal{B}_4$. We define the monotonic in $\leq_k$ immediate consequence operator $\Phi_P : \mathcal{B}_4^{H_P} \rightarrow \mathcal{B}_4^{H_P}$ such that for each $A \in H_P$,*
1. if A is not the head of any member of P^*, $\Phi_P(I)(A) = f$,
2. otherwise, $\Phi_P(I)(A) = \bigvee\{I(B) \mid A \leftarrow B$ is in $P^*\}$.

Ginsberg [14] defined a world-based bilattices, considering a collection of worlds W, where by world we mean some possible way of things might be:

Definition 3. *[14] A pair $[U, V] \in \mathcal{P}(W) \times \mathcal{P}(W)$ of subsets of W (here $\mathcal{P}(W)$ denotes the powerset of the set W) express truth of some sentence p, with $\leq_t, \leq_k$ truth and knowledge preorders relatively, as follows:*
1. U is a set of worlds where p is true, V is a set of worlds where p is false, $P = U \bigcap V$ is a set where p is inconsistent (both true and false), and $W - (U \bigcup V)$ is a set where p is unknown.
2. $[U, V] \leq_t [U_1, V_1]$ iff $U \subseteq U_1$ and $V_1 \subseteq V$
3. $[U, V] \leq_k [U_1, V_1]$ iff $U \subseteq U_1$ and $V \subseteq V_1$.
The bilattice operations associated with $\leq_t$ and $\leq_k$ are:
4. $[U, V] \wedge [U_1, V_1] = [U \bigcap U_1, V \bigcup V_1]$, $[U, V] \vee [U_1, V_1] = [U \bigcup U_1, V \bigcap V_1]$

5. $[U, V] \bullet [U_1, V_1] \ = \ [U \bigcap U_1, V \bigcap V_1]$ *,* $[U, V] + [U_1, V_1] \ = \ [U \bigcup U_1, V \bigcup V_1]$
6. $\sim [U, V] \ = \ [V, U]$*.*
Let denote by $\mathcal{B}_W$ *the set* $\mathcal{P}(W) \times \mathcal{P}(W)$*, then the structure* $(\mathcal{B}_W, \wedge, \vee, \bullet, +, \sim)$ *is a bilattice.*

This definition is well suited for the 3-valued Kleene logic, but not for the 4-valued logic used to overcome "localizable" inconsistencies. It is not useful, mainly for two following reasons:
1. The *inconsistent* (both true and false) top knowledge value $\top$ in the Belnap's bilattice can't be assigned to sentences, otherwise we will obtain an inconsistent logic theory where *all* sentences are inconsistent; because of that, consistent logics in this interpretation can have only three remaining values. Thus, we interpret $\top$ as *possible* value, which will be assigned to mutually inconsistent sentences, in order to obtain the consistent 4-valued logic theories that overcome such 3-valued inconsistencies.
2. Let us denote by $T = U - N$, $F = V - N$, where N is a set of worlds where p has a logic value 'possible'. Then we obtain that $[U, V] \ \leq_t \ [U_1, V_1]$ also when $T \supset T_1$, which is in contrast with our intuition. For example, let $W = [1, 100]$ be the closed interval of integers (indexes for a collection of worlds), $U = [1, 60]$, $V = [50, 100]$, and $U_1 = [1, 60]$, $V_1 = [40, 100]$: then $[U, V] \ \leq_t \ [U_1, V_1]$ while $T = U - N = [1, 50] \supset [1, 40] = U_1 - N_1 = T_1$, and $F = F_1 = [60, 100]$!
Consequently, we adopt a triple $[T, N, F]$ of mutually disjoint subsets of W to express truth of some sentence p (the $W - T \bigcup N \bigcup F$ is a set of worlds where p is unknown), with the following definition for their truth and knowledge orders:
2.1 $[T, N, F] \leq_t [T_1, N_1, F_1]$ iff $T \subseteq T_1$ and $F_1 \subseteq F$
2.2 $[T, N, F] \leq_k [T_1, N_1, F_1]$ iff $T \subseteq T_1$, $N \subseteq N_1$ and $F \subseteq F_1$.
The meet and join truth and knowledge operations for this extended bilattice can be found in [20]. In this way we consider the *possible* value as weak true value and *not as inconsistent* (that is both true and false). We have more knowledge for ground atom with such value, w.r.t. the true ground atom, because we know also that if we assign the true value to such atom we may obtain an inconsistent database.
The difference of a possible and unknown value may be explained also intuitively as follows: if we consider a 3-valued Kleene's strong logic, and try to use it in order to give a semantics for databases with inconsistencies, then we will obtain a number of stable 3-valued models (minimal 'repairs') for it. In each of such stable model the set of unknown ground atoms is *invariant*: if one atom is unknown in some model it remains unknown in all other stable models. But we will have some atom true in some and false in some other stable model: to such atoms we can assign the *possible* logic value in a framework of this 4-valued logic, in order to obtain a minimal Herbrand model. Thus, "both true and false" of Belnap's interpretation for $\top$ can be relaxed in "true in some possible world and false in some *other* possible world". Because of that, we prefer the Lukasiewicz's term "possible" for top-knowledge logic value $\top$, and Kleene's term "unknown" for bottom-knowledge logic value $\perp$.
In [21] is given the definition for modal operators on bilattices, which generalizes both Kripke's possible world approach and Moore's autoepistemic logic: a modal operator is any n-ary function from the bilattice $\mathcal{B}$ to itself, with the following property:

Definition 4. *[21] A modal operator on a bilattice $\mathcal{B}$ will be called* deductive *if and only if it commute with $\otimes$ and $\oplus$. All other modal operators will be called* nondeductive.

For example, *nondeductive* modal operator [22] is Moore's operator M, where $M(p)$ is intended to capture the notion of, "I know that p" i.e.,
$M(\alpha) = t$ if $\alpha \in \{t, \top\}$; f otherwise.
This nondeductive autoepistemic Moore's modal operator will be used for logic programming in presence of (mutually) inconsistent information, and is a reason to denominate such programs as Autoepistemic logic programs: in order to relax a belief on ground facts and be able to reason in the presence of the inconsistent information also, a ground fact, $p \leftarrow t$, is substituted by a modal formula $M(p \leftarrow t)$.

3 From inconsistency toward possibility

In order to obtain a new bilattice abstraction rationality, useful to manage logic programs with *inconsistencies*, we need to consider more deeply the *fundamental phenomena* in such one framework. In the process of derivation of new facts, for a given logic program, based on the 'immediate consequence operator', we have the following three truth transformations for ground atoms in a Herbrand base of such program:
1. When a ground atom pass from *unknown* to *true* logic value (it means that the value of an atom was unknown and in the next iteration it becomes true), without generating inconsistence. Let us denote this action by $\uparrow_1\colon \perp \rightarrowtail t$. The preorder of this 2-valued sublattice of $\mathcal{B}$, $L_1 = \{\perp, t\}$, defined by the direction of this transformation, 'truth increasing', is $\leq_1 \equiv \leq_t$. The meet and join operators for this lattice are $\wedge, \vee$ respectively. It is also knowledge increasing.

2. When some ground atom, tries to pass from unknown to true/false value, generating an inconsistency, then is applied the *inconsistency repairing*, that is, the *true* value of the literal of this atom, in a body of a violated clause with built-in predicate, is replaced by *possible* value. Let us denote this action by $\uparrow_2\colon t \rightarrowtail \top$. The preorder of this 2-valued sublattice of $\mathcal{B}$, $L_2 = \{t, \top\}$, defined by the direction of this transformation, is 'knowledge increasing'. The meet and join operators for this lattice, w.r.t. this ordering, are $\otimes, \oplus$ respectively. Notice that this transformation *does not change* the truth ordering because the ground atom pass from unknown to possible value.

3. When a ground atom pass from *unknown* to *false* logic value, without generating inconsistence. Let us denote this action by $\uparrow_3\colon \perp \rightarrowtail f$. The preorder of this 2-valued sublattice of $\mathcal{B}$, $L_3 = \{\perp, f\}$, defined by the direction of this transformation, 'falsehood increasing' (inverse of 'truth increasing'), is $\leq_3 \equiv \leq_t^{-1}$. The meet and join operators for this lattice are $\vee, \wedge$ respectively. It is also knowledge increasing.

Thus, any truth transformation in some multi-valued logic theory (program) can be seen as composition of these three orthogonal dimensional transformations,i.e. by triples (or *multi-actions*), $[a_1, a_2, a_3]$, acting on the idle (default) state $[\perp, t, \perp]$; for instance the multi-action $[\_, \_, \uparrow_3]$, composed by the single action $\uparrow_3$, applied to the default state generates the "false" state $[\perp, t, f]$. The default state $[\perp, t, \perp]$ in this 3-dimensional space has role as unknown value for single-dimensional bilattice transformations, that is it is a "unknown" state. Consequently, we define this space of states by the cartesian product of single-dimensional lattices, $L_1 \times L_2 \times L_3$, composed by triples $[x, y, z]$, $x \in L_1 = \{\perp, t\}$, $y \in L_2 = \{t, \top\}$ and $z \in L_3 = \{\perp, f\}$.

Definition 5. *By $L_1 \odot L_2 \odot L_3$ we mean the bilattice $< L_1 \times L_2 \times L_3, \leq_t^B, \leq_k^B >$ where, given any $X = [x, y, z]$, and $X_1 = [x_1, y_1, z_1]$:*
1. Considering that the second transformation does not influence the truth ordering,
$X \leq_t^B X_1$ *if* $x \leq_1 x_1$ *and* $z \leq_3 z_1$ *, i.e., if* $x \leq_t x_1$ *and* $z \geq_t z_1$
2. Considering that all three transformations are knowledge increasing, we have
$X \leq_k^B X_1$ *if* $x \leq_k x_1$ *and* $y \leq_k y_1$ *and* $z \leq_k z_1$
3. $X \wedge_B X_1 =_{def} [(x \wedge_1 x_1,\ y \wedge_1 y_1), z \wedge_3 z_1] = [x \wedge x_1,\ y \wedge y_1,\ z \vee z_1]$
4. $X \vee_B X_1 =_{def} [x \vee_1 x_1, (y \vee_3 y_1,\ z \vee_3 z_1)] = [x \vee x_1,\ y \wedge y_1,\ z \wedge z_1]$
5. $X \otimes_B X_1 =_{def} [x \otimes x_1,\ y \otimes y_1,\ z \otimes z_1]$, $X \oplus_B X_1 =_{def} [x \oplus x_1,\ y \oplus y_1,\ z \oplus z_1]$

The item 1 of this definition corresponds to the fact that the second transformation does not influence the truth ordering, while the item 2 corresponds to the fact that all three transformations are knowledge increasing.
These three bilattice transformations can be formally defined by the lattice homomorphisms.

Proposition 1 *The following three lattice homomorphisms define the 3-dimensional truth transformations:*
1. Truth dimension, $\theta_1 = \_ \vee \perp : (\mathcal{B}, \wedge, \vee, \otimes, \oplus) \rightarrow (L_1, \wedge_1, \vee_1, \otimes, \oplus)$,
with $\wedge_1 = \wedge$, $\vee_1 = \vee$. *This is a strong positive transformation, which transforms falsehood into unknown and possibility in truth.*
2. Possibility dimension, $\theta_2 = \_ \vee \sim \_ \vee \top : (\mathcal{B}, \otimes, \oplus) \rightarrow (L_2, \otimes, \oplus)$.
This is a weak knowledge transformation which transforms unknown into possibility.
3. Falsehood dimension, $\theta_3 = \_ \wedge \perp : (\mathcal{B}, \vee, \wedge, \otimes, \oplus) \rightarrow (L_3, \wedge_3, \vee_3, \otimes, \oplus)$,
with $\wedge_3 = \vee$, $\vee_3 = \wedge$. *This is a strong negative transformation, which transforms truth into unknown and possibility into falsehood.*
We define the following two mappings between Belnap's and its derived bilattice:
Dimensional partitioning: $\theta =< \theta_1, \theta_2, \theta_3 >: \mathcal{B} \rightarrow L_1 \odot L_2 \odot L_3$,
Collapsing: $\vartheta : L_1 \odot L_2 \odot L_3 \rightarrow \mathcal{B}$*, such that* $\vartheta(x_1, x_2, x_3) =_{def} (x_1 \oplus x_3) \wedge x_2$.

These three lattice homomorphisms preserve the bilattice structure of $\mathcal{B}$ into the space of states $L_1 \odot L_2 \odot L_3$. That is we have that ($'\_'$ represents no action)

$\theta(\perp) = [\_, \_, \_]([\perp, t, \perp]) = [\perp, t, \perp]$, unknown state
$\theta(f) = [\_, \_, \uparrow_3]([\perp, t, \perp]) = [\perp, t, f]$, false state
$\theta(t) = [\uparrow_1, \_, \_]([\perp, t, \perp]) = [t, t, \perp]$, true state
$\theta(\top) = [\uparrow_1, \uparrow_2, \uparrow_3]([\perp, t, \perp]) = [t, \top, f]$, possible state.

Notice that the multi-action $[\uparrow_1, \uparrow_2, \uparrow_3]$ represents two cases for repairing inconsistencies: first, when unknown value of some ground atom tries to become true (action $\uparrow_1$) but makes inconsistency, then is applied also action $\uparrow_2$ to transform it into the possible value $\top$; second, when unknown value of some ground atom tries to become false (action $\uparrow_3$) but makes inconsistency, then is applied also action $\uparrow_2$ to transform it into the possible value $\top$. Notice that the isomorphism between the set of states and the set of multi-actions $\{[a_1, a_2, a_3] \mid a_1 \in \{\uparrow_1, \_\},\ a_2 \in \{\uparrow_2, \_\},\ a_3 \in \{\uparrow_3, \_\}\}$ defines the *semantics* to the bilattice $L_1 \odot L_2 \odot L_3$.

Proposition 2 *Let $Im\theta \subseteq L_1 \odot L_2 \odot L_3$ be the bilattice obtained by image of Dimensional partitioning. It has also unary operators:*
Negation, $\sim_B = \theta \sim \vartheta$, and conflation, $-_B = \theta - \vartheta$.

It is easy to verify that $\vartheta \circ \theta = id_{\mathcal{B}}$ is an identity on $\mathcal{B}$, and that ϑ is surjective with $\theta \circ \vartheta = id_{Im\theta}$. The negation $\sim_B$ preserves knowledge and inverts truth ordering and $\sim_B \sim_B X = X$; the conflation $-_B$ preserves truth and inverts knowledge ordering and $-_B -_B X = X$; and holds the commutativity $\sim_B -_B = -_B \sim_B$. (for example, $\sim_B -_B = \theta \sim \vartheta\theta - \vartheta = \theta \sim id_{\mathcal{B}} - \vartheta = \theta \sim -\vartheta = \theta - \sim \vartheta = \theta - \vartheta\theta \sim \vartheta = -_B \sim_B$). So, for any $X = [x, y, z]$, hold $\sim_B X =_{def} [\sim z,\ y, \sim x]$ and $-_B X =_{def} [\theta_1(-z),\ \theta_2(-\vartheta(X)),\ \theta_3(-x)]$.

Theorem 1. *(Representation theorem) If $\mathcal{B}$ is a 4-valued distributive lattice then there are its distributive sublattices, L_1, L_2, L_3, such that $\mathcal{B}$ is isomorphic to the sublattice of $L_1 \odot L_2 \odot L_3$ defined by image of Dimensional partitioning $Im\theta$. Moreover the following diagram (on the left) of bilattice homomorphisms commutes*

$$\begin{tikzcd}
L_1 \odot L_2 \odot L_3 \arrow[r, "\theta\vartheta"] \arrow[dr, "\vartheta"'] & Im\theta \arrow[r, "\simeq_{\pi_1 \times \pi_3}"] \arrow[d, "\simeq_4"] & L_1 \odot L_3 \arrow[dl, "\simeq_3"] \\
& \mathcal{B} &
\end{tikzcd}
\qquad
\begin{tikzcd}
& L_2(possibility) & \\
(falsehood)L_3 & \mathcal{B} \arrow[l, "\theta_3"'] \arrow[r, "\theta_1"] \arrow[u, "\theta_2"] & L_1(truth)
\end{tikzcd}$$

where $\simeq_{\pi_1 \times \pi_3}$ is a projection isomorphism, $\simeq_3$ is the isomorphism (restriction of ϑ to the projection $L_1 \odot L_3$) of Fitting's representation Th. [23] valid for a 3-valued logics, and $\simeq_4$ is the new 4-valued isomorphism (restriction of ϑ to $Im\theta$, and inverse to θ).
If $\mathcal{B}$ has negation and conflation operators that commute with each other, they are preserved by all isomorphisms of the right commutative triangle.

Proof. It is easy to verify that all arrows are homomorphisms (w.r.t. binary bilattice operators). The following table represents the correspondence of elements of these bilattices defined by homomorphisms:

$Multi-actions$	$L_1 \odot L_2 \odot L_3$	$Im\theta$	$L_1 \odot L_3$	$\mathcal{B}$
$[\_, \_, \_]$	$[\bot, t, \bot]$	$[\bot, t, \bot]$	$[\bot, \bot]$	$\bot$
$[\_, \uparrow_2, \_]$	$[\bot, \top, \bot]$			
$[\_, \_, \uparrow_3]$	$[\bot, t, f]$	$[\bot, t, f]$	$[\bot, f]$	f
$[\_, \uparrow_2, \uparrow_3]$	$[\bot, \top, f]$			
$[\uparrow_1, \_, \_]$	$[t, t, \bot]$	$[t, t, \bot]$	$[t, \bot]$	t
$[\uparrow_1, \uparrow_2, \_]$	$[t, \top, \bot]$			
$[\uparrow_1, \uparrow_2, \uparrow_3]$	$[t, \top, f]$	$[t, \top, f]$	$[t, f]$	$\top$
$[\uparrow_1, \_, \uparrow_3]$	$[t, t, f]$			

Let us prove, for example, that the isomorphism $\theta : \mathcal{B} \to Im\theta$ preserves negation and conflation: $\sim_B \theta(x) = \theta \sim \vartheta\theta(x) = \theta \sim id_B(x) = \theta(\sim x)$, and
$-_B\theta(x) = \theta - \vartheta\theta(x) = \theta - id_B(x) = \theta(-x)$.

4 Many-valued intuitionistic implication

One of the key insights behind bilattices [14,15] was the interplay between the truth values assigned to sentences and the (non classic) notion of *implication*. The problem was to study how truth values should be propagated "across" implications. The constructivism is surprisingly close to logic programming [24]. The features of logic programming that are unconventional from the classical point of view find immediate constructivistic explanations. Constructivism does not allow indefiniteness in proofs: from a constructivistic viewpoint implications are not "hidden disjunctions". Constructivism is causalistic: implications are viewed as infering new information from already proved information, like in logic programming.

In this paragraph we will try to introduce the formal definition of *many-valued* implication for logic programs. Notice that logic *implication* and logic *entailment*, as pointed by Belnap, for the 4-valued logic are strongly connected: the implication has to be the principal structure for *inference* (entailment) capabilities. Let denote by $\models_B$ this bilattice 4-valued entailment, so that this paradigm can be defined as follows: " $p \models_B q$ iff $p \rightarrow q$ is true".

From this point of view, we are fundamentally interested only for the cases when an implication is *true*. That is probably the reason of why in many-valued logic programming (e.x., Fitting, Przymusinski 3-valued logic) the implication is, differently from other logic connectives, defined as two-valued connective, which preserves the truth: $p \rightarrow q$ holds just in case when for each assignment of one of the four values to the variables, the truth value of p does not exceed the value of q (in symbols: $v(p) \leq_t v(q)$ for each truth assignment $v : \mathcal{L} \rightarrow \mathcal{B}$).

In order to obtain such many-valued definition, which generalizes the 2-valued definition given above, we will consider the conservative extensions of Lukasiewicz's and Kleene's strong 3-valued matrices (where third logic value $\perp$ is considered as unknown) in the *intuitionistic* way. Such conservative extensions are based on the observation we explained in precedence: the problem to study how the truth values should be propagated "across" implications can be restricted only to *true* implications (in fact, the 'immediate consequence operator' derives new facts only for *true* clauses, i.e. when implication is true). Thus, what we must guarantee is that, if $b \leftarrow a$ and $c \leftarrow b$ are true than $c \leftarrow a$ also must be true, in order to guarantee the *reflexivity* and the *transitivity* of the logic entailment $\models_B$.

In other words, that low is intimately connected with inference fixpoint semantics for logic programs: let consider than in the i-th step the ground clause $b \leftarrow a$ become true , so that we derive the new fact b from body a, and that in some $i + k$-th step the ground clause $c \leftarrow b$ become true, so that we derive the new fact c: it means that $c \leftarrow a$ has to be true.

With such constructivistic considerations, Heyting produced an axiomatic system of propositional logic which claimed to generate as theorems precisely those sentences that are valid according to the intuitionistic conception of truth. It is well known that the implication for *intuitionistic logic* satisfy the upper conditions. It is defined by the relative pseudo-complements [25] as follows:

the logical value of intuitionistic implication $a \rightarrow b$ is the greatest member c of $\mathcal{B}$ (w.r.t. the truth ordering) such that $a \wedge c \leq_t b$ (that is, $a \rightarrow b \equiv \bigvee\{c \mid a \wedge c \leq_t b\}$).

Thus we obtain that $a \to b$ iff $a \leq_t b$ (for $c = t$), and that holds the modus ponens inference rule: if $a \to b$, that is, $a \leq_t b$, and a is true, then $t \leq_t b$, that is, b must be true.
The relative pseudo-complement for finite distributive lattices always exist. The 4-vlued implication is given by the following truth-matrix [26], $f_{\leftarrow} : \mathcal{B} \times \mathcal{B} \to \mathcal{B}$

$\to$	t	$\bot$	f	$\top$
t	t	$\bot$	f	$\top$
$\bot$	t	t	$\top$	$\top$
f	t	t	t	t
$\top$	t	$\bot$	$\bot$	t

Remark: The negation in the intuitionistic logic is defined by the pseudo-complement, that is, $\neg\alpha$ is equivalent to $\alpha \to f$, i.e., $\neg\alpha$ is the l.u.b. of $\{\beta \mid \alpha \wedge \beta = f\}$, so that "$\neg$" is different from the epistemic negation " $\sim$ " (it is also different from the knowledge bilattice negation " $-$ ") and, consequently, we do not need it for the logic programming. For our purpose we obtained the Lukasiewicz's extension and a tautology $a \leftarrow a$ for any formula a. Based on this intuitionistic semantics for this implication we guarantee the truth of a clause (implication) $p(\mathbf{c}) \leftarrow B$, whenever (iff) $v_B(p(\mathbf{c})) \geq_t v_B(B)$, as used in the fixpoint semantics for the 'immediate consequence operators'.

5 Autoepistemic many-valued logic

By a standard 3-valued logic program we mean a finite set of universally quantified clauses of the form $\forall(A \leftarrow L_1 \wedge ... \wedge L_m)$, and a set of constraints $\forall(\leftarrow L_1 \wedge ... \wedge L_m)$, where $m \geq 0$, A is an atom and L_i are positive or negative literals (see [27]). Following a standard convention, such clauses will be simply written as clauses of the form $A \leftarrow L_1 \wedge ... \wedge L_m$.
The Autoepistemic 4-valued Logic Programs (ALP) are defined by introducing of modal formulae for facts (ground clauses with $m = 0$) and by modifying the head of constraints from f (false) to $\top$ (possible) logic value:

Definition 6. *The syntax for the autoepistemic many-valued logic programs is defined as follows (logical values $t, \top \in \mathcal{B}$ are considered as ground atoms):*
1. A finite set of universally quantified many-valued clauses of the form
$\forall(A \leftarrow L_1 \wedge ... \wedge L_m)$ *, where $m \geq 1$, A is an atom and L_i are positive or negative literals (see [27]).*
2. A finite set of autoepistemic modal *formulae for ground facts* $M(A \leftarrow t)$ *, where A is a ground (variable free) atom and M is Moore's modal operator.*
3. A finite set of universally quantified many-valued autoepistemic *constraint-clauses*
$\forall(\top \leftarrow L_1 \wedge ... \wedge L_m)$ *, where $m \geq 1$, and L_i are positive or negative literals (with bulid-in predicates also).*
Following a standard convention, such clauses will be simply written as clauses of the form $A \leftarrow L_1 \wedge ... \wedge L_m$.

The *alphabet* of an autoepistemic program P consists of all constants, predicates and functional symbols that appear in P, a countably infinite set of variable symbols, connectives ($\wedge, \vee, \sim, \leftarrow, M$ i.e., and, or, not, intuitionistic logic implication, and modal

Moore's operator, respectively), and the usual punctuation symbols. We assume that if P does not contain any constant, then one is added to the alphabet. The *language* $\mathcal{L}$ of P consists of all the well-formed formulae of the so obtained theory. We assume that the Herbrand universe is Γ_U, an ordinary domain of database constants, and, for a given logic program P composed by a set of predicates and function symbols, P_S, F_S respectively, we define a set of all terms, $\mathcal{T}_S$, and its subset of ground terms $\mathcal{T}_0$. Then the atoms are defined as: $\mathcal{A}_S = \{p(c_1, .., c_n) \mid p \in P_S,\ n = arity(p)\ and\ c_i \in \mathcal{T}_S\}$
The Herbrand base, H_P, is the set of all ground (i.e., variable free) atoms.
The *modal formulae* for "external" ground facts are good means for consistent repairs of inconsistent facts coming from different sources: the freedom, that the autoepistemic reasoning system (Knowledge base) has to *believe* that the external facts are not *absolutely true*, as presented to it from an external world (source databases), preserves its internal consistency. So, the "local" *truth* of facts in locally-consistent sources and the internal belief of a (global) autoepistemic reasoning system, which integrates the information from different sources, can coexist together.
Informally, the *autoepistemic integrity constraints* with the head $\top$ are needed to guarantee that a revision of the truth value of external facts, from true into possible value, is consistently accepted by the reasoning system.

Example 1: Let $r(x, y)$ be a predicate for a relation-table of a source database D_1 and $s(x, y)$ be a predicate for relation-table of a source database D_2. We define the data integration of these two source databases into the global schema predicate $p(x, y)$ and we define the key-constraint for attributes in x by the clause $f \leftarrow p(x, y), p(x, z), \sim (y = z)$:

	Program P	Autoepistemic program P^A
Knowledege Base	$p(x, y) \leftarrow r(x, y)$ $p(x, y) \leftarrow s(x, y)$ $f \leftarrow p(x, y),\ p(x, z),\ \sim (y = z)$	$p(x, y) \leftarrow r(x, y)$ $p(x, y) \leftarrow s(x, y)$ $\top \leftarrow p(x, y),\ p(x, z),\ \sim (y = z)$
External Facts	$r(a, b) \leftarrow t$ $s(a, c) \leftarrow t$ $s(d, c) \leftarrow t$	$M(r(a, b) \leftarrow t)$ $M(s(a, c) \leftarrow t)$ $M(s(d, c) \leftarrow t)$

It is easy to verify that the standard logic program P is inconsistent, that is, there is no Herbrand model for it, because the constraint $f \leftarrow p(a, b),\ p(a, c),\ \sim (b = c)$ can not be satisfied (the logic value of the body is true). Instead, the autoepistemic program P^A has the following truth-minimal Herbrand models, w.r.t. the consistent belief-revision of "external" ground facts (the ground atoms of a Herbrand base of a program P which are not enumerated have the unknown logic value $\perp$):

1. The following set of true ground atoms $\{r(a, b), p(a, b), s(d, c), p(d, c)\}$, and the set $\{s(a, c), p(a, c)\}$ of ground atoms of the logic value $\top$ (possible), define the minimal model, $m_1 : H_P \rightarrow \mathcal{B}$, obtained by the single belief-revision of the fact $s(a, c)$ (it is believed as possible, and not as true fact, by the reasoning autoepistemic system).
The other models, with the same set of "external" facts which have the value $\top$, are not truth-minimal: for instance, the model obtained by the same revision, with the set of true ground atoms $\{r(a, b), p(a, b), p(a, c), s(d, c), p(d, c)\}$, and the set $\{s(a, c)\}$ of

ground atoms of logic value $\top$ is greater then m_1 (consider the logical value of $p(a,c)$ and the fact that $\top \leq_t t$).
2. The set of true ground atoms $\{s(a,c), p(a,c), s(d,c), p(d,c)\}$, and the set $\{r(a,b), p(a,b)\}$ of ground atoms of logic value $\top$, define the minimal model, $m_2 : H_P \rightarrow \mathcal{B}$, obtained by the single belief-revision of the fact $r(a,b)$ (it is believed as possible, and not as true fact, by the reasoning autoepistemic system).
The models m_1 and m_2 are minimal, that is, there is no model m such that $m <_k m_i$ and $m <_t m_i$, $1 \leq i \leq 2$.
The model m_3, composed by the set of true ground atoms $\{s(d,c), p(d,c)\}$, and the set $\{r(a,b), s(a,c), p(a,b), p(a,c)\}$ of ground atoms of logic value $\top$ is not belief-minimal model: it is obtained by the belief-revision of the facts, $s(a,c)$ and $r(a,b)$, which are the belief-revisions used for a model m_1 and m_2, respectively. That is, $m_1 <_k m_3$ and $m_2 <_k m_3$.
It is easy to verify that $m_3 = \oplus\{m_1, m_2\} = m_1 \oplus m_2$, where, for any ground atom $A \in H_P$, $m_1 \oplus m_2(A) = m_1(A) \oplus m_2(A)$.

Definition 7. *Let P be a logic program, with the monotonic in $\leq_k$ immediate consequence operator $\Phi_P : \mathcal{B}_4^{H_P} \rightarrow \mathcal{B}_4^{H_P}$, (Def.2) and S be the set of constraints.*
Then ***fixp***(P, I, I_F, err) *is the algorithm which, for a consistent $\mathcal{B}$-valuation $I \in \mathcal{B}_4^{H_P}$, that is $I : H_P \rightarrow \mathcal{B}_4$, returns with the least fixpoint $I_F : H_P \rightarrow \mathcal{B}_4$ of Φ_P, such that $I_F = \Phi_P(I_F)$, and with $err = 0$, if in all steps of this calculation all intermediate $\mathcal{B}$-valuations satisfy all constraints in S; otherwise returns with $err = 1$.*

Remark: The immediate consequence operator Φ_P of Definition 2 is well defined also for the our *intuitionistic* implication introduced in Section 3, because any ground rule $A \leftarrow B$ is satisfied (true) if in a given many-valued Herbrand interpretation I holds that $I(A) \geq_t I(B)$, as for the classical material 2-valued implication used in Fitting's fixpoint semantics for logic programs. We substituted the material 2-valued implication by the intuitionistic many-valued implication not only because of the many-valued connective theoretic point of view (That is, coherently, *all* logic connectives has to be many-valued), but because it is *fundamental* for the semantics of Autoepistemic logic modal formulae (item 2 in Definition 6). Notice that also such intuitionistic logic can be used for extended logic programs with implication also in bodies of rules (nested implications for hypothetic reasoning).
From [28] holds that the least fixpoint is generalized well-founded model of P. Now we present the simple algorithm for calculation of epistemic (generalized well-founded) models of autoepistemic logic programs:

Definition 8. ALGORITHM: COMPUTE A LEAST 4-VALUED HERBRAND MODEL
Input: Autoepistemic logic program P^A with Herbrand base H_P.
Output: 4-valued minimal Herbrand Model $I : H_P \rightarrow \mathcal{B}_4$ of P^A.
1. $I(A) = \perp$, for all $A \in H_P$; // initialize Herbrand model.
2. $P = \{Clause \mid$ non modal Clause in $P^A\}$; // P is the ordinary (non modal) part of P^A (Knowledge base).
3. $S = \{A \mid M(A \leftarrow t) \in P^A\}$; // initialize the set of non epistemically elaborated external ground facts.
4. ***While*** *$S \neq \{\}$, select one external fact $A \in S$;*

5. $\quad P = P \bigcup \{A \leftarrow t\}$*; // enlarge a program by new fact.*
6. $\quad$ *execute* ***fixp***(P, I, I_F, err) *;*
7. $\quad$ *if* $err \neq 0$ *then* $P = (P - \{A \leftarrow t\}) \bigcup \{A \leftarrow \top\}$*;*
. $\quad$ *execute* ***fixp***(P, I, I_F, err) *;*
8. $\quad S = S - \{A \leftarrow t\}$*;* $I = I_F$*;*
9. ***Return***
END-ALGORITHM

As we can see, this algorithm make a number of *program transformations* in order to avoid inconsistency caused by external ground facts. The different orderings in which we select facts for S will possibly determinate different minimal Herbrand models for P^A. The following properties for autoepistemic logic programs hold:

Proposition 3 *Each autoepistemic logic program P^A is consistent.*
We denote by m_B the banal *model of an autoepistemic program, that is, the model where all ground facts (not only the subset which is mutually inconsistent) are believed as possible.*
Let $\mathcal{M}$ be the set of all minimal epistemic models of P_A. Then $m = \oplus\mathcal{M}$ is the model of P^A, such that $m \leq_k m_B$, and we denominate it `canonical` *model.*

Proof. Let prove that the banal model m_B is a model of the autoepistemic program P^A: if we take all "external" ground facts with logic value $\top$ then all modal formulae in P^A are satisfied (i.e., are true). In that case we derive only ground facts of clauses in P^A with the logic value $\top$ (the logic value of their body), so that all ground instantiations of autoepistemic clauses can be reduced in the form $\top \leftarrow \top \wedge B$, where B is a conjunction of built-in literals: thus the logic value of B can be false or true, and the value of $\top \wedge B$ can be false or $\top$, so that the autoepistemic clauses are satisfied (i.e., true).
Thus, a program P^A is consistent, and must have at least one minimal model (in the worst case it is equal to m_B).
It is easy to verify that for any two models m_1, m_2, also their sum $m_1 \oplus m_2$ is a model of P^A, thus, for a given set of minimal models $\mathcal{M}$, their sum $m = \oplus\mathcal{M}$ is a model.

In the example above, the banal model has the empty set of true ground facts and the set $\{r(a,b), s(a,c), s(d,c), p(a,b), p(a,c), p(d,c)\}$ of ground atoms with the logic value $\top$, but it is not minimal; the canonical model in the example is the model m_3.
The model theoretic semantics of autoepistemic logic programs can be given by means of an ontological encapsulation of the many-valued logic into the 2-valued 'meta' logic (see [29,30], for the *model theoretic* Herbrand semantics for the ontologically encapsulated many-valued (EMV) logic programs).
We can consider the *canonical model* of an autoepistemic logic program P^A as *the semantics* for such program: it corresponds to the view that all mutually-inconsistent information which comes from external sources, if we are not able to distinguish which sources have the better quality of information, must have the same belief-value $\top$. That is, if we have two mutually-inconsistent facts for knowledge reasoner, we do not assume that one of them is true and other is (only) possible: they are both a possible information. Such consideration is analog to the query-answering approach used in the 2-valued data-integration repairing, where the answer to a query for some inconsistent

data-integration system is considered only if such answer is true in all minimal 2-valued repairs of a database.
Only in some practical realizations of data integration, given in the example 1, if, for instance, the source with relation r(x,y) has better information quality w.r.t. the other source, then we may adopt the preferential minimal model m_1 as good semantics for such data integration system.

6 Conclusion

We have presented an autoepistemic programming logic capable of handling inconsistent beliefs, based on the reinterpreted 4-valued Belnap's bilattice, with the intuitionistic semantics for logic implication and the modal Moore's operator for ground facts.
The reinterpreted top knowledge logic value, from *inconsistent* (both true and false) into *possible* value, is used for all mutually inconsistent ground atoms.
We are able to define the canonical 4-valued Herbrand model for such autopeistemic program, which is the complete belief-revision of the inconsistent information which comes from external sources, and, for any user query, to obtain known (certain) answers, which are true in all minimal 4-valued models of a many-valued autoepistemic logic program, and also the possible answers.
The future work will be dedicated to explore the fixpoint algorithms for minimal 4-valued Herbrand models of the autoepistemic logic programs, in particulary for data integration systems with incomplete and inconsistent data source information, w.r.t. the key and foreign-key integrity constraints over a virtual (global) schema. We argue that in this framework, the query answering will be less complex than currently applied methods. The example of a program with the key-constraints, given in this work, is taken by these practical considerations.

References

1. M.Gelfond and V.Lifshitz, "The stable model semantics for logic programming," *In Proc. of the Fifth Logic Programming Symposium, Cambridge, MA. MIT Press*, pp. 1070–1080, 1988.
2. K.Fine, "The justification of negation as failure," *In Logic, Methodology and Philosophy of Science VIII, Amsterdam, North-Holland*, pp. 263–301, 1989.
3. T.Przymusinski, "Every logic program has a natural stratification and an iterated fixed point model," *In Eighth ACM Symposium on Principles of Databases Systems*, pp. 11–21, 1989.
4. T.Przymusinski, "Well-founded semantics coincides with thre-valued stable-semantics," *Fundamenta Informaticae 13*, pp. 445–463, 1990.
5. M.C.Fitting, "A Kripke/Kleene semantics for logic programs," *Journal of Logic Programming 2*, pp. 295–312, 1985.
6. K.Kunen, "Negation in logic programming," *Journal of Logic Programming 4*, pp. 289–308, 1987.
7. A. Van Gelder, "Negation as failure using tight derivations for general logic programs," *Foundations of Deductive Databases and Logic Programming, Morgan Kaufmann, Los Altos*, pp. 149–176, 1987.

8. K.R.Apt and H. Van Emden, "Contributions to the theory of logic programming," *J.Assoc. Comput. Mach.*, vol. 29, pp. 841–863, 1982.
9. G.Escalada Imaz and F.Manyá, "The satisfiability problem for multiple-valued Horn formulae," *In Proc. International Symposium on Multiple-Valued Logics (ISMVL), Boston, IEEE Press, Los Alamitos*, pp. 250–256, 1994.
10. B.Beckert, R.Hanhle, and F.Manyá, "Transformations between signed and classical clause logic," *In Proc. 29th Int.Symposium on Multiple-Valued Logics, Freiburg,Germany*, pp. 248–255, 1999.
11. H.A.Blair and V.S.Subrahmanian, "Paraconsistent logic programming," *Theoretical Computer Science,68*, pp. 135–154, 1989.
12. M.Kifer and E.L.Lozinskii, "A logic for reasoning with inconsistency," *Journal of Automated reasoning 9(2)*, pp. 179–215, 1992.
13. M.Kifer and V.S.Subrahmanian, "Theory of generalized annotated logic programming and its applications," *Journal of Logic Programming 12(4)*, pp. 335–368, 1992.
14. M.Ginsberg, "Multivalued logics: A uniform approach to reasoning in artificial intelligence," *Computational Intelligence, vol.4*, pp. 265–316, 1988.
15. M.C.Fitting, "Bilattices and the semantics of logic programming," *Journal of Logic Programming,11*, pp. 91–116, 1991.
16. M.H.van Emden, "Quantitative deduction and its fixpoint theory," *Journal of Logic Programming,4,1*, pp. 37–53, 1986.
17. V.S.Laksmanan and N.Shiri, "A parametric approach to deductive databases with uncertainty," *IEEE Transactions on Knowledge and Data Engineering,13(4)*, pp. 554–570, 2001.
18. Z.Majkić, "Fixpoint semantics for query answering in data integration systems," *AGP03 - 8.th Joint Conference on Declarative Programming, Reggio Calabria*, pp. 135–146, 2003.
19. N.D.Belnap, "A useful four-valued logic," *In J-M.Dunn and G.Epstein, editors, Modern Uses of Multiple-Valued Logic. D.Reidel*, 1977.
20. Z.Majkić, "Meta many-valued logic programming for incomplete and locally inconsistent databases," *8th International Database Engineering and Application Symposium (IDEAS),July 7-9, Coimbra, Portugal*, 2004.
21. M.L.Ginsberg, "Bilattices and modal operators," *Tech.Rep.N.94305, Comp.Science Dept. Stanford University, California*, 1990.
22. Robert C. Moore, "Semantical considerations on nonmonotonic logic," *Acta Informatica*, vol. 25, pp. 75–94, 1985.
23. M.C.Fitting, "Kleene's three valued logics and their children," *Fundamenta Informaticae*, vol. 20, pp. 113–131, 1994.
24. F.Bry, "Logic programming as constructivism: a formalization and its application to databases," *In Proc. of the Symposium on Principles of Database Systems, ACM SIGACT-SIGMOD*, 1989.
25. H.Rasiowa and R.Sikorski, "The matematics of metamatematics," *PWN- Polisch Scientific Publishers, Warsaw, 3rd edition*, 1970.
26. Z.Majkić, "Many-valued intuitionistic implication and inference closure in a bilattice based logic," *35th International Symposium on Multiple-Valued Logic (ISMVL 2005), May 18-21, Calgary, Canada*, 2005.
27. J.W.Lloyd, *Foundations of Logic Programming*, Springer-Verlag, New York, 1984.
28. M.C.Fitting, "Well-founded semantics, generalized," *in Logic Programming, Proc. of the 1991 International Symposium, Cambridge*, pp. 71–84, 1991.
29. Z.Majkić, "Coalgebraic semantics for logic programming," *18th Worshop on (Constraint) Logic Programming, W(C)LP 2004, March 04-06, Berlin, Germany*, 2004.
30. Z.Majkić, "Truth and knowlwdge fixpoint semantics for many-valued logic programming," *19th Workshop on (Constraint) Logic Programming (W(C)LP 2005), February 21-25, Ulm, Germany*, 2005.