On the notions of residuated-based coherence and bilattice-based consistence

Carlos V. Damásio[1], Nicolás Madrid[2], and M. Ojeda-Aciego[2]

[1] CENTRIA. Universidade Nova de Lisboa. Portugal
cd@di.fct.unl.pt
[2] Dept. Matemática Aplicada. Universidad de Málaga. Spain
{nmadrid,aciego}@ctima.uma.es

Abstract. Different notions of coherence and consistence have been proposed in the literature on fuzzy systems. In this work we focus on the relationship between some of the approaches developed, on the one hand, based of residuated lattices and, on the other hand, based on the theory of bilattices.

1 Introduction

Although inconsistency is essentially considered as an undesirable feature, it arises naturally when considering databases and, in many cases, seems to be unavoidable. As a result, some efforts have been made in order to develop some mechanism to tolerate inconsistent information.

Paraconsistent logics were introduced several decades ago as an *inconsistency-tolerant approach* which allows for efficiently handling inconsistent information. Among the different approaches in the literature, we emphasize the approaches related to *consistence restoring* [2, 3] (focused on how to write repair programs), *fix-point semantics* [9, 15] and *inconsistent information measuring* [11, 14].

It is noticeable that there is not a consensus on the notion of inconsistency in the fuzzy logic framework: one approach, given in [7] considers that a knowledge base is potentially inconsistent or incoherent if there exists a piece of input data that respects integrity constraints and that leads to logical inconsistency when added to the knowledge base; in [15] the authors consider the problem of revising extended programs, and base their approach on the coherence theory initially advocated by Gardenfors for belief revision.

Our contribution in this work is based on two additional approaches, previously developed separately by the authors, the notion of coherence [13] in residuated logic programming [5] and the notions of consistence on a paraconsistent extension of logic programming [1].

2 Preliminary definitions

In order to make this paper self-contained, the notions of residuated-based coherence and of bilattice-based consistence are recalled here.

2.1 L-interpretations and coherence

Definition 1. *A* residuated lattice with negation *is a tuple* $\mathcal{L} = (L, \leq, *, \leftarrow, n)$ *such that:*

1. $(L, \leq)$ *is a complete bounded lattice, with top and bottom elements 1 and 0.*
2. $(L, *, 1)$ *is a commutative monoid with unit element 1.*
3. $(*, \leftarrow)$ *forms an adjoint pair, i.e.* $z \leq (x \leftarrow y)$ *iff* $y * z \leq x \quad \forall x, y, z \in L$.
4. n *is an antitonic mapping* $n : L \to L$ *satisfying* $n(0) = 1$ *and* $n(1) = 0$.

The operator n in a residuated lattice with negation $\mathcal{L} = (L, \leq, *, \leftarrow, n)$, is called the *negation operator* of $\mathcal{L}$. Let us define now the syntax of our logic. Let Π be a set of propositional symbols, then the set of *well-formed formulas* is defined inductively as follows:

- every propositional symbol is a well-formed formula (wff).
- if p is a propositional symbols, then $\sim p$ is a wff.
- if ϕ and ψ are wffs then $\neg\phi, \phi * \psi$ and $\phi \leftarrow \psi$ are wffs.

Note that we use four propositional connectives; $*$ to represent the conjunction, $\leftarrow$ to represent the implication, $\sim$ to represent the strong negation and $\neg$ to represent the default negation.

Definition 2. *A literal* ℓ *is either a propositional symbol* p *or a propositional symbol negated by the strong negation* $\sim p$. *The set of literals is denoted by* Lit.

Let us describe the semantics for the syntax described below.

Definition 3. *Let* $\mathcal{L} = (L, \leq, *, \leftarrow, n)$ *be a residuated lattice with negation, an* L*-interpretation is a mapping* $I : Lit \to L$.

The domain of each L*-interpretation can be inductively extended to every wff as follows:*

- *for every literal* ℓ *the truth value assigned by* I *is* $I(\ell)$.
- *if* ϕ *and* ψ *are wff then:*
  - $I(\neg\phi) = n(\phi)$
  - $I(\phi * \psi) = I(\phi) * I(\psi)$
  - $I(\phi \leftarrow \psi) = I(\phi) \leftarrow I(\psi)$

It is important to point out the semantical difference between strong and default negation in this logic framework. The semantics is compositional (i.e the truth value of $\neg p$ depends univocally of the truth value of p) with respect to default negation but not necessarily with respect to the strong negation (i.e the truth values of p and $\sim p$ are, a priori, independent). As a result, it might happen that the truth-value of two opposite literals, which are assigned directly by one L-interpretation, represent contradictory information and consider the possibility of rejecting those cases; for instance, in classical logic programming inconsistent interpretations are rejected. In [13], we introduced the notion of *coherence* as a suitable generalization of consistence in the residuated framework.

Definition 4. *Let $\mathcal{L} = (L, \leq, *, \leftarrow, n)$ be a residuated lattice with negation, an L-interpretation is* coherent *if $I(\sim p) \leq \dot{\sim} I(p)$ holds for every propositional symbol $p \in \Pi$.*

It is remarkable that the formula $\neg p \ \leftarrow \sim p$ has the value 1 as truth-value with respect to every coherent L-interpretation; since $I(p \leftarrow q) = 1$ if and only if $I(p) \geq I(q)$ holds in every residuated lattice [12]. The formula above states a relationship between the two types of negation, specifically, it states that *strong negation implies default negation.*

2.2 Bilattices and consistence

Other approaches to deal with default negation and consistency are based on the notion of bilattice, instead of on a residuated lattice with negation as in the previous section. For instance, [1] proposed a framework which extends a previous approach to generalized logic programming to an arbitrary complete bilattice of truth-values, where belief and doubt are explicitly represented, as well as a precise definition of important operators found in logic programming, such as explicit and default negation

Definition 5. *A* bilattice *is a tuple $\mathcal{B} = (B, \leq_t, \leq_k)$ where B is a nonempty set, and $(B, \leq_t)$ and $(B, \leq_k)$ are both bounded lattices.*

Given a bounded lattice, two standard orderings can be defined:

- $\langle a, b\rangle \leq_1 \langle c, d\rangle$ if and only if $a \leq c$ and $d \leq b$
- $\langle a, b\rangle \leq_2 \langle c, d\rangle$ if and only if $a \leq c$ and $b \leq d$

Note that the subscript in the ordering relations occurring in the definition stands for *truth* and for *knowledge* as this will be their underlying meaning. Therefore, $(B, \leq_1, \leq_2)$ and $(B, \leq_2, \leq_1)$ represent two different bilattices.

The two possible constructions above have been used in the literature; on the one hand, $(B, \leq_1, \leq_2)$ was used by Ginsberg in [10], who proved that it was able to represent both the standard notion of inference and that of assumption-based truth maintenance systems. Specifically, given a bounded lattice $(L, \leq)$, the bilattice $\mathcal{G}(L)$ is constructed as $\{L \times L, \leq_1, \leq_2\}$. The underlying idea in this bilattice consists in constructing pairs of the form $\langle a, b\rangle$ where a is interpreted as the degree of truth and b as the degree of falsity.

The second construction can be seen in [10] again, as well as in [8] and the idea here is to represent intervals. Specifically, this bilattice is defined as the tuple $\mathcal{F}(L) = (L \times L, \leq_2, \leq_1)$.The underlying idea in this bilattice is to assign to every propositional symbol a set of possible values of truth by providing the infimum and supremum of a such set.

Bilattices have been widely used as useful tools to deal with incomplete and/or inconsistent information [4, 6]. This is due to the fact that bilattices provide a natural framework in which one can define the notions of *consistence* and *default negation* [1]. In order to introduce these notions in bilattices is necessary to define before the concepts of *negation* and *conflation* operator.

Now that we have the notion of bilattice, we can introduce the bilattice-based semantics for strong and default negated propositional symbols.

Definition 6. *Let $\mathcal{B} = (B, \leq_t, \leq_k)$ be a bilattice, a $\mathcal{B}$-interpretation is a mapping $I \colon \Pi \to B$.*

Note that the domain of $\mathcal{B}$-interpretation is defined on the set of propositional symbols Π while the domain of L-interpretations (where L is a lattice) is the set of literals Lit. As in the case of L-interpretations, we have to generalize the concept of consistence; this lead to the definition of negation, conflation and default negation operators.

Definition 7. *Let $\mathcal{B}$ be a bilattice $(B, \leq_t, \leq_k)$*

1. *A* negation *operator over $\mathcal{B}$ is a mapping $n \colon B \to B$ such that:*
   (a) $a \leq_k b$ implies $n(a) \leq_k n(b)$;
   (b) $a \leq_t b$ implies $n(b) \leq_t n(a)$;
   (c) $n(n(a)) = a$
2. *A* conflation *operator over $\mathcal{B}$ is a mapping $- \colon B \to B$ such that:*
   (a) $a \leq_k b$ implies $-b \leq_k -a$;
   (b) $a \leq_t b$ implies $-a \leq_t -b$;
   (c) $- - a = a$
   If $-$ satisfies just items (2a)-(2b) above, it is called a weak-conflation.

Notice that a negation operator (resp. conflation operator) reverses the true-ordering (resp. knowledge ordering) but preserves the knowledge ordering (resp. true ordering). Once the definitions of negation and weak-conflation have been introduced, we can provide the notion of default negation.

Definition 8. *Let $\mathcal{B} = (B, \leq_t, \leq_k)$ be a bilattice, let n and $-$ be a negation and a weak-conflation operator defined on $\mathcal{B}$ respectively. Then, the* default negation *operator is defined as $not(x) = -(n(x))$.*

We are now in condition to recall the notion of bilattice-based consistence:

Definition 9. *Let $\mathcal{B} = (B, \leq_t, \leq_k)$ be a bilattice, let $-$ be a conflation defined on $\mathcal{B}$ and let I be an $\mathcal{B}$-interpretation. Then, I is* consistent *if and only if for every propositional symbol $I(p) \leq_k -I(p)$.*

Given a $\mathcal{B}$-interpretation I, the truth values assigned to propositional symbols negated by the strong and default negation are defined as follows:

1. $I(\sim p) = n(I(p))$
2. $I(\neg p) = not(I(p))$

Note that under this semantics the truth-value assigned to both strong and default negation, is given compositionally.

3 Default negation, consistence and coherence

We start this section by showing how we can link the residuated-based semantics (given in Section 2.1) to the bilattice-based semantics (given in Section 2.2).

Let $\mathcal{L} = (L, \leq, *, \leftarrow, n)$ be a residuated lattice with negation, every L-interpretation can be considered as a $\mathcal{G}(L)$-interpretation, and vice versa, via the following (reversible) transformation:

$$\begin{array}{c} \Omega\colon L\text{-interpretations} \rightarrow \mathcal{G}(L)\text{-interpretations} \\ \left.\begin{array}{r} p \mapsto I(p) \\ \sim p \mapsto I(\sim p) \end{array}\right\} \Rightarrow p \mapsto (I(p), I(\sim p)) \end{array} \tag{1}$$

Once the relationship between L-interpretations and $\mathcal{G}(L)$-interpretations has been fixed, let us relate the underlying mathematical structures in both frameworks. Specifically, the operators $*, \leftarrow$ and n of $\mathcal{L}$ can be extended to $\mathcal{G}(L)$ as follows:

1. $\langle a, b\rangle * \langle c, d\rangle = \langle a * c, b * d\rangle$
2. $\langle a, b\rangle \leftarrow \langle c, d\rangle = \langle a \leftarrow c, b \leftarrow d\rangle$
3. $n(\langle a, b\rangle) = \langle n(a), n(b)\rangle$

Proposition 1. *Let $(L, \leq, *, \leftarrow, n)$ be a residuated lattice, and let $\mathcal{G}(L) = (L \times L, \leq_t, \leq_k)$ be the bilattice associated to $(L, \leq)$. Then, with the extensions described above, the tuples $(L \times L, \leq_t, *, \leftarrow, n)$ and $(L \times L, \leq_k, *, \leftarrow, n)$ are residuated lattices as well.*

Therefore, we can define a residuated semantics on $(L \times L, \leq_t, *, \leftarrow, n)$ and $(L \times L, \leq_k, *, \leftarrow, n)$. Note, however, that n does not define a negation operator on $\mathcal{G}(L)$, since it is antitonic with respect to the knowledge ordering as well. That is not really a problem since we can always define in $\mathcal{G}(L)$ a "natural" negation operator by $\overline{n}(\langle a, b\rangle) = \langle b, a\rangle$.

The advantage of embedding a residuated logic into a bilattice structure with negation and conflation, in this case in $\mathcal{G}(L)$, is that we can compare the semantics for strong and default propositional symbols. But to do that, it is necessary to define also a conflation in $\mathcal{G}(L)$. We recall that, given the negation operator n on the residuated lattice $(L, \leq, *, \leftarrow, n)$ we can define the following weak-conflation in $\mathcal{G}(L)$:

$$-\langle a, b\rangle = \langle n(b), n(a)\rangle$$

Anyway, by using $-$, we can consider on $\mathcal{G}(L)$ the default negation operator $not(x) = -(\overline{n}(x))$. It is important to note that $not(x) = n(x)$ for every element in $L \times L$ since $not(\langle a, b\rangle) = - \circ \overline{n}(\langle a, b\rangle) = -(\langle b, a\rangle) = \langle n(a), n(b)\rangle$. In other words:

Proposition 2. *The default negation is semantically equivalent when is interpreted on the residuated logic $(L, \leq, *, \leftarrow, n)$ and when is interpreted on $\mathcal{G}(L)$ with $\overline{n}$ and $-$ as negation and conflation respectively.*

In order to establish consistence in $\mathcal{G}(L)$, and consider consistent elements in $L \times L$, a (strong) conflation is needed. Thus, to ensure that the operator $-$ defined above is actually a conflation, we have to assume that the negation operator n defined on L is involutive. In that case the notions of coherence and consistence are equivalent, that is:

Proposition 3. *Let $(L, \leq, *, \leftarrow, n)$ be a residuated lattice where n is an involutive operator. Then I a coherent interpretation in L if and only if $\Omega(I)$, as defined by* (1)*, is a consistent interpretation in $\mathcal{G}(L)$ with respect to the conflation operator $-$.*

Note that, at first sight, the definition of consistence in a $\mathcal{G}(L)$-interpretation $\Omega(I)$ implies two different inequalities in I, namely:

1. $I(\sim p) \leq n(I(p))$ (coherence)
2. $I(p) \leq n(I(\sim p))$ (dual-coherence)

but under the hypothesis of Proposition 3, that is when n is involutive, coherence implies dual-coherence. So, the second inequality imposed by the definition of consistence in $\mathcal{G}(L)$ is unnecessary, in that case.

The following question concerning the previous proposition arises now: What is the relationship between coherence and consistence when the negation operator in $\mathcal{L}$ is not involutive?

The answer is not straightforward, as there is not a natural conflation in $\mathcal{G}(L)$ in the sense of the negation $\overline{n}$ which is independent of the negation in L. Obviously, defining a conflation on $\mathcal{G}(L)$ by using a negation operator different from n seems inadequate. We have opted by the use of a bilattice structure which admits a natural conflation: $\mathcal{F}(L)$. The problem here is that we cannot identify one-to-one L-interpretations with $\mathcal{F}(L)$-interpretations, as in the case of $\mathcal{G}(L)$. But, by using the negation operator defined on $\mathcal{L}$, we can identify every element in $\mathcal{F}(L)$ with another in $\mathcal{G}(L)$ by preserving both orderings. On other words, we can define the following operator:

$$\begin{aligned} \Lambda\colon \mathcal{F}(L) &\to \mathcal{G}(L) \\ \langle a, b\rangle &\mapsto \langle a, n(b)\rangle \end{aligned}$$

Note that Λ is not necessarily a one-to-one mapping, since n could be one-to-many mapping. Note also that by using the mapping Ω we can assign to each $\mathcal{F}(L)$-interpretation an L- interpretation.

The advantage of using $\mathcal{F}(L)$ is that we can define a natural conflation operator without using the operator n:

$$-(\langle a, b\rangle) = \langle b, a\rangle$$

So the definition of consistence is "cristal clear" in this structure, a pair $\langle a, b\rangle$ is consistent in the bilattice of interval if and only if $a \leq b$ (that is, if and only if $\langle a, b\rangle$ defines actually a interval). The following proposition shows the relationship between coherence in L and consistence in $\mathcal{F}(L)$:

Proposition 4.

1. *If J is a consistent $\mathcal{F}(L)$-interpretation, then there is a coherent L-interpretation I such that $\Omega(I) = \Lambda(J(p))$ is a coherent.*
2. *Given a coherent L-interpretation I, there is a consistent $\mathcal{F}(L)$-interpretation J such that $\Omega(I(p)) \leq_k \Lambda(J(p))$.*

Proof. The first item is straightforward. For the second item, consider a coherent L-interpretation I, then $I(\sim p) \leq n(I(p))$ and $\Omega(I)(p) = \langle I(p), I(\sim p)\rangle$. Consider the $\mathcal{F}(L)$-interpretation J defined as $J(p) = \langle I(p), I(p)\rangle$ for every propositional symbol $p \in \Pi$; obviously J is consistent. Then, $\Lambda(J)(p) = \langle I(p), n(I(p))\rangle$. Note that $\langle I(p), n(I(p))\rangle \geq_k \langle I(p), I(\sim p)\rangle$ since $I(\sim p) \leq n(I(p))$.

It is important to recall that inconsistency is linked to the knowledge ordering in the following sense: let a and b be two elements in a bilattice such that $a \leq_k b$ and b is consistent, then a is consistent as well. Thus, we have the following corollary:

Corollary 1. *Assume that we have a conflation in $\mathcal{G}(L)$ such that Λ assigns consistent $\mathcal{F}(L)$-interpretations to consistent $\mathcal{G}(L)$-interpretations, then I is a coherent L-interpretation if and only if $\Omega(I)$ is a consistent $\mathcal{G}(L)$-interpretation.*

It is convenient to show that the necessary condition to apply Corollary 1 is weak, since considering the opposite seems unreasonable. Take into account that Λ applies conveniently intervals in $\mathcal{F}(L)$ to elements in $\mathcal{G}(L)$: assume that the real value of p is an element in the interval $[a, b]$. Then we have for sure that the value for "p is true" is at least a and the value of "p is false" is at least $n(b)$. Thus, assuming the existence of an inconsistent element $a \in \mathcal{G}(L)$ coming from a consistent interval $b \in \mathcal{F}(L)$ does not seem reasonable.

Let us finish the section by considering the dual-coherence inequality; namely $I(p) \leq n(I(\sim p))$. Although, after reading Proposition 3, the dual-coherence inequality seem necessary when the negation in the residuated lattice is not involutive, the following example shows that none of the implications of Corollary 1 hold when the dual-coherence inequality hold.

Example 1. Consider, on the lattice $([0, 1], \leq)$, the negation operator defined by $n(x) = 1$ if $x = 0$ and $n(x) = 0$ otherwise. Then the inconsistent $\mathcal{L}$- interpretation J which assigns to p the interval $\langle 1, 0.5\rangle$ is assigned by Λ to the $\mathcal{G}(L)$-interpretation $I(p) = (1, 0)$ which satisfies the inequality $I(p) \leq n(I(\sim p)$.

On the other hand, consider the negation operator defined by $n(x) = 0$ if $x = 1$ and $n(x) = 1$ otherwise. Then the L-interpretation defined by $I(p) = 0.5$ and $I(\sim p) = 1$ does not satisfy the inequality $I(p) \leq n(I(\sim p))$ $(0.5 > n(1) = 0)$ but we can assign via Λ the consistent interval $\langle 0.5, 1\rangle$.

As a consequence of Corollary 1 and Example 1, in order to represent the idea of consistence of $\mathcal{F}(L)$ in a residuated lattice it is only necessary the coherence inequality, as if we consider the dual-coherence inequality then the equivalence might not hold.

4 Conclusions

The relationship between some of the approaches developed, on the one hand, based of residuated lattices and, on the other hand, based on the theory of bilattices. Specifically, the notions of coherence in the residuated-based approach and consistence in the bilattice-based approaches (those based on Ginsberg's $\mathcal{G}(L)$ and on Fitting's $\mathcal{F}(L)$) have been thoroughly studied.

References

1. J. Alcântara, C. V. Damásio, and L. M. Pereira. An encompassing framework for paraconsistent logic programs. *J. Applied Logic*, 3(1):67–95, 2005.
2. M. Balduccini and M. Gelfond. Logic programs with consistency-restoring rules. In *International Symposium on Logical Formalization of Commonsense Reasoning, AAAI 2003 Spring Symposium Series*, pages 9–18, 2003.
3. P. Barceló and L. Bertossi. Logic programs for querying inconsistent databases. *Lecture Notes in Computer Science*, 2562:208–222, 2003.
4. C. Cornelis, O. Arieli, G. Deschrijver, and E. E. Kerre. Uncertainty modeling by bilattice-based squares and triangles. *IEEE T. Fuzzy Systems*, 15(2):161–175, 2007.
5. C. V. Damásio and L. M. Pereira. Monotonic and residuated logic programs. *Lect. Notes in Artificial Intelligence*, 2143:748–759, 2001.
6. G. Deschrijver, O. Arieli, C. Cornelis, and E. E. Kerre. A bilattice-based framework for handling graded truth and imprecision. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(1):13–41, 2007.
7. D. Dubois, H. Prade, and L. Ughetto. Checking the coherence and redundancy of fuzzy knowledge bases. *IEEE T. Fuzzy Systems*, 5(3):398–417, 2002.
8. M. Fitting. Bilattices and the semantics of logic programming. *Journal of Logic Programming*, 11:91–116, 1991.
9. M. Fitting. Fixpoint semantics for logic programming - a survey. *Theoretical Computer Science*, 278:25–51, 1999.
10. M. Ginsberg. Multivalued logics: A uniform approach to inference in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.
11. J. Grant and A. Hunter. Measuring inconsistency in knowledgebases. *Journal of Intelligent Information Systems*, 27(2):159–184, 2006.
12. P. Hájek. *Metamathematics of Fuzzy Logic.* Trends in Logic. Kluwer Academic, 1998.
13. N. Madrid and M. Ojeda-Aciego. Towards a fuzzy answer set semantics for residuated logic programs. In *Proc of WI-IAT'08. Workshop on Fuzzy Logic in the Web*, pages 260–264, 2008.
14. N. Madrid and M. Ojeda-Aciego. Measuring inconsistency in fuzzy answer set semantics. *IEEE Transactions on Fuzzy Systems*, 2011. Accepted.
15. L. Yuan and J. You. Coherence approach to logic program revision. *IEEE Tr. Knowledge and Data Engineering*, 10(1):108–119, 2002.