# On the Notion of Coherence in Fuzzy Answer Set Semantics

**Nicolás Madrid Labrador, Manuel Ojeda-Aciego**
Dept. Matemática Aplicada. Universidad de Málaga
{nmadrid,aciego}@ctima.uma.es

## Abstract

The notion of coherence, introduced in the context of fuzzy answer set programming (FASP) [4], provides a metalogic condition on the obtained models in FASP. In this work, we relate it with the concept of $N$-contradiction which is used in the definition of antonyms.

## 1 Introduction

Answer set semantics is an intuitive and elegant generalisation of the stable model semantics which provides a powerful solution for knowledge representation and non-monotonic reasoning problems. Its applicability has been widely manifested by a lot of researches in different areas.

Originally, they were intended to deal with two negations, one strong and one default negation. The use of these two types of negation is advocated in many contexts of interest, in particular in [8] their use is justified in relation to web rules.

In [4], in order to generalize the answer set semantics to general residuated logic programs, a generalization of consistency was proposed, it was called coherence in order to distinguish it from other generalizations in the fuzzy framework.

In this approach, we provide additional insight and motivation in favour of the notion of coherence instead of other generalizations of consistency. We show that non-coherent interpretations contradict the negation meta-rule by an excess of information and we relate the use of the strong negated propositional symbols with computing with $N$-contradictory concepts.

In this paper, we start by briefly recalling the preliminary definitions needed in order to introduce coherent interpretations in the framework of general residuated logic programs. Then we further motivate the coherent condition as generalization of consistency and the relationship with the concept of $N$-contradictory $L$ fuzzy sets studied in [6].

## 2 Preliminary definitions

As stated in the introduction, the notion of coherence was introduced in the framework of general residuated logic programming, a generalization of logic programming in which the underlying mathematical basis is that of residuated lattices.

The syntax of general residuated logic programs assumes a set $\Pi$ of propositional symbols. If $p \in \Pi$, then both $p$ and $\sim p$ are called *literals*. We will denote arbitrary literals with the symbol $\ell$ (possible subscripted), and the set of all literals as $Lit$.

**Definition 1** *Given a residuated lattice with two negations* $(L, *, \leftarrow, \sim, \neg)$*, a* general residuated logic program $\mathbb{P}$ *is a set of weighted rules of the form*

$$\langle \ell \leftarrow \ell_1 * \cdots * \ell_m * \neg \ell_{m+1} * \cdots * \neg \ell_n; \quad \vartheta \rangle$$

*where* $\vartheta$ *is an element of* $L$ *and* $\ell, \ell_1, \dots, \ell_n$ *are literals.*

**Definition 2** *A fuzzy* $L$-interpretation *is a mapping* $I\colon Lit \to L$*; that is, an* $L$*-fuzzy subset of literals.*

*We say that* $I$ satisfies *a rule* $\langle \ell \leftarrow \mathcal{B}; \quad \vartheta \rangle$ *if and only if* $I(\mathcal{B}) * \vartheta \leq \mathcal{I}(\ell)$ *or, equivalently,* $\vartheta \leq I(\ell \leftarrow \mathcal{B})$.

*Finally,* $I$ *is a* model *of* $\mathbb{P}$ *if it satisfies all rules (and facts) in* $\mathbb{P}$.

The fuzzy answer set semantics described in [4, 5] is defined in two steps. Firstly, a treatment of strong negation in the context of residuated logic programming is provided in terms of the notion of coherence as a generalization in the fuzzy framework of the concept of consistence. Then, fuzzy answer sets for general residuated logic programs are defined as a suitable generalization of the *Gelfond-Lifschitz* reduct.

As our interest in this work is on the notion of coherence, our natural environment is that of extended residuated logic programs, that is, those which do not contain default negation.

Note that, as our interpretations are defined on the set of literals, every extended program has a least model which can be obtained, for instance, by iterating the immediate consequence operator, see [1]. However, one has to take into account the interaction between opposite literals. For example, in the classical case we reject the inconsistent models, i.e $p$ and $\sim p$ cannot be true at the same time.

This notion of inconsistency is generalized to the fuzzy framework, where one can allow that two opposite literals live together ... under some requirements.

**Definition 3** *A fuzzy $L$-interpretation $I$ over $Lit$ is* coherent *if the inequality $I(\sim p) \leq \sim I(p)$ holds for every propositional symbol $p$.*

Now, the key definition in extended residuated logic programs is the following:

**Definition 4** *Let $\mathbb{P}$ be an extended residuated logic program, we say that $\mathbb{P}$ is* coherent *if its least model is coherent.*

Now, a natural question arises: Why coherent interpretations provide an interesting generalization? We have three main reasons. Firstly, it is easy to implement, because it only depends on the negation operator; secondly, it allows to formalise lack of knowledge. For example $I$ such that $I(\ell) = 0$ for all $\ell \in Lit$ is always coherent. And finally, coherence coincides with consistency in the classical framework.

In the next section we present additional motivations on the notion of coherence and we compare it with another approach in the fuzzy framework: the $N$-contradictory fuzzy sets.

# 3 On coherence

In the previous section, the notions of coherent interpretation and coherent program were introduced. Here, we focus on widely explaining the concept of coherence. Specifically, we start by recalling some properties of coherent interpretations, then we present additional motivations to consider the coherence as an adequate generalization of consistency and, finally, we relate it with other approaches.

## 3.1 Recalling some properties of coherence

We recall that a coherent interpretation is an $L$-interpretation such that $I(\sim p) \leq \sim(I(p))$ for all literal $\ell \in Lit$ and that an extended residuated logic program $\mathbb{P}$ is *coherent* if its least model is coherent. Although the definition of coherent program might look a hard restriction, the following property of coherent interpretations shows that a program is coherent if and only if it has, at least, one coherent model.

**Proposition 1** *Let $I$ and $J$ be two interpretations satisfying $I \leq J$. If $J$ is coherent, then $I$ is coherent.*

**Proof** Let $p$ be a propositional symbol. By using $I \leq J$, coherence of $J$ and the decreasing property of $\dot\sim$ we have $I(\sim p) \leq J(\sim p) \leq \dot\sim J(p) \leq \dot\sim I(p)$ $\square$

**Corollary 1** *An extended residuated logic program is coherent if and only if it has at least one coherent model.*

Proposition 1 and its contrapositive, which states that one cannot recover coherence by adding knowledge, play an important role in the usefulness of coherent interpretations.

In order to continue with some properties of the notion of coherence, take into account that an interpretation $I$ assigns a truth degree to any negative literal $\sim p$ independently from the negation operator. This way, if we have two different negation operators ($\sim_1$ and $\sim_2$) we can talk about the coherence of $I$ wrt any of these operators.

**Proposition 2** *Let $\sim_1$ and $\sim_2$ be two negation operators such that $\sim_1 \leq \sim_2$, then any interpretation $I$ that is coherent wrt $\sim_1$ is coherent wrt $\sim_2$.*

**Proof** Let $p$ be a propositional symbol, then $I(\sim p) \leq \dot\sim_1 I(p) \leq \dot\sim_2 I(p)$. $\square$

The following example shows us the importance in the negation operator selected as strong negation to determine the coherence of a residuated logic program.

**Example** Consider the lattice $[0,1]$ with its usual order, Gödel connectives, and the following program $\mathbb{P}$:

$$
\begin{aligned}
r_1 &: \langle p \leftarrow; \quad 1\rangle \\
r_2 &: \langle q \leftarrow p; \quad 0.8\rangle \\
r_3 &: \langle \sim q \leftarrow; \quad 0.7\rangle
\end{aligned}
$$

The least model is $M = \{(p,1); (q,0.8); (\sim q, 0.7)\}$. If we consider the usual negation $n(x) = 1 - x$ to determine the coherence of the program we obtain that $\mathbb{P}$ is not coherent, and the least model semantics fails in this case. However, if we consider the negation:

$$
\overline{n}(x) = \begin{cases} 1 & \text{if} \quad x \leq 0.8 \\ 0 & \text{if} \quad x \geq 0.8 \end{cases}
$$

the program is coherent and the least model semantics provides a meaning to the program. □

We define an ordering among extended residuated logic programs as follows: Let $\mathbb{P}_1$ and $\mathbb{P}_2$ be two extended programs, then $\mathbb{P}_1 \subseteq \mathbb{P}_2$ if and only if for each rule $\langle r_i; \vartheta_1 \rangle$ in $\mathbb{P}_1$ there exists another rule[1] $\langle r_i; \vartheta_2 \rangle$ in $\mathbb{P}_2$ such that $\vartheta_1 \leq \vartheta_2$.

**Proposition 3** *Let $\mathbb{P}_1 \subseteq \mathbb{P}_2$ be two extended programs then the least model of $\mathbb{P}_1$ is smaller than the least model of $\mathbb{P}_2$.*

**Proof** Since $\mathbb{P}_1 \subseteq \mathbb{P}_2$, we can affirm that for all $L$-interpretation $I$

$$T_{\mathbb{P}_1}(I) \leq T_{\mathbb{P}_2}(I)$$

This implies that $lfp(T_{\mathbb{P}_1}) \leq lfp(T_{\mathbb{P}_2})$ □

Therefore we can say that the greater a program is the more information it provides. Now it is easy to prove that the coherence is also decreasingly conserved under the order of extended residuated logic programs.

**Corollary 2** *Let $\mathbb{P}_1 \subseteq \mathbb{P}_2$ be two extended programs. If $\mathbb{P}_2$ is coherent then $\mathbb{P}_1$ is coherent as well.*

### 3.2 Motivation for the condition of coherence

There are many ideas underlying the concept of inconsistence: conflicting inference, inferring contradiction formulas, lack of models, etc. Coherence focuses on the idea of excess of information, which leads to a conflict with the negation meta-rule. Let $n$ be a negation operator, the negation meta-rule is defined as follows: *"if $p$ has truth value $\vartheta$ then $n(p)$ has assigned the truth value $n(\vartheta)$"*. Contradicting the negation meta-rule by excess of information means that the program rules infer more information for a propositional symbol $p$ than it could be inferred using the negation meta-rule. Let us see, using the properties described in the last section, that non-coherent interpretations are affected by this excess of information.

Notice first that a coherent $L$-interpretation never contradicts the negation meta-rule (wrt $\sim$) by excess of information. Given a coherent interpretation $I$, if $I$ contradicts the negation metarule (wrt $\sim$) it is easy to prove that every interpretation less than or equal to $I$ contradicts this meta-rule as well. Contrariwise, if we consider the following $L$-interpretation:

$$\overline{I}(\ell) = \begin{cases} I(p) & if \quad \ell \text{ is the propositional symbol } p \\ \sim I(p) & if \quad \ell \text{ is the literal symbol } \sim p \end{cases}$$

we obtain a non-contradictory interpretation including more information in $I$. Therefore a coherent interpretation either satisfies the negation meta-rule or contradicts it by lack of information. According to the explanation described above, considering coherent interpretation seems a convenient option to avoid a contradiction with the negation metarule by excess of information.

When $I$ is a non-coherent $L$-interpretation, then there exists a propositional symbol $p$ such that either $I(\sim p) > \sim I(p)$ or $I(\sim p)$ and $\sim I(p)$ are incomparable elements in $L$. In both cases a non-coherent interpretation implies a contradiction with the negation meta-rule. Is this contradiction given by an excess of information? Certainly. The contrapositive of Proposition 1 tells us that by adding information to $I$ we will never obtain a coherent interpretation, thus the lack of information is not the reason of this contradiction. Therefore the only possibility to obtain a non contradictory[2] interpretation is by removing information. Thus if we obtain an incoherent interpretation as least fixpoint of a logic program, it has been due to an excess of information in the program (possibly erroneous information). As a result, rejecting non-coherent interpretations seems convenient as well.

An important remark is that coherence can be interpreted with an empirical sense and that the strong negation operator can (and should) be fixed at the beginning. More precisely, when the propositional symbol are fixed, each one has an empirical motivation, i.e a definition given in a natural language. At this point, the programmer has to determine what the strong negated propositional symbols mean. A possibility is described in the next section and is related to computing with antonyms. This way, $\sim p$ could symbolize an antonym of $p$ and the negation operator used represents the degree of contradiction between $p$ and $\sim p$ (see section 3.3). Therefore the choice of the strong negation operator is determined by empirical motivation and not for technical stuff.

Of course this is not the only generalization of consistency in the fuzzy framework. In several approaches is usual to find the following, and popular, generalization.

**Definition 5** *Let $*$ be a t-norm and $\sim$ a negation operator. We say that an interpretation $I\colon Lit \to L$ on the set of literals is $\alpha$-*consistent *if for all propositional symbol $p$ we have that $I(p) * I(\sim p) \leq \alpha$.*

Note that, by the adjoint condition, $I(p) * I(\sim p) \leq \alpha$ iff $I(\sim p) \leq \alpha \leftarrow I(p)$. In other words, $\alpha$-consistence provides an upper bound to the value of $I(\sim p)$ in

[1] Note that the only difference between both rules is the assigned weight.

[2] with respect to the negation meta-rule

terms of $I(p)$ and the parameter $\alpha$. On its turn, recall that a coherent interpretation directly provides such an upper bound, namely $\sim I(p)$, which depends only on the operator intended to interpret the strong negation. Obviously, in a classical context, both terms are equivalent. There is not a universal motivation to prefer one instead of another, the choice depends on the context. In the next section we relate the coherence with computing with antonyms, so if we want to use antonyms in our residuated logic programs it seems convenient to use the concept of coherence as generalization of inconsistency.

### 3.3 Coherence and $N$-contradiction

The concept of $N$-contradiction is defined over fuzzy sets in [6] and is closely related to computing with antonyms [7]; in fact, one condition for $(A, B)$ to be an antonym pair is that $A$ is $N$-contradictory with $B$.

The aim of this section is to connect coherence and strong negation with the concept of $N$-contradiction and, consequently, with computing with antonyms.

Recall that, given a lattice $L$, an $L$-fuzzy set $A$ defined over the universe $X \neq \emptyset$ is a set $A = \{(x, \mu_A(x)) \colon x \in X\}$ such that $\mu_A(x) \in L$ for all $x \in X$. The function $\mu_A$ is called the membership function of $A$ and usually an $L$-fuzzy set is denoted directly by its membership function. A fuzzy answer set is an $L$-fuzzy answer set where $L$ is the unit real interval $[0, 1]$.

The following definition describes when an $L$-fuzzy set is $N$-contradictory with respect to another $L$-fuzzy set.

**Definition 6** *Let $N$ be a negation operator, an $L$-fuzzy set $A$ is $N$-contradictory with respect to the $L$-fuzzy set $B$ if and only if $A(x) \leq N(B(x))$ for all element $x$ in the universe*

The following lemma shows that the negation operator used to establish the $N$-contradiction also represents a grade of contradiction:

**Lemma 1** *Let $N_1$ and $N_2$ be two negations operator such that $N_1(x) \leq N_2(x)$ for all $x \in L$. Let $A$ and $B$ be two $L$-fuzzy sets. If $A$ is $N_1$-contradictory wrt $B$ then $A$ is $N_2$-contradictory wrt $B$*

**Proof** As $A$ is $N_1$-contradictory wrt $B$ then $A(x) \leq N_1(B(x))$ for all $x \in L$. Using the hypotesis $N_1 \leq N_2$, we obtain the inequality: $A(x) \leq N_1(B(x)) \leq N_2(B(x))$ for all $x \in L$.

□

As we said above, the negation operator in Definition 6 indicates a grade of contradiction between both $L$-fuzzy sets. For example, consider the greatest negation operator:

$$N_\top(x) = \begin{cases} 0 & if \quad x = 1 \\ 1 & if \quad x < 1 \end{cases}$$

Then, an $L$-fuzzy set is $N$-contradictory wrt another $L$-fuzzy set ($N$ is a non fixed negation operator) if and only if is $N_\top$ contradictory. That means that $N_\top$ determines the lowest level of contradiction. On the other hand, the least negation operator:

$$N_\perp(x) = \begin{cases} 0 & if \quad x \neq 1 \\ 1 & if \quad x = 1 \end{cases}$$

determines the bigest level of contradiction in the sense that if an $L$-fuzzy set is $N_\perp$-contradictory wrt another $L$-fuzzy set then is $N$-contradictory for all negation operator $N$. This last level of contradiction establishes that an element in the universe cannot has a positive value in both $L$-fuzzy sets.

Let us clarify the concept of $N$-contradiction through an example:

**Example** Consider the following fuzzy set defined over the interval $[0, 100]$:

$$CloseTo20(x) = \begin{cases} 0 & if \quad x \leq 10 \\ \frac{x-10}{5} & if \quad 10 \leq x \leq 15 \\ 1 & if \quad 15 \leq x \leq 25 \\ \frac{30-x}{5} & if \quad 25 \leq x \leq 30 \\ 0 & if \quad x \geq 30 \end{cases}$$

which represents the numbers of $[0, 100]$ which are close to the number 20. It is not difficult to obtain $N$-contradictory sets wrt $closeTo20$. A $N_\perp$-contradictory set wrt $CloseTo20$ is $CloseTo100$:

$$CloseTo100(x) = \begin{cases} 0 & if \quad x \leq 90 \\ \frac{x-90}{5} & if \quad 90 \leq x \leq 95 \\ 1 & if \quad x \geq 95 \end{cases}$$

Moreover, is possible to find $L$-fuzzy sets which are also contradictories with $closeTo20$ but in a lesser level. For example the following fuzzy set is $(1-x)$-contradictory wrt $closeTo20$:

$$CloseTo35(x) = \begin{cases} 0 & if \quad x \leq 25 \\ \frac{x-25}{5} & if \quad 25 \leq x \leq 30 \\ 1 & if \quad 30 \leq x \leq 40 \\ \frac{45-x}{5} & if \quad 40 \leq x \leq 45 \\ 0 & if \quad x \geq 45 \end{cases}$$

□

To finish the section, we show that the strong negation and the coherence conditions are related with the concept of $N$-contradiction. Let $A$ and $B$ be two $L$-fuzzy sets. Assume that $B$ is $N$-contradictory wrt $A$. Suppose also that the general residuated logic program $\mathbb{P}$ contains the following propositional symbols (together them meta-interpretations):

$$p \equiv \textit{The element } \delta \textit{ belongs to } A$$

$$q \equiv \textit{The element } \delta \textit{ belongs to } B$$

If we want to represent the $N$-contradictory relation between $A$ and $B$ for the element $\delta$ in $\mathbb{P}$, and then in the information inferred by it, we identify the strong negation operator with $N$ and we include the following rule:

$$\langle \sim p \leftarrow q \quad ; \quad 1 \rangle$$

Observe that all fuzzy answer set of this program (with the above rule includes in it) holds $I(q) \leq I(\sim p) \leq \sim I(p) = N(I(p))$. Hence the information inferred by the program holds the relation of $N$-contradictory between $A$ and $B$ for the element $\delta$.

The same representation of the notion of $N$-contradiction can be done by identifying $q$ with $\sim p$. Each choice has good and bad features. Identifying $q$ with $\sim p$ reduces the number of literals and thus it reduces the computational complexity. However not doing it supplies the possibility to represent various $N$-contradictory $L$-fuzzy sets with respect to a given one.

## 4 Conclusion and future work

We have recalled the basic definition of answer set semantics for general residuated logic programs. We have focused on motivating the use of the notion of coherence as a generalization of consistency in the fuzzy framework. We have presented several reasons to consider coherent interpretations: they do not contradict the negation metarule by an excess of information and are useful to reasoning with $N$-contradictory concepts.

We have started with the first steps towards reasoning with antonyms in general residuated logic programs. Therefore, future work should go towards the complete introduction of antonyms in this framework.

## References

[1] C. V. Damásio and L. M. Pereira. Monotonic and residuated logic programs. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'01*, pages 748–759. Lect. Notes in Artificial Intelligence, 2143, 2001.

[2] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP-88*, pages 1070–1080, 1988.

[3] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[4] 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.

[5] N. Madrid and M. Ojeda-Aciego. On coherence and consistence in fuzzy answer set semantics for residuated logic programs. *Lect. Notes in Computer Science*, 5571:60–67, 2009.

[6] E. Trillas, C. Alsina, and J. Jacas. On contradiction in fuzzy logic. *Soft Computing*, 3(4):197–199, 1999.

[7] E. Trillas, C. Moraga, S. Guadarrama, S. Cubillo, and E. Castiñeira. Computing with antonyms. In *Forging New Frontiers: Fuzzy Pioneers I*, volume 217 of *Studies in Fuzziness and Soft Computing*, pages 133–153. Springer, 2007.

[8] G. Wagner. Web rules need two kinds of negation. In *Principles and Practice of Semantic Web Reasoning*, volume 2901 of *Lecture Notes in Computer Science*, pages 33–50, 2003.