On coherence and consistence in fuzzy answer set semantics for residuated logic programs*

Nicolás Madrid Manuel Ojeda-Aciego

Dept. Matemática Aplicada
Univ. Málaga, Spain

Abstract. In this work we recall the first steps towards the definition of an answer set semantics for residuated logic programs with negation, and concentrate on the development of relationships between the notions of coherence and consistence of an interpretation.

1 Introduction

An answer set semantics has been recently introduced in the framework of residuated logic programming in response of the need of developing new reasoning paradigms for knowledge representation and non-monotonic deduction problems.

One of the most important areas on which this kind of research can be applied is in the development of the *semantic web*. Mainly due to its vast nature, reasoning with current technologies has been deliberately chosen to be monotonic. Although this choice seems to be the right one when dealing with the semantic web as a whole, the benefits of non-monotonic reasoning become apparent in a local sense: for instance, when a small number of agents communicate they need to consider only their own knowledge bases as all the relevant knowledge, regardless the big amount of information out there. In this case, non-monotonic reasoning is advantageous over monotonic reasoning, i.e. one can retract previous inferences on the discovery of new knowledge by one of the agents, or one can safely assume everything that is not known to the agents as false.

Originally, answer sets semantics was intended to deal with non-monotonic reasoning, in that it provides a method to handle negation in logic programming. Moreover, two kind of negations, one strong negation and one default negation, were allowed in the programs. The use of these two types of negation is advocated in many contexts of interest, in particular in [10] their use is justified in relation to web rules. Moreover, the overall framework of answer set programming has important links with description logics, as stated in [2, 5]. Specifically, [2] have proposed a combination of logic programming under the answer set semantics with some description logics in order to build rules on top of ontologies and, to a limited extent, build ontologies on top of rules; on its turn, [5] introduces

* Partially supported by the Spanish Science Ministry grant TIN06-15455-C03-01 and by Junta de Andalucía grant P06-FQM-02049.

a language that unifies both answer set programming and expressive description logics as an alternative for intuitive non-monotonic reasoning with possibly infinite knowledge.

It is convenient to note that stable models, from which the answer set semantics arose, were initially aimed at formalizing the use of negation in logic programming as negation-as-failure and, thus, are closely related to reasoning under uncertainty. For instance, the closed world assumption for a given predicate P allows for extracting negative knowledge about P from the absence of positive information about it.

The ideal environment for developing a theory of management of uncertainty is fuzzy logic in any of its flavours. This is why we chose to introduce negations in a particular fuzzy logic programming paradigm, specifically the framework of residuated logic programs (which is negation-free) [1], and consider it as our target theory for a suitable generalization of answer set semantics.

In this paper, we start by recalling the basic definitions of stable model, answer set and coherent interpretation in the framework of residuated logic programs introduced in [7]. Then we initiate the analysis of the relationships between the notions of coherence of an interpretation with the more common notion of consistence.

2 On fuzzy answer set semantics for residuated programs

In this section we include the definitions needed to recall the answer set semantics for residuated logic programs with negation. Let us start with the definition of residuated lattice:

Definition 1. *A* residuated lattice *is a tuple* $(L, \leq, *, \leftarrow)$ *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* $y * z \leq x \quad \forall x, y, z \in L$.

In residuated lattices one can interpret the operator $*$ like a conjunction and the operator $\leftarrow$ like an implication.

In the rest of the paper we will consider a residuated lattice enriched with two negation operators, $(L, \leq, *, \leftarrow, \sim, \neg)$. The two negations will modelize the notions of strong negation $\sim$ and default negation $\neg$ often used in logic programming. As usual, a negation operator, over L, is any decreasing mapping $n\colon L \to L$ satisfying $n(0) = 1$ and $n(1) = 0$.

The difference between strong and default negation in our context is essentially semantical, and relates to the method used to infer the truth value of one negated propositional symbol.

In order to introduce our logic programs, we will assume a set Π of propositional symbols. If $p \in \Pi$, then both p and $\sim p$ are called *literals*. We will denote arbitrary literals with the symbol ℓ (possible subscripted), and the set of all literals as Lit.

Definition 2. *Given a residuated lattice with negations* $(L, \leq, *, \leftarrow, \sim, \neg)$*, 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 ϑ *is an element of* L *and* $\ell, \ell_1, \ldots, \ell_n$ *are literals.*

Rules will be frequently denoted as $\langle \ell \leftarrow \mathcal{B}; \vartheta \rangle$. As usual, the formula $\mathcal{B}$ is called the *body* of the rule whereas ℓ is called its *head*. We consider *facts* as rules with empty body, which are interpreted as a rule $\langle \ell \leftarrow 1; \quad \vartheta \rangle$.

Definition 3. *A fuzzy* L-interpretation *is a mapping* $I \colon Lit \to L$*; note that the domain of the interpretation can be lifted to any rule by homomorphic extension.*

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

A general residuated logic program $\mathbb{P}$ is said to be:

- *positive* if it does not contain negation operators.
- *normal* if it does not contain strong negation.
- *extended* if it does not contain default negation.

2.1 Extended logic programs and coherence

In this section, we concentrate on strong negation and, therefore, we will consider extended residuated logic programs.

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. The advantage of working in a fuzzy framework is that one can allow that two opposite literals, such as p and $\sim p$, live together ... under some requirements.

Our approach will be based on a generalization of the concept of consistency which we have called *coherence*, to distinguish it from other existing definitions of consistency in a fuzzy setting.

Definition 4. *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, a natural question arises: why the definition above provides an acceptable generalization? There are three main reasons: firstly, it is easy to implement, since it only depends on the negation operator (whereas other definitions use both a t-norm and a negation); secondly, it allows to handle missing information (i.e. I such that $I(\ell) = 0$ for all $\ell \in Lit$ is always coherent); thirdly, our notion of coherence coincides with consistency in the classical framework (it is not difficult to check this).

We will also apply hereafter the term "coherent" to refer to a logic program, as stated by the following definition.

Definition 5. *Let $\mathbb{P}$ be an extended residuated logic program, we say that $\mathbb{P}$ is* coherent *if its least model is coherent.*

In Section 3 we will introduce some results about coherence, in particular that an extended program is coherent if and only if it has *some* coherent model, and we will compare it with other approaches to the generalization of consistency to a fuzzy framework.

Example 1. Consider the following extended residuated logic program

$$\mathbb{P} = \{\langle p \leftarrow; 1\rangle, \langle \sim p \leftarrow; 0.3\rangle\}$$

over the unit interval and strong negation $\sim x = 1 - x$:

This program is not coherent because its unique minimal model $M = \{(p, 1), (\sim p, 0.3)\}$ is not a coherent interpretation, since $0.3 = M(\sim p) > \sim M(p) = 0$.

2.2 Fuzzy answer sets

Once the concept of coherence has been presented, we can introduce the notion of *fuzzy answer set* for extended logic programs. Such a set is a fuzzy set of literals, similarly to the classical case, the difference is that in our framework it will be considered a particular case of fuzzy L-interpretation.

Definition 6. *Let $\mathbb{P}$ be a coherent extended residuated logic program; the* fuzzy answer set *of $\mathbb{P}$ is its least coherent model of $\mathbb{P}$.*

Our aim in this section is to adapt the approach given in [3, 4] to the *general* residuated logic programs defined above.

Let us consider a general residuated logic program $\mathbb{P}$ together with a fuzzy L-interpretation I. To begin with, we will construct a new normal program $\mathbb{P}_I$ by substituting each rule in $\mathbb{P}$ of the form

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

by the rule[1]

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

Notice that the new program $\mathbb{P}_I$ is extended, that is, does not contain default negation; in fact, the construction closely resembles that of a reduct in the classical case, this is why we introduce the definition below.

Definition 7. *The program $\mathbb{P}_I$ is called the* reduct *of $\mathbb{P}$ wrt the interpretation I.*

It is not difficult to prove that every model M of the program $\mathbb{P}$ is a model of the reduct $\mathbb{P}_M$.

[1] Note the overloaded use of the negation symbol, as a syntactic function in the formulas and as the algebraic negation in the truth-values.

Remark 1. As a result, note that given two fuzzy L-interpretations I and J, then the reducts $\mathbb{P}_I$ and $\mathbb{P}_J$ have the same rules, and might only differ in the values of the weights. By the monotonicity properties of $*$ and $\neg$, we have that if $I \leq J$ then the weight of a rule in $\mathbb{P}_I$ is greater or equal than its weight in $\mathbb{P}_J$.

Now we are ready to introduce our notion of (fuzzy) answer set for general residuated logic program.

Definition 8. *Let $\mathbb{P}$ be a general residuated logic program and let I be a coherent fuzzy L-interpretation; I is said to be an* answer set[2] *of $\mathbb{P}$ iff I is a minimal model of $\mathbb{P}_I$.*

Theorem 1. *Any answer set of $\mathbb{P}$ is a minimal model of $\mathbb{P}$.*

Obviously, this approach is a conservative extension of the classical approach. In the following example we use a simple normal logic program with just one rule in order to clarify the definition of answer set.

Example 2. Consider the program $\langle p \leftarrow \neg q; \quad \vartheta \rangle$. Given a fuzzy L-interpretation $I\colon \mathit{\Pi} \to L$, the reduct $\mathbb{P}_I$ is the rule (actually, the fact) $\langle p; \quad \vartheta * \neg I(q) \rangle$ for which the least model is $M(p) = \vartheta * \neg I(q)$, and $M(q) = 0$. As a result, I is an answer set of $\mathbb{P}$ if and only if $I(p) = \vartheta * \neg I(0) = \vartheta * 1 = \vartheta$ and $I(q) = 0$.

3 On coherence and consistence

Let us start this section by introducing the usual extension of the concept of consistent interpretation to the fuzzy case, which needs both a t-norm and a negation operator.

Definition 9. *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* α-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, α-consistence provides an upper bound to the value of $I(\sim p)$ in terms of $I(p)$ and the parameter α. On its turn, recall that a coherent interpretation (Definition 4) 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 as stated in the proposition below:

Proposition 1. *In classical logic, an interpretation is coherent if and only if it is α-consistent for all $\alpha \in [0, 1)$.*

Example 3. Let us study the set of coherent interpretations associated to two extreme cases of negation. Firstly, for the least negation operator

[2] For normal residuated logic programs, this definition reduces to that of stable set.

$$\sim(x) = \begin{cases} 1 \text{ if } x = 0 \\ 0 \text{ if } x > 0 \end{cases}$$

we have that a coherent interpretation cannot assign a positive value to a propositional symbol and its negation,i.e if $I(p) > 0$ then $I(\sim p) = 0$.

Now, consider the greatest negation operator:

$$\sim(x) = \begin{cases} 1 \text{ if } x < 1 \\ 0 \text{ if } x = 1 \end{cases}$$

For this negation, the coherence condition states that if $I(p) = 1$ then $I(\sim p) = 0$ or, alternatively, that any positive value of $\sim p$ (arbitrarily small), implies that p cannot be certainly true. □

As shown in the previous example, although in a fuzzy context both definitions differ in general, there exist some relations between them. For instance, given $*$ and $\sim$, consider the value

$$\alpha^*_\sim = \sup\{x * \sim x \mid x \in L\}$$

which will be called the *consistence bound*. Let us see some examples on the unit interval.

Example 4. Consider the negation given by $\sim x = 1 - x$

1. For Gödel t-norm, $\min(x, y)$, the consistence bound is 0.5.
2. For product t-norm, $x \cdot y$, the consistence bound is 0.25.
3. For Łukasiewicz t-norm, $\max(0, x + y - 1)$, the consistence bound is 0. □

Proposition 2. *Let $*$ be a t-norm and $\sim$ a negation operator, then any coherent interpretation is $\alpha^*_\sim$-consistent.*

Remark 2. Note that, in the above example, any coherent interpretation is 0-consistent wrt Łukasiewicz t-norm, which requires the strongest type of consistence; however, in a fuzzy context this does not mean that either p or $\sim p$ should be evaluated as 0 since, for instance, if $I(p) = 0.5$ and $I(\sim p) = 0.5$ one still has $I(p) * I(\sim p) = 0$.

One of the main features of the notion of coherence is that it uniquely depends on the negation operator in use, contrariwise to the definition of α-consistence, which involves as well the underlying t-norm and the consistence level. Of course, Proposition 2 above helps the programmer to implement the intended behaviour regarding strong negation regarding the maximum common level that both a propositional symbol and its negation can have maintaining coherence.

Example 5. Assume that we are working with the standard negation operator $\sim x = 1 - x$. In order to find what is the maximum possible common value for $I(p)$ and $I(\sim p)$ in a coherent interpretation I firstly note that, the value of $I(\sim p)$ should reach its upper bound, that is $I(\sim p) = \sim I(p)$. By definition of the negation operator, this amounts to $I(\sim p) = 1 - I(p)$ and, as the values of

$I(p)$ and $I(\sim p)$ are assume to be the same, the previous equation leads that $I(p) = I(\sim p) = 0.5$.

As a result, independently of the underlying t-norm, the use of the standard negation operator and coherent interpretations does not allow that $I(p) = I(\sim p) > 0.5$.

Should we wish a bound different from 0.5, the solution would be to fix a different negation operator. A more restrictive bound is obtained by using $\sim x = 1 - \sqrt{x}$; In effect, by the same reasoning as with the standard negation, we are led to the equation $1 - \sqrt{x} = x$, whose positive solution is $x \approx 0.38$. On the other hand, a less restrictive one is obtained with the operator $\sim x = 1 - x^2$, in this case the equation to be solved is $1 - x^2 = x$, whose positive solution is $x \approx 0.62$. □

The previous example presented some negations which grant more or less restrictive consistence values; however, one would like to be able to construct the corresponding negation to a prescribed consistence value. In the unit interval, this is given by the following

Proposition 3. *Consider $\varepsilon \in [0, 1]$, then the following negation operator*

$$\sim x = \begin{cases} 1 - \frac{1-\varepsilon}{\varepsilon} x & \text{if} \quad x < \varepsilon \\ \frac{\varepsilon}{1-\varepsilon}(1 - x) & \text{if} \quad x \geq \varepsilon \end{cases}$$

satisfies that in all coherent interpretation

$$\sup\{\alpha \in [0, 1] \mid \exists p \text{ such that } I(p) = I(\sim p) = \alpha\} \leq \varepsilon.$$

Note that the operator given in the previous proposition is not the only one satisfying the statement, we have just provided a continuous negation operator with the intended behaviour.

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 4. *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 $\sim_2$.*

Another formulation of the previous property of coherence can be given, this time in terms of two interpretations and just one negation operator, as follows:

Proposition 5. *Let I and J be two interpretations satisfying $I \leq J$. If J is coherent, then I is coherent as well.*

Corollary 1. *If M is a fuzzy coherent model of $\mathbb{P}$, then any other model T such that $T \leq M$ is a coherent model.*

Corollary 2. *A extended residuated logic program is coherent if and only if it has at least one coherent model.*

4 Conclusions and future work

We have recalled the basic definitions of the answer set semantics of general residuated logic programs. We have concentrated on the relationships between the notions of coherence and consistence of an interpretation.

The notion of *consistence bound* has been introduced for a given t-norm $*$ and strong negation $\sim$, and its relationship with coherent interpretations has been presented. Then, coherence has been studied in terms of the ordering relation between interpretations.

A number of issues still have to be studied: for instance, the epistemological implications of the concept of coherence. We have only taken into account that the resulting fuzzy answer sets should be validated for coherence, as a consistency-related notion, and developed some of its initial properties. Future work, should go towards imbricating this notion with threshold computation which turns out to be an important issue for negation-as-failure. For instance, the absence of evidence of p could be interpreted that the value of p is at most a threshold value which cannot be detected by the sensors which provide our information.

Finally, it is important to further relate our approach with other existing approaches [6, 9], and study their possible interactions, as well as studying the modifications needed in order to extend the answer set semantics to multi-adjoint logic programs [8].

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. T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. In *Principles of Knowledge Representation and Reasoning (KR'04)*, pages 141–151, 2004.
3. M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP-88*, pages 1070–1080, 1988.
4. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
5. S. Heymans and D. Vermeir. Integrating semantic web reasoning and answer set programming. In *Answer Set Programming*, pages 195–209, 2003.
6. T. Lukasiewicz. Fuzzy description logic programs under the answer set semantics for the semantic web. *Fundamenta Informaticae*, 82(3):289–310, 2008.
7. N. Madrid and M. Ojeda Aciego. Towards an answer set semantics for residuated logic programs. In *IEEE/WIC/ACM Intl Conf on Web Intelligence and Intelligent Agent Technology, WI-IAT '08*, pages 260-264, 2008.
8. J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Similarity-based unification: a multi-adjoint approach. *Fuzzy Sets and Systems*, 146(1):43–62, 2004.
9. D. Van Nieuwenborgh, M. De Cock, and D. Vermeir. An introduction to fuzzy answer set programming. *Ann. Math. Artif. Intell.*, 50(3-4):363–388, 2007.
10. 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.