RESEARCH ARTICLE

On the existence and unicity of stable models in normal residuated logic programs

Nicolás Madrid[a*] and Manuel Ojeda-Aciego[b*]

Dept. Matemática Aplicada. ETSI Informática, Campus de Teatinos. Málaga, Spain;
[a]*e-mail: nmadrid@ctima.uma.es*   [b]e-mail: aciego@ctima.uma.es

(*Received 00 Month 200x; in final form 00 Month 200x*)

We introduce a sufficient condition which guarantees the existence of stable models for a normal residuated logic program interpreted on the truth-space $[0, 1]^n$. Specifically, the continuity of the connectives involved in the program ensures the existence of stable models. Then, we study conditions which guarantee the uniqueness of stable models in the particular case of the product t-norm, its residuated implication, and the standard negation.

1. Introduction

Similarly to classical logic programming, the existence of fuzzy stable models cannot be guaranteed for an arbitrary normal residuated logic program [13]. Necessary conditions to ensure the existence of stable models have been widely studied in classical logic programming. In fact, a syntactic condition on crisp normal programs to have stable models can be found in [3].

However the characterization in the fuzzy framework is much more complex since it involves two different dimensions: "the syntactic structure of the normal program" and "the choice of suitable connectives in the residuated lattice". For short, we will call them the *syntactic* and the *semantic dimension*, respectively.

In classical logic programming only syntactic conditions are available since the connectives are fixed. However, for normal residuated logic program the semantic dimension plays a crucial role as well; for example the program with only one rule

$$\mathbb{P} = \{\langle p \leftarrow \neg p; 1\rangle\}$$

has a stable model if and only if the operator associated with $\neg$ has a fixpoint. To the best of our knowledge, the problem of establishing semantic conditions for guaranteeing the existence of fuzzy stable models has not been attempted so far, although sufficient conditions underlie in some approaches; for example [16] proves that every normal logic program has stable models in the 3-valued Kleene logic and, more generally, [6, 10, 17–19] show that every normal residuated logic program has stable models if the underlying residuated lattice has an appropriate bilattice structure [8]. Concerning the latter, it is worth to note that not only multivalued and fuzzy logics are often used in computer science and artificial intelligence, there are other non-classical logics such as the temporal, modal, multimodal logics which are receiving lots of attention in the recent years [1, 2, 9, 15].

In this paper we provide another condition on the residuated lattice to ensure the existence of stable models, more specifically: if the underlying lattice is an Euclidean space and the connectives $*$ and $\neg$ in the residuated lattice are continuous, then the existence of at least a fuzzy stable model is guaranteed. Then, the problem of uniqueness is studied and sufficient conditions for uniqueness, in the particular framework of product t-norm and standard negation, have been obtained. These conditions are stated in terms of a set of inequalities between several parameters which can be easily obtained from a program $\mathbb{P}$.

2.   Preliminaries

Let us start this section by recalling the definition of residuated lattice, which fixes the set of truth values and the relationship between the conjunction and the implication (the adjoint condition) occurring in our logic programs.

DEFINITION 2.1 *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* $y * z \leq x$ $\forall x, y, z \in L$.

Remark 1 The adjoint pair is uniquely determined by the chosen t-norm $*$. In other words, fixed a left-continuous t-norm $*$, the only operator $\leftarrow$ which forms an adjoint pair together with $*$ is that defined by:

$$x \leftarrow y = \sup\{z \in L : y * z \leq x\}$$

This is the reason why we usually present residuated lattices by simply mentioning the operator $*$.

In the rest of the paper we will consider a residuated lattice enriched with a negation operator, $(L, *, \leftarrow, \neg)$. The negation $\neg$ will model the notion of default negation 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 $n(1) = 0$.

DEFINITION 2.2 *Given a residuated lattice with negation* $(L, \leq, *, \leftarrow, \neg)$, *a* normal residuated logic program $\mathbb{P}$ *is a finite set of weighted rules of the form*

$$\langle p \leftarrow p_1 * \cdots * p_m * \neg p_{m+1} * \cdots * \neg p_n; \quad \vartheta \rangle$$

where ϑ *is an element of* L *and* $p, p_1, \ldots, p_n$ *are propositional symbols verifying* $p_i \neq p_j$ *for* $i, j = 1, \ldots, n$.

It is usual to denote the rules as $\langle p \leftarrow \mathcal{B}; \vartheta \rangle$. The formula $\mathcal{B}$ is usually called the *body* of the rule, p is called its *head* and ϑ is called its *weight*.

A *fact* is a rule with empty body, i.e facts are rules with the form $\langle p \leftarrow \ ; \vartheta \rangle$. The set of propositional symbols appearing in $\mathbb{P}$ is denoted by $\Pi_{\mathbb{P}}$.

DEFINITION 2.3 *A* fuzzy L-interpretation *is a mapping* $I \colon \Pi_{\mathbb{P}} \to L$; *note that the domain of the interpretation can be lifted to any rule by homomorphic extension.*
    We say that I satisfies *a rule* $\langle p \leftarrow \mathcal{B}; \quad \vartheta \rangle$ *if and only if* $I(\mathcal{B}) * \vartheta \leq I(p)$ *or, equivalently,* $\vartheta \leq I(p \leftarrow \mathcal{B})$. *Finally,* I *is a* model *of* $\mathbb{P}$ *if it satisfies all rules (and facts) in* $\mathbb{P}$.

Note that the ordering relation in the residuated lattice $(L, \leq)$ can be extended over the set of all L-interpretations as follows: *Let* I *and* J *be two* L-*interpretations,*

then $I \leq J$ if and only if $I(p) \leq J(p)$ for all $p \in \Pi_{\mathbb{P}}$.

2.1   *On the Immediate Consequence Operator*

The immediate consequence operator was successfully generalized for positive residuated programs in [5] and can be applied straightforwardly to normal residuated programs. Its definition is as follows:

DEFINITION 2.4 *Let $\mathbb{P}$ be a normal residuated logic program. The immediate consequence operator maps every L-interpretation I to the L-interpretation $T_{\mathbb{P}}(I)$ defined below:*

$$T_{\mathbb{P}}(I)(p) = \sup\{I(\mathcal{B}) * \vartheta \colon \langle p \leftarrow \mathcal{B}; \ \vartheta \rangle \in \mathbb{P}\}$$

where $p \in \mathbb{P}$.

Similarly to the positive case, the operator $T_{\mathbb{P}}$ can be used to characterize the models of normal residuated logic programs:

PROPOSITION 2.5 *Let $\mathbb{P}$ be a residuated logic program and let M be an L-interpretation. M is a model of $\mathbb{P}$ if and only if $T_{\mathbb{P}}(M) \leq M$.*

Proof Let M be a model of $\mathbb{P}$. Then for every rule $\langle p \leftarrow \mathcal{B}; \ \vartheta \rangle \in \mathbb{P}$:

$$M(p) \geq M(\mathcal{B}) * \vartheta$$

This inequality implies that for every propositional symbol p, $M(p)$ is an upper bound of the set $\{M(\mathcal{B}) * \vartheta \colon \langle p \leftarrow \mathcal{B}; \ \vartheta \rangle \in \mathbb{P}\}$ and thus:

$$M(p) \geq \sup\{M(\mathcal{B}) * \vartheta \colon \langle p \leftarrow \mathcal{B}; \ \vartheta \rangle \in \mathbb{P}\} = T_{\mathbb{P}}(M)(p)$$

Assume now that $M(p) \geq T_{\mathbb{P}}(M)(p)$ for every propositional symbol p, then for every rule $\langle p \leftarrow \mathcal{B}; \ \vartheta \rangle$ in $\mathbb{P}$:

$$M(p) \geq T_{\mathbb{P}}(M)(p) = \sup\{M(\mathcal{B}') * \vartheta' \colon \ \langle p \leftarrow \mathcal{B}'; \ \vartheta' \rangle \in \mathbb{P}\} \geq M(\mathcal{B}) * \vartheta$$

■

The immediate consequence operator is monotonic when is defined on positive residuated logic programs [5]:

THEOREM 2.6 *Let $\mathbb{P}$ be a positive residuated logic program, then $T_{\mathbb{P}}$ is monotonic.*

The theorem above together with Knaster-Tarski's fix-point theorem ensure that the operator $T_{\mathbb{P}}$ has a least fix-point; furthermore this least fix-point coincides with the least model of $\mathbb{P}$.

The main difference in the case of normal residuated logic programs, is that $T_{\mathbb{P}}$ is not necessarily monotonic. That feature implies that we cannot make use of the least model semantics in arbitrary normal residuated logic programs.

Example 2.7 Consider the logic program $\langle p \leftarrow \neg q \ \ ; 1 \rangle$ interpreted on the residuated lattice with negation $([0,1], \leq, \min, \leftarrow, 1-x)$. Then the immediate consequence operator is the mapping:

$$T_{\mathbb{P}}(I)(x) = \begin{cases} 1 - I(q) & \text{if} \ \ x = p \\ 0 & \text{otherwise} \end{cases}$$

where I is a $[0, 1]$-interpretation. Clearly this mapping is not monotonic with respect to the order between $[0, 1]$-interpretations.

Certainly, the definition of $T_{\mathbb{P}}$ can be simplified if for each propositional symbol p, there exists only one rule whose head is p, since the operator sup can be removed from the definition. Although that condition on a program $\mathbb{P}$ does not usually hold, we can always obtain a partition of $\mathbb{P}$ such that the condition holds for each partition and, then, the immediate consequence operator of $\mathbb{P}$ can be obtained by using the immediate consequence operator of each partition. Formally:

PROPOSITION 2.8 *Let $\mathbb{P}$ be a normal residuated logic program. Then there exist a partition $\{\mathbb{P}_i\}_{i \in I}$ of the program $\mathbb{P}$ satisfying:*

- *For all $i \in I$, there are no rules in $\mathbb{P}_i$ with the same head.*
- $T_{\mathbb{P}}(I)(p) = \sup_{i \in I}\{T_{\mathbb{P}_i}(I)(p)\}$.

Proof The finest partition of $\mathbb{P}$ satisfies the statement of the proposition. Explicitly, for each rule $r_i \in \mathbb{P}$ we consider the normal residuated logic program with just one rule $\mathbb{P}_i = \{r_i\}$. Then the partition $\{\mathbb{P}_i\}_{i \in I}$ satisfies the first item. Now, for each $\mathbb{P}_i$ the immediate consequence operator has the form

$$T_{\mathbb{P}_i}(I)(x) = \begin{cases} I(\mathcal{B}) * \vartheta & \text{if } x = p \\ 0 & \text{otherwise} \end{cases}$$

where $\langle p \leftarrow \mathcal{B};\ \vartheta \rangle$ is the only rule in $\mathbb{P}_i$. Then:

$$T_{\mathbb{P}}(I)(p) = \sup\{I(\mathcal{B}) * \vartheta\ :\ \langle p \leftarrow \mathcal{B};\ \vartheta \rangle \in \mathbb{P}\} = \sup_{i \in I}\{T_{\mathbb{P}_i}(I)(p)\}$$

∎

This proposition will be crucial in the proof of the main result in Section 4.

2.2 *Stable Models*

We recall here the approach given in [12], which generalizes the stable model semantics [7] to normal residuated logic programs.

Let us consider a normal 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 $\mathbb{P}$ such as

$$\langle p \leftarrow p_1 * \cdots * p_m * \neg p_{m+1} * \cdots * \neg p_n;\quad \vartheta \rangle$$

by the rule[1]

$$\langle p \leftarrow p_1 * \cdots * p_m;\quad \neg I(p_{m+1}) * \cdots * \neg I(p_n) * \vartheta \rangle$$

Notice that the new program $\mathbb{P}_I$ is positive, that is, does not contain any negation; in fact, the construction closely resembles that of a reduct in the classical case, this is why we introduce the following:

DEFINITION 2.9 *The program $\mathbb{P}_I$ is called the* reduct *of $\mathbb{P}$ wrt the interpretation I.*

---

[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.

As a result of the definition, 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 properties of $*$ and $\neg$, we have that if $I \leq J$, then the weight of a rule in $\mathbb{P}_I$ is greater than or equal to its weight in $\mathbb{P}_J$.

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

Recall that a *fuzzy interpretation* can be interpreted as an L-fuzzy subset. Now, as usual, the notion of reduct allows for defining a *stable set* for a program.

DEFINITION 2.10 *Let $\mathbb{P}$ be a normal residuated logic program and let I be a fuzzy L-interpretation; I is said to be a* stable set *of $\mathbb{P}$ iff I is a minimal model of $\mathbb{P}_I$.*

THEOREM 2.11 *Any stable set of $\mathbb{P}$ is a minimal model of $\mathbb{P}$.*

Thanks to Theorem 2.11 we know that every stable set is a model, therefore we will be able to use the term *stable model* to refer to a stable set. 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 stable set (stable model).

Example 2.12 Consider the program $\langle p \leftarrow \neg q\ \ ; \vartheta \rangle$. Given a fuzzy L-interpretation $I \colon \Pi \to L$, the reduct $\mathbb{P}_I$ is the rule (actually, the fact) $\langle p\ \ ; \vartheta * \neg I(q) \rangle$ for which the least model is $M(p) = \vartheta * \neg I(q)$, and $M(q) = 0$. As a result, I is a stable model of $\mathbb{P}$ if and only if $I(p) = \vartheta * \neg I(0) = \vartheta * 1 = \vartheta$ and $I(q) = 0$.

An important feature of the stable models, that holds as well in our extended framework, is that a stable model is always a minimal fix-point of $T_\mathbb{P}$.

PROPOSITION 2.13 *Any stable model of $\mathbb{P}$ is a minimal fix-point of $T_\mathbb{P}$.*

Proof We will refer here a rule $\langle p \leftarrow p_1 * \cdots * p_m * \neg p_{m+1} * \cdots * \neg p_n;\ \ \vartheta \rangle$ by writing $\langle p \leftarrow \mathcal{B}^+ * \mathcal{B}^-;\ \vartheta \rangle$ where $\mathcal{B}^+$ is identified with $p_1 * \cdots * p_m$ and $\mathcal{B}^-$ is identified with $\neg p_{m+1} * \cdots * \neg p_n$. With this notation, the reduct $\mathbb{P}_I$ can be seen as the transformation which substitutes each rule $\langle p \leftarrow \mathcal{B}^+ * \mathcal{B}^-;\ \vartheta \rangle$ in $\mathbb{P}$ by $\langle p \leftarrow \mathcal{B}^+;\ I(\mathcal{B}^-) * \vartheta \rangle$.

Let us see firstly that for every L-interpretation I, $T_\mathbb{P}(I) = T_{\mathbb{P}_I}(I)$:

$$T_\mathbb{P}(I)(p) = \sup\{I(\mathcal{B}^+) * I(\mathcal{B}^-) * \vartheta \colon \langle p \leftarrow \mathcal{B}^+ * \mathcal{B}^-;\ \vartheta \rangle \in \mathbb{P}\} =$$
$$= \sup\{I(\mathcal{B}^+) * I(\mathcal{B}^-) * \vartheta \colon \langle p \leftarrow \mathcal{B}^+;\ I(\mathcal{B}^-) * \vartheta \rangle \in \mathbb{P}_I\} = T_{\mathbb{P}_I}(I)(p)$$

Now, let M be a stable model of $\mathbb{P}$. Then, by definition, $M = T_{\mathbb{P}_M}(M)$. By using the equality above, we obtain $M = T_{\mathbb{P}_M}(M) = T_\mathbb{P}(M)$; in other words, M is a fix-point of $T_\mathbb{P}$.

Let us prove the minimality of M. Let N be a fix-point of $T_\mathbb{P}$ such that $N \leq M$, then N is a model of $\mathbb{P}$ by Proposition 2.5 . Now, applying Theorem 2.11, we obtain $N = M$. ∎

Notice, however, that a minimal fix-point of $T_\mathbb{P}$ is not necessarily a stable model of $\mathbb{P}$, as shown in the following example:

Example 2.14 Let $\mathbb{P} = \{\langle p \leftarrow p; 1 \rangle, \langle q \leftarrow \neg p; 1 \rangle\}$ be a normal residuated logic program defined on $([0, 1], \leq, \min, \leftarrow, 1 - x)$. Let us obtain firstly the stable models of $\mathbb{P}$. Let $I = \{(p, \alpha), (q, \beta)\}$ be a $[0, 1]$-interpretation, then the reduct $\mathbb{P}_I$ is the program $\mathbb{P}_I = \{\langle p \leftarrow p; 1 \rangle, \langle q \leftarrow; 1 - \alpha \rangle\}$. The least model of $\mathbb{P}_I$ is the $[0, 1]$-interpretation $M = \{(p, 0), (q, 1 - \alpha)\}$. So I is a stable model of $\mathbb{P}$ if and only if $I = M$, that is, if and only if $I = \{(p, 0), (q, 1)\}$.

Let us obtain now the set of fix-points of $T_\mathbb{P}$. The immediate consequence operator of $\mathbb{P}$ is:

$$T_\mathbb{P}(I)(x) = \begin{cases} I(p) & \text{if } x = p \\ 1 - I(p) & \text{if } x = q \end{cases}$$

A $[0,1]$-interpretation $I = \{(p, \alpha), (q, \beta)\}$ is a fix-point of $T_\mathbb{P}$ if and only if $I(p) = I(p)$ and $I(q) = 1 - I(p)$. Therefore the set of fix-points of $T_\mathbb{P}$ is given by the interpretations I_α such that $I_\alpha(p) = \alpha$ and $I_\alpha(q) = 1 - \alpha$ for all $\alpha \in [0,1]$. Note that every $[0,1]$-interpretation I_α is a minimal fix-point but only one of them is a stable model.

3. On the existence of stable models in $[0, 1]$

The existence of stable models can be guaranteed by simply imposing conditions on the underlying residuated lattice [14]:

THEOREM 3.1 *Let $\mathcal{L} \equiv ([0,1], \leq, *, \leftarrow, \neg)$ be a residuated lattice with negation. If $*$ and $\neg$ are continuous operators, then every finite normal program $\mathbb{P}$ defined over $\mathcal{L}$ has at least a stable model.*

Proof The idea is to apply Brouwer's fix-point theorem. Specifically, we show that the operator defined by $\mathcal{R}(I) = \mathrm{lfp}(T_{\mathbb{P}_I})$, for a given interpretation I, is continuous. Note that this operator can be seen as a composition of two operators $\mathcal{F}_1(I) = \mathbb{P}_I$ and $\mathcal{F}_2(\mathbb{P}) = \mathrm{lfp}(T_\mathbb{P})$. Actually, we will show that $\mathcal{F}_1$ and $\mathcal{F}_2$ are continuous.

To begin with, note that $\mathcal{F}_1$ can be seen as an operator from the set of $[0,1]$-interpretations to the Euclidean space $[0,1]^k$ where k is the number of rules in $\mathbb{P}$. This is due to the fact that $\mathcal{F}_1$ just changes the weights of $\mathbb{P}$, and nothing else. Now, the continuity of $\mathcal{F}_1$ is trivial since the weight of each rule in $\mathbb{P}$ is changed only by using the continuous operators $\neg$ and $*$.

Concerning $\mathcal{F}_2$, the syntactic part of $\mathbb{P}$ can be considered fixed and positive. This is due to the fact that its only inputs are of the form $\mathbb{P}_I$, therefore, the number of rules is fixed, negation does not occur in $\mathbb{P}$, and the only elements which can change are the weights. As a result, $\mathcal{F}_2$ can be seen as a function from $[0,1]^k$ to the set of interpretations. Note that this restriction over $\mathcal{F}_2$ does not affect the composition between $\mathcal{F}_1$ and $\mathcal{F}_2$. To prove that $\mathcal{F}_2$ is continuous note, firstly, that the immediate consequences operator is continuous with respect to the weights in $\mathbb{P}$, since every operator in the definition of $T_\mathbb{P}$ (namely sup and $*$) is continuous. Secondly, a direct consequence of the termination result introduced in [4, see Cor. 29] ensures that if $\mathbb{P}$ is a finite positive program, then $\mathrm{lfp}(T_\mathbb{P})$ can be obtained by iterating finitely many times the immediate consequence operator on the bottom interpretation $I_\perp$; in other words, $\mathrm{lfp}(T_\mathbb{P}) = T_\mathbb{P}^k(I_\perp)$ where k is the number of rules in $\mathbb{P}$. Therefore, as the operator $\mathcal{F}_2$ is a finite composition of continuous operators, $\mathcal{F}_2$ is also continuous.

Finally, as $\mathcal{R}(I) = \mathrm{lfp}(T_{\mathbb{P}_I})$ is a composition of two continuous operators, $\mathcal{R}(I)$ is continuous as well. Hence we can apply Brouwer's fix-point theorem to $\mathcal{R}(I)$ and ensure that it has at least a fix-point. To conclude, we only have to note that every fix-point of $\mathcal{R}(I)$ is actually a stable model of $\mathbb{P}$.   ∎

Example 3.2 The existence of stable models for the normal residuated logic program given below

$$\langle p \leftarrow \neg q \; ; 0.8\rangle$$

$$\langle q \leftarrow \neg r \; ; 0.7\rangle$$

$$\langle r \leftarrow \neg p \; ; 0.9\rangle$$

is not always guaranteed. For example, if we consider the residuated lattice $L = ([0,1], *, \leftarrow, \neg)$ determined by $x * y = x \cdot y$ and

$$\neg(x) = \begin{cases} 0 & \text{if } x > 0.5 \\ 1 & \text{if } x \le 0.5 \end{cases}$$

then the program has no stable model. However, if we consider the residuated lattice $L = ([0,1], *, \leftarrow, \neg)$ determined by $x * y = x \cdot y$ and $\neg(x) = 1 - x$ the normal residuated logic program has the following stable model[1]

$$M = \{(p, 0.49); (q, 0.38); (r, 0.45)\}$$

Obviously, the sufficient condition provided in Theorem 3.1 is not a necessary condition. Considering the residuated lattice $L = ([0,1], *, \leftarrow, \neg)$ determined by

$$x * y = \begin{cases} x & \text{if } y = 1 \\ y & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \neg(x) = \begin{cases} 0 & \text{if } x > 0.9 \\ 1 & \text{if } x \le 0.9 \end{cases}$$

the program above has one stable model, $M = \{(p, 0.8); (q, 0.7); (r, 0.9)\}$; although the connectives $*$ and $\neg$ are not continuous.

Remark 1 It is important to recall that most connectives in fuzzy logic are defined on the unit interval $[0, 1]$. Thus the condition about continuity on a Euclidean space as sets of truth-values is not excessively restrictive. Moreover, most t-norms used currently in fuzzy logic are continuous (Gödel, Łukasiewicz, product, . . .), therefore the theorem establishes that in the most used fuzzy frameworks, the existence of fuzzy stable models is always guaranteed, at least when considering the standard negation.

4.   On the unicity of stable models in a logic based on the product t-norm

In this section we introduce a condition which guarantees the unicity of stable models for normal residuated logic programs defined with the product adjoint pair and the standard negation. It is important to point out that there are just a few results in crisp logic programming which guarantee the unicity of stable models (some of them later extended in [11] for fuzzy logic programming). The condition presented below combines the weights in the program and the syntax, allowing that cycles with negation appear, provided that its rules have convenient weights.

---

[1]The values are approximated to two digits precision.

In the rest of the paper we use the residuated lattice $\mathcal{L} = ([0, 1], *, \leftarrow, n)$ given by:

$$x * y = x \cdot y \qquad x \leftarrow y = \begin{cases} \frac{y}{x} & \text{if } x \geq y \\ 1 & \text{if } x < y \end{cases} \qquad n(x) = 1 - x$$

Note that, as the operator $*$ and $\neg$ are continuous, the existence of at least a stable model is guaranteed.

An advantage of working in $[0, 1]$ is that, for finite programs, the operator $T_{\mathbb{P}}$ can be seen as a real function from $[0, 1]^n$ to $[0, 1]^n$ where n is the number of propositional symbols in $\mathbb{P}$, that is $n = \#\Pi_{\mathbb{P}}$. This can be done by paying attention to the following considerations, which will be used hereafter:

- The propositional symbols occurring in $\mathbb{P}$ can be sorted as a list of n elements, denoted by $p_1, \ldots, p_n$.
- Each $[0, 1]$-interpretation can be seen as a tuple $(I(p_1), \cdots, I(p_n)) \in [0, 1]^n$. Hence, the i-th component in a tuple $\overline{a} = (a_1, \ldots, a_n)$ represents the value of the propositional symbol p_i.
- As $T_{\mathbb{P}}$ assigns $[0, 1]$-interpretations to $[0, 1]$-interpretations, with the above consideration, $T_{\mathbb{P}}$ assigns tuples in $[0, 1]^n$ to tuples in $[0, 1]^n$; in other words, $T_{\mathbb{P}}$ can be seen as a real function from $[0, 1]^n$ to $[0, 1]^n$.
- Finally, although the value of $T_{\mathbb{P}}(I)$ for the i-th propositional symbol p_i is usually denoted by $T_{\mathbb{P}}(I)(p_i)$ within the logic programming community, we will use here the usual notation used in real analysis. That is, $T_{\mathbb{P}}(I)$ will be written as a tuple $\left((T_{\mathbb{P}})_1(I), \ldots, (T_{\mathbb{P}})_n(I)\right)$ and, thus, we will write $(T_{\mathbb{P}})_i(I)$ instead of $T_{\mathbb{P}}(I)(p_i)$.

If at most one rule whose head is p_i appears in $\mathbb{P}$, then the immediate consequence operator has the following simple form:

$$(T_{\mathbb{P}})_i(I) = I(q_1) \cdot \cdots \cdot I(q_k) \cdot (1 - I(q_{k+1})) \cdot \cdots \cdot (1 - I(q_m)) \cdot \vartheta \qquad (1)$$

where we assume that $\langle p_i \leftarrow q_1 * \cdots * q_k * \neg q_{k+1} * \cdots * \neg q_m, \ \vartheta \rangle$ is the only rule in $\mathbb{P}$ with head p_i.

Note that the occurrence of I as argument of $(T_{\mathbb{P}})_i$ in equation (1) above, actually means the tuple formed by $\left(I(p_1), \ldots, I(p_n)\right)$ for I being considered as a variable interpretation. Note as well that the use of $I(p_i)$ to represent one variable argument easily might lead to misunderstandings. In order to solve this potential problem, we introduce the following

Notational conventions:

(1) Two different notations will be used, in order to denote whether we are referring to a propositional symbol as an element of the domain of $T_{\mathbb{P}}$ (we will use the family p_i in this case, as stated in the previous considerations) or to a symbol occurring in the body of a rule (we will use q_j). Note that a given propositional symbol could be represented in two different forms depending on the use we want to stress.

(2) As $(T_{\mathbb{P}})_i$ will be used as a real valued function in the proofs of this section, we will denote its variables by using directly the propositional symbols p_i, i.e. the formula (1) will be written as:

$$(T_{\mathbb{P}})_i(p_1, \ldots, p_n) = q_1 \cdot \ldots \cdot q_k \cdot (1 - q_{k+1}) \cdot \ldots \cdot (1 - q_m) \cdot \vartheta \qquad (2)$$

where each q_j is actually some p_i.

The following lemma, which states a certain inequality when considering the immediate consequence operator as a differentiable real function, will be used to prove the main result of this section.

LEMMA 4.1 *Let $\mathbb{P}$ be a normal residuated logic program such that at most one rule which head is p appears in $\mathbb{P}$. Let I and J be two $[0,1]$-interpretations such that $J \leq I$, then:*

$$\sum_{j=1}^{n} \left| \frac{\partial(T_{\mathbb{P}})_i}{\partial p_j}(J(p_1),\ldots,J(p_n)) \right| \leq \sum_{j=1}^{h} I(q_1) \cdot \ldots \cdot I(q_{j-1}) \cdot I(q_{j+1}) \cdot \ldots \cdot I(q_h) \cdot \vartheta +$$

$$+ (k-h)(I(q_1) \cdot \ldots \cdot I(q_h) \cdot \vartheta)$$

where $\langle p_i \leftarrow q_1 * \cdots * q_h * \neg q_{h+1} * \cdots * \neg q_k \; ; \vartheta \rangle$ is the rule in $\mathbb{P}$ whose head is p_i.

Proof We will use here the expression of $(T_{\mathbb{P}})_i$ given by Equation (2) above. Clearly $(T_{\mathbb{P}})_i$ is differentiable, and its partial derivatives are:

$$\frac{\partial(T_{\mathbb{P}})_i}{\partial p_j} = \begin{cases} q_1 \cdots q_{t-1} \cdot q_{t+1} \cdots q_h \cdot (1-q_{h+1}) \cdots (1-q_k) \cdot \vartheta & \text{if } p_j = q_t \text{ for } t \leq h \\ -q_1 \cdots q_h \cdot (1-q_{h+1}) \cdots (1-q_{t-1}) \cdot (1-q_{t+1}) \cdots (1-q_k) \cdot \vartheta & \text{if } p_j = q_t \text{ for } t > h \\ 0 & \text{otherwise} \end{cases}$$

Hence, the sum of all the partial derivatives of $(T_{\mathbb{P}})_i$ evaluated on the point $(J(p_1),\ldots,J(p_n))$ satisfies the following inequality:

$$\sum_{j=1}^{n} \left| \frac{\partial(T_{\mathbb{P}})_i}{\partial p_j}(J(p_1),\ldots,J(p_n)) \right| =$$

$$= \sum_{j=1}^{h} |J(q_1) \cdots J(q_{j-1}) \cdot J(q_{j+1}) \cdots J(q_h) \cdot (1-J(q_{h+1})) \cdots (1-J(q_k)) \cdot \vartheta| +$$

$$+ \sum_{j=h+1}^{k} |-J(q_1) \cdots J(q_h) \cdot (1-J(q_{h+1})) \cdots (1-J(q_{j-1})) \cdot (1-J(q_{j+1})) \cdots (1-J(q_k)) \cdot \vartheta|$$

$$\leq \left(\sum_{j=1}^{h} J(q_1) \cdot \ldots \cdot J(q_{j-1}) \cdot J(q_{j+1}) \cdot \ldots \cdot J(q_h) \cdot \vartheta \right) + (k-h)(J(q_1) \cdot \ldots \cdot J(q_h) \cdot \vartheta)$$

$$\leq \left(\sum_{j=1}^{h} I(q_1) \cdot \ldots \cdot I(q_{j-1}) \cdot I(q_{j+1}) \cdot \ldots \cdot I(q_h) \cdot \vartheta \right) + (k-h)(I(q_1) \cdot \ldots \cdot I(q_h) \cdot \vartheta)$$

∎

Now we are able to state and prove the main result of this section.

THEOREM 4.2 *Let $\mathbb{P}$ be a finite normal residuated logic program, and p a propositional symbol occurring in $\mathbb{P}$, let us write $\vartheta_p = \max\{\vartheta_j : \langle p \leftarrow \mathcal{B} \; ; \vartheta_j \rangle \in \mathbb{P}\}$.[1] If, for every rule $\langle p \leftarrow q_1 * \cdots * q_h * \neg q_{h+1} * \cdots * \neg q_k \; ; \vartheta \rangle \in \mathbb{P}$, the inequality below holds*

$$\left(\sum_{j}^{h} \vartheta_{q_1} \cdot \ldots \cdot \vartheta_{q_{j-1}} \cdot \vartheta_{q_{j+1}} \cdot \ldots \cdot \vartheta_{q_h} \cdot \vartheta \right) + (k-h)(\vartheta_{q_1} \cdot \ldots \cdot \vartheta_{q_h} \cdot \vartheta) < 1$$

---

[1] If p does not appear in the head of any rule then $\vartheta_p = 0$.

then there is only one stable model of $\mathbb{P}$.

Proof The structure of the proof is as follows:

- We will show that $T_{\mathbb{P}}$ is a contractive map with respect to the norm $||.||_{\infty}$ in a specific subset $A \subseteq [0,1]^n$, for this part we will use that each element in A can be seen as one $[0,1]$-interpretation. Then, by applying Banach's fix-point theorem, $T_{\mathbb{P}}$ has only one fix-point in A.
- We will show as well that, if $I \in [0,1]^n$ is a fix-point of $T_{\mathbb{P}}$, then I necessarily belongs to A. Therefore $T_{\mathbb{P}}$ has only one fix-point in $[0,1]^n$.
- Finally, taking into account that:
    - (1) Every stable model of $\mathbb{P}$ is actually a fix-point of $T_{\mathbb{P}}$ (Lemma 2.13)
    - (2) By Theorem 3.1, there exists at least one stable model of $\mathbb{P}$

    then the unique fix-point of $T_{\mathbb{P}}$ has to be its unique stable model.

Let I be the $[0,1]$-interpretation which assigns the value ϑ_p to each propositional symbol p. Let us prove that $T_{\mathbb{P}}$ is contractive in the set $A = \{J \in [0,1]^n \colon J \leq I\}$.

To do that, we distinguish two cases, firstly that for every propositional symbol $p \in \Pi_{\mathbb{P}}$, there is at most one rule whose head is p in $\mathbb{P}$; and secondly we prove the general case for an arbitrary normal logic program.

Let us asume that $\mathbb{P}$ is a normal residuated logic program such that at most one rule whose head is p appears in $\mathbb{P}$. Then the hypothesis of Lemma 4.1 holds, and it provides the following inequality for every $J \in A$:

$$\sum_{j=1}^{n} \left| \frac{\partial (T_{\mathbb{P}})_i}{\partial p_j}(J(p_1), \ldots, J(p_n)) \right| \leq \sum_{j=1}^{h} I(q_1) \cdot \ldots \cdot I(q_{j-1}) \cdot I(q_{j+1}) \cdot \ldots \cdot I(q_h) \cdot \vartheta \ +$$

$$+(k-h)(I(q_1) \cdot \ldots \cdot I(q_h) \cdot \vartheta)$$

$$= \left(\sum_{j=1}^{h} \vartheta_{q_1} \cdot \ldots \cdot \vartheta_{q_{j-1}} \cdot \vartheta_{q_{j+1}} \cdot \ldots \cdot \vartheta_{q_h} \cdot \vartheta \right) + (k-h)(\vartheta_{q_1} \cdot \ldots \cdot \vartheta_{q_h} \cdot \vartheta)$$

As a result, by hypothesis, we obtain for every $J \in A$

$$\sum_{j=1}^{n} \left| \frac{\partial (T_{\mathbb{P}})_i}{\partial p_j}(J(p_1), \ldots, J(p_n)) \right| < 1 \tag{3}$$

Now we make use of the *mean value theorem* for vector fields on $\mathbb{R}^n$ w.r.t $||.||_{\infty}$ (see [20]): Given a differentiable function f on a set X, if the line segment $[a,b] = \{(1-t) \cdot a + t \cdot b \colon 0 \leq t \leq 1\}$ is contained in X then:

$$||f(b) - f(a)||_{\infty} \leq ||b - a||_{\infty} \sup\{||Df(z)||_{\infty} \colon z \in [a,b]\}$$

where Df denotes the differential of f, i.e the linear map associated to the *Jacobian* matrix of $f = (f_1, \ldots, f_n)$:

$$\begin{pmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{pmatrix}$$

and the norm $||Df(z)||_\infty$ is defined as:

$$||Df(z)||_\infty = \sup\{||\big(Df(z)\big)(x)||_\infty \colon ||x||_\infty \le 1\}$$

Before applying the mean value theorem to $T_\mathbb{P}$, let us consider the following:

- Note that for every $[0,1]$-interpretation $J \in A$ and for every vector $x \in \mathbb{R}^n$ such that $||x||_\infty = \max\{|x_i| \colon x = (x_1, \ldots, x_n)\} \le 1$:

$$||\big(DT_\mathbb{P}(J)\big)(x)||_\infty = \max_i \left\{ \sum_{j=1}^n \left| \frac{\partial (T_\mathbb{P})_i}{\partial p_j}(J(p_1), \ldots, J(p_n)) \cdot x_j \right| \right\} \le$$

$$\le \max_i \left\{ \sum_{j=1}^n \left| \frac{\partial (T_\mathbb{P})_i}{\partial p_j}(J(p_1), \ldots, J(p_n)) \right| \right\} < 1$$

  where the last inequality follows from (3).
- Note that A can be written as the n-cube $[0, \vartheta_{p_1}] \times \cdots \times [0, \vartheta_{p_n}]$. As a result, A is a *compact* set, and the supremum:

$$\sup\{||DT_\mathbb{P}(J)||_\infty \colon J \in A\}$$

  is in fact a maximum.
- Note as well that for every $J_1, J_2 \in A$ the line segment $[J_1, J_2]$ is included in A.

Let us see now that $T_\mathbb{P}$ is a contractive map in A. Let J_1, J_2 be two $[0,1]$-interpretations in A. Then by the mean value theorem:

$$||T_\mathbb{P}(J_1) - T_\mathbb{P}(J_2)||_\infty \le ||J_1 - J_2||_\infty \sup\{||DT_\mathbb{P}(J)||_\infty \colon J \in [J_1, J_2]\}$$

$$\text{(by using the third consideration above)}$$

$$\le ||J_1 - J_2||_\infty \sup\{||DT_\mathbb{P}(J)||_\infty \colon J \in A\}$$

$$\text{(by the second note above)}$$

$$\le ||J_1 - J_2||_\infty \max\{||DT_\mathbb{P}(J)||_\infty \colon J \in A\}$$

$$\text{(by the first note above)}$$

$$\le ||J_1 - J_2||_\infty \cdot \lambda \qquad \text{for } \lambda < 1$$

Therefore the function $T_\mathbb{P}$ is a contractive mapping in A, as we wanted to prove.

We consider now the general case. Let us assume that $\mathbb{P}$ is an arbitrary normal logic program. Then we consider a partition of normal logic programs $\{\mathbb{P}_i\}_{i \in I}$ as described in Lemma 2.8, that is,

- in each $\mathbb{P}_i$ there are not two rules with the same head.
- the equality $T_\mathbb{P}(I)(p) = \sup_{i \in I}\{T_{\mathbb{P}_i}(I)(p)\}$ holds.

Moreover as $\mathbb{P}$ is a finite normal logic program, the supremum is actually a maximum; i.e $T_\mathbb{P}(I)(p) = \max_{i \in I}\{T_{\mathbb{P}_i}(I)(p)\}$. By using the case above, we can state that each $T_{\mathbb{P}_i}$ is a contractive map in A (by using the values ϑ_{p_i} defined globally for $\mathbb{P}$) with Lipschitz constant λ_i. Therefore, it is straightforward to prove that $T_\mathbb{P}$ is a contractive map in A with Lipschitz constant $\max_{i \in \mathbb{I}}\{\lambda_i\}$.

Finally, to conclude the proof, we only have to show that every fix-point of $T_\mathbb{P}$ belongs to A. But this is easy, since for every $[0,1]$-interpretation I, $T_\mathbb{P}(I)(p) \le$

$\max\{\vartheta_j \colon \langle p \leftarrow \mathcal{B} \,; \vartheta_j \rangle \in \mathbb{P}\}$; i.e $T_{\mathbb{P}}([0,1]^n) \subseteq A$.

∎

The following example shows how we can apply Theorem 4.2 to a given normal residuated logic program.

Example 4.3 Consider the following program

$$r_1 \colon \langle p \leftarrow q * \neg s \,; \qquad 0.7\rangle \qquad\qquad r_2 \colon \langle q \leftarrow \neg t \,; \qquad 0.4\rangle$$

$$r_3 \colon \langle t \leftarrow \neg q \,; \qquad 0.8\rangle \qquad\qquad r_4 \colon \langle s \leftarrow t * u * \neg p \,; \qquad 0.5\rangle$$

$$r_5 \colon \langle p \leftarrow q * t \,; \qquad 0.7\rangle \qquad\qquad r_6 \colon \langle u \leftarrow \,; \qquad 0.5\rangle$$

In order to check that this program satisfies the hypotheses of Theorem 4.2, firstly we need to obtain the values $\vartheta_x = \max\{\vartheta_j \colon \langle x \leftarrow \mathcal{B} \,; \vartheta_j \rangle \in \mathbb{P}\}$ for each propositional symbol x.

x	p	q	s	t	u
ϑ_x	0.7	0.4	0.5	0.8	0.5

Secondly we must check that the inequality shown in Theorem 4.2 holds for every rule in the program: by writing ϑ_{r_i} for the weight of each rule r_i in the program, the following expressions are obtained for any rule:
For r_1:

$$\vartheta_{r_1} + \vartheta_{r_1} \cdot \vartheta_q = 0.7 + 0.7 \cdot 0.4 = 0.98 < 1$$

For r_2 and r_3:

$$\vartheta_{r_2} = 0.4 < 1 \qquad\qquad \vartheta_{r_3} = 0.8 < 1$$

For r_4:

$$\vartheta_{r_4} \cdot \vartheta_u + \vartheta_{r_4} \cdot \vartheta_t + \vartheta_{r_4} \cdot \vartheta_t \cdot \vartheta_u = 0.5 \cdot 0.5 + 0.8 \cdot 0.5 + 0.8 \cdot 0.5 \cdot 0.5 = 0.85 < 1$$

For r_5:

$$\vartheta_{r_5} \cdot \vartheta_t + \vartheta_{r_5} \cdot \vartheta_q = 0.7 \cdot 0.4 + 0.7 \cdot 0.8 = 0.84 < 1$$

As r_6 is a fact, the inequality does not impose any restriction.

Therefore, we can apply Theorem 4.2 and obtain that this program has only one stable model.

At first sight, the hypothesis required in Theorem 4.2 might seem too involved. The following corollary provides a sufficient condition for the inequality required in Theorem 4.2.

COROLLARY 4.4 *Consider a rule* $\langle p \leftarrow q_1 * \cdots * q_h * \neg q_1 * \cdots * \neg q_k \,; \vartheta \rangle$ *in a finite normal residuated logic program* $\mathbb{P}$*. If the following inequality holds*

$$h \cdot (\max\{\vartheta_{q_1}, \ldots, \vartheta_{q_h}, \vartheta\})^h + k \cdot (\max\{\vartheta_{q_1}, \ldots, \vartheta_{q_h}, \vartheta\})^{h+1} < 1$$

then the rule satisfies the inequality required in the statement of Theorem 4.2.

Proof The result follows from the following inequality:

$$\left(\sum_{j}^{h} \vartheta_{q_1} \cdot \ldots \cdot \vartheta_{q_{j-1}} \cdot \vartheta_{q_{j+1}} \cdot \ldots \cdot \vartheta_{q_h} \cdot \vartheta \right) + k(\vartheta_{q_1} \cdot \ldots \cdot \vartheta_{q_h} \cdot \vartheta) \le$$

$$\le \left(\sum_{j}^{h} (\max\{\vartheta_{q_1}, \ldots, \vartheta_{q_h}, \vartheta\})^h \right) + k(\max\{\vartheta_{q_1}, \ldots, \vartheta_{q_h}, \vartheta\})^{h+1}$$

$$= h \cdot (\max\{\vartheta_{q_1}, \ldots, \vartheta_{q_h}, \vartheta\})^h + k \cdot (\max\{\vartheta_{q_1}, \ldots, \vartheta_{q_h}, \vartheta\})^{h+1}$$

■

COROLLARY 4.5 *Let $\mathbb{P}$ be a finite normal residuated logic program. If every rule in $\mathbb{P}$ has a weight strictly less than 1 and at most one propositional symbol appears in the body of each rule, then $\mathbb{P}$ has only one stable model.*

Proof It follows as an application of Corollary 4.4.

Let r be a rule in $\mathbb{P}$. If r is a fact, the hypothesis holds trivially.

If r has any of the forms $\langle p \leftarrow q \,; \vartheta \rangle$ or $\langle p \leftarrow \neg q \,; \vartheta \rangle$, the inequality to be verified is $\vartheta < 1$; which holds by the hypothesis of this corollary.   ■

In the following examples we stress the fact that the underlying residuated lattice should be based on the product t-norm for the uniqueness results stated in this section.

Example 4.6 Corollary 4.5 guarantees that the program $\mathbb{P}$

$$\langle p \leftarrow \neg q \,; 0.9 \rangle \qquad\qquad \langle q \leftarrow \neg p \,; 0.9 \rangle$$

has only one stable model when considering the product t-norm.

However, the same normal logic program interpreted with Gödel t-norm, its residuated implication, and the standard negation, has infinitely many stable models, specifically every interpretation in the set $\{M_\alpha = \{(p, \alpha); (q, 1-\alpha)\} \colon 0.1 \le \alpha \le 0.9\}$ would be a stable model.

Example 4.7 Consider the following normal residuated logic program on $\mathcal{L}_P$:

$$\langle p \leftarrow \neg q * \neg r \,; 0.9 \rangle$$
$$\langle q \leftarrow \neg r * \neg p \,; 0.9 \rangle$$
$$\langle r \leftarrow \neg p * \neg q \,; 0.9 \rangle$$

It is not difficult to check that it has only one stable model M which is given, approximately, by $\{(p, 0.3608); (q, 0.3608); (r, 0.3608)\}$. However the program does not satisfy the hypothesis of Theorem 4.2 since for the first rule we have that $2 \cdot 0.9 = 1.8 \ge 1$.

Computing the unique stable model of $\mathbb{P}$

Banach's fix-point theorem plays a crucial role in the proof of Theorem 4.2; moreover, Banach's theorem not only provides a proof of uniqueness, but a method to compute it as well. Specifically, if $\mathbb{P}$ is a normal residuated logic program which satisfies the hypothesis of Theorem 4.2, we can compute its unique stable model

by computing the limit of the following sequence:

$$I_{i+1} = T_{\mathbb{P}}(I_i)$$

where I_0 is any $[0, 1]$-interpretation.

THEOREM 4.8 *Under the hypothesis of Theorem 4.2, the sequence $I_0 = I_\perp$, $I_{i+1} = T_{\mathbb{P}}(I_i)$ converges to the unique stable model of $\mathbb{P}$, although it may require ω many steps.*

Example 4.9 Continuing with Example 4.3, the computation of the unique stable model of $\mathbb{P}$ by using the above sequence is given in the following table:

x	$T_{\mathbb{P}}(x)(p)$	$T_{\mathbb{P}}(x)(q)$	$T_{\mathbb{P}}(x)(s)$	$T_{\mathbb{P}}(x)(t)$	$T_{\mathbb{P}}(x)(u)$
I_0	0	0	0	0	0
I_1	0	0.4	0	0.8	0.5
I_2	0.28	0.079999999999	0.2	0.48	0.5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_{100}	0.068820224719	0.11764705882	0.16432584269	0.70588235294	0.5
I_{101}	0.068820224719	0.11764705882	0.16432584269	0.70588235294	0.5

Thus I_{100} is the unique stable model of $\mathbb{P}$.

5. Conclusions

We have introduced results on existence and uniqueness of stable models for normal residuated logic programs. The first one determines that every normal residuated logic program defined in $[0, 1]$ in terms of continuous operators has at least one stable model; as a result, inconsistencies (that is, lack of stable models) can be avoided by simply considering continuous operators. The second result provides sufficient conditions under which a normal logic program has only one stable model which can be computed as a fix-point of the immediate consequences operator.

References

[1] A. Burrieza, I. P. de Guzmán, and E. Muñoz-Velasco. Generalization of some properties of relations in the context of functional temporal×modal logic. *Intl J of Computer Mathematics* 85(3):371–383, 2007.
[2] A. Burrieza, A. Mora, M. Ojeda-Aciego, and E. Orłowska. An implementation of a dual tableaux system for order-of-magnitude qualitative reasoning. *Intl J of Computer Mathematics* 86(10):1852–1866, 2009.
[3] S. Costantini. On the existence of stable models of non-stratified logic programs. *Journal of Theory and Practice of Logic Programming*, 6(1-2):169–212, 2006.
[4] C. Damásio, J. Medina, and M. Ojeda-Aciego. Termination of logic programs with imperfect information: applications and query procedure. *Journal of Applied Logic*, 5(3):435–458, 2007.
[5] 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.
[6] M. Fitting. The family of stable models. *The Journal of Logic Programming*, 17(2-4):197 – 225, 1993.
[7] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of ICLP-88*, pages 1070–1080, 1988.
[8] M. L. Ginsberg. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 4:265–316, 1988.
[9] J. Golińska-Pilarek and E. Muñoz-Velasco. Dual tableau for a multimodal logic for order of magnitude qualitative reasoning with bidirectional negligibility. *Intl J of Computer Mathematics* 86(10):1707–1718, 2009.
[10] Y. Loyer and U. Straccia. Epistemic foundation of stable model semantics. *Journal of Theory and Practice of Logic Programming*, 6:355–393, 2006.

[11] T. Lukasiewicz. Fuzzy description logic programs under the answer set semantics for the semantic web. *Fundamenta Informaticae*, 82(3):289–310, 2008.

[12] N. Madrid and M. Ojeda-Aciego. Towards a fuzzy answer set semantics for residuated logic programs. In *Web Intelligence/IAT Workshops*, pages 260–264, 2008.

[13] 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.

[14] N. Madrid and M. Ojeda-Aciego. On the existence of stable models in normal residuated logic programs. In Proc of *Computational Methods in Mathematics, Science, and Engineering*. CMMSE, pages 598-604, 2010.

[15] A. Mora, E. Muñoz-Velasco, and J. Golińska-Pilarek. Implementing a relational theorem prover for modal logic K. *Intl J of Computer Mathematics*, 2011.
iFirst `http://dx.doi.org/10.1080/00207160.2010.493211`

[16] T. Przymusinski. Well-founded semantics coincides with three-valued stable semantics. *Fundamenta Informaticae*, 13:445–463, 1990.

[17] U. Straccia. Query answering in normal logic programs under uncertainty. *Lect. Notes in Computer Science*, 3571:687–700, 2005.

[18] U. Straccia. Query answering under the any-world assumption for normal logic programs. *Lect. Notes in Computer Science*, 3571:687–700, 2006.

[19] U. Straccia. A top-down query answering procedure for normal logic programs under the any-world assumption. *Proc. of the 10th Intl Conf on Principles of Knowledge Representation*, 329–339, AAAI Press, 2006.

[20] M. Tsoy-Wo. *Classical Analysis on Normed Spaces*. World Scientific Publishing, 1995.