Multi-Adjoint Logic Programming with Continuous Semantics

Jesús Medina,[1] Manuel Ojeda-Aciego,[1] and Peter Vojtáš[2]

[1] Dept. Matemática Aplicada. Universidad de Málaga. [***]
{jmedina,aciego}@ctima.uma.es
[2] Inst. Computer Science. Academy of Science of Czech Republic.[†]
vojtas@cs.cas.cz

Abstract. Considering different implication operators, such as Łukasiewicz, Gödel or product implication in the same logic program, naturally leads to the allowance of several adjoint pairs in the lattice of truth-values. In this paper we apply this idea to introduce multi-adjoint logic programs as an extension of monotonic logic programs. The continuity of the immediate consequences operators is proved and the assumptions required to get continuity are further analysed.

1 Introduction

One can find several papers in the literature on applications of definite fuzzy logic programming which are based either on Łukasiewicz, or product, or Gödel implications on the unit real interval (an overview can be seen in [9]); for more complex systems it is reasonable to allow room for several different implications. In [2] an extension was presented in which the set of truth-values is generalised to a residuated lattice (in order to embed hybrid probabilistic logic programs). Another generalisation of the set of truth-values is that given by the structure of bilattice, which has been used to handle negation in logic programming [5].

The purpose of this work is to provide a further generalisation of the framework given in [2, 3] so that: (1) it is possible to use a number of different implications in the rules of our programs, (2) the algebraic requirements on residuated lattices are weaken and (3) we focus on the continuity of the immediate consequences operator by providing sufficient conditions for continuity.

A general theory of logic programming which allows the simultaneous use of different implications in the rules and rather general connectives in the bodies is presented. Models of these programs are post-fixpoints of the immediate consequences operator, which is proved to be monotonic under very general hypotheses.

The final part of the paper deals with the continuity of the immediate consequences operator, which is proved under the assumption of continuity of all

*** Partially supported by Spanish DGI project BFM2000-1054-C02-02 and Junta de Andalucía project TIC-115.
† Supported by Grant GAČR 201/00/1489

© Springer-Verlag. Lect. Notes in Artificial Intelligence 2173:351–364, 2001

the operators in the program (but, possibly, the implications). This theorem is also re-stated in terms of lower-semicontinuity of the operators.

2 Preliminary definitions

We will make extensive use of the constructions and terminology of universal algebra, in order to define formally the syntax and the semantics of the languages we will deal with. A minimal set of concepts from universal algebra, which will be used in the sequel in the style of [2], are introduced below.

2.1 Some Definitions from Universal Algebra

Definition 1 (Graded set). *A* graded set *is a set Ω with a function which assigns to each element $\omega \in \Omega$ a number $n \geq 0$, called the arity of ω.*

Definition 2 (Ω-Algebra). *Given a graded set Ω, an Ω-*algebra *$\mathfrak{A}$ is a pair $\langle A, I \rangle$ where A is a nonempty set called the carrier, and I is a function which assigns maps to the elements of Ω as follows:*

1. *Each element $\omega \in \Omega_n$, $n > 0$, is interpreted as a map $I(\omega)\colon A^n \to A$, denoted by $\omega_{\mathfrak{A}}$.*
2. *Each element $c \in \Omega_0$ (i.e., c is a constant) is interpreted as an element $I(c)$ in A, denoted by $c_{\mathfrak{A}}$.*

Finally, the last definition needed will be that of *subalgebra* of an Ω-algebra, which generalises the concept of substructure of an algebraic structure. The definition is straightforward.

Definition 3 (Subalgebra of an Ω-algebra). *Given an Ω-algebra $\mathfrak{A} = \langle A, I \rangle$, an Ω-*subalgebra *$\mathfrak{B}$, is a pair $\langle B, J \rangle$, such that $B \subset A$ and*

1. *$J(c) = I(c)$ for all $c \in \Omega_0$.*
2. *Given $\omega \in \Omega_n$, then $J(\omega)\colon B^n \to B$ is the restriction of $I(\omega)\colon A^n \to A$.*

2.2 Multi-Adjoint Lattices and Multi-Adjoint Algebras

The main concept we will need in this section is that of *adjoint pair*, firstly introduced in a logical context by Pavelka [8], who interpreted the poset structure of the set of truth-values as a category, and the relation between the connectives of implication and conjunction as functors in this category. The result turned out to be another example of the well-known concept of adjunction, introduced by Kan in the general setting of category theory in 1950.

Definition 4 (Adjoint pair). *Let $\langle P, \preceq \rangle$ be a partially ordered set and $(\leftarrow, \&)$ a pair of binary operations in P such that:*

(a1) Operation $\&$ is increasing in both arguments, i.e. if $x_1, x_2, y \in P$ such that $x_1 \preceq x_2$ then $(x_1 \,\&\, y) \preceq (x_2 \,\&\, y)$ and $(y \,\&\, x_1) \preceq (y \,\&\, x_2)$;

(a2) *Operation $\leftarrow$ is increasing in the first argument (the consequent) and decreasing in the second argument (the antecedent), i.e. if $x_1, x_2, y \in P$ such that $x_1 \preceq x_2$ then $(x_1 \leftarrow y) \preceq (x_2 \leftarrow y)$ and $(y \leftarrow x_2) \preceq (y \leftarrow x_1)$;*

(a3) *For any $x, y, z \in P$, we have that $x \preceq (y \leftarrow z)$ holds if and only if $(x \mathbin{\&} z) \preceq y$ holds.*

Then we say that $(\leftarrow, \&)$ forms an adjoint pair *in $\langle P, \preceq\rangle$.*

The need of the monotonicity of operators $\leftarrow$ and $\&$ is clear, if they are to be interpreted as generalised implications and conjunctions. The third property in the definition, which corresponds to the categorical adjointness; but can be adequately interpreted in terms of multiple-valued inference as asserting that the truth-value of $y \leftarrow z$ is the maximal x satisfying $x \mathbin{\&} z \preceq_P y$, and also the validity of the following generalised modus ponens rule [6]:

> If x is a lower bound of $\psi \leftarrow \varphi$, and z is a lower bound of φ then a lower bound y of ψ is $x \mathbin{\&} z$.

In addition to $(a1)$-$(a3)$ it will be necessary to assume the existence of bottom and top elements in the poset of truth-values (the zero and one elements), and the existence of joins (suprema) for every directed subset; that is, we will assume a structure of complete lattice but nothing about associativity, commutativity and general boundary conditions of $\&$. In particular, the requirement that $(L, \&, \top)$ has to be a commutative monoid in a residuated lattice is too restrictive, in that commutativity needn't be required in the proofs of soundness and correctness [9]. Here in this generality we are able to work with approximations of t-norms and/or conjunctions learnt from data by a neural net like in [7].

Extending the results in [2, 3, 9] to a more general setting, in which different implications (Łukasiewicz, Gödel, product) and thus, several modus ponens-like inference rules are used, naturally leads to considering several *adjoint pairs* in the lattice. More formally,

Definition 5 (Multi-Adjoint Lattice). *Let $\langle L, \preceq\rangle$ be a lattice. A* multi-adjoint lattice *$\mathcal{L}$ is a tuple $(L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ satisfying the following items:*

(l1) *$\langle L, \preceq\rangle$ is bounded, i.e. it has bottom ($\bot$) and top ($\top$) elements;*

(l2) *$(\leftarrow_i, \&_i)$ is an adjoint pair in $\langle L, \preceq\rangle$ for $i = 1, \ldots, n$;*

(l3) *$\top \mathbin{\&_i} \vartheta = \vartheta \mathbin{\&_i} \top = \vartheta$ for all $\vartheta \in L$ for $i = 1, \ldots, n$.*

Remark 1. Note that residuated lattices are a special case of multi-adjoint lattice, in which the underlying poset has a lattice structure, has monoidal structure wrt $\otimes$ and $\top$, and only one adjoint pair is present.

From the point of view of expressiveness, it is interesting to allow extra operators to be involved with the operators in the multi-adjoint lattice. The structure which captures this possibility is that of a multi-adjoint algebra.

Definition 6 (Multi-Adjoint Ω-Algebra). *Let Ω be a graded set containing operators $\leftarrow_i$ and $\&_i$ for $i = 1, \ldots, n$ and possibly some extra operators, and let $\mathfrak{L} = (L, I)$ be an Ω-algebra whose carrier set L is a lattice under $\preceq$.*

We say that $\mathfrak{L}$ is a multi-adjoint Ω-algebra *with respect to the pairs $(\leftarrow_i, \&_i)$ for $i = 1, \ldots, n$ if $\mathcal{L} = (L, \preceq, I(\leftarrow_1), I(\&_1), \ldots, I(\leftarrow_n), I(\&_n))$ is a multi-adjoint lattice.*

In practice, we will usually have to assume some properties on the extra operators considered. These extra operators will be assumed to be either conjunctors or disjunctors or aggregators.

Example 1. Consider $\Omega = \{\leftarrow_P, \&_P, \leftarrow_G, \&_G, \wedge_L, @\}$, the real unit interval $U = [0, 1]$ with its lattice structure, and the interpretation function I defined as:

$$I(\leftarrow_P)(x, y) = \min(1, x/y) \qquad I(\&_P)(x, y) = x \cdot y$$

$$I(\leftarrow_G)(x, y) = \begin{cases} 1 & \text{if } x \le y \\ 0 & \text{otherwise} \end{cases} \qquad I(\&_G)(x, y) = \min(x, y)$$

$$I(@)(x, y, z) = \tfrac{1}{6}(x + 2y + 3z) \qquad I(\wedge_L)(x, y) = \max(0, x + y - 1)$$

that is, connectives are interpreted as product and Gödel connectives, a weighted sum and Łukasiewicz implication; then $\langle U, I\rangle$ is a multi-adjoint Ω-algebra with one aggregator and one additional conjunctor (denoted $\wedge_L$ to make explicit that its adjoint implicator is not in the language).

Note that the use of aggregators as weighted sums somehow covers the approach taken in [1] when considering the evidential support logic rules of combination.

□

2.3 General Approach to the Syntax of Propositional Languages

The syntax of the propositional languages we will work with will be defined by using the concept of Ω-algebra. To begin with, the concept of alphabet of the language is introduced below.

Definition 7 (Alphabet). *Let Ω be a graded set, and Π a countably infinite set. The* alphabet $A_{\Omega,\Pi}$ *associated to Ω and Π is defined to be the disjoint union $\Omega \cup \Pi \cup S$, where S is the set of auxiliary symbols "(", ")" and ",".*

In the following, we will use only A_Ω to designate an alphabet, for deleting the reference to Π cannot lead to confusion.

Definition 8 (Expressions). *Given a graded set Ω and alphabet A_Ω. The Ω-algebra $\mathfrak{E} = \langle {A_\Omega}^*, I\rangle$ of* expressions *is defined as follows:*

1. *The carrier ${A_\Omega}^*$ is the set of strings over A_Ω.*

2. *The interpretation function I satisfies the following conditions for strings $a_1, \dots, a_n$ in ${A_\Omega}^*$:*
   - $c_{\mathfrak{E}} = c$, *where c is a constant operation ($c \in \Omega_0$).*
   - $\omega_{\mathfrak{E}}(a_1) = \omega\, a_1$, *where ω is an unary operation ($\omega \in \Omega_1$).*
   - $\omega_{\mathfrak{E}}(a_1, a_2) = (a_1 \omega\, a_2)$, *where ω is a binary operation ($\omega \in \Omega_2$).*
   - $\omega_{\mathfrak{E}}(a_1, \dots, a_n) = \omega(a_1, \dots, a_n)$, *where ω is a n-ary operation ($\omega \in \Omega_n$) and $n > 2$.*

Note that an expression is only a string of letters of the alphabet, that is, it needn't be a well-formed formula. Actually, the well-formed formulas is the subset of the set of expressions defined as follows:

Definition 9 (Well-formed formulas). *Let Ω be a graded set, Π a countable set of propositional symbols and $\mathfrak{E}$ the algebra of expressions corresponding to the alphabet $A_{\Omega,\Pi}$. The* well-formed formulas *(in short, formulas) generated by Ω over Π is the least subalgebra $\mathfrak{F}$ of the algebra of expressions $\mathfrak{E}$ containing Π.*

The set of formulas, that is the carrier of $\mathfrak{F}$, will be denoted F_Ω. It is well-known that least subalgebras can be defined as an inductive closure, and it is not difficult to check that it is freely generated, therefore it satisfies the *unique homomorphic extension theorem* stated below:

Theorem 1. *Let Ω be a graded set, Π a set of propositional symbols, $\mathfrak{F}$ the corresponding Ω-algebra of formulas. Let $\mathfrak{L}$ be an arbitrary Ω-algebra with carrier L. Then, for every function $J \colon \Pi \to L$ there is a unique homomorphism $\hat{J} \colon F_\Omega \to L$ such that:*

1. *For all $p \in \Pi$, $\hat{J}(p) = J(p)$;*
2. *For each constant $c \in \Omega_0$, $\hat{J}(c_{\mathfrak{F}}) = c_{\mathfrak{L}}$;*
3. *For every $\omega \in \Omega_n$ with $n > 0$ and for all $F_i \in F_\Omega$ with $i = 1, \dots, n$*

$$\hat{J}(\omega_{\mathfrak{F}}(F_1, \dots, F_n)) = \omega_{\mathfrak{L}}(\hat{J}(F_1), \dots, \hat{J}(F_n)).$$

3 Syntax and Semantics of Multi-Adjoint Logic Programs

Multi-adjoint logic programs will be constructed from the abstract syntax induced by a multi-adjoint algebra on a set of propositional symbols. Specifically, we will consider a multi-adjoint Ω-algebra $\mathfrak{L}$ whose extra operators are either conjunctors, denoted $\wedge_1, \dots, \wedge_k$, or disjunctors, denoted $\vee_1, \dots, \vee_l$, or aggregators, denoted $@_1, \dots, @_m$. (This algebra will host the manipulation the truth-values of the formulas in our programs.)

In addition, let Π be a set of propositional symbols and the corresponding algebra of formulas $\mathfrak{F}$ freely generated from Π by the operators in Ω. (This algebra will be used to define the syntax of a propositional language.)

Remark 2. As we are working with two Ω-algebras, and to discharge the notation, we introduce a special notation to clarify which algebra an operator belongs to, instead of continuously using either $\omega_{\mathfrak{L}}$ or $\omega_{\mathfrak{F}}$. Let ω be an operator symbol in Ω, its interpretation under $\mathfrak{L}$ is denoted $\dot{\omega}$ (a dot on the operator), whereas ω itself will denote $\omega_{\mathfrak{F}}$ when there is no risk of confusion.

3.1 Syntax of Multi-Adjoint Logic Programs

The definition of multi-adjoint logic program is given, as usual, as a set of rules and facts. The particular syntax of these rules and facts is given below:

Definition 10 (Multi-Adjoint Logic Programs). *A* multi-adjoint logic program *is a set $\mathbb{P}$ of rules of the form $\langle (A \leftarrow_i \mathcal{B}), \vartheta \rangle$ such that:*

1. *The* rule $(A \leftarrow_i \mathcal{B})$ *is a formula of* $\mathfrak{F}$*;*
2. *The* confidence factor ϑ *is an element (a truth-value) of* L*;*
3. *The* head *of the rule* A *is a propositional symbol of* Π*.*
4. *The* body *formula* $\mathcal{B}$ *is a formula of* $\mathfrak{F}$ *built from propositional symbols* $B_1, \dots, B_n$ $(n \geq 0)$ *by the use of conjunctors* $\&_1, \dots, \&_n$ *and* $\wedge_1, \dots, \wedge_k$*, disjunctors* $\vee_1, \dots, \vee_l$ *and aggregators* $@_1, \dots, @_m$ *.*
5. Facts *are rules with body* $\top$*.*
6. *A* query *(or* goal*) is a propositional symbol intended as a question* $?A$ *prompting the system.*

Note that an arbitrary composition of conjunctors, disjunctors and aggregators is also an aggregator.

Sometimes, we will represent the above pair as $A \overset{\vartheta}{\leftarrow}_i @[B_1, \dots, B_n]$, where[1] $B_1, \dots, B_n$ are the propositional variables occurring in the body and @ is the aggregator obtained as a composition.

3.2 Semantics of Multi-Adjoint Logic Programs

Definition 11 (Interpretation). *An* interpretation *is a mapping* $I\colon \Pi \to L$*. The set of all interpretations of the formulas defined by the* Ω*-algebra* $\mathfrak{F}$ *in the* Ω*-algebra* $\mathfrak{L}$ *is denoted* $\mathcal{I}_{\mathfrak{L}}$*.*

Note that by the unique homomorphic extension theorem, each of these interpretations can be uniquely extended to the whole set of formulas F_Ω.

The ordering $\preceq$ of the truth-values L can be easily extended to the set of interpretations as usual:

Definition 12 (Lattice of interpretations). *Consider two interpretations* $I_1, I_2 \in \mathcal{I}_{\mathfrak{L}}$*. Then,* $\langle \mathcal{I}_{\mathfrak{L}}, \sqsubseteq \rangle$ *is a lattice where* $I_1 \sqsubseteq I_2$ *iff* $I_1(p) \preceq I_2(p)$ *for all* $p \in \Pi$*. The least interpretation* $\vartriangle$ *maps every propositional symbol to the least element* $\perp$ *of* L*.*

A rule of a multi-adjoint logic program is satisfied whenever the truth-value of the rule is greater or equal than the confidence factor associated with the rule. Formally:

Definition 13 (Satisfaction, Model). *Given an interpretation* $I \in \mathcal{I}_{\mathfrak{L}}$*, a weighted rule* $\langle A \leftarrow_i \mathcal{B}, \vartheta \rangle$ *is* satisfied *by* I *iff* $\vartheta \preceq \hat{I}\,(A \leftarrow_i \mathcal{B})$*. An interpretation* $I \in \mathcal{I}_{\mathfrak{L}}$ *is a* model *of a multi-adjoint logic program* $\mathbb{P}$ *iff all weighted rules in* $\mathbb{P}$ *are satisfied by* I*.*

[1] Note the use of square brackets in this context.

Note the following equalities

$$\hat{I}(A \leftarrow_i \mathcal{B}) = \hat{I}(A) \dot{\leftarrow}_i \hat{I}(\mathcal{B}) = I(A) \dot{\leftarrow}_i \hat{I}(\mathcal{B})$$

and the evaluation of $\hat{I}(\mathcal{B})$ proceeds inductively as usual, till all propositional symbols in $\mathcal{B}$ are reached and evaluated under I. For the particular case of a fact (a rule with $\top$ in the body) satisfaction of $\langle A \leftarrow_i \top, \vartheta \rangle$ means

$$\vartheta \preceq \hat{I}(A \leftarrow_i \top) = I(A) \dot{\leftarrow}_i \top$$

by property *(a3)* of adjoint pairs this is equivalent to $\vartheta \,\dot{\&}_i\, \top \preceq I(A)$ and this by assumption *(l3)* of multi-adjoint lattices gives $\vartheta \preceq I(A)$.

Definition 14. *An element $\lambda \in L$ is a* correct answer *for a program $\mathbb{P}$ and a query $?A$ if for an arbitrary interpretation $I\colon \Pi \to L$ which is a model of $\mathbb{P}$ we have $\lambda \preceq I(A)$.*

4 Fix-point semantics

It is possible to generalise the immediate consequences operator, given by van Emden and Kowalski in [4], to the framework of multi-adjoint logic programs as follows:

Definition 15. *Let $\mathbb{P}$ be a multi-adjoint logic program. The* immediate consequences operator *$T_{\mathbb{P}}^{\mathfrak{L}}\colon \mathcal{I}_{\mathfrak{L}} \to \mathcal{I}_{\mathfrak{L}}$, mapping interpretations to interpretations, is defined by considering*

$$T_{\mathbb{P}}^{\mathfrak{L}}(I)(A) = \sup\left\{ \vartheta \,\dot{\&}_i\, \hat{I}(\mathcal{B}) \mid A \overset{\vartheta}{\leftarrow}_i \mathcal{B} \in \mathbb{P} \right\}$$

Note that all the suprema involved in the definition do exist because L is assumed to be a lattice.

As it is usual in the logic programming framework, the semantics of a multi-adjoint logic program is characterised by the post-fixpoints of $T_{\mathbb{P}}^{\mathfrak{L}}$.

Theorem 2. *An interpretation I of $\mathcal{I}_{\mathfrak{L}}$ is a model of a multi-adjoint logic program $\mathbb{P}$ iff $T_{\mathbb{P}}^{\mathfrak{L}}(I) \sqsubseteq I$.*

Proof: Assume we have an interpretation I for the program $\mathbb{P}$, then we have the following chain of equivalent statements for all rule $A \overset{\vartheta}{\leftarrow}_i \mathcal{B}$ in $\mathbb{P}$

$$\begin{aligned}
\vartheta &\preceq \hat{I}(A \leftarrow_i \mathcal{B}) \\
\vartheta &\preceq \hat{I}(A) \dot{\leftarrow}_i \hat{I}(\mathcal{B}) \\
\vartheta \,\dot{\&}_i\, \hat{I}(\mathcal{B}) &\preceq \hat{I}(A) = I(A) \\
\sup\{\vartheta \,\dot{\&}_i\, \hat{I}(\mathcal{B}) \mid A \overset{\vartheta}{\leftarrow}_i \mathcal{B} \in \mathbb{P}\} &\preceq I(A) \\
T_{\mathbb{P}}^{\mathfrak{L}}(I)(A) &\preceq I(A)
\end{aligned}$$

Thus, if I is a model of $\mathbb{P}$, then for every A occurring in the head of a rule we have $T_{\mathbb{P}}^{\mathfrak{L}}(I)(A) \preceq I(A)$. If A is not the head of any rule, we have $T_{\mathbb{P}}^{\mathfrak{L}}(I)(A) = \sup \varnothing = \bot \leq I(A)$ and, therefore, I is a post-fixpoint for $T_{\mathbb{P}}^{\mathfrak{L}}$.

Reciprocally, assume that I is a post-fixpoint for $T_{\mathbb{P}}^{\mathfrak{L}}$, then any rule $A \overset{\vartheta}{\longleftarrow}_i \mathcal{B}$ is fulfilled.

□

Note that the fixpoint theorem works even without any further assumptions on conjunctors (definitely they need not be commutative and associative).

The monotonicity of the operator $T_{\mathbb{P}}^{\mathfrak{L}}$, for the case of only one adjoint pair, has been shown in [3]. The proof for the general case is similar.

Theorem 3 (Monotonicity of $T_{\mathbb{P}}^{\mathfrak{L}}$). *The operator $T_{\mathbb{P}}^{\mathfrak{L}}$ is monotonic.*

Proof: Consider I and J two elements of $\mathcal{I}_{\mathfrak{L}}$ such that $I \sqsubseteq J$. We have to show that

$$T_{\mathbb{P}}^{\mathfrak{L}}(I) \sqsubseteq T_{\mathbb{P}}^{\mathfrak{L}}(J)$$

Let A be a propositional symbol in Π,

$$T_{\mathbb{P}}^{\mathfrak{L}}(I)(A) = \sup \left\{ \vartheta \mathbin{\dot{\&}_i} \hat{I}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P} \right\}$$

If we had $\hat{I}(\mathcal{B}) \leq \hat{J}(\mathcal{B})$ for all $\mathcal{B}$, then we would also have $\vartheta \mathbin{\dot{\&}_i} \hat{I}(\mathcal{B}) \preceq \vartheta \mathbin{\dot{\&}_i} \hat{J}(\mathcal{B})$ for all i, since operators $\dot{\&}_i$ are increasing. Now, by taking suprema

$$T_{\mathbb{P}}^{\mathfrak{L}}(I)(A) \preceq T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \quad \text{for all } A$$

Therefore, it is sufficient to prove that $\hat{I}(\mathcal{B}) \preceq \hat{J}(\mathcal{B})$ for all $\mathcal{B}$. We will use structural induction:

If $\mathcal{B}$ is an atomic formula, then it is obvious, ie

$$\hat{I}(\mathcal{B}) = I(\mathcal{B}) \preceq J(\mathcal{B}) = \hat{J}(\mathcal{B})$$

For the inductive case, consider $\mathcal{B} = @[\mathcal{B}_1, \dots, \mathcal{B}_n]$ and assume that $\hat{I}(\mathcal{B}_i) \preceq \hat{J}(\mathcal{B}_i)$ for all $i = 1, \dots, n$. By definition of the rules, we know that @ behaves as an aggregator, and therefore, using the induction hypothesis

$$\begin{aligned} \hat{I}(\mathcal{B}) &= \dot{@}[\hat{I}(\mathcal{B}_1), \dots, \hat{I}(\mathcal{B}_n)] \\ &\preceq \dot{@}[\hat{J}(\mathcal{B}_1), \dots, \hat{J}(\mathcal{B}_n)] \\ &= \hat{J}(\mathcal{B}) \end{aligned}$$

□

Due to the monotonicity of the immediate consequences operator, the semantics of $\mathbb{P}$ is given by its least model which, as shown by Knaster-Tarski's theorem, is exactly the least fixpoint of $T_{\mathbb{P}}^{\mathfrak{L}}$, which can be obtained by transfinitely iterating $T_{\mathbb{P}}^{\mathfrak{L}}$ from the least interpretation $\triangle$.

The proof of the monotonicity of the $T_{\mathbb{P}}^{\mathfrak{L}}$ operator in [2] is accompanied by the following statement, surely due to their wanting to stress the embedding of different logic programming paradigms:

> The major difference to classical logic programming is that our $T_{\mathbb{P}}^{\mathfrak{L}}$ may not be continuous, and therefore more than countably many iterations may be necessary to reach the least fixpoint.

In the line of the previous quotation, we would like to study sufficient conditions for the continuity of the $T_{\mathbb{P}}^{\mathfrak{L}}$ operator.

5 On the continuity of the $T_{\mathbb{P}}^{\mathfrak{L}}$ operator

A first result in this approach is that whenever every operator in Ω turns out to be continuous in the lattice, then $T_{\mathbb{P}}$ is also continuous and, consequently, its least fixpoint can be obtained by a countably infinite iteration from the least interpretation.

Let us state the definition of continuous function which will be used.

Definition 16. *Let L be a complete upper lattice and let $f\colon L \to L$ be a mapping. We say that f is* continuous *if it preserves suprema of directed sets, that is, given a directed set X one has*

$$f(\sup X) = \sup\{f(x) \mid x \in X\}$$

A mapping $g\colon L^n \to L$ is said to be continuous *provided that it is continuous in each argument separately.*

Definition 17. *Let $\mathfrak{F}$ be a language interpreted on a multi-adjoint Ω-algebra $\mathfrak{L}$, and let ω be any operator symbol in the language. We say that ω is* continuous *if its interpretation under $\mathfrak{L}$, that is $\dot{\omega}$, is continuous in L.*

Now we state and prove a technical lemma which will allow us to prove the continuity of the immediate consequences operator.

Lemma 1. *Let $\mathbb{P}$ be a program interpreted on a multi-adjoint Ω-algebra $\mathfrak{L}$, and let $\mathcal{B}$ be any body formula in $\mathbb{P}$. Assume that all the operators @ in $\mathcal{B}$ are continuous, let X be a directed set of interpretations, and write $S = \sup X$; then*

$$\hat{S}(\mathcal{B}) = \sup\{\hat{J}(\mathcal{B}) \mid J \in X\}$$

Proof: Follows by induction. □

Theorem 4. *If all the operators occurring in the bodies of the rules of a program $\mathbb{P}$ are continuous, and the adjoint conjunctions are continuous in their second argument, then $T_{\mathbb{P}}^{\mathfrak{L}}$ is continuous.*

Proof: We have to check that for each directed subset of interpretations X and each atomic formula A

$$T_{\mathbb{P}}^{\mathfrak{L}}(\sup X)(A) = \sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\}$$

Let us write $S = \sup X$, and consider the following chain of equalities:

$$\begin{aligned} T_{\mathbb{P}}^{\mathfrak{L}}(\sup X)(A) &= \sup\{\vartheta \mathbin{\dot{\&}_i} \hat{S}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\} \\ &\overset{(1)}{=} \sup\{\vartheta \mathbin{\dot{\&}_i} \sup\{\hat{J}(\mathcal{B}) \mid J \in X\} \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\} \\ &\overset{(2)}{=} \sup\{\vartheta \mathbin{\dot{\&}_i} \hat{J}(\mathcal{B}) \mid J \in X, \text{ and } A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\} \\ &= \sup\{\sup\{\vartheta \mathbin{\dot{\&}_i} \hat{J}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\} \mid J \in X\} \\ &= \sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\} \end{aligned}$$

where equality (1) follows from Lemma 1 and equality (2) follows from the continuity of the operators $\dot{\&}_i$. □

In some sense, it is possible to reverse the implication in the theorem above.

Theorem 5. *If the operator $T_{\mathbb{P}}^{\mathfrak{L}}$ is continuous for all program $\mathbb{P}$ on $\mathfrak{L}$, then any operator in the body of the rules is continuous.*

Proof: Let @ be an n-ary connective. Assume an ordering on L^n defined on components. Denoting a tuple $(y_1, \dots, y_n) \in L^n$ as $\bar{y}$, the ordering in L^n is: $\bar{y} \leq \bar{z}$ iff $y_i \preceq z_i$ for $i = 1, \dots, n$.

Let Y be a directed set in L^n, and let us check that

$$\dot{@}(\sup Y) = \sup\{\dot{@}(y_1, \dots, y_n) \mid (y_1, \dots, y_n) \in Y)\}$$

The inequality

$$\sup\{\dot{@}(y_1, \dots, y_n) \mid (y_1, \dots, y_n) \in Y\} \preceq \dot{@}(\sup Y) \tag{1}$$

follows directly by monotonicity of @ and the definition of supremum.

For the other inequality, given n propositional symbols $A_1, \dots, A_n \in \Pi$ and a tuple $\bar{y} = (y_1, \dots, y_n) \in L^n$, consider the interpretation $I_{\bar{y}}$ defined as $I(A_i) = y_i$ for $i = 1, \dots, n$ and $\bot$ otherwise. This way we have $I_{\bar{y}} \sqsubseteq I_{\bar{z}}$ if and only if $\bar{y} \leq \bar{z}$.

Consider, now, the set X_Y of interpretations $I_{\bar{y}}$ for all $\bar{y} \in Y$, and also consider its supremum, $S_Y = \sup X_Y$. By the ordering in L^n we have, for all $\bar{y} \in Y$

$$(y_1, \dots, y_n) = \big(I_{\bar{y}}(A_1), \dots, I_{\bar{y}}(A_n)\big) \leq \big(S_Y(A_1), \dots, S_Y(A_n)\big)$$

therefore we have

$$\sup Y \leq \big(S_Y(A_1), \dots, S_Y(A_n)\big)$$

now, by the monotonicity of $\dot{@}$ we have

$$\dot{@}(\sup Y) \preceq \dot{@}\,\big(S_Y(A_1), \dots, S_Y(A_n)\big) = \widehat{S_Y}(@(A_1, \dots, A_n)) \tag{2}$$

On the other hand, consider the program $\mathbb{P}$ below consisting of only a rule

$$\mathbb{P} = \left\{ A \overset{\top}{\longleftarrow}_i @(A_1, \dots, A_n) \right\}$$

by the assumption of monotonicity of $T_{\mathbb{P}}^{\mathfrak{L}}$ we have the following chain of equalities

$$\begin{aligned}
\widehat{S_Y}(@(A_1, \dots, A_n)) &= \top \mathbin{\dot{\&}_i} \widehat{S_Y}(@(A_1, \dots, A_n)) \\
&= \sup\{\vartheta \mathbin{\dot{\&}_i} \widehat{S_Y}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\} \\
&= T_{\mathbb{P}}^{\mathfrak{L}}(S_Y)(A) \\
&= \sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J_{\bar{y}})(A) \mid J_{\bar{y}} \in X_Y\} \\
&= \sup\{\sup\{\vartheta \mathbin{\dot{\&}_i} J_{\bar{y}}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\} \mid J_{\bar{y}} \in X_Y\} \\
&= \sup\{\top \mathbin{\dot{\&}_i} \hat{J}_{\bar{y}}(@(A_1, \dots, A_n)) \mid J_{\bar{y}} \in X_Y\} \\
&= \sup\{\dot{@}(J_{\bar{y}}(A_1), \dots, J_{\bar{y}}(A_n)) \mid J_{\bar{y}} \in X_Y\} \\
&= \sup\{\dot{@}(y_1, \dots, y_n) \mid (y_1, \dots, y_n) \in Y\}
\end{aligned}$$

Finally, by Eqns. (2) and (1) and this result we have

$$\sup\{\dot{@}(y_1, \dots, y_n)\} \preceq \dot{@}(\sup Y) \preceq \widehat{S_Y}(@(A_1, \dots, A_n)) = \sup\{\dot{@}(y_1, \dots, y_n)\}$$

□

Another Approach to the Continuity of $T_{\mathbb{P}}^{\mathfrak{L}}$

It is possible to generalise the previous theorem by requiring weaker continuity conditions on the operators but, at the same time, restricting the structure of the set of truth-values.

Definition 18. *Let L be a poset and $f\colon L^n \to L$ a function. We say that f is* lower-semicontinuous*, for short LSC, in $(\vartheta_1, \dots, \vartheta_n) \in L^n$ if for all $\varepsilon < f(\vartheta_1, \dots, \vartheta_n)$ there exist δ_i for $i = 1, \dots, n$ such that whenever $(\mu_1 \dots, \mu_n)$ satisfies $\delta_i < \mu_i \leq \vartheta_i$ then $\varepsilon < f(\mu_1, \dots, \mu_n) \leq f(\vartheta_1, \dots, \vartheta_n)$.*

A function f is said to be lower-semicontinuous (or LSC) *if it is lower-semicontinuous in every point in its domain.*

It is obvious that the composition of two lower-semicontinuous functions is also lower-semicontinuous.

Definition 19. *A cpo L is said to satisfy the* supremum property *if for all set $X \subset L$ and for all ε we have that if $\varepsilon < \sup X$ then there exists $\delta \in X$ such that $\varepsilon < \delta \leq \sup X$.*

Lemma 1 also holds assuming LSC and the supremum property and, therefore, the continuity of the $T_{\mathbb{P}}^{\mathfrak{L}}$ operator is obtained from the combined hypotheses of LSC of the operators and the supremum property of the lattice of truth-values.

Lemma 2. *Let $\mathbb{P}$ be a program interpreted on a multi-adjoint Ω-algebra $\mathfrak{L}$ whose carrier has the supremum property for directed sets. Let $\mathcal{B}$ be any body formula in $\mathbb{P}$, and a assume that all the operators in $\mathcal{B}$ are LSC. Let X be a directed set of interpretations, and write $S = \sup X$; then*

$$\hat{S}(\mathcal{B}) = \sup\{\hat{J}(\mathcal{B}) \mid J \in X\}$$

Proof sketch: The following inequality is straightforward.

$$\sup\{\hat{J}(\mathcal{B}) \mid J \in X\} \preceq \hat{S}(\mathcal{B})$$

Now, assume the strict inequality and get a contradiction, using LSC and the supremum property separately on each argument to obtain elements $J_i(\mathcal{B})$, then apply directedness to get an uniform interpretation $J_0(\mathcal{B})$, finally use once again LSC to get a contradiction. □

Theorem 6. *If L satisfies the supremum property, and all the operators in the body are LSC and $\dot{\&}_i$ are LSC in their second argument, then the operator $T_{\mathbb{P}}^{\mathfrak{L}}$ is continuous.*

Proof: Let us prove that for a directed set X and $S = \sup X$ we have that

$$T_{\mathbb{P}}^{\mathfrak{L}}(S)(A) = \sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\}$$

by showing that $T_{\mathbb{P}}^{\mathfrak{L}}(S)(A)$ fulfils the properties of a supremum for the set $\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\}$.

1. Clearly, by monotonicity of the operator $T_{\mathbb{P}}^{\mathfrak{L}}$ and the fact that $S = \sup X$, we have that $T_{\mathbb{P}}^{\mathfrak{L}}(S)(A)$ is an upper bound for all the $T_{\mathbb{P}}^{\mathfrak{L}}(J)(A)$ with $J \in X$ and, therefore
$$\sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\} \preceq T_{\mathbb{P}}^{\mathfrak{L}}(S)(A)$$
2. Reasoning by contradiction, assume the strict inequality
$$\sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\} \prec T_{\mathbb{P}}^{\mathfrak{L}}(S)(A)$$
As $T_{\mathbb{P}}^{\mathfrak{L}}(S)(A) = \sup\{\vartheta \,\dot{\&}_i\, \hat{S}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}\}$ by the supremum property taking $\varepsilon = \sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\}$ we have that there exist a rule $A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P}$ such that
$$\sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\} = \varepsilon \prec \vartheta \,\dot{\&}_i\, \hat{S}(\mathcal{B}) \preceq T_{\mathbb{P}}^{\mathfrak{L}}(S)(A)$$
By using lower-semicontinuity of $\vartheta \,\dot{\&}_i\, \_$ on the strict inequality, we have that there exists $\delta \prec \hat{S}(\mathcal{B})$ such that whenever $\delta \prec \lambda \preceq \hat{S}(\mathcal{B})$ then $\varepsilon \prec \vartheta \,\dot{\&}_i\, \lambda \preceq \vartheta \,\dot{\&}_i\, \hat{S}(\mathcal{B})$.
Now, by Lemma 2, we have that $\hat{S}(\mathcal{B}) = \sup\{\hat{J}(\mathcal{B}) \mid J \in X\}$, we can apply once again the supremum property and select an element $J_0 \in X$ such that $\delta \prec \hat{J}_0(\mathcal{B}) \preceq \hat{S}(\mathcal{B})$. For this element, by LSC of $\vartheta \,\dot{\&}_i\, \_$ we have that
$$\varepsilon \prec \vartheta \,\dot{\&}_i\, \hat{J}_0(\mathcal{B}) \preceq \vartheta \,\dot{\&}_i\, \hat{S}(\mathcal{B})$$
But this is contradictory with the fact that $\varepsilon = \sup\{T_{\mathbb{P}}^{\mathfrak{L}}(J)(A) \mid J \in X\} = \sup\{\vartheta \,\dot{\&}_i\, \hat{J}(\mathcal{B}) \mid A \overset{\vartheta}{\longleftarrow}_i \mathcal{B} \in \mathbb{P} \text{ and } J \in X\}$.

□

6 Conclusions and future work

We have presented a general theory of logic programming which allows the simultaneous use of different implications in the rules and rather general connectives in the bodies.

We have shown that models of our programs are post-fixpoints of the immediate consequences operator $T_{\mathbb{P}}^{\mathfrak{L}}$, and the it is monotonic under very general hypotheses. In addition we have proved the continuity of $T_{\mathbb{P}}^{\mathfrak{L}}$ under the assumption of continuity of the operators in the language (but, possibly, the implications). This hypothesis of continuity of the operators can be relaxed to lower-semicontinuity, whenever we are working with a lattice with the supremum property. As future work we are planning to develop a complete procedural semantics for multi-adjoint programs and further investigate lattice with the supremum property.

Acknowledgements

We thank C. Damásio and L. Moniz Pereira for communicating the existence of first drafts of their papers, on which this research began.

References

1. J.F. Baldwin, T.P. Martin, and B.W. Pilsworth. *FRIL-Fuzzy and Evidential Reasoning in AI.* Research Studies Press (John Wiley), 1995.
2. C.V. Damásio and L. Moniz Pereira. Hybrid probabilistic logic programs as residuated logic programs. In *Logics in Artificial Intelligence, JELIA'00*, pages 57–73. Lect. Notes in Artificial Intelligence, 1919, Springer-Verlag, 2000.
3. C.V. Damásio and L. Moniz Pereira. Monotonic and residuated logic programs. In *Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'01*, pages 748–759. Lect. Notes in Artificial Intelligence 2173, Springer-Verlag, 2001.
4. M. van Emden and R. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, 1976.
5. M.C. Fitting. Bilattices and the semantics of logic programming. *Journal of Logic Programming*, 11:91–116, 1991.
6. P. Hájek. *Metamathematics of Fuzzy Logic.* Trends in Logic. Studia Logica Library. Kluwer Academic Publishers, 1998.
7. E. Naito, J. Ozawa, I. Hayashi, and N. Wakami. A proposal of a fuzzy connective with learning function. In P. Bosc and J. Kaczprzyk, editors, *Fuzziness Database Management Systems*, pages 345–364. Physica Verlag, 1995.
8. J. Pavelka. On fuzzy logic I, II, III. *Zeitschr. f. Math. Logik und Grundl. der Math.*, 25, 1979.
9. P. Vojtáš. Fuzzy logic programming. *Fuzzy sets and systems*, 2001. Accepted.