Revisiting immediate consequences operator on first-order logic programming*

Jesús Medina and José Antonio Torné-Zambrano

Department of Mathematics. University of Cádiz. Spain
{jesus.medina, joseantonio.torne}@uca.es

Abstract. Logic programming is a fundamental paradigm based on formal logic with relevant features in the design of automatic intelligence systems. The computation of the consequences from a given dataset, modeled by logic rules, is given by the immediate consequences operator. This paper clarifies the definition of this operator in the first-order logic programming framework and remarks the consideration of two sets in the computation of the immediate consequences operator.

1 Introduction

Among all the generalizations of logic programming, it stands out the residuated and multi-adjoint approaches [4,14], which are able to capture different extensions developed for different purposes such as Generalized Annotated Logic Programs [11], Fuzzy Logic Programming [17], Hybrid Probabilistic Logic Programs [5], and Possibilistic Logic Programming [?]. In [14], a propositional approach considering the multi-adjoint philosophy was studied. Later, a first-order extension arose [13] as a generalization of the propositional one, which enables to deal with more complex sentences and reasonings. For example, it allows the use of a huge variety of different adjoint pairs in the construction of the rules of a logic program. Consequently, a much more flexible framework is achieved for the processing of information and the modeling of different scenarios.

The immediate consequences operator is essential to define the semantics of a logic program [12]. Indeed, it is basic for the fixed-point semantics, in which models of a program are characterized through the post-fixed points of this operator and the least model of the program can be obtained by iterating the immediate consequences operator an ordinal number of times, starting from the least interpretation.

The use of substitution mappings (or simply substitutions) is also a fundamental notion in first-order logic [7,?,16]. The definition of an arbitrary interpretation on general formulas considers substitutions; the matching of formulas also

* Partially supported by the project PID2019-108991GB-I00 funded by MICIU/AEI/10.13039/501100011033, the project PID2022-137620NB-I00 funded by MICIU/AEI/10.13039/501100011033 and FEDER, UE, by the grant TED2021-129748B-I00 funded by MCIN/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR, and by the research and transfer program of the University of Cádiz in project PR2022-057

needs substitutions (unifiers); and the definition of the immediate consequences operator on first-order logic [16] requires substitutions too, among others. However, the kind of these substitutions in this last definition and the proper definition of immediate consequences operator need diverse clarifications. This paper will prove that the substitutions cannot be arbitrary ones, but ground in the variables of the rules. Moreover, we will show that the definition of the immediate consequences operator is based on the computation of two suprema on two notable sets. These clarifications will be essential in future extensions and advances in residuated and multi-adjoint first-order logic programming approaches.

The structure of the paper is as follows. In Section 2 some preliminaries results about multi-adjoint algebra and first-order multi-adjoint logic programming will be recalled. Section 3 presents the main contribution of the paper with more clarified definitions of the immediate consequences operator, together with the characterization of the models as post-fixed points of the immediate consequence operator and an illustrative example. Finally, in Section 4 some conclusions and future work are given.

2 Preliminaries

In this section we will present the algebraic structure that we are going to use in the paper and we will recall the main notions of multi-adjoint logic programming.

2.1 Multi-adjoint algebra

The considered operators in this algebra are the adjoint pairs. Each adjoint pair is composed of a conjunctor and an implication, which is a generalization of a t-norm and its residuated implication. We will write the operators defined on an algebraic structure (poset or lattice) with a dot, and their symbols in a graded set without dot.

Definition 1 ([16]). *Let $\langle P, \preceq \rangle$ be a poset and $(\dot{\leftarrow}, \dot{\&})$ a pair of binary operators in P such that*

1. *The operator $\dot{\&}$ is increasing in both arguments.*
2. *The operator $\dot{\leftarrow}$ is increasing in the first argument (*consequent*) and decreasing in the second one (*antecedent*)*
3. *The adjoint property holds, i.e., $x \preceq (z \dot{\leftarrow} y)$ if and only if $(x \dot{\&} y) \preceq z$, for all $x, y, z \in P$.*

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

Now, the definitions of graded set and multi-adjoint lattice are recalled.

Definition 2 ([14]). *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 3 ([14]). *Let $\langle L, \preceq \rangle$ be a complete lattice. A* multi-adjoint lattice *$\mathcal{L}$ is a tuple $(L, \preceq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ satisfying*

1. *$\langle L, \preceq\rangle$ is bounded, i.e., it has a bottom $\bot$ and a top $\top$ elements.*
2. *$(\dot{\leftarrow}_i, \dot{\&}_i)$ is an adjoint pair in $\langle L, \preceq\rangle$, for all $i \in \{1, \ldots, n\}$.*
3. *$\top \dot{\&}_i \vartheta = \vartheta \dot{\&}_i \top = \vartheta$, for all $\vartheta \in L$ and $i \in \{1, \ldots, n\}$.*

Next, we recall the definition of multi-adjoint Ω-algebra, where the definition of Ω-algebra can be consulted in [14].

Definition 4 ([16]). *Let Ω be a graded set containing the operator symbols $\leftarrow_i$ and $\&_i$, for $i \in \{1, \ldots, n\}$, and possibly some extra operators, and let $\mathfrak{L}$ be an Ω-algebra on the complete lattice $\langle L, \preceq\rangle$. We say that $\mathfrak{L}$ is a* multi-adjoint Ω-algebra *with respect to the pairs $(\leftarrow_i, \&_i)$, for $i \in \{1, \ldots, n\}$, if $\mathcal{L} = (L, \preceq, \dot{\leftarrow}_1, \dot{\&}_1, \ldots, \dot{\leftarrow}_n, \dot{\&}_n)$ is a multi-adjoint lattice.*

Now, an example of multi-adjoint Ω-algebra is introduced.

Example 1. Let $\Omega = \left\{\leftarrow_{\mathrm{P}}, \&_{\mathrm{P}}, \leftarrow_{\mathrm{G}}, \&_{\mathrm{G}}, \wedge_{\mathrm{L}}, @_{(2,1)}\right\}$ be a graded set and $\mathfrak{L} = \langle [0,1], I\rangle$ an Ω-algebra, where $\dot{\&}_{\mathrm{P}}$ and $\dot{\&}_{\mathrm{G}}$ are the product and Gödel t-norms respectively, together with their residuated implications. In addition, we are considering the Łukasiewicz conjunctor $\dot{\wedge}_{\mathrm{L}}$ and the aggregator $\dot{@}_{(2,1)}$. It is easy to prove that $\mathcal{L} = ([0,1], \leq, \dot{\leftarrow}_{\mathrm{P}}, \dot{\&}_{\mathrm{P}}, \dot{\leftarrow}_{\mathrm{G}}, \dot{\&}_{\mathrm{G}})$ is a multi-adjoint lattice, and then $\mathfrak{L}$ is a multi-adjoint Ω-algebra. These operators are defined as

$$\dot{\&}_{\mathrm{G}}(x,y) = \min\{x,y\} \qquad z \dot{\leftarrow}_{\mathrm{G}} y = \begin{cases} 1 & \text{if } \ y \leq z \\ z & \text{otherwise} \end{cases}$$

$$\dot{\&}_{\mathrm{P}}(x,y) = x \cdot y \qquad z \dot{\leftarrow}_{\mathrm{P}} y = \begin{cases} 1 & \text{if } \ y \leq z \\ \frac{z}{y} & \text{otherwise} \end{cases}$$

$$\dot{\wedge}_{\mathrm{L}}(x,y) = \max\{0, x+y-1\} \qquad \dot{@}_{(2,1)}(x,y) = \frac{2x+y}{3}$$

for all x, y, $z \in [0,1]$. □

2.2 Multi-adjoint logic programming

First of all, we will present the syntax of multi-adjoint logic programming. Diverse relevant notions such as alphabet, language and well-formed formulas can be consulted in [12,16]. Next, in order to fix the notation and the starting point, we recall the definition of logic program, which is a set of rules and facts in a considered first-order extended language $\mathfrak{F}_e$, which is the corresponding Ω-algebra of formulas freely generated from the disjoint union of the set of the symbols from the first-order language and the lattice [9,16]. From now on, a graded set Ω containing conjunctor, disjunctor and aggregator symbols, a multi-adjoint lattice $\mathcal{L} = (L, \preceq, \dot{\leftarrow}_1, \dot{\&}_1, \ldots, \dot{\leftarrow}_n, \dot{\&}_n)$ and its Ω-algebra $\mathfrak{L}$ will be fixed.

Definition 5 ([16]). *A* multi-adjoint logic program *is a set $\mathbb{P}$ of weighted rules of the form $\langle H \leftarrow_i \mathcal{B}; \vartheta\rangle$, also denoted as $H \overset{\vartheta}{\leftarrow}_i \mathcal{B}$, such that:*

1. *The consequent of the implication, H, is an atom called the* head.
2. *The antecedent of the implication, $\mathcal{B}$, is called the* body*, and it is a formula built from atoms $B_1, \dots, B_n$, with $0 \le n$, by the use of conjunctors, disjunctors and aggregators.*
3. *The* confidence factor *ϑ is an element, a truth-value of L.*

A program is completed with particular observations in the form of facts*, which are rules with body $\top$. Moreover, free occurrences of variables in the program are considered to be universally quantified.*

Note that the $\top$ element in the body of a fact is considered a constant of the first-order language. Furthermore, the formula $\mathcal{B}$ of the body of a rule can be written from its more basic elements as $@\,[B_1, \dots, B_n]$, where $@$ is the aggregation of the symbols of operators appearing in $\mathcal{B}$, and $B_1, \dots, B_n$ are the atoms of $\mathcal{B}$. Next, we present an example of logic program, which will be used later in the next section.

Example 2. Consider the multi-adjoint Ω-algebra $\mathfrak{L}$ introduced in Example 1, where each operator symbol ω in the graded set Ω has been associated with its respective interpretation $\dot{\omega}$, with the aim of simplifying the notation. With the algebraic structure fixed, we can consider the following program $\mathbb{P}$.

$$\begin{aligned}
\mathcal{R}_1\colon &\quad p(a,Y) \overset{0.8}{\longleftarrow}_{\mathrm{G}} \quad @_{(2,1)}\,(q(X)\&_{\mathrm{G}} r(Y), s(X,b))\\
\mathcal{R}_2\colon &\quad p(X,b) \overset{0.9}{\longleftarrow}_{\mathrm{P}} \quad t(X) \wedge_{\mathrm{L}} r(Y)\\
\mathcal{R}_3\colon &\quad q(X) \overset{0.2}{\longleftarrow}_{\mathrm{P}} \quad t(X)
\end{aligned}$$

In each particular case, specific values are "measured" to some variables. These variables are called *hypotheses* and are formally expressed as facts. For example, we can consider the following facts associated with the ground atoms $r(b)$, $s(a,b)$ and $t(a)$,

$$\begin{aligned}
\mathcal{R}_4\colon &\quad r(b) \overset{0.6}{\longleftarrow} \quad \top\\
\mathcal{R}_5\colon &\quad s(a,b) \overset{0.8}{\longleftarrow} \quad \top\\
\mathcal{R}_6\colon &\quad t(a) \overset{0.6}{\longleftarrow} \quad \top
\end{aligned}$$

The set composed of the previous rules $\mathcal{R}_1, \dots, \mathcal{R}_6$ is an example of multi-adjoint logic program. □

Now, the semantics of first-order logic programming will be recalled. The following notion provides a meaning (representative truth-value) to syntactic formulas of the logic framework.

Definition 6 ([16]). *Let $\mathbb{P}$ be a multi-adjoint logic program. An* interpretation *is a mapping from the set of ground atoms to the lattice of truth-values $\langle L, \preceq \rangle$. The set of all interpretations of the formulas defined by the Ω-algebra $\mathfrak{F}$ in the Ω-algebra $\mathfrak{L}$ is denoted as $\mathcal{I}_{\mathfrak{L}}$.*

Although interpretations are defined for ground atoms, thanks to the homomorphic extension theorem [14], we can extend the definition of an interpretation I for all ground formulas of the language, what we will denote as $\hat{I}$. Besides, taking into account that all formulas are universally closed, we can extend the definition of an interpretation I for a non-ground formula A as follows.

$$\hat{I}(A) = \inf\{\hat{I}(A\xi) \mid A\xi \text{ is a ground instantiation of } A\}$$

The main notion of the semantics in logic programming is the definition of model.

Definition 7 ([16]). *Given an interpretation $I \in \mathcal{I}_{\mathfrak{L}}$, a weighted rule $\langle A \leftarrow_i \mathcal{B}, \vartheta\rangle$ is* satisfied *by I if and only if $\vartheta \preceq \hat{I}\,(A \leftarrow_i \mathcal{B})$. An interpretation $I \in \mathcal{I}_{\mathfrak{L}}$ is a* model *of a multi-adjoint logic program $\mathbb{P}$ if and only if all weighted rules in $\mathbb{P}$ are satisfied by I.*

3 Analyzing immediate consequences operator

Fixed-point semantics was studied in the Boolean and fuzzy settings [12,16] in order to compute the least model of a logic program, which provides the most relevant information that it is possible to extract from the program, and can be used for instance in automatic recommendation systems.

This special semantics is based on the immediate consequences operator. This operator aggregates, for every ground atom A, the information given by each rule with head H satisfying that A and H unify. Notice that, given a ground substitution ξ and a rule $H \leftarrow_i \mathcal{B}$, we obtain the following chain of equalities

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

As a consequence, the operator $T_{\mathbb{P}}\colon \mathcal{I}_{\mathfrak{L}} \to \mathcal{I}_{\mathfrak{L}}$ defined as

$$T_{\mathbb{P}}(I)(A) = \sup\left\{\vartheta\dot{\&}_i\hat{I}(\mathcal{B}\theta) \mid \langle C \leftarrow_i \mathcal{B}, \vartheta\rangle \in \mathbb{P} \text{ and } A = C\theta\right\} \tag{1}$$

for all interpretation I and ground atom A, was introduced in [16], as a generalization of the Boolean immediate consequences operator given by van Emden and Kowalski in [6].

However, in Expression (1) it is unclear whether the substitution θ should be ground in some sense or not. The following example shows that an arbitrary non-ground substitution cannot be considered in the previous expression, since in this case the models are not characterized by the fixed-points of $T_{\mathbb{P}}$, as it happens in the classical case.

Example 3. Consider the program $\mathbb{P}$ introduced in Example 2 and the interpretation I defined on the ground atoms as

$$\begin{array}{rrr}
I(r(a)) = 0.1 & I(r(b)) = 0.6 & I(s(a,b)) = 0.8 \\
I(s(b,b)) = 0.1 & I(t(a)) = 0.6 & I(t(b)) = 0.1 \\
I(q(a)) = 0.12 & I(q(b)) = 0.02 & I(p(a,a)) = 0.1 \\
I(p(b,b)) = 0.1 & I(p(a,b)) = 0.1 &
\end{array}$$

Furthermore, we can extend this definition for non-ground atoms, following Definition 6, as follows.

$$\hat{I}(q(X)) = \inf\{I(q(a)), I(q(b))\} = \inf\{0.12, 0.02\} = 0.02$$
$$\hat{I}(s(X,b)) = \inf\{I(s(a,b)), I(s(b,b))\} = \inf\{0.8, 0.1\} = 0.1$$
$$\hat{I}(r(Y)) = \inf\{I(r(a)), I(r(b))\} = \inf\{0.1, 0.6\} = 0.1$$

First of all, we will see that I satisfies that $T_{\mathbb{P}}(I) \sqsubseteq I$. We will start proving it for $p(a,b)$. For this, we need to compute the value $T_{\mathbb{P}}(I)(p(a,b))$. According to Expression (1), we consider the substitutions $\theta_1 = \{X/X', Y/b\}$[1] applied to $\mathcal{R}_1$; and $\theta_2 = \{X/a, Y/Y'\}$ applied to the second rule $\mathcal{R}_2$.

$$\begin{aligned}
T_{\mathbb{P}}(I)(p(a,b)) &= \sup\Big\{0.8 \dot{\&}_{\mathrm{G}} \hat{I}(\mathcal{B}_1\theta_1), 0.9 \dot{\&}_{\mathrm{P}} \hat{I}(\mathcal{B}_2\theta_2)\Big\} \\
&= \sup\Big\{0.8 \dot{\&}_{\mathrm{G}} \dot{@}_{(2,1)}\Big(\hat{I}\left(q\left(X'\right)\right) \dot{\&}_{\mathrm{G}} I\left(r\left(b\right)\right), \hat{I}\left(s\left(X', b\right)\right)\Big), \\
&\qquad 0.9 \dot{\&}_{\mathrm{P}}\Big(I\left(t\left(a\right)\right) \dot{\wedge}_{\mathrm{L}} \hat{I}\left(r\left(Y'\right)\right)\Big)\Big\} \\
&= \sup\Big\{0.8 \dot{\&}_{\mathrm{G}} \dot{@}_{(2,1)}(0.02 \dot{\&}_{\mathrm{G}} 0.6, 0.1), 0.9 \dot{\&}_{\mathrm{P}}\left(0.6 \dot{\wedge}_{\mathrm{L}} 0.1\right)\Big\} \\
&= \sup\{0.047, 0\} \\
&\leq 0.1 = I(p(a,b))
\end{aligned}$$

It is easy to prove the inequality $T_{\mathbb{P}}(I) \sqsubseteq I$ for the rest of the ground atoms, as we can see below.

$$\begin{array}{cc}
T_{\mathbb{P}}(I)(r(a)) = 0 \leq I(r(a)) & T_{\mathbb{P}}(I)(r(b)) = 0.6 \leq I(r(b)) \\
T_{\mathbb{P}}(I)(s(a,b)) = 0.8 \leq I(s(a,b)) & T_{\mathbb{P}}(I)(s(b,b)) = 0 \leq I(s(b,b)) \\
T_{\mathbb{P}}(I)(t(a)) = 0.6 \leq I(t(a)) & T_{\mathbb{P}}(I)(t(b)) = 0 \leq I(t(b)) \\
T_{\mathbb{P}}(I)(q(a)) = 0.12 \leq I(q(a)) & T_{\mathbb{P}}(I)(q(b)) = 0.02 \leq I(q(b)) \\
T_{\mathbb{P}}(I)(p(a,a)) = 0.046 \leq I(p(a,a)) & T_{\mathbb{P}}(I)(p(b,b)) = 0 \leq I(p(b,b))
\end{array}$$

Now, we are going to see that I is not a model for $\mathbb{P}$. Following Definition 2, we have that I is a model for $\mathbb{P}$ if, for every rule $\langle A \leftarrow_i \mathcal{B}_i\rangle \in \mathbb{P}$, the following inequality holds.

$$\vartheta \preceq \hat{I}(A \leftarrow_i \mathcal{B}_i) = \inf\{\hat{I}((A \leftarrow_i \mathcal{B}_i)\xi) \mid \xi \text{ provides a ground instantiation}\} \quad (2)$$

Notice that Expression (2) must be satisfied for every substitution ξ satisfying that $(A \leftarrow_i \mathcal{B}_i)\xi$ is a ground instantiation and, in particular, it should be satisfied if we consider $\xi = \{X/a, Y/b\}$ and the rule $\mathcal{R}_2 = \langle p(X,b) \leftarrow_{\mathrm{P}} t(X) \wedge_{\mathrm{L}} r(Y); 0.9\rangle$.

[1] Notice that the substitution θ_1 interchanges in the formula the variable X by the variable X' and Y by b.

Nevertheless, the following chain of inequalities arises.

$$\begin{aligned}\hat{I}\left((p(X,b) \leftarrow_{\mathrm{P}} t(X) \wedge_{\mathrm{L}} r(Y))\xi\right) &= \hat{I}((p(a,b) \leftarrow_{\mathrm{P}} t(a) \wedge_{\mathrm{L}} r(b))) \\ &= I(p(a,b)) \dot{\leftarrow}_{\mathrm{P}} (I(t(a)) \dot{\wedge}_{\mathrm{L}} I(r(b))) \\ &= 0.1 \dot{\leftarrow}_{\mathrm{P}} (0.6 \dot{\wedge}_{\mathrm{L}} 0.6) \\ &= 0.5 \\ &\not\geq 0.9\end{aligned}$$

Therefore, Expression (2) does not hold and we have found an interpretation I, such that $T_{\mathbb{P}}(I) \sqsubseteq I$, but I is not a model for $\mathbb{P}$.

Thus, we have obtained that the characterization of models based on the post-fixed points of the immediate consequences operator does not hold. □

As a consequence, it is clear that the substitution θ must provide a ground instantiation of the rule in Expression (1). The following definition clarifies this fact in the definition of immediate consequences operator given in [16].

Definition 8. *The* immediate consequences operator $T_{\mathbb{P}}$ *maps interpretations to interpretations and it is defined in such a way that, given an interpretation I and a ground atom A, $T_{\mathbb{P}}(I)(A)$ is computed as*

$$\sup\left\{\vartheta \dot{\&}_i \hat{I}(\mathcal{B}\xi) \mid \langle C \leftarrow_i \mathcal{B}, \vartheta\rangle \in \mathbb{P}, A = C\xi,\ \textit{and } (C \leftarrow_i \mathcal{B})\xi \textit{ is ground}\right\}$$

The set of variables in a rule $\mathcal{R}$ will be denoted as $\mathcal{V}(\mathcal{R})$. Hence, the last property in the definition of $T_{\mathbb{P}}(I)(A)$ can be written as: "ξ is a substitution ground in $\mathcal{V}(\langle C \leftarrow_i \mathcal{B}, \vartheta\rangle)$". The following result provides a different point of view of the definition of the $T_{\mathbb{P}}$ operator. We need to remark that Definition 8 really computes two suprema, one on the substitutions and the other one on the rules of the program. Computationally, it does not change, but it is interesting for studying more properties and future generalizations of the operator.

Theorem 1. *Let $\mathbb{P}$ be a program, an interpretation I and a ground atom A. We obtain that*

$$T_{\mathbb{P}}(I)(A) = \sup\left\{S_I(\vartheta \&_i \mathcal{B}\theta) \mid \langle C \leftarrow_i \mathcal{B}, \vartheta\rangle \in \mathbb{P},\ \textit{and } \theta = \textit{mgu}\{A, C\}\right\}$$

where the mapping $S_I \colon \mathfrak{F}_e \to L$ is defined for every $\phi \in \mathfrak{F}_e$ as $S_I(\phi) = \sup\{\hat{I}(\phi\xi) \mid \xi$ is a ground substitution$\}$.

Proof. Given a ground atom A and an interpretation I, for each rule $\mathcal{R} = \langle C \leftarrow_i \mathcal{B}, \vartheta\rangle \in \mathbb{P}$, such that there exists $\theta = \mathrm{mgu}\{A, C\}$, we must compute in Definition 8 the value $\sup\left\{\vartheta \dot{\&}_i \hat{I}(\mathcal{B}\theta\xi) \mid \xi \text{ is a ground substitution}\right\}$, which is given by the mapping $S_I \colon \mathfrak{F} \to L$. Finally, the computation of $T_{\mathbb{P}}(I)(A)$ arises as the supremum of this value, $S_I(\vartheta \&_i \mathcal{B}\theta)$, for all such rules $\mathcal{R} = \langle C \leftarrow_i \mathcal{B}, \vartheta\rangle \in \mathbb{P}$, satisfying that there exists $\theta = \mathrm{mgu}\{A, C\}$. □

Notice that it is not necessary for θ to be a ground substitution. This last definition is taken into consideration to assert that this operator just allows the characterization of models of a given program $\mathbb{P}$ through the post-fixed points of the immediate consequences operator, which also clarifies the proof given in [16].

Theorem 2. *An interpretation I is a model of a multi-adjoint logic program $\mathbb{P}$ if and only if $T_{\mathbb{P}}(I) \sqsubseteq I$.*

Proof. Given a model I for $\mathbb{P}$ and a ground atom A, we will prove that $T_{\mathbb{P}}(I) \sqsubseteq I$. On the one hand, if there is no rule in $\mathbb{P}$ matching with A, then $T_{\mathbb{P}}(I)(A) = \sup \varnothing = \bot \preceq I(A)$, and the result is trivially obtained.

On the other hand, consider an arbitrary rule $\mathcal{R} = \langle C \leftarrow_i \mathcal{B}, \vartheta \rangle$ of $\mathbb{P}$, such as there exists $\theta = \mathrm{mgu}\{A, C\}$. Moreover, given a ground substitution ξ, we have that $\theta\xi$ is also ground and, by hypothesis, the following chain of inequalities holds, where σ denotes a substitution.

$$\begin{aligned}
\vartheta &\overset{(*)}{\preceq} \hat{I}\,(C \leftarrow_i \mathcal{B}) \\
&= \inf\{\hat{I}\,((C \leftarrow_i \mathcal{B})\sigma) \mid (C \leftarrow_i \mathcal{B})\sigma \text{ is a ground instantiation of } C \leftarrow_i \mathcal{B}\} \\
&\preceq \hat{I}\,((C \leftarrow_i \mathcal{B})\theta\xi) \\
&= \hat{I}\,(C\theta \leftarrow_i \mathcal{B}\theta\xi) \\
&= \hat{I}(C\theta) \dot{\leftarrow}_i \hat{I}(\mathcal{B}\theta\xi)
\end{aligned}$$

where $(*)$ holds due to the fact that I is a model for $\mathbb{P}$. Applying the adjoint property, we can write

$$\hat{I}((\vartheta \&_i \mathcal{B}\theta)\xi) = \vartheta \dot{\&}_i \hat{I}(\mathcal{B}\theta\xi) \preceq \hat{I}(C\theta\xi) = I(A\xi) = I(A)$$

where the last equality holds because the atom A is ground. Therefore, taking suprema on the set of ground substitutions, we reach

$$S_I(\vartheta \&_i \mathcal{B}\theta) = \sup\{\hat{I}((\vartheta \&_i \mathcal{B}\theta)\xi) \mid \xi \text{ is a ground substitution}\} \preceq I(A)$$

Now, taking suprema on the set of rules such that there exists $\theta = \mathrm{mgu}\{A, C\}$, the following inequality arises.

$$\begin{aligned}
T_{\mathbb{P}}(I)(A) &= \sup\{S_I(\vartheta \&_i \mathcal{B}\theta) \mid \langle C \leftarrow_i \mathcal{B}, \vartheta \rangle \in \mathbb{P}, \text{ and } \theta = \mathrm{mgu}\{A, C\}\} \\
&\preceq I(A)
\end{aligned}$$

Consequently, the inequality $T_{\mathbb{P}}(I) \sqsubseteq I$ holds. Now, we consider an interpretation I satisfying that $T_{\mathbb{P}}(I) \sqsubseteq I$, and we will see that I is a model for $\mathbb{P}$. For this, we consider an arbitrary weighted rule $\mathcal{R} = \langle C \leftarrow_i \mathcal{B}, \vartheta \rangle$ in $\mathbb{P}$ and we need to prove that

$$\begin{aligned}
\vartheta &\preceq \hat{I}(C \leftarrow_i \mathcal{B}) \\
&= \inf\{\hat{I}((C \leftarrow_i \mathcal{B})\sigma) \mid (C \leftarrow_i \mathcal{B})\sigma \text{ is a ground instantiation of } C \leftarrow_i \mathcal{B}\}
\end{aligned}$$

that is, $\vartheta \preceq \hat{I}((C \leftarrow_i \mathcal{B})\sigma)$ for all substitution σ such that $(C \leftarrow_i \mathcal{B})\sigma$ is a ground instantiation of $C \leftarrow_i \mathcal{B}$.

Hence, given one of these substitutions σ and a ground substitution ξ, applying that $T_{\mathbb{P}}(I) \sqsubseteq I$, we have that

$$\vartheta \dot{\&}_i \hat{I}(\mathcal{B}\sigma) = \vartheta \dot{\&}_i \hat{I}(\mathcal{B}\sigma\xi) = \hat{I}((\vartheta \&_i \mathcal{B}\sigma)\xi) \preceq S_I(\vartheta \&_i \mathcal{B}\sigma) \overset{(*)}{\preceq} T_{\mathbb{P}}(I)(C\sigma) \preceq I(C\sigma)$$

where $(*)$ follows from the definition of $T_{\mathbb{P}}$. Now, by the adjoint property, we obtain $\vartheta \preceq I(C\sigma) \dot{\leftarrow}_i \hat{I}(\mathcal{B}\sigma) = \hat{I}((C \leftarrow_i \mathcal{B})\sigma)$ and finally, taking infima on the substitutions σ, we obtain the required inequality. $\square$

Notice that we cannot ensure the following expressions are equal:

$$\sup\{\vartheta \dot{\&}_i \hat{I}(\mathcal{B}\theta\xi) \mid \xi \text{ is a ground substitution}\} \tag{3}$$
$$\vartheta \dot{\&}_i \sup\{\hat{I}(\mathcal{B}\theta\xi) \mid \xi \text{ is a ground substitution}\} \tag{4}$$

because of the operator $\dot{\&}_i$ could not preserve the supremum in the second argument (see for example [1,2,3]). This fact caused the need to consider the extended language $\mathfrak{F}_e$, including the truth-values in the language, due to $\hat{I}(\mathcal{B}\xi)$ is an element of L. With the following example, we will show how the value of the immediate consequences operator is obtained according to Theorem 1.

Example 4. Consider Example 2 and the interpretation J given by

$J(r(a)) = 0$	$J(r(b)) = 0.6$	$J(s(a,b)) = 0.8$
$J(s(b,b)) = 1$	$J(t(a)) = 0.6$	$J(t(b)) = 0$
$J(q(a)) = 1$	$J(q(b)) = 1$	

We are going to compute the value $T_{\mathbb{P}}(J)(p(a,b))$ considering the substitutions $\theta_1 = \{X/X', Y/b\}$ and $\theta_2 = \{X/a, Y/Y'\}$, which are ground on the variables of rules $\mathcal{R}_1$ and $\mathcal{R}_2$.

$$\begin{aligned}
T_{\mathbb{P}}(J)(p(a,b)) &= \sup\{S_J(0.8\&_{\mathrm{G}}\mathcal{B}_1\theta_1), S_J(0.9\&_{\mathrm{P}}(\mathcal{B}_2\theta_2))\} \\
&= \sup\big\{S_J\big(0.8\&_{\mathrm{G}}\big(@_{(2,1)}(q(X)\&_{\mathrm{G}}r(Y), s(X,b))\,\theta_1\big)\big), \\
&\qquad S_J(0.9\&_{\mathrm{P}}((t(X) \wedge_{\mathrm{L}} r(Y))\,\theta_2))\} \\
&= \sup\big\{S_J\big(0.8\&_{\mathrm{G}}@_{(2,1)}(q(X')\&_{\mathrm{G}}r(b), s(X',b))\big), \\
&\qquad S_J(0.9\&_{\mathrm{P}}(t(a) \wedge_{\mathrm{L}} r(Y')))\}
\end{aligned}$$

$$
\begin{aligned}
&= \sup\Big\{\sup\Big\{0.8\dot{\&}_{\mathrm{G}}\hat{J}\left(@_{(2,1)}\left(q(a)\&_{\mathrm{G}} r(b), s(a,b)\right)\right),\\
&\qquad 0.8\dot{\&}_{\mathrm{G}}\hat{J}\left(@_{(2,1)}\left(q(b)\&_{\mathrm{G}} r(b), s(b,b)\right)\right)\Big\},\\
&\qquad \sup\Big\{0.9\dot{\&}_{\mathrm{P}}\hat{J}\left(t(a)\wedge_{\mathrm{L}} r(a)\right), 0.9\dot{\&}_{\mathrm{P}}\hat{J}\left(t(a)\wedge_{\mathrm{L}} r(b)\right)\Big\}\Big\}\\
&= \sup\Big\{\sup\Big\{0.8\dot{\&}_{\mathrm{G}}\dot{@}_{(2,1)}\Big(J(q(a))\dot{\&}_{\mathrm{G}}J(r(b)), J(s(a,b))\Big),\\
&\qquad 0.8\dot{\&}_{\mathrm{G}}\dot{@}_{(2,1)}\Big(J(q(b))\dot{\&}_{\mathrm{G}}J(r(b)), J(s(b,b))\Big)\Big\},\\
&\qquad \sup\Big\{0.9\dot{\&}_{\mathrm{P}}\left(J(t(a))\dot{\wedge}_{\mathrm{L}}J(r(a))\right), 0.9\dot{\&}_{\mathrm{P}}\left(J(t(a))\dot{\wedge}_{\mathrm{L}}J(r(b))\right)\Big\}\Big\}\\
&= \sup\Big\{\sup\Big\{0.8\dot{\&}_{\mathrm{G}}\dot{@}_{(2,1)}\Big(1\dot{\&}_{\mathrm{G}}0.6, 0.8\Big), 0.8\dot{\&}_{\mathrm{G}}\dot{@}_{(2,1)}\Big(1\dot{\&}_{\mathrm{G}}0.6, 1\Big)\Big\},\\
&\qquad \sup\Big\{0.9\dot{\&}_{\mathrm{P}}\left(0.6\dot{\wedge}_{\mathrm{L}}0\right), 0.9\dot{\&}_{\mathrm{P}}\left(0.6\dot{\wedge}_{\mathrm{L}}0.6\right)\Big\}\Big\}\\
&= \sup\left\{\sup\left\{0.67, 0.73\right\}, \sup\left\{0, 0.18\right\}\right\}\\
&= 0.73
\end{aligned}
$$

□

4 Conclusions and future work

In this paper, we have revised the notion of the immediate consequences operator given in [16] and we have clarified the use of ground substitutions in its definition. This clarification ensures the characterization of the models for a program $\mathbb{P}$ as the post-fixed points of the immediate consequences operator, i.e., $T_{\mathbb{P}}(I) \sqsubseteq I$. Furthermore, we have remarked the use of two suprema in the definition of the immediate consequences operator, which will be fundamental in future extensions and studies of this operator. For example, a procedural semantics [15] based on reductants [8,9,10] will be analyzed and the approaches based on generalized quantifiers [**?**,**?**] and ordered weighted averaging (OWA) operators [**?**] will be studied in the first-order setting. Moreover, the comparison with recent approaches, such as fuzzy order-sorted feature logic [**?**], will be discussed.

References

1. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. A comparative study of adjoint triples. *Fuzzy Sets and Systems*, 211:1–14, 2013.
2. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. Multi-adjoint algebras versus non-commutative residuated structures. *International Journal of Approximate Reasoning*, 66:119–138, 2015.
3. M. E. Cornejo, J. Medina, and E. Ramírez-Poussa. Adjoint negations, more than residuated negations. *Information Sciences*, 345:355 – 371, 2016.
4. 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. Lecture Notes in Artificial Intelligence, 2143, 2001.

5. A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. *Journal of Logic Programming*, 43:187–250, 2000.
6. M. v. Emden and R. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, 1976.
7. F. Formato, G. Gerla, and M. Sessa. Similarity-based unification. *Fundamenta Informaticae*, 41(4):393–414, 2000.
8. P. Julián, G. Moreno, and J. Penabad. An improved reductant calculus using fuzzy partial evaluation techniques. *Fuzzy Sets and Systems*, 160:162–181, 2009.
9. P. Julián-Iranzo, J. Medina, and M. Ojeda-Aciego. On reductants in the framework of multi-adjoint logic programming. *Fuzzy Sets and Systems*, 317:27 – 43, 2017.
10. P. Julián-Iranzo, J. Medina, and M. Ojeda-Aciego. Revisiting reductants in the multi-adjoint logic programming framework. *Lecture Notes in Artificial Intelligence*, 8761:694–702, 2014.
11. M. Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *Journal of Logic Programming*, 12:335–367, 1992.
12. J. Lloyd. *Foundations of Logic Programming.* Springer Verlag, 1987.
13. J. Medina and M. Ojeda-Aciego. On first-order multi-adjoint logic programming. In *XI Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2002; León (España)*, 2002.
14. J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. In *Logic Programming and Non-Monotonic Reasoning, LPNMR'01*, pages 351–364. Lecture Notes in Artificial Intelligence 2173, 2001.
15. J. Medina, M. Ojeda-Aciego, and P. Vojtáš. A procedural semantics for multi-adjoint logic programming. In *Progress in Artificial Intelligence, EPIA'01*, pages 290–297. Lecture Notes in Artificial Intelligence 2258, 2001.
16. J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Similarity-based unification: a multi-adjoint approach. *Fuzzy Sets and Systems*, 146:43–62, 2004.
17. P. Vojtáš. Fuzzy logic programming. *Fuzzy sets and systems*, 124(3):361–370, 2001.