

Extended multi-adjoint logic programming

M. Eugenia Cornejo, David Lobo, Jesús Medina

Department of Mathematics, University of Cádiz. Spain
{mariaeugenia.cornejo,david.lobo,jesus.medina}@uca.es

Abstract

Extended multi-adjoint logic programming arises as an extension of multi-adjoint normal logic programming where constraints and a special type of aggregator operator have been included. The use of this general aggregator operator permits to consider, for example, different negation operators in the body of the rules of a logic program. We have introduced the syntax and the semantics of this new paradigm, as well as an interesting mechanism for obtaining a multi-adjoint normal logic program from an extended multi-adjoint logic program. This mechanism will allow us to establish technical properties relating the different stable models of both logic programming frameworks. Moreover, it makes possible that the already developed and future theory associated with stable models of multi-adjoint normal logic programs can be applied to extended multi-adjoint logic programs.

Keywords: multi-adjoint logic programming, adjoint triples, negation operator, stable models.

1. Introduction

Multi-adjoint logic programming is an interesting logical theory introduced by Medina et al. [20] which has gained a lot of popularity [2, 6, 12, 13, 19, 22, 23, 25]. This logical theory arises as a generalization of different non-classical logic programming approaches [9, 27], removing their particular details and preserving only the minimal mathematical requirements in order to guarantee the operability. Specifically, a multi-adjoint logic program is defined from an algebraic structure composed of a lattice together with different conjunctors and implications making up adjoint pairs. Notice that the conjunctors do not need to be neither commutative nor associative.

The multi-adjoint logic programming framework has been recently extended by considering a negation operator in the logic programs, which has

given rise to a new logic programming approach called multi-adjoint normal logic programming [4, 6, 19]. One of the most interesting properties of multi-adjoint normal logic programs is associated with the existence of stable models, which allows us to check whether the logic program is related to a solvable problem. Note that, when multi-adjoint normal logic programs correspond to some search problem, the stable models are actually its possible solutions. A detailed study on the syntax and semantics of multi-adjoint normal logic programs, including important results about the existence and the unicity of stable models, was introduced in [6]. Now, we are interested in extending the multi-adjoint logic normal programming framework in order to increase its flexibility and the range of real applications.

Concerning the syntax of multi-adjoint normal logic programs, the main novelty presented in this paper will be the inclusion of different negation operators in the body of the rules and the consideration of a special type of rule called constraint. This new and more versatile logic framework will be called extended multi-adjoint logic programming and it can be considered as one further step in order to allow an easier transformation from the natural language to decision rules.

The classical notion of a constraint is related to a rule whose head is empty (or only contains the bottom element) and it is used to specify that the body of the rule should not be satisfied for any valid solution. This definition was extended to the fuzzy case in order to consider in the head any element in the lattice [11]. This consideration enriches the language of the multi-adjoint framework and so, this special kind of rules will be adapted to this framework. On the other hand, taking into account the multi-adjoint philosophy, one can expect that different negation operators appear in the body of the rules of a logic program. This fact will be simulated by the use of general operators which are order-preserving in some arguments and order-reversing in the remainders.

In what regards to the semantics of extended multi-adjoint logic programs, we will continue with the stable model semantics. Based on the motivation and work developed by Janssen et al. [11], we have introduced a mechanism to translate an arbitrary extended multi-adjoint logic program into an extended multi-adjoint logic program without constraints, which does not increase the number of rules of the original program. Additionally, a procedure to translate an arbitrary extended multi-adjoint logic program without constraints into a multi-adjoint normal logic program will be also given. Both mechanisms will allow us to establish the relationships among

the stable models of each mentioned logic programs. From these relationships, we will be capable of ensuring that there exists a one-to-one correspondence between the stable models of an extended multi-adjoint logic program and the multi-adjoint normal logic program obtained from the translation mechanisms.

In summary, the advances provided by this work have a strong impact on the syntax and the semantics of the multi-adjoint logic programming framework. The possibility of including different negation operators constraints in the body of the rules and considering constraints makes the rules more flexible and therefore, it makes easier the process of translating the information contained in a text or in a database into a logical program. The more flexible the rules, the easier it will be to translate the information into decision rules and the easier interpretation of those decision rules will be. It is also essential to highlight the importance of the theoretical results developed in this work. This fact lies in the possibility of translating different results given in the multi-adjoint framework [6] or in other more particular environments, as the residuated one [15, 18], into the new flexible framework of extended multi-adjoint logic programs.

This paper is laid out in the following way. Section 2 recalls the main concepts and results related to the multi-adjoint normal logic programming framework. Section 3 presents the syntax and semantics of extended multi-adjoint logic programs. The procedure for translating an arbitrary extended multi-adjoint logic program into an extended multi-adjoint logic program without constraints is included in Section 4, whereas the one given for translating an arbitrary extended multi-adjoint logic program without constraints into a multi-adjoint normal logic program is presented in Section 5. The technical properties introduced in Sections 4 and 5 allow us to obtain a multi-adjoint normal logic program with the same stable models as a given extended multi-adjoint logic program. Section 6 ends with some conclusions and prospects for future work.

2. Multi-adjoint normal logic programming

Multi-adjoint normal logic programming was introduced in [6] as a non-monotonic multi-adjoint logic programming framework in which the considered algebraic structure is enriched with a negation operator. The syntax and the semantics of multi-adjoint normal logic programs are recalled in the sequel.

First of all, we will include the notion of adjoint pair, which arises as a generalization of a t-norm and its residuated implication.

Definition 1. Let (P, \leq) be a partially ordered set and $(\&, \leftarrow)$ a pair of binary operations in P , we say that $(\&, \leftarrow)$ forms an *adjoint pair* in (P, \leq) , if the following properties hold:

1. $\&$ is order-preserving¹ in both arguments, that is, if $x_1, x_2, x, y_1, y_2, y \in P$ and $x_1 \leq x_2, y_1 \leq y_2$, then $(x_1 \& y) \leq (x_2 \& y)$ and $(x \& y_1) \leq (x \& y_2)$.
2. \leftarrow is order-preserving in the first argument (the consequent) and order-reserving in the second argument (the antecedent). That is, if $y_1, y_2, y, z_1, z_2, z \in P$ and $y_1 \leq y_2, z_1 \leq z_2$, then $(z_1 \leftarrow y) \leq (z_2 \leftarrow y)$, $(z \leftarrow y_2) \leq (z \leftarrow y_1)$.
3. $(\&, \leftarrow)$ satisfies the adjoint property, that is,

$$x \leq (z \leftarrow y) \quad \text{if and only if} \quad (x \& y) \leq z$$

for all $x, y, z \in P$.

Example of adjoint pairs defined on $[0, 1]$ are the Gödel, product and Lukasiewicz t-norms together with their residuated implications:

$$\begin{aligned} x \&_G y &= \min\{x, y\} & z \leftarrow_G y &= \begin{cases} 1 & \text{if } y \leq z \\ z & \text{otherwise} \end{cases} \\ x \&_P y &= x \cdot y & z \leftarrow_P y &= \min\{1, z/y\} \\ x \&_L y &= \max\{0, x + y - 1\} & z \leftarrow_L y &= \min\{1, 1 - y + z\} \end{aligned}$$

The algebraic structure on which multi-adjoint normal logic programs are defined is usually known as multi-adjoint normal lattice. This notion is formally stated as follows.

Definition 2. The tuple $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, \neg)$ is a *multi-adjoint normal lattice*, if the following properties are verified:

1. (L, \preceq) is a bounded lattice, i.e. it has a bottom (\perp) and a top (\top) element;

¹Order-preserving, monotonic and increasing mappings are equivalent notions.

2. $(\&_i, \leftarrow_i)$ is an adjoint pair in (L, \preceq) , for $i \in \{1, \dots, n\}$;
3. $\top \&_i \vartheta = \vartheta \&_i \top = \vartheta$, for all $\vartheta \in L$ and $i \in \{1, \dots, n\}$;
4. $\neg: L \rightarrow L$ is a negation operator, that is, an order-reversing mapping satisfying the equalities $\neg(\perp) = \top$ and $\neg(\top) = \perp$.

Once the underlying structure has been established, a multi-adjoint normal logic program is defined as a set of rules where different implications may appear in different rules of the program, and the body of the rules is composed of aggregation operators, which will usually be denoted as $@$, together with a negation operator. In this paper, the usual boundary conditions of the aggregator operator will be not required, hence, the monotonicity will be the needed basic property. Notice that particular cases of aggregators are conjunctive operators (which in particular can be the adjoint conjunctors of the implications, that is, $\&_1, \&_2, \dots, \&_n$), disjunctive operators (denoted as \vee), average and hybrid operators, etc. Moreover, it can be the composition of different of these operators. In this case, we will write the propositional symbols q_1, \dots, q_n appearing in the body of the rule between brackets, that is, $@[q_1, \dots, q_n]$. The set of propositional symbols will be denoted as Π . Therefore, for example, the expression $(p \&_P q) \vee_L r$, depending on $p, q, r \in \Pi$, can be written from the operator $@: \Pi \times \Pi \rightarrow \Pi$, defined as $@[p, q, r] = (p \&_P q) \vee_L r$, for all $p, q, r \in \Pi$. For more details see [20].

Definition 3. Let $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, \neg)$ be a multi-adjoint normal lattice. A *multi-adjoint normal logic program (MANLP)* \mathbb{P} is a finite set of weighted rules of the form:

$$\langle p \leftarrow_i @[p_1, \dots, p_m, \neg p_{m+1}, \dots, \neg p_n]; \vartheta \rangle$$

where $i \in \{1, \dots, n\}$, $@$ is an aggregator operator, ϑ is an element of L and p, p_1, \dots, p_n are propositional symbols such that $p_j \neq p_k$, for all $j, k \in \{1, \dots, n\}$, with $j \neq k$. The propositional symbol p is called *head* of the rule, $@[p_1, \dots, p_m, \neg p_{m+1}, \dots, \neg p_n]$ is called *body* of the rule and the value ϑ is its *weight*.

The set of propositional symbols appearing in a MANLP \mathbb{P} is usually denoted as $\Pi_{\mathbb{P}}$. As far as the semantics of MANLPs is concerned, it is based on the stable model semantics. In the following, the notion of interpretation is presented, which plays a fundamental role in the stable models semantics.

Definition 4. Given a complete lattice (L, \preceq) , a mapping $I: \Pi_{\mathbb{P}} \rightarrow L$, which assigns to every propositional symbol appearing in $\Pi_{\mathbb{P}}$ an element of the lattice L , is called *L-interpretation*. The set of all *L-interpretations* is denoted as $\mathcal{I}_{\mathcal{L}}$.

In order to distinguish a syntactical symbol in a rule from the operator that it represents, we will fix the next notation. Given a symbol ω , its interpretation under a multi-adjoint normal lattice will be denoted as $\dot{\omega}$. Likewise, the evaluation of a formula \mathcal{B} under an interpretation I will be given by the uniquely extended interpretation $\hat{I}(\mathcal{B})$ defined from I using the unique homomorphic extension theorem.

Considering this notation, the notions of model and satisfiability are defined below.

Definition 5. Given a MANLP \mathbb{P} and an interpretation $I \in \mathcal{I}_{\mathcal{L}}$, we say that:

- (1) A weighted rule $\langle p \leftarrow_i @ [p_1, \dots, p_m, \neg p_{m+1}, \dots, \neg p_n]; \vartheta \rangle$ in \mathbb{P} is *satisfied* by I if and only if

$$\vartheta \preceq \hat{I}(p \leftarrow_i @ [p_1, \dots, p_m, \neg p_{m+1}, \dots, \neg p_n])$$

- (2) An *L-interpretation* $I \in \mathcal{I}_{\mathcal{L}}$ is a *model* of \mathbb{P} if and only if all weighted rules in \mathbb{P} are satisfied by I .
- (3) An *L-interpretation* $I \in \mathcal{I}_{\mathcal{L}}$ is the *least model* of \mathbb{P} , if the inequality $I(p) \preceq J(p)$ holds, for all $p \in \Pi_{\mathbb{P}}$ and for each model $J \in \mathcal{I}_{\mathcal{L}}$ of \mathbb{P} .

The following definition recalls the immediate consequence operator for the multi-adjoint normal logic programming framework considered in [6].

Definition 6. Let \mathbb{P} be a MANLP. The *immediate consequence operator* is the mapping $T_{\mathbb{P}}^{\mathcal{L}}: \mathcal{I}_{\mathcal{L}} \rightarrow \mathcal{I}_{\mathcal{L}}$ defined for every *L-interpretation* I and $p \in \Pi_{\mathbb{P}}$ as follows:

$$T_{\mathbb{P}}(I)(p) = \sup\{\vartheta \&_i \dot{\hat{I}}(\mathcal{B}) \mid \langle p \leftarrow_i \mathcal{B}; \vartheta \rangle \in \mathbb{P}\}$$

It is convenient to mention that, when \mathbb{P} does not contain any negation operator [20], we obtain that $T_{\mathbb{P}}$ is monotonic and it has a least fix-point by Knaster-Tarski fix-point theorem [26]. From these facts, we can deduce that this least fix-point is the least model of \mathbb{P} . For more details, see [20].

However, the immediate consequence operator is not necessarily monotonic in MANLPs and consequently, the existence of the least model cannot be ensured. For that reason, the semantics of MANLPs is based on a special type of models called stable models. This notion is closely related to the reduct of a MANLP with respect to a given interpretation. Namely, given a MANLP \mathbb{P} and an L -interpretation I , the reduct of \mathbb{P} with respect to I , denoted as \mathbb{P}_I , is defined by substituting each rule

$$\langle p \leftarrow_i @ [p_1, \dots, p_m, \neg p_{m+1}, \dots, \neg p_n]; \vartheta \rangle$$

in \mathbb{P} by the rule

$$\langle p \leftarrow_i @_I [p_1, \dots, p_m]; \vartheta \rangle \quad (1)$$

where the operator $@_I: L^m \rightarrow L$ is defined as

$$@_I[\vartheta_1, \dots, \vartheta_m] = @[\vartheta_1, \dots, \vartheta_m, \dot{\neg} I(p_{m+1}), \dots, \dot{\neg} I(p_n)]$$

for all $\vartheta_1, \dots, \vartheta_m \in L$.

The concept of stable model is then defined as follows.

Definition 7. Given a MANLP \mathbb{P} and an L -interpretation I , we say that I is a *stable model* of \mathbb{P} if and only if I is the least model of \mathbb{P}_I .

It is important to highlight that the semantics of MANLPs is defined in terms of the stable models of the program. Therefore, ensuring the existence of stable models becomes a crucial task in order to define the semantics of a MANLP. The next result provides a sufficient condition to come to this target. In particular, it is related to the continuity of the operators involved in the rules of the MANLP.

Theorem 8 ([6]). *Let $(K, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, \neg)$ be a multi-adjoint normal lattice, where K is a non-empty convex compact subset of an euclidean space, and \mathbb{P} be a finite MANLP defined on this lattice. If $\&_1, \dots, \&_n, \neg$ and the aggregator operators in the body of the rules of \mathbb{P} are continuous operators, then \mathbb{P} has at least a stable model.*

After presenting the main notions corresponding to the multi-adjoint normal logic programming framework, we will focus on extending this logic programming environment to a more general one, including several negation operators in the body of the rules and a new type of rules called constraints. From now on, we will consider a fixed MANLP \mathbb{P} defined on a multi-adjoint normal lattice $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, \neg)$.

3. Extending multi-adjoint normal logic programs

It is very usual in real cases that some property, attribute or characteristic be limited by an upper bound. Indeed, this boundary value can be given by a combination of properties. This situation can be simulated in logic programming by the use of constraints. Specifically, in fuzzy logic programming, given a program \mathbb{P} , a formula \mathcal{B} and an upper bound $c \in L$, we need to ensure that the inequality $M(\mathcal{B}) \preceq c$ holds for every stable model M of \mathbb{P} .

In order to capture this condition in multi-adjoint logic programming, we propose the inclusion of the rule $\langle c \leftarrow_i \mathcal{B}; \top \rangle$ in the program \mathbb{P} , being $(\&_i, \leftarrow_i)$ any adjoint pair belonging to the multi-adjoint normal lattice in which the program \mathbb{P} is defined, that is, $i \in \{1, \dots, n\}$. This kind of rules, in which an element of the carrier of the multi-adjoint normal lattice appears in the head of the rule, are usually known as *constraints*.

Next, we will see that the inclusion of constraints provides the required upper bound limitation. If a stable model M of the program \mathbb{P} satisfies the rule $\langle c \leftarrow_i \mathcal{B}; \top \rangle$, then the inequality $\top \preceq M(c \leftarrow_i \mathcal{B})$ should be satisfied, which is equivalent to $M(\mathcal{B}) \&_i \top \preceq c$, by the adjoint property. Since \top is the identity element of $\&_i$, we obtain the inequality² $M(\mathcal{B}) \preceq c$. Therefore, the satisfiability of the constraint $\langle c \leftarrow_i \mathcal{B}; \top \rangle$ provides that any stable model M of \mathbb{P} verifies the inequality $M(\mathcal{B}) \preceq c$.

On the other hand, according to the multi-adjoint philosophy, the inclusion of different negation operators in the body of the rules would also be interesting. This proposal will be carried out by using a special type of aggregator operator in the body of the rules being order-preserving in some arguments and order-reversing in the remainders.

Hence, hereinafter, an extended multi-adjoint logic programming framework which includes both constraints and aggregator operators with order-reversing arguments will be developed. Namely, this kind of aggregator operators will be called extended aggregators.

Definition 9. Let (L, \preceq) be a complete lattice. An *extended aggregator* $@^e: L^n \rightarrow L$ is any order-preserving mapping on the first i -th arguments when $i \in \{1, \dots, m\}$ and order-reversing on the j -th arguments when $j \in$

²Notice that we consider $M(c) = c$, for each $c \in L$, as it is formally stated further in Definition 12.

$\{m + 1, \dots, n\}$. The image of an element $(x_1, \dots, x_n) \in L^n$ under $@^e$ will be denoted as

$$@^e(x_1, \dots, x_n) = @^e[x_1, \dots, x_m; x_{m+1}, \dots, x_n]$$

A multi-adjoint lattice $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n)$ enriched with a family of extended aggregators $@_1^e, \dots, @_k^e$, will be called *extended multi-adjoint lattice*.

This general algebraic structure together with constraints are the pillars from which we will extend MANLP to a more flexible framework.

Definition 10. Let $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, @_1^e, \dots, @_k^e)$ be an extended multi-adjoint lattice. An *extended multi-adjoint logic program (EMALP)* \mathbb{P}^e is a finite set of weighted rules of the form

$$\langle p \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

and rules of the form

$$\langle c \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \top \rangle$$

where $i \in \{1, \dots, n\}$, $@^e \in \{@_1^e, \dots, @_k^e\}$, ϑ and c are elements of L and p, p_1, \dots, p_n are propositional symbols such that $p_{s_1} \neq p_{s_2}$, for all $s_1, s_2 \in \{1, \dots, n\}$, with $s_1 \neq s_2$.

Notice that, the extended aggregators considered in the extended multi-adjoint lattice and used in the program, can be obtained from the composition of different monotonic operators (such as, conjunctors and disjunctions) and negations operators.

As it was mentioned at the beginning of this section, we can simulate more than just one negation operator. Namely, consider a multi-adjoint lattice $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n)$ enriched with the negation operators \neg_1, \dots, \neg_k , and a set of rules of the form

$$\langle p \leftarrow_i @ [p_1, \dots, p_m, \neg_{j_{m+1}} p_{m+1}, \dots, \neg_{j_n} p_n]; \vartheta \rangle$$

where $i \in \{1, \dots, n\}$, $j_l \in \{1, \dots, k\}$ for each $l \in \{m + 1, \dots, n\}$, $@$ is an aggregator, $\vartheta \in L$ and p, p_1, \dots, p_n are propositional symbols such that $p_{s_1} \neq p_{s_2}$, for all $s_1, s_2 \in \{1, \dots, n\}$, with $s_1 \neq s_2$.

Then, the MANLP \mathbb{P} composed of such set of rules can be rewritten as an EMALP as follows. For each rule r of the form

$$\langle p \leftarrow_i @ [p_1, \dots, p_m, \neg_{j_{m+1}} p_{m+1}, \dots, \neg_{j_n} p_n]; \vartheta \rangle$$

with $j_l \in \{1, \dots, k\}$ for each $l \in \{m+1, \dots, n\}$, occurring in \mathbb{P} , we consider the rule r^e given by

$$\langle p \leftarrow_i @^e [p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

being

$$@^e [p_1, \dots, p_m; p_{m+1}, \dots, p_n] = @[p_1, \dots, p_m, \neg_{j_{m+1}} p_{m+1}, \dots, \neg_{j_n} p_n]$$

Due to the fact that $@$ is an aggregator and \neg_1, \dots, \neg_k are negation operators, the mapping $@^e$ is undeniably an extended aggregator. Hence, the program

$$\mathbb{P}^e = \{r^e \mid r \in \mathbb{P}\}$$

is clearly an EMALP. Therefore, if we want to consider MANLPs in which more than one negation operator appears, we can do it by means of EMALPs.

In order to illustrate the expressivity power of the extended multi-adjoint logic programming framework, a practical toy example will be developed in the sequel.

Example 11. A group of experts stated that the level of water, the level of oil and the temperature are critical features for the acceptable behaviour of a motor. Specifically, they reached the following conclusions:

- (a) The level of water and the level of oil seriously affect to the temperature of the motor. In particular, if either the level of water or the level of oil is low, then the temperature of the motor increases significantly, being the level of water more damaging than the level of oil.
- (b) The level of oil and the temperature of the motor must be controlled, since in case that these two features are high at the same time, then the motor is broken.

Consider the variables $w, o, t \in [0, 1]$, which represent the level of water, the level of oil and the temperature of the motor, respectively. Namely, the value 1 (respectively 0) for the variable w/o/t corresponds to a high (respectively low) level of water/level of oil/temperature.

The behaviour detailed in Statement (a) can be modelled by means of the following rule:

$$\langle t \leftarrow_{\mathbb{P}} \max\{\neg_1(o), \neg_2(w)\}; 0.9 \rangle$$

where the negation operators $\neg_1, \neg_2: [0, 1] \rightarrow [0, 1]$ are defined as $\neg_1(x) = 1 - x$ and $\neg_2(x) = (1 - x^2)^{1/2}$, for all $x \in [0, 1]$.

Observe that, the rule has a high weight due to the level of water and the level of oil have a big impact on the temperature of the motor. We make use of different negations in order to model that this impact is different, being stronger in the case of low water than in the case of low oil. Indeed, $\neg_1(x) \leq \neg_2(x)$, for all $x \in [0, 1]$. Finally, we consider the maximum between $\neg_1(o)$ and $\neg_2(w)$, because a low level of any of these two variables, oil and water, is enough in order to increase the temperature of the motor.

In what regards Statement (b), since the motor cannot work with a high level of oil and a high temperature, then we need to include this requirement through the following constraint:

$$\langle 0.8 \leftarrow_{\mathbb{P}} o \&_{\mathbb{P}} t; 1 \rangle$$

in which the threshold 0.8 prevents the level of oil and the temperature of the motor, from being high at the same time. In this case, the operator $\&_{\mathbb{P}}$ has been selected in order to represent the conjunction of the natural language. Notice that, the implication can be any residuated implications because the truth value is the top element 1. \square

As far as the semantics of EMALPs is concerned, the notions of interpretation, satisfaction of a rule and model are defined in a similar way to the semantics of MANLPs.

Definition 12. Let \mathbb{P}^e be an EMALP on an extended multi-adjoint lattice $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, @_1^e, \dots, @_k^e)$. An interpretation of \mathbb{P}^e is any mapping $I^e: \Pi_{\mathbb{P}^e} \rightarrow L$, where $\Pi_{\mathbb{P}^e}$ is the set of propositional symbols appearing in \mathbb{P}^e . The set of interpretations of \mathbb{P}^e is usually denoted as $\mathcal{I}_{\mathbb{P}^e}^e$.

As usual, the evaluation of a formula under an interpretation proceeds inductively. For the sake of completeness, an interpretation I^e is extended to the set L as $I^e(x) = x$ for each $x \in L$.

Basing on the complete lattice (L, \preceq) , an order relation can be defined in the set of interpretations of \mathbb{P}^e . Namely, given two interpretations $I_1^e, I_2^e \in \mathcal{I}_{\mathbb{P}^e}^e$, we say that $I_1^e \sqsubseteq I_2^e$ if and only if $I_1^e(p) \preceq I_2^e(p)$, for all $p \in \Pi_{\mathbb{P}^e}$.

Definition 13. Given an EMALP \mathbb{P}^e and an interpretation $I^e \in \mathcal{I}_{\mathfrak{L}}^e$, we say that:

- (1) A rule $\langle p \leftarrow_i \textcircled{\@}^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$ of \mathbb{P}^e is *satisfied* by I^e if and only if $\vartheta \preceq \hat{I}^e(p \leftarrow_i \textcircled{\@}^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n])$.

Likewise, a constraint $\langle c \leftarrow_i \textcircled{\@}^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$ of \mathbb{P}^e is *satisfied* by I^e if and only if $\vartheta \preceq c \leftarrow_i \hat{I}^e(\textcircled{\@}^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n])$.

- (2) An interpretation $I^e \in \mathcal{I}_{\mathfrak{L}}^e$ is a *model* of \mathbb{P}^e if and only if all weighted rules in \mathbb{P}^e are satisfied by I^e .

Similarly to the multi-adjoint normal logic programming framework, the notion of stable model for EMALPs is stated by means of the concept of reduct with respect to an interpretation. In this case, given an interpretation, each rule in the EMALP is substituted in the reduct by an analogous rule in which the atoms occurring in any order-reversing argument of the aggregator in the body of the rule are evaluated under the given interpretation.

Formally, given an EMALP \mathbb{P}^e and an interpretation M^e , the reduct of \mathbb{P}^e with respect to M^e , denoted as $\mathbb{P}_{M^e}^e$, is built substituting each rule in \mathbb{P}^e

$$\langle l \leftarrow_i \textcircled{\@}^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

where l is either a propositional symbol or a value in L , and $\textcircled{\@}^e \in \{\textcircled{\@}_1^e, \dots, \textcircled{\@}_k^e\}$, by the rule

$$\langle l \leftarrow_i \textcircled{\@}_{M^e}^e[p_1, \dots, p_m]; \vartheta \rangle$$

where the aggregator $\textcircled{\@}_{M^e}^e: L^m \rightarrow L$ is defined as

$$\textcircled{\@}_{M^e}^e[\vartheta_1, \dots, \vartheta_m] = \textcircled{\@}^e[\vartheta_1, \dots, \vartheta_m; M^e(p_{m+1}), \dots, M^e(p_n)] \quad (2)$$

Evidently, as $\textcircled{\@}^e$ is an order-preserving mapping in the first m arguments, $\textcircled{\@}_{M^e}^e$ is an aggregator operator. Hence, we conclude that the program $\mathbb{P}_{M^e}^e$ is a monotonic multi-adjoint logic program. The notion of stable model is then established as follows

Definition 14. Let \mathbb{P}^e be an EMALP. An interpretation M^e is said to be a stable model of \mathbb{P}^e if M^e is the least model of $\mathbb{P}_{M^e}^e$.

The following proposition shows an interesting property of stable models.

Proposition 15. *Any stable model of an EMANLP \mathbb{P}^e is a minimal model of \mathbb{P}^e .*

PROOF. Consider that M^e is a stable model of \mathbb{P}^e . In other words, M^e is the least model of the program $\mathbb{P}_{M^e}^e$. We will prove by reductio ad absurdum that M^e is a minimal model of \mathbb{P}^e .

Thus, suppose that there exists a model N^e of \mathbb{P}^e such that $N^e \sqsubset M^e$. We will see that N^e is a model of $\mathbb{P}_{M^e}^e$.

Since N^e is a model of \mathbb{P}^e , for each rule in \mathbb{P}^e of the form

$$\langle l \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

where l is either a propositional symbol or a value in L , we obtain that

$$\vartheta \preceq \hat{N}^e(l \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n])$$

Equivalently,

$$\vartheta \preceq N^e(l) \leftarrow_i \dot{@}^e[N^e(p_1), \dots, N^e(p_m); N^e(p_{m+1}), \dots, N^e(p_n)]$$

Now, as $\dot{@}^e$ is order-reversing in its last $n - m$ arguments and $N^e \sqsubset M^e$, the following inequality holds:

$$\dot{@}^e[N^e(p_1), \dots, N^e(p_m); M^e(p_{m+1}), \dots, M^e(p_n)] \preceq \dot{@}^e[N^e(p_1), \dots, N^e(p_m); N^e(p_{m+1}), \dots, N^e(p_n)]$$

Therefore, according to the fact that \leftarrow_i is order-reversing in the antecedent, we deduce that

$$\begin{aligned} \vartheta &\preceq N^e(l) \leftarrow_i \dot{@}^e[N^e(p_1), \dots, N^e(p_m); N^e(p_{m+1}), \dots, N^e(p_n)] \\ &\preceq N^e(l) \leftarrow_i \dot{@}^e[N^e(p_1), \dots, N^e(p_m); M^e(p_{m+1}), \dots, M^e(p_n)] \\ &= N^e(l) \leftarrow_i \dot{@}_{M^e}^e[N^e(p_1), \dots, N^e(p_m)] \end{aligned}$$

As this inequality is true for every rule, we obtain that N^e is a model of $\mathbb{P}_{M^e}^e$, which leads us to a contradiction, since M^e is the least model of $\mathbb{P}_{M^e}^e$ by hypothesis. \square

The next example illustrates the concepts defined in this section. In addition, it shed lights on how flexible the extended multi-adjoint logic programming framework is.

Example 16. Let us consider the extended multi-adjoint lattice given by $([0, 1], \leq, \leftarrow_G, \&_G, \leftarrow_P, \&_P, \leftarrow_L, \&_L, @_1^e, @_2^e, @_3^e, @_4^e, @_5^e)$, where the adjoint pairs are the well-known Gödel, product and Łukasiewicz pairs [7] and the extended aggregators are defined on $[0, 1]^4$ as follows

$$\begin{aligned} @_1^e[x, y, z, t] &= \min \left\{ \frac{y}{z + t + 0.1}, 1 \right\} \\ @_2^e[x, y, z, t] &= \max \{ \neg_1 z, \neg_2 t \} \\ @_3^e[x, y, z, t] &= \neg_1 y \\ @_4^e[x, y, z, t] &= 1 \\ @_5^e[x, y, z, t] &= \max \{ z, 0.7 \} \end{aligned}$$

for all $x, y, z, t \in [0, 1]$, and the negation operators $\neg_1, \neg_2: [0, 1] \rightarrow [0, 1]$ defined as $\neg_1(x) = 1 - x$ and $\neg_2(x) = (1 - x^2)^{1/2}$, for all $x \in [0, 1]$.

Notice that, the aggregators $@_1^e$ and $@_2^e$ are both order-preserving on the two first arguments, and order-reversing in the last two arguments. The aggregator $@_3^e$ is order-reversing on the second argument. Since the rest of variables not appear in the computation, we can consider the semicolon in between x and y . We can also consider that the aggregators $@_4^e$ and $@_5^e$ are order-preserving in all the arguments. Hence, in order to simplify the notation we have written $[x, y, z, t]$ instead of $[x, y, z, t;]$.

In this setting, we consider the EMALP \mathbb{P}^e composed of the following three weighted rules, one constraint and one fact.

$$\begin{aligned} r_1^e &: \langle p \leftarrow_P @_1^e[p, q, s, t] ; 0.5 \rangle & r_4^e &: \langle s \leftarrow_G @_4^e[p, q, s, t] ; 0.8 \rangle \\ r_2^e &: \langle q \leftarrow_P @_2^e[p, q, s, t] ; 0.6 \rangle & r_5^e &: \langle t \leftarrow_G @_5^e[p, q, s, t] ; 0.8 \rangle \\ r_3^e &: \langle 0.7 \leftarrow_L @_3^e[p, q, s, t] ; 1 \rangle \end{aligned}$$

It is important to highlight the crucial role that the rule r_3^e plays with respect to the value of q in the models of \mathbb{P}^e . Namely, any interpretation M^e satisfying the rule r_3^e in \mathbb{P}^e verifies the inequality $1 \leq \dot{M}^e(0.7 \leftarrow_L \neg_1 q)$, that is, $1 \leq 0.7 \leftarrow_L \neg_1 M^e(q)$. Therefore, according to the adjoint property, satisfied by $(\&_L, \leftarrow_L)$, and the definition of \neg_1 , we obtain that $1 - M^e(q) \leq 0.7$. Equivalently, $0.3 \leq M^e(q)$. Hence, the least value that q can take under any model of \mathbb{P} is 0.3. In other words, we demand the models of \mathbb{P} to satisfy that the evaluation of q is greater or equal than 0.3.

For instance, we will see in the following that the interpretation given by $M^e \equiv \{(p, 0.25), (q, 0.4), (s, 0.9), (t, 0.85)\}$ is a model of \mathbb{P}^e . For the rule

r_1^e , we obtain that

$$\begin{aligned}
\hat{M}^e\left(p \leftarrow_{\text{P}} \min\left\{\frac{q}{s+t+0.1}, 1\right\}\right) &= M^e(p) \leftarrow_{\text{P}} \min\left\{\frac{M^e(q)}{M^e(s) + M^e(t) + 0.1}, 1\right\} \\
&= 0.25 \leftarrow_{\text{P}} \min\left\{\frac{0.4}{0.9 + 0.85 + 0.1}, 1\right\} \\
&= 0.25 \leftarrow_{\text{P}} \min\left\{\frac{8}{37}, 1\right\} \\
&= 0.25 \leftarrow_{\text{P}} \frac{8}{37} = 1
\end{aligned}$$

Therefore $0.5 \leq \hat{M}^e\left(p \leftarrow_{\text{P}} \min\left\{\frac{q}{s+t+0.1}, 1\right\}\right)$, that is, M^e satisfies the rule r_1^e . With regard to the rule r_2^e , the following chain of equalities holds

$$\begin{aligned}
\hat{M}^e(q \leftarrow_{\text{P}} \max\{\neg_1 s, \neg_2 t\}) &= M^e(q) \leftarrow_{\text{P}} \max\{\dot{\neg}_1 M^e(s), \dot{\neg}_2 M^e(t)\} \\
&= 0.4 \leftarrow_{\text{P}} \max\{\dot{\neg}_1 0.9, \dot{\neg}_2 0.85\} \\
&= 0.4 \leftarrow_{\text{P}} \max\left\{0.1, \frac{\sqrt{111}}{20}\right\} \\
&= 0.4 \leftarrow_{\text{P}} \frac{\sqrt{111}}{20} = \frac{8}{\sqrt{111}} \approx 0.76
\end{aligned}$$

As a consequence, $0.6 \leq \hat{M}^e(q \leftarrow_{\text{P}} \max\{\neg_1 s, \neg_2 t\})$. Thus, the rule r_2^e is satisfied by M^e . Moreover, as $0.3 \leq 0.4 = M^e(q)$, the interpretation M^e satisfies the rule r_3^e as well. Similarly, as $0.8 \leq 0.9 = M^e(s)$, M^e satisfies the rule r_4^e . Finally, the computations corresponding to the satisfaction of r_5^e are given below.

$$\begin{aligned}
\hat{M}^e(t \leftarrow_{\text{G}} \max\{s, 0.7\}) &= M^e(t) \leftarrow_{\text{G}} \max\{M^e(s), 0.7\} \\
&= 0.85 \leftarrow_{\text{G}} \max\{0.9, 0.7\} \\
&= 0.85 \leftarrow_{\text{G}} 0.9 = 0.85
\end{aligned}$$

Hence, we conclude that the interpretation M^e satisfies all rules appearing in \mathbb{P}^e , and thus it is a model of \mathbb{P}^e .

In spite of this fact, one can easily check that M^e is not a stable model of the EMALP \mathbb{P}^e . Indeed, it is not a minimal model of \mathbb{P}^e . For instance, the interpretation N^e given by $N^e \equiv \{(p, \frac{9}{85}), (q, 0.36), (s, 0.8), (t, 0.8)\}$ is also a model of \mathbb{P}^e and clearly satisfies that $N^e \sqsubseteq M^e$. Therefore, applying Proposition 15, we deduce that M^e is not a stable model of \mathbb{P}^e . In the following, we show that the interpretation N^e is not only a model of \mathbb{P}^e ,

but a stable model. That is, N^e is the least model of the reduct of \mathbb{P}^e with respect to N^e , denoted as $\mathbb{P}_{N^e}^e$.

We must recall that the least model of a positive program $\mathbb{P}_{N^e}^e$ is equivalent to the least fix-point of the immediate consequence operator $T_{\mathbb{P}_{N^e}^e}$ [6, 14, 20]. Therefore, we will prove that the interpretation N^e is a stable model of \mathbb{P}^e , proving that the least fix-point of $T_{\mathbb{P}_{N^e}^e}$ is N^e .

Following the Knaster-Tarski's theorem, we obtain the least fix-point iterating the operator $T_{\mathbb{P}_{N^e}^e}$ from the least interpretation: $I_{\perp} \equiv \{(p, 0), (q, 0), (s, 0), (t, 0)\}$ until a fix-point arises, which is the least one. Hence, from the reduct $\mathbb{P}_{N^e}^e$, which is composed of the rules

$$\begin{aligned} r_1^{e*} &: \langle p \leftarrow_{\text{P}} \min \left\{ \frac{y}{s+t+0.1}, 1 \right\} ; 0.5 \rangle & r_4^{e*} &: \langle s \leftarrow_{\text{G}} 1 ; 0.8 \rangle \\ r_2^{e*} &: \langle q \leftarrow_{\text{P}} \max\{0.2, 0.6\} ; 0.6 \rangle & r_5^{e*} &: \langle t \leftarrow_{\text{G}} \max\{s, 0.7\} ; 0.8 \rangle \\ r_3^{e*} &: \langle 0.7 \leftarrow_{\text{L}} 0.64 ; 1 \rangle \end{aligned}$$

we compute the $T_{\mathbb{P}_{N^e}^e}$, obtaining the following iterations:

	p	q	s	t
I_{\perp}	0	0	0	0
$T_{\mathbb{P}_{N^e}^e}(I_{\perp})$	0	0.36	0.8	0.7
$T_{\mathbb{P}_{N^e}^e}^2(I_{\perp})$	9/80	0.36	0.8	0.8
$T_{\mathbb{P}_{N^e}^e}^3(I_{\perp})$	9/85	0.36	0.8	0.8
$T_{\mathbb{P}_{N^e}^e}^4(I_{\perp})$	9/85	0.36	0.8	0.8

Therefore, after the third iteration the fix-point arises and we can ensure that N^e is the least model of $\mathbb{P}_{N^e}^e$ and so, a stable model of \mathbb{P}^e . \square

The notion of stable model plays a crucial role in the definition of the semantics of an EMALP. Given the set of all stable models of an EMALP, one can check whether a statement is a consequence of the program by simply computing the truth value of the statement on each stable model. Hence, finding conditions which provide information on the number and the form of stable models becomes a critical task in order to define the semantics of an EMALP.

The following sections will focus on obtaining a MANLP equivalent to a given EMALP, from a semantical point of view. We will see that any condition related to the stable models in the multi-adjoint normal logic programming framework can be used in extended multi-adjoint logic programming.

4. Translating EMALPs into constraint-free EMALPs.

The main goal of this section is to obtain a semantically equivalent program to the original one, transforming the constraints to weighted rules. Hence, given an EMALP, we will see how to build a new constraint-free EMALP with the same stable models as the given EMALP. First of all, we will show that the mechanism given in [11] can be adapted to our framework and then we will improve it avoiding the inclusion of a bigger number of rules.

Given an EMALP \mathbb{P}^e defined on an extended multi-adjoint lattice $(L, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, @_1^e, \dots, @_k^e)$, observe that the main obstacle in order to include a constraint r in \mathbb{P}^e

$$r : \langle c \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \top \rangle$$

in a MANLP is that c is not a propositional symbol, but an element of the lattice L . This obstacle is overcome if there exists a propositional symbol p_c such that any stable model M^e of \mathbb{P}^e satisfies $M^e(p_c) = c$, since in that case the rule r is semantically equivalent to the rule

$$r^* : \langle p_c \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \top \rangle$$

Therefore, the stable models of the EMALP $\mathbb{P}^{\tilde{e}} = \mathbb{P}^e \setminus \{r\} \cup \{r^*\}$ coincide with the stable models of \mathbb{P}^e . Hence, we can include in the program a new propositional symbol p_c such that $M^e(p_c) = c$ for each stable model M^e of \mathbb{P}^e . Janssen et al. [11] create such propositional symbol by adding the following two rules in the program:

$$\begin{aligned} r_c^1 & : \langle p_c \leftarrow_j c; \top \rangle \\ r_c^2 & : \langle p_{\perp} \leftarrow_j g_{\perp}(\neg p_{\perp}) \& g_c(p_c); \top \rangle \end{aligned}$$

where $j \in \{1, \dots, n\}$, p_{\perp} is a new propositional symbol, $\&$ is a conjunctor of the multi-adjoint lattice, \neg any negation operator and the mapping $g_c: L \rightarrow L$ is defined, for each $c \in L$ and $x \in L$, as

$$g_c(x) = \begin{cases} \top & \text{if } c \prec x \\ \perp & \text{otherwise} \end{cases}$$

Notice that g_c is an order-preserving mapping, for each $c \in L$, and thus the mapping $@^e: L^2 \rightarrow L$ given by

$$@^e[x; y] = g_{\perp}(\neg y) \& g_c(x)$$

is order-preserving in the first argument and order-reversing in the second argument. Consequently, $@^e$ is an extended aggregator operator and, as a result, r_c^2 is a rule which can be included in an extended multi-adjoint logic program. Let $\mathcal{C}_{\mathbb{P}^e}$ be the set of constraints of \mathbb{P}^e and $\mathcal{K}_{\mathbb{P}^e}$ the set of elements of the lattice that occur in the head of constraints, that is $\mathcal{K}_{\mathbb{P}^e} = \{c \mid \langle c \leftarrow_i \mathcal{B}; \top \rangle \in \mathcal{C}_{\mathbb{P}^e}\}$, where \mathcal{B} represents the body of a rule. The corresponding constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ of \mathbb{P}^e is then given by

$$\begin{aligned} \mathbb{P}^{\tilde{e}} &= \{r \mid r \in \mathbb{P}^e \setminus \mathcal{C}_{\mathbb{P}^e}\} \\ &\cup \{\langle p_c \leftarrow_i \mathcal{B}; \top \rangle \mid \langle c \leftarrow_i \mathcal{B}; \top \rangle \in \mathcal{C}_{\mathbb{P}^e}\} \\ &\cup \{\langle p_c \leftarrow_j c; \top \rangle \mid c \in \mathcal{K}_{\mathbb{P}^e}\} \\ &\cup \{\langle p_{\perp} \leftarrow_j g_{\perp}(\neg p_{\perp}) \ \& \ g_c(p_c); \top \rangle \mid c \in \mathcal{K}_{\mathbb{P}^e}\} \end{aligned}$$

with $i, j \in \{1, \dots, n\}$. Following an analogous reasoning to the one given in [11], it can be proved that the stable models of \mathbb{P}^e are equivalent to the stable models of $\mathbb{P}^{\tilde{e}}$.

Notice that, in order to obtain the constraint-free EMALP $\mathbb{P}^{\tilde{e}}$, the number of new rules that we need to add (for transforming the constraints) to the EMALP \mathbb{P}^e does not depend on the number of constraints, but in the cardinal of $\mathcal{K}_{\mathbb{P}^e}$, specifically, it is equal to $|\mathcal{C}_{\mathbb{P}^e}| + 2|\mathcal{K}_{\mathbb{P}^e}|$. As a consequence, the program $\mathbb{P}^{\tilde{e}}$ has $2|\mathcal{K}_{\mathbb{P}^e}|$ more rules than the original EMALP \mathbb{P}^e .

In the sequel, we provide a new procedure from which one can build a constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ from \mathbb{P}^e whose number of rules coincides with the number of rules in \mathbb{P}^e . Furthermore, this new EMALP only requires one new propositional symbol whilst the strategy suggested in [11] demands $|\mathcal{K}_{\mathbb{P}^e}| + 1$ new propositional symbols.

To reach this goal, given $c \in L$, consider the mapping $f_c: L \rightarrow L$ defined for each $x \in L$ as

$$f_c(x) = \begin{cases} \perp & \text{if } x \preceq c \\ \top & \text{otherwise} \end{cases}$$

Note that, g_c and f_c are different since in a general complete lattice $c \not\preceq x$ is not equivalent to $x \preceq c$, since x and c can be incomparables.

Now, taking into account that f_c is an order-preserving mapping, for each $c \in L$, we can establish the definition of the corresponding constraint-free EMALP of a given EMALP as follows.

Definition 17. Let \mathbb{P}^e be an EMALP and $\mathcal{C}_{\mathbb{P}^e}$ the set of constraints of \mathbb{P}^e . The corresponding constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ of \mathbb{P}^e is defined as the

following set of rules:

$$\begin{aligned} \mathbb{P}^{\tilde{e}} &= \{r \mid r \in \mathbb{P}^e \setminus \mathcal{C}_{\mathbb{P}^e}\} \\ &\cup \{\langle p_{\perp} \leftarrow_i f_{\perp}(\neg p_{\perp}) \& f_c(\mathcal{B}); \top \rangle \mid \langle c \leftarrow_i \mathcal{B}; \top \rangle \in \mathcal{C}_{\mathbb{P}^e}\} \end{aligned}$$

where $i \in \{1, \dots, n\}$, p_{\perp} is a new propositional symbol, $\&$ is any conjunction of the extended multi-adjoint lattice and \neg is any negation operator.

In the following, we will see that the program built in Definition 17 is well-defined, that is, $\mathbb{P}^{\tilde{e}}$ is a EMALP, which evidently has no constraint. Notice that, for each rule $r^{\tilde{e}} \in \mathbb{P}^{\tilde{e}}$ of the form

$$r^{\tilde{e}} : \langle p_{\perp} \leftarrow_i f_{\perp}(\neg p_{\perp}) \& f_c(@^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]); \top \rangle$$

as \neg is an order-reversing mapping and, for each $c \in L$, f_c is order-preserving, we deduce that the mapping $\mathring{@}^{\tilde{e}}: L^{n+1} \rightarrow L$ defined as

$$\mathring{@}^{\tilde{e}}[p_1, \dots, p_m; p_{m+1}, \dots, p_n, p_{\perp}] = f_{\perp}(\mathring{\neg} p_{\perp}) \& f_c(\mathring{@}^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n])$$

is order-preserving in the first m arguments and order-reversing in the last $n - m + 1$ arguments. As a result, $\mathring{@}^{\tilde{e}}$ is an extended aggregator operator, and therefore the corresponding program $\mathbb{P}^{\tilde{e}}$ of an EMALP \mathbb{P}^e is, in fact, a constraint-free EMALP.

Now, we will show how the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ of the program $\mathbb{P}^{\tilde{e}}$ with respect to a given interpretation $M^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_{\perp}\} \rightarrow L$ is defined. Each rule in $\mathbb{P}^{\tilde{e}}$ of the form

$$\langle p_{\perp} \leftarrow_i f_{\perp}(\neg p_{\perp}) \& f_c(@^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]); \top \rangle$$

is substituted in the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ by the rule

$$\langle p_{\perp} \leftarrow_i f_{\perp}(\hat{M}^{\tilde{e}}(\neg p_{\perp})) \& f_c(@^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]); \top \rangle \quad (3)$$

since $\mathring{@}^{\tilde{e}}$ is order-preserving in the first m arguments and order-reversing in the last $n - m + 1$ arguments.

Notice that, each rule in $\mathbb{P}^{\tilde{e}}$ belonging to $\mathbb{P}^e \setminus \mathcal{C}_{\mathbb{P}^e}$ is substituted in the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ following the same procedure to the one given in Section 3 (Equation (2)).

The next result shows the existing relation between the stable models of an EMALP and the stable models of its corresponding constraint-free EMALP.

Theorem 18. *Let \mathbb{P}^e be an EMALP, $\mathbb{P}^{\tilde{e}}$ the corresponding constraint-free EMALP of \mathbb{P}^e , $M^e: \Pi_{\mathbb{P}^e} \rightarrow L$ an interpretation, and $M^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_{\perp}\} \rightarrow L$ the mapping defined as $M^{\tilde{e}}(p) = M^e(p)$ if $p \in \Pi_{\mathbb{P}^e}$ and $M^{\tilde{e}}(p_{\perp}) = \perp$. We have that M^e is a stable model of \mathbb{P}^e if and only if $M^{\tilde{e}}$ is a stable model of $\mathbb{P}^{\tilde{e}}$.*

PROOF. Given an interpretation $N^e: \Pi_{\mathbb{P}^e} \rightarrow L$, we define $N^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_{\perp}\} \rightarrow L$ as $N^{\tilde{e}}(p) = N^e(p)$ if $p \in \Pi_{\mathbb{P}^e}$ and $N^{\tilde{e}}(p_{\perp}) = \perp$.

First of all, we will see that, given two interpretations M and N , we have that N^e is a model of the reduct $\mathbb{P}_{M^e}^e$ if and only if $N^{\tilde{e}}$ is a model of the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$.

Given a rule $r^e \in \mathbb{P}^e$, we denote by $r^{\tilde{e}}$ its corresponding rule in $\mathbb{P}^{\tilde{e}}$. Observe that, if $r^e \in \mathbb{P}^e \setminus \mathcal{C}_{\mathbb{P}^e}$, then $r^{\tilde{e}} = r^e$. Clearly, as the propositional symbol p_{\perp} does not occur in the rule r^e , it neither does in $r^{\tilde{e}}$, and therefore N^e satisfies the rule r^e if and only if $N^{\tilde{e}}$ satisfies the rule $r^{\tilde{e}}$.

Now, we suppose that $r^e \in \mathcal{C}_{\mathbb{P}^e}$, that is, r^e is a rule of the form

$$\langle c \leftarrow_i \textcircled{e}[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \top \rangle$$

Its corresponding rule in the reduct $\mathbb{P}_{M^e}^e$, denoted as $r_{M^e}^e$, is then given by

$$\langle c \leftarrow_i \mathcal{B}_{M^e}; \top \rangle$$

being $\mathcal{B}_{M^e} = \textcircled{e}[p_1, \dots, p_m; M^e(p_{m+1}), \dots, M^e(p_n)]$.

Notice that, N^e satisfies the rule $r_{M^e}^e$ if and only if $\top \preceq N^e(c) \leftarrow_i N^e(\mathcal{B}_{M^e}) = c \leftarrow_i N^e(\mathcal{B}_{M^e})$. As a result, we can assert that N^e satisfies the rule $r_{M^e}^e$ if and only if $N^e(\mathcal{B}_{M^e}) \preceq c$.

Concerning the rule $r^{\tilde{e}} \in \mathbb{P}^{\tilde{e}}$, by Definition 17, it is given by

$$\langle p_{\perp} \leftarrow_i f_{\perp}(\neg p_{\perp}) \& f_c(\textcircled{e}[p_1, \dots, p_m; p_{m+1}, \dots, p_n]); \top \rangle$$

Consequently, we obtain that the corresponding rule of $r^{\tilde{e}}$ in the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, denoted as $r_{M^{\tilde{e}}}^{\tilde{e}}$, is given by

$$\langle p_{\perp} \leftarrow_i f_{\perp}(M^{\tilde{e}}(\neg p_{\perp})) \& f_c(\mathcal{B}_{M^{\tilde{e}}}); \top \rangle$$

being $\mathcal{B}_{M^{\tilde{e}}} = \textcircled{e}[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]$. Since $p_i \neq p_{\perp}$, for each $i \in \{m+1, \dots, n\}$, we deduce that $M^e(p_i) = M^{\tilde{e}}(p_i)$, and thus $\mathcal{B}_{M^e} = \mathcal{B}_{M^{\tilde{e}}}$. Moreover, by definition of the interpretation $M^{\tilde{e}}$, $M^{\tilde{e}}(p_{\perp}) = \perp$, and therefore the following chain of equalities holds

$$f_{\perp}(M^{\tilde{e}}(\neg p_{\perp})) = f_{\perp}(\overset{\cdot}{\neg} M^{\tilde{e}}(p_{\perp})) = f_{\perp}(\overset{\cdot}{\neg} \perp) = f_{\perp}(\top) = \top$$

Hence, since $\&$ satisfies the boundary condition with the top element, the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ is semantically equivalent to

$$\langle p_{\perp} \leftarrow_i f_c(\mathcal{B}_{M^e}); \top \rangle$$

Therefore, the interpretation $N^{\tilde{e}}$ satisfies the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ if and only if $\top \preceq N^{\tilde{e}}(p_{\perp}) \leftarrow_i f_c(N^{\tilde{e}}(\mathcal{B}_{M^e}))$. Due to p_{\perp} does not appear in \mathcal{B}_{M^e} , we deduce that $N^e(\mathcal{B}_{M^e}) = N^{\tilde{e}}(\mathcal{B}_{M^e})$, whence it is followed that $N^{\tilde{e}}$ satisfies the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ if and only if $f_c(N^e(\mathcal{B}_{M^e})) \preceq N^{\tilde{e}}(p_{\perp}) = \perp$. That is, $f_c(N^e(\mathcal{B}_{M^e})) = \perp$. Hence, according to the definition of the mapping f_c , we can assert that $N^{\tilde{e}}$ satisfies the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ if and only if $N^e(\mathcal{B}_{M^e}) \preceq c$, or equivalently, N^e satisfies the rule $r_{M^e}^e$.

Consequently, we conclude that an interpretation N^e is a model of the program $\mathbb{P}_{M^e}^e$ if and only if $N^{\tilde{e}}$ is a model of the program $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$. In particular, M^e is a model of $\mathbb{P}_{M^e}^e$ if and only if $M^{\tilde{e}}$ is a model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$.

Now, we will demonstrate that, given an interpretation M , we have that M^e is the least model of $\mathbb{P}_{M^e}^e$ if and only if $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, which will be proved by reductio ad absurdum.

Suppose that $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ but M^e is not the least model of $\mathbb{P}_{M^e}^e$. Then, there exists $N^e: \Pi_{\mathbb{P}^e} \rightarrow L$ with $N^e \sqsubset M^e$ such that N^e is a model of $\mathbb{P}_{M^e}^e$. Therefore, by the property proved above, the interpretation $N^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_{\perp}\} \rightarrow L$ given by $N^{\tilde{e}}(p) = N^e(p)$ if $p \in \Pi_{\mathbb{P}^e}$ and $N^{\tilde{e}}(p_{\perp}) = \perp$ is a model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, and the inequality $N^{\tilde{e}} \sqsubset M^{\tilde{e}}$ is straightforwardly satisfied. Therefore, we obtain a contradiction, since $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ by hypothesis.

Finally, suppose that M^e is the least model of $\mathbb{P}_{M^e}^e$ but $M^{\tilde{e}}$ is not the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, that is, there exists a model $N^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_{\perp}\} \rightarrow L$ of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ verifying $N^{\tilde{e}} \sqsubset M^{\tilde{e}}$. Since, N^e is equal to the restriction of $N^{\tilde{e}}$ to $\Pi_{\mathbb{P}^e}$, that is, $N^e = N^{\tilde{e}}|_{\Pi_{\mathbb{P}^e}}$ and $M^{\tilde{e}}(p_{\perp}) = \perp = N^{\tilde{e}}(p_{\perp})$, we obtain that

$$N^e = N^{\tilde{e}}|_{\Pi_{\mathbb{P}^e}} \sqsubset M^{\tilde{e}}|_{\Pi_{\mathbb{P}^e}} = M^e$$

Hence, according to the fact that $N^{\tilde{e}}$ is a model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, we obtain that N^e is a model of $\mathbb{P}_{M^e}^e$, and therefore M^e is not the least model of $\mathbb{P}_{M^e}^e$, in contradiction with the hypothesis.

Hence, we conclude that M^e is the least model of $\mathbb{P}_{M^e}^e$ if and only if $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, that is, M^e is a stable model of \mathbb{P}^e if and only if $M^{\tilde{e}}$ is a stable model of $\mathbb{P}^{\tilde{e}}$. \square

Given an EMALP \mathbb{P}^e , one can wonder if there exist stable models of the constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ such that the propositional symbol p_{\perp} is not

assigned to the bottom element in the lattice L . In that case, those stable models are not included in the characterization provided in Theorem 18.

Nevertheless, from Definition 17, we can deduce that any stable model maps the element p_\perp to the bottom element in the lattice L , as the following result shows.

Theorem 19. *Let \mathbb{P}^e be an EMALP and $\mathbb{P}^{\tilde{e}}$ be the corresponding constraint-free EMALP of \mathbb{P}^e . If $M^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_\perp\} \rightarrow L$ is a stable model of $\mathbb{P}^{\tilde{e}}$, then $M^{\tilde{e}}(p_\perp) = \perp$.*

PROOF. Suppose that $M^{\tilde{e}}: \Pi_{\mathbb{P}^e} \cup \{p_\perp\} \rightarrow L$ is a stable model of $\mathbb{P}^{\tilde{e}}$. First and foremost, notice that, if $\mathcal{C}_{\mathbb{P}^e} = \emptyset$, then there are no rule in $\mathbb{P}^{\tilde{e}}$ with head p_\perp , and thus neither are in the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$. Therefore, as $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, we straightforwardly obtain that $M^{\tilde{e}}(p_\perp) = \perp$.

Now, we assume that $\mathcal{C}_{\mathbb{P}^e} \neq \emptyset$. Clearly, as far as $\hat{M}^{\tilde{e}}(\neg p_\perp)$ is concerned, only the next two options are feasible: $\hat{M}^{\tilde{e}}(\neg p_\perp) = \perp$ or $\perp \prec \hat{M}^{\tilde{e}}(\neg p_\perp)$. From each of them, we will deduce that $M^{\tilde{e}}(p_\perp) = \perp$.

Suppose that $\hat{M}^{\tilde{e}}(\neg p_\perp) = \perp$. Then, according to Equation (3), for each rule $r^{\tilde{e}}$ in $\mathbb{P}^{\tilde{e}}$ of the form³

$$\langle p_\perp \leftarrow_i f_\perp(\neg p_\perp) \& f_c(@^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]); \top \rangle$$

its corresponding rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ in the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ is defined as

$$\langle p_\perp \leftarrow_i f_\perp(\perp) \& f_c(@^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]); \top \rangle$$

Equivalently, as $f_\perp(\perp) = \perp$ and $\&$ is a conjunctor of the extended multi-adjoint lattice, we obtain that $\perp \dot{\&}_i x = \perp$ for each $x \in L$. Hence, the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ in $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ can be rewritten as

$$r_{M^{\tilde{e}}}^{\tilde{e}} : \quad \langle p_\perp \leftarrow_i \perp; \top \rangle$$

Finally, as p_\perp only appears in the rules in $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ which come from constraints and $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, we conclude that⁴

$$\begin{aligned} M^{\tilde{e}}(p_\perp) &= \inf\{c \in L \mid \top \preceq c \dot{\leftarrow}_i M^{\tilde{e}}(\perp)\} = \inf\{c \in L \mid \top \preceq c \dot{\leftarrow}_i \perp\} \\ &= \inf\{c \in L \mid \top \dot{\&}_i \perp \preceq c\} = \inf\{c \in L \mid \perp \preceq c\} = \inf L = \perp \end{aligned}$$

³Since $\mathcal{C}_{\mathbb{P}^e} \neq \emptyset$, we can ensure the existence of at least one rule of the form of $r^{\tilde{e}}$.

⁴Notice that in this case, the lattice L should be a singleton.

Now, suppose that $\perp \prec \hat{M}^{\tilde{e}}(\neg p_{\perp})$. By definition of f_{\perp} , we obtain $f_{\perp}(\hat{M}^{\tilde{e}}(\neg p_{\perp})) = \top$. As a result, according to Equation (3) we can assert that given a rule $r^{\tilde{e}}$ in $\mathbb{P}^{\tilde{e}}$ of the form

$$\langle p_{\perp} \leftarrow_i f_{\perp}(\neg p_{\perp}) \& f_c(@^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]); \top \rangle$$

its corresponding rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ in the reduct $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ is defined as

$$\langle p_{\perp} \leftarrow_i f_c(@^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]); \top \rangle$$

Suppose now that the body of the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ is equal to \top , that is

$$f_c(@^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]) = \top$$

Then the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ can be rewritten as

$$\langle p_{\perp} \leftarrow_i \top; \top \rangle$$

As $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$, it satisfies the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$, and thus

$$\top \preceq M^{\tilde{e}}(p_{\perp}) \leftarrow_i \top$$

Making the corresponding computations and considering the adjoint conjunctive $\&_i$ of the implication, we deduce that

$$\top = \top \&_i \top \preceq M^{\tilde{e}}(p_{\perp})$$

This leads us to a contradiction, since, according to the fact that \neg is a negator operator, the chain $\perp \prec \hat{M}^{\tilde{e}}(\neg p_{\perp}) = \dot{\neg} M^{\tilde{e}}(p_{\perp})$ implies that $M^{\tilde{e}}(p_{\perp}) \neq \top$. Consequently, we can ensure that

$$f_c(@^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]) = \perp$$

for each rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ in $\mathbb{P}_{M^{\tilde{e}}}^{\tilde{e}}$ of the form

$$\langle p_{\perp} \leftarrow_i f_c(@^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]); \top \rangle$$

Hence, we obtain that the rule $r_{M^{\tilde{e}}}^{\tilde{e}}$ is equivalent to

$$\langle p_{\perp} \leftarrow_i \perp; \top \rangle$$

Therefore, following an analogous reasoning to the previous one, we can assert that any stable model $M^{\tilde{e}}$ of the program $\mathbb{P}^{\tilde{e}}$ verifies $M^{\tilde{e}}(p_{\perp}) = \perp$. \square

Theorems 18 and 19 highlights the close connection between an EMALP and its corresponding constraint-free EMALP given in Definition 17. Indeed, one can ensure that the stable models of an EMALP are equivalent to the stable models of its corresponding constraint-free EMALP.

A straightforward outcome from these previous results is the fact that the number of stable models of \mathbb{P}^e is equal to the number of stable models of $\mathbb{P}^{\tilde{e}}$. This fact gives rise to the next result, which concerns the existence and uniqueness of stable models.

Corollary 20. *Let \mathbb{P}^e be an EMALP and $\mathbb{P}^{\tilde{e}}$ the corresponding constraint-free EMALP of \mathbb{P}^e . Then, the following statements hold:*

- *There exists a stable model of \mathbb{P}^e if and only if there exists a stable model of $\mathbb{P}^{\tilde{e}}$.*
- *There exists a unique stable model of \mathbb{P}^e if and only if there exists a unique stable model of $\mathbb{P}^{\tilde{e}}$.*

In the next example, we carry out the translation detailed in Definition 17 in order to obtain a constraint-free EMALP from the EMALP given in Example 16.

Example 21. Coming back to Example 16 and considering the same EMALP \mathbb{P}^e , we have by the procedure in Definition 17, that the rules r_1^e, r_2^e, r_4^e and r_5^e are included in the corresponding constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ of \mathbb{P}^e . In what regards the rules r_3^e , it is transformed in the rule:

$$\langle p_{\perp} \leftarrow_{\mathbb{L}} f_{\perp}(\neg_{\mathbb{1}} p_{\perp}) \&_G f_{0.7}(@_3^e[p; q, s, t]) ; 1 \rangle$$

Notice that, we have arbitrarily chosen the conjunction $\&_G$, but we can make use of any different conjunction whenever it satisfies the boundary condition with \top .

The constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ is then given by the following four rules and one fact

$$\begin{array}{ll} r_1^{\tilde{e}} : \langle p \leftarrow_{\mathbb{P}} @_1^e[p, q, s, t] ; 0.6 \rangle & r_4^{\tilde{e}} : \langle s \leftarrow_{\mathbb{G}} @_4^e[p, q, s, t] ; 0.8 \rangle \\ r_2^{\tilde{e}} : \langle q \leftarrow_{\mathbb{P}} @_2^e[p, q, s, t] ; 0.5 \rangle & r_5^{\tilde{e}} : \langle t \leftarrow_{\mathbb{G}} @_5^e[p, q, s, t] ; 0.8 \rangle \\ r_3^{\tilde{e}} : \langle p_{\perp} \leftarrow_{\mathbb{L}} f_{\perp}(\neg_{\mathbb{1}} p_{\perp}) \&_G f_{0.7}(@_3^e[p; q, s, t]) ; 1 \rangle & \end{array}$$

Now, according to Theorem 18, due to $N^e \equiv \{(p, 9/85), (q, 0.36), (s, 0.8), (t, 0.8)\}$ is a stable model of \mathbb{P}^e , we can assert that the interpretation

$$N^{\tilde{e}} \equiv \{(p, 9/85), (q, 0.36), (s, 0.8), (t, 0.8), (p_{\perp}, 0)\}$$

is a stable model of $\mathbb{P}^{\tilde{e}}$. □

To summarize, given an EMALP \mathbb{P}^e , Definition 17 provides a method in order to obtain a constraint-free EMALP $\mathbb{P}^{\bar{e}}$ whose stable models coincide with the stable models of \mathbb{P}^e . One of the most interesting consequences of this method is that, if one knows that there exists at least a stable model (resp. a unique stable model) of $\mathbb{P}^{\bar{e}}$, then the existence of stable models (resp. a unique stable model) of \mathbb{P}^e is guaranteed. However, there are no results related to the existence or the uniqueness of stable models for either EMALPs or constraint-free EMALP.

Recently, these kind of results were introduced in the particular case of normal residuated logic programming in [18] and in the general case of the multi-adjoint framework in [6]. Therefore, we should try to introduce a transformation mechanism from constraint-free EMALPs to MANLPs, which preserve the semantics (the stable models). As a consequence, we may apply the already introduced results in constraint-free EMALPs, and by the transformation already given in this section they can also be considered in the general case of EMALPs.

5. Translating constraint-free EMALP into MANLP

This section will present a transformation from a constraint-free EMALP to a MANLP, such that the stable models of the original constraint-free EMALP are the same as the stable models of the obtained MANLP. The underlying idea of this translation method is provided next.

Let $\mathbb{P}^{\bar{e}}$ be a constraint-free EMALP and $\mathcal{N}_{\mathbb{P}^{\bar{e}}}$ be the set of propositional symbols that appear in an order-reversing argument of an extended aggregator in the body of some rule in $\mathbb{P}^{\bar{e}}$. Given an operation symbol \neg associated with an involutive negation operator $\dot{\neg}: L \rightarrow L$, suppose that for each $q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}$ there exists a propositional symbol not_q such that any stable model $M^{\bar{e}}$ of $\mathbb{P}^{\bar{e}}$ verifies the equality $M^{\bar{e}}(\text{not}_q) = \hat{M}^{\bar{e}}(\neg q)$. As $\dot{\neg}$ is an involutive negation operator, we obtain that $\dot{\neg} M^{\bar{e}}(\text{not}_q) = \dot{\neg} \hat{M}^{\bar{e}}(\neg q) = \dot{\neg} \dot{\neg} M^{\bar{e}}(q) = M^{\bar{e}}(q)$.

Therefore, for each rule $\langle p \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle \in \mathbb{P}^{\bar{e}}$, we deduce that

$$@^e[p_1, \dots, p_m; M^{\bar{e}}(p_{m+1}), \dots, M^{\bar{e}}(p_n)] = @^e[p_1, \dots, p_m; \dot{\neg} M^{\bar{e}}(\text{not}_{p_{m+1}}), \dots, \dot{\neg} M^{\bar{e}}(\text{not}_{p_n})]$$

This fact leads us to assert that the rule

$$\langle p \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

can be rewritten as the rule

$$\langle p \leftarrow_i @ [p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

where the mapping $\dot{@}: L^n \rightarrow L$ is defined, for all $\vartheta_1, \dots, \vartheta_n \in L$, as

$$\dot{@}[\vartheta_1, \dots, \vartheta_m, \vartheta_{m+1}, \dots, \vartheta_n] = \dot{@}^e[\vartheta_1, \dots, \vartheta_m; \dot{\neg} \vartheta_{m+1}, \dots, \dot{\neg} \vartheta_n]$$

Notice that, as $\dot{@}^e$ is an extended aggregator, it is an order-preserving mapping in the first m arguments, and so $\dot{@}$ is order-preserving in the first m arguments. Furthermore, since $\dot{@}^e$ is an order-reversing mapping in the last $n - m$ arguments and $\dot{\neg}$ is also an order-reversing mapping, we obtain that $\dot{@}$ is an order-preserving mapping in the last $n - m$ arguments. Hence, we conclude that $\dot{@}$ is an aggregator. Therefore, the rule

$$\langle p \leftarrow_i @ [p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

is a weighted rule without negations and so, a particular case of rule allowed in a MANLP.

Last but not least, suppose that the assumption of existing, for each $q \in \mathcal{N}_{\mathbb{P}^{\dot{e}}}$, a propositional symbol not_q such that any stable model $M^{\dot{e}}$ of $\mathbb{P}^{\dot{e}}$ verifies the equality $M^{\dot{e}}(\text{not}_q) = \hat{M}^{\dot{e}}(\neg q)$ does not hold. In that case, we include the next rule in \mathbb{P} for each propositional symbol $q \in \mathcal{N}_{\mathbb{P}^{\dot{e}}}$:

$$r_q : \quad \langle \text{not}_q \leftarrow_j \neg q; \top \rangle$$

It is straightforward that the rule r_q can belong to a MANLP, for each $q \in \mathcal{N}_{\mathbb{P}^{\dot{e}}}$. Furthermore, any stable model M of \mathbb{P} is, by definition, the least model of the reduct \mathbb{P}_M . Since r_q is the unique rule in \mathbb{P} with head not_q , the interpretation of not_q under M is equal to $\hat{M}(\neg q)$, as it will be demonstrated in Proposition 24.

The following definition collects the previous comments in order to formally introduce the proposed transformation.

Definition 22. Let $\mathbb{P}^{\dot{e}}$ be a constraint-free EMALP and $\mathcal{N}_{\mathbb{P}^{\dot{e}}}$ the set of propositional symbols that appear in an order-reversing argument of an extended aggregator in the body of some rule in $\mathbb{P}^{\dot{e}}$. Given the symbol \neg associated with an involutive negation operator $\dot{\neg}: L \rightarrow L$, the corresponding MANLP \mathbb{P} of $\mathbb{P}^{\dot{e}}$ is defined as the following set of rules

$$\begin{aligned} \mathbb{P} = & \{ \langle p \leftarrow_i @ [p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle \mid \\ & \langle p \leftarrow_i @^e [p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle \in \mathbb{P}^{\dot{e}} \} \\ \cup & \{ \langle \text{not}_q \leftarrow_j \neg q; \top \rangle \mid q \in \mathcal{N}_{\mathbb{P}^{\dot{e}}} \} \end{aligned}$$

where $i, j \in \{1, \dots, n\}$, $\text{not}_q \notin \Pi_{\mathbb{P}^{\tilde{e}}}$, for each $q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}$, and $\dot{\@}: L^n \rightarrow L$ is an aggregator which is defined, for all $\vartheta_1, \dots, \vartheta_n \in L$, as

$$\dot{\@}[\vartheta_1, \dots, \vartheta_m, \vartheta_{m+1}, \dots, \vartheta_n] = \dot{\@}^e[\vartheta_1, \dots, \vartheta_m; \dot{\neg} \vartheta_{m+1}, \dots, \dot{\neg} \vartheta_n]$$

Given an interpretation $M: L^n \rightarrow L$, the reduct of \mathbb{P} with respect to M is defined analogously to the procedure explained in Section 2 (Equation (1)).

Now, the stable models of the MANLP \mathbb{P} will be proved to be, as expected, equivalent to the stable models of the EMALP $\mathbb{P}^{\tilde{e}}$, improving and complementing the results given in [11]. To reach this conclusion, two results will be introduced. The first one provides a characterization of the stable models of $\mathbb{P}^{\tilde{e}}$ in terms of a family of stable models of \mathbb{P} .

Theorem 23. *Let $\mathbb{P}^{\tilde{e}}$ be a constraint-free EMALP and \mathbb{P} the corresponding MANLP of $\mathbb{P}^{\tilde{e}}$, $M^{\tilde{e}}: \Pi_{\mathbb{P}^{\tilde{e}}} \rightarrow L$ an interpretation, and $M: \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\} \rightarrow L$ given by $M(p) = M^{\tilde{e}}(p)$ if $p \in \Pi_{\mathbb{P}^{\tilde{e}}}$ and $M(\text{not}_q) = \hat{M}(\neg q)$ for all $q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}$. Then, $M^{\tilde{e}}$ is a stable model of $\mathbb{P}^{\tilde{e}}$ if and only if M is a stable model of \mathbb{P} .*

PROOF. Given two interpretations $M^{\tilde{e}}: \Pi_{\mathbb{P}^{\tilde{e}}} \rightarrow L$ and $N^{\tilde{e}}: \Pi_{\mathbb{P}^{\tilde{e}}} \rightarrow L$, we define $N_M: \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\} \rightarrow L$ as $N_M(p) = N^{\tilde{e}}(p)$ if $p \in \Pi_{\mathbb{P}^{\tilde{e}}}$ and $N_M(\text{not}_q) = \hat{M}^{\tilde{e}}(\neg q)$ for each $q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}$. In the particular case of $M^{\tilde{e}} = N^{\tilde{e}}$, we will have that $M_M: \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\} \rightarrow L$ is defined as $M_M(p) = M^{\tilde{e}}(p)$ if $p \in \Pi_{\mathbb{P}^{\tilde{e}}}$ and $M_M(\text{not}_q) = \hat{M}^{\tilde{e}}(\neg q) = \dot{\neg} \hat{M}^{\tilde{e}}(q) = \dot{\neg} M_M(q) = \hat{M}_M(\neg q)$ for each $q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}$, and we simply write M instead of M_M .

First of all, we will prove that $N^{\tilde{e}}$ is a model of the reduct $\mathbb{P}_{M^{\tilde{e}}}$ if and only if N_M is a model of the reduct \mathbb{P}_M . The first step for getting this equivalence will be to prove that an interpretation $N: \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\} \rightarrow L$ satisfies the rule

$$\langle \text{not}_q \leftarrow_j \hat{M}(\neg q); \top \rangle$$

in the reduct \mathbb{P}_M if and only if $M(\text{not}_q) \preceq N(\text{not}_q)$. Since the following chain straightforwardly holds

$$\top \&_j N(\hat{M}(\neg q)) = N(\hat{M}(\neg q)) = \hat{M}(\neg q) = M(\text{not}_q)$$

and $(\&_j, \leftarrow_j)$ is an adjoint pair, we obtain that

$$\top = \hat{N}(\text{not}_q \leftarrow_j \hat{M}(\neg q)) \quad \text{iff} \quad M(\text{not}_q) = \top \&_j N(\hat{M}(\neg q)) \preceq N(\text{not}_q)$$

which leads to the proof of this first claim.

Consequently, in particular, as $N_M(\text{not}_q) = \hat{M}^{\tilde{e}}(-q) = M(\text{not}_q)$, the interpretation N_M satisfies the rules with head not_q in the reduct \mathbb{P}_M . Now, notice that a rule

$$\langle p \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

belongs to the program $\mathbb{P}^{\tilde{e}}$ if and only if the rule

$$\langle p \leftarrow_i @[p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

belongs to \mathbb{P} , and thus, due to $@$ is order-preserving in every argument, this is equivalent to this rule belongs to the reduct \mathbb{P}_M . Therefore, regarding the reducts $\mathbb{P}_{M^{\tilde{e}}}$ and \mathbb{P}_M , we deduce that the rule

$$\langle p \leftarrow_i @^e[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]; \vartheta \rangle$$

belongs to $\mathbb{P}_{M^{\tilde{e}}}$ if and only if

$$\langle p \leftarrow_i @[p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

belongs to \mathbb{P}_M . Taking into account this fact, we will see that $N^{\tilde{e}}$ is a model of $\mathbb{P}_{M^{\tilde{e}}}$ if and only if N_M is a model of the reduct \mathbb{P}_M . Indeed, the interpretation N_M satisfies the rule

$$\langle p \leftarrow_i @[p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

if and only if

$$\vartheta \preceq \hat{N}_M(p \leftarrow_i @[p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}])$$

that is

$$\vartheta \preceq N_M(p) \dot{\leftarrow}_i \dot{@}[N_M(p_1), \dots, N_M(p_m), N_M(\text{not}_{p_{m+1}}), \dots, N_M(\text{not}_{p_n})]$$

which is equivalent, according to the definition of $@$, to the inequality

$$\vartheta \preceq N_M(p) \dot{\leftarrow}_i @^e[N_M(p_1), \dots, N_M(p_m); \dot{\neg} N_M(\text{not}_{p_{m+1}}), \dots, \dot{\neg} N_M(\text{not}_{p_n})]$$

Furthermore, as $N_M(p) = N^{\tilde{e}}(p)$ and $N_M(\text{not}_q) = \dot{\neg} M^{\tilde{e}}(q)$, for each $q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}$, we obtain that N_M satisfies the inequality above if and only if $N^{\tilde{e}}$ satisfies the following one

$$\vartheta \preceq N^{\tilde{e}}(p) \dot{\leftarrow}_i \dot{@}^e[N^{\tilde{e}}(p_1), \dots, N^{\tilde{e}}(p_m); \dot{\neg} \dot{\neg} M^{\tilde{e}}(p_{m+1}), \dots, \dot{\neg} \dot{\neg} M^{\tilde{e}}(p_n)]$$

Taking into account that $\dot{\neg}$ is by hypothesis an involutive negation, the last inequality is equivalent to

$$\vartheta \preceq N^{\tilde{e}}(p) \dot{\leftarrow}_i \textcircled{\text{e}}[N^{\tilde{e}}(p_1), \dots, N^{\tilde{e}}(p_m); M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)] \quad (4)$$

Furthermore, as $N^{\tilde{e}}(x) = x$ for each $x \in L$ and $M^{\tilde{e}}(p_i) \in L$ for each $i \in \{m+1, \dots, n\}$, we can assert that $N^{\tilde{e}}(M^{\tilde{e}}(p_i)) = M^{\tilde{e}}(p_i)$, for each $i \in \{m+1, \dots, n\}$. Therefore, Equation (4) can be rewritten as

$$\vartheta \preceq N^{\tilde{e}}(p) \dot{\leftarrow}_i \textcircled{\text{e}}[N^{\tilde{e}}(p_1), \dots, N^{\tilde{e}}(p_m); N^{\tilde{e}}(M^{\tilde{e}}(p_{m+1})), \dots, N^{\tilde{e}}(M^{\tilde{e}}(p_n))]$$

Equivalently

$$\vartheta \preceq \hat{N}^{\tilde{e}}(p \leftarrow_i \textcircled{\text{e}}[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)])$$

Hence, we conclude that N_M satisfies the rule

$$\langle p \leftarrow_i \textcircled{\text{e}}[p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

in the reduct \mathbb{P}_M if and only if $N^{\tilde{e}}$ satisfies the rule

$$\langle p \leftarrow_i \textcircled{\text{e}}[p_1, \dots, p_m; M^{\tilde{e}}(p_{m+1}), \dots, M^{\tilde{e}}(p_n)]; \vartheta \rangle$$

in the program $\mathbb{P}_{M^{\tilde{e}}}$. Therefore, $N^{\tilde{e}}$ is a model of $\mathbb{P}_{M^{\tilde{e}}}$ if and only if N_M is a model of the reduct \mathbb{P}_M .

To finish with this demonstration, we will see that $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}$ if and only if M is the least model of \mathbb{P}_M by reductio ad absurdum. In fact, suppose that M is the least model of \mathbb{P}_M but there exists $N^{\tilde{e}}: \Pi_{\mathbb{P}^{\tilde{e}}} \rightarrow L$ with $N^{\tilde{e}} \sqsubset M^{\tilde{e}}$ such that $N^{\tilde{e}}$ is a model of $\mathbb{P}_{M^{\tilde{e}}}$. Then, by the previous proved equivalence, the interpretation $N_M: \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\} \rightarrow L$, given by $N_M(p) = N^{\tilde{e}}(p)$ if $p \in \Pi_{\mathbb{P}^{\tilde{e}}}$ and $N_M(\text{not}_q) = \hat{M}^{\tilde{e}}(\neg q)$ otherwise, is a model of \mathbb{P}_M . As a consequence, we obtain that $N_M \sqsubset M$ is straightforwardly satisfied and so, we obtain that M is not the least model of \mathbb{P}_M , in contradiction with the hypothesis.

Finally, suppose that $M^{\tilde{e}}$ is the least model of $\mathbb{P}_{M^{\tilde{e}}}$ but M is not the least model of \mathbb{P}_M . As a result, we can assert that there exists a model $N: \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\} \rightarrow L$ of \mathbb{P}_M verifying $N \sqsubset M$. Then, there exists $p \in \Pi_{\mathbb{P}^{\tilde{e}}} \cup \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\}$ such that $N(p) \prec M(p)$. Notice that, if $p \in \{\text{not}_q \mid q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}\}$, that is, there exists $q \in \mathcal{N}_{\mathbb{P}^{\tilde{e}}}$ such that $p = \text{not}_q$, then the inequality $N(\text{not}_q) \prec M(\text{not}_q) = \hat{M}^{\tilde{e}}(\neg q)$ holds, which contradicts

that N satisfies the rule $\langle \text{not}_q \leftarrow_j \hat{M}(\neg q); \top \rangle$ in the reduct \mathbb{P}_M , as we showed at the beginning of this proof. As a consequence, we also have that $N(\text{not}_q) = \hat{M}^{\bar{e}}(\neg q)$, for all $q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}$.

Hence, we can assert that there exists $p' \in \Pi_{\mathbb{P}^{\bar{e}}}$ such that $N(p') \prec M(p')$. As a result, the interpretation $N^{\bar{e}}: \Pi_{\mathbb{P}^{\bar{e}}} \rightarrow L$ given by $N^{\bar{e}}(p) = N(p)$, for all $p \in \Pi_{\mathbb{P}^{\bar{e}}}$, satisfies $N^{\bar{e}} \sqsubset M^{\bar{e}}$. Since N is a model of \mathbb{P}_M , with $N(\text{not}_q) = \hat{M}^{\bar{e}}(\neg q)$, for all $q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}$, we deduce that $N^{\bar{e}}$ is a model of $\mathbb{P}_{M^{\bar{e}}}^{\bar{e}}$, and thus $M^{\bar{e}}$ is not the least model of $\mathbb{P}_{M^{\bar{e}}}^{\bar{e}}$, in contradiction with the hypothesis.

Therefore, we conclude that $M^{\bar{e}}$ is the least model of $\mathbb{P}_{M^{\bar{e}}}^{\bar{e}}$ if and only if M is the least model of \mathbb{P}_M , that is, $M^{\bar{e}}$ is a stable model of $\mathbb{P}^{\bar{e}}$ if and only if M is a stable model of \mathbb{P} . \square

Theorem 23 establishes that each stable model of $\mathbb{P}^{\bar{e}}$ is equivalent to a stable model of \mathbb{P} in the set

$$S = \{M \in \mathcal{I}_{\mathcal{L}} \mid M(\text{not}_q) = \hat{M}(\neg q), \text{ for all } q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}\}$$

In the following, we will see that any stable model of the MANLP \mathbb{P} belongs to S . As a result, any stable model of \mathbb{P} is taken into account in Theorem 23, and thus we can conclude that the set of stable models of $\mathbb{P}^{\bar{e}}$ actually coincide with the set of stable models of \mathbb{P} .

Theorem 24. *Let $\mathbb{P}^{\bar{e}}$ be a constraint-free EMALP, \mathbb{P} be its corresponding MANLP and $\mathcal{N}_{\mathbb{P}^{\bar{e}}}$ be the set of propositional symbols that appear in an order-reversing argument position of an extended aggregator in the body of some rule in $\mathbb{P}^{\bar{e}}$. Then, any stable model M of the MANLP \mathbb{P} satisfies $M(\text{not}_q) = \hat{M}(\neg q)$, for all $q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}$.*

PROOF. Given a stable model M of the MANLP \mathbb{P} and $q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}$, in particular, M satisfies the rule

$$\langle \text{not}_q \leftarrow_j \hat{M}(\neg q); \top \rangle$$

in the reduct \mathbb{P}_M . Hence, from the first equivalence proved in the proof of Theorem 23, we deduce that $\hat{M}(\neg q) \preceq M(\text{not}_q)$. On the other hand, according to the fact that M is the least model of \mathbb{P}_M and $\langle \text{not}_q \leftarrow_j \hat{M}(\neg q); \top \rangle$ is the unique rule with head not_q in \mathbb{P}_M , we have that $M(\text{not}_q)$ takes the least value such that $\hat{M}(\neg q) \preceq M(\text{not}_q)$. Hence, we can assert that $M(\text{not}_q) = \hat{M}(\neg q)$, and this equality holds for each $q \in \mathcal{N}_{\mathbb{P}^{\bar{e}}}$. \square

From Theorems 23 and 24, we can assert that the stable models of an EMALP coincide with the stable models of its transformed MANLP. As a consequence, the number of stable models of $\mathbb{P}^{\tilde{e}}$ is equal to the number of stable models of \mathbb{P} .

This fact leads us to deduce that there exists at least a stable model of $\mathbb{P}^{\tilde{e}}$ if and only if there exists at least a stable model of \mathbb{P} . An equivalent outcome is obtained regarding the uniqueness of stable models of $\mathbb{P}^{\tilde{e}}$ and \mathbb{P} . These results are formalized as follows.

Corollary 25. *Let $\mathbb{P}^{\tilde{e}}$ be a constraint-free EMALP and \mathbb{P} the corresponding MANLP of $\mathbb{P}^{\tilde{e}}$. Then, the following statements hold:*

- *There exists a stable model of $\mathbb{P}^{\tilde{e}}$ if and only if there exists a stable model of \mathbb{P} .*
- *There exists a unique stable model of $\mathbb{P}^{\tilde{e}}$ if and only if there exists a unique stable model of \mathbb{P} .*

As a continuation of Examples 16 and 21, we will complete the translation from an EMALP into a MANLP with the same stable models.

Example 26. Consider the EMALP \mathbb{P}^e defined in Example 16 and its corresponding constraint-free EMALP $\mathbb{P}^{\tilde{e}}$ given in Example 21. Notice that, the propositional symbols that appear in an order-reversing argument of an extended aggregator in the body of the rules in $\mathbb{P}^{\tilde{e}}$ are s, t (rules $r_1^{\tilde{e}}$ and $r_2^{\tilde{e}}$) and q, p_{\perp} (rule $r_3^{\tilde{e}}$). Hence, we obtain that $\mathcal{N}_{\mathbb{P}^{\tilde{e}}} = \{q, s, t, p_{\perp}\}$.

According to Definition 22, in order to define the corresponding MANLP \mathbb{P} of $\mathbb{P}^{\tilde{e}}$ we only need an involutive negation in $[0, 1]$. For the sake of simplicity, as the negation \neg_1 is an involutive negation, we will make use of this operator to define the MANLP \mathbb{P} . Hence, we obtain that the corresponding MANLP \mathbb{P} of $\mathbb{P}^{\tilde{e}}$ is defined as the following seven rules and one fact

$$\begin{array}{ll}
r_1 : \langle p \leftarrow_{\text{P}} @_1[p, q, \text{not}_s, \text{not}_t] ; 0.5 \rangle & r_6 : \langle \text{not}_q \leftarrow_{\text{G}} \neg_1 q ; \top \rangle \\
r_2 : \langle q \leftarrow_{\text{P}} @_2[p, q, \text{not}_s, \text{not}_t] ; 0.6 \rangle & r_7 : \langle \text{not}_s \leftarrow_{\text{G}} \neg_1 s ; \top \rangle \\
r_3 : \langle p_{\perp} \leftarrow_{\text{L}} @_3[p, \text{not}_q, \text{not}_{p_{\perp}}] ; 1 \rangle & r_8 : \langle \text{not}_t \leftarrow_{\text{G}} \neg_1 t ; \top \rangle \\
r_4 : \langle s \leftarrow_{\text{G}} @_4[p, q, s, t] ; 0.8 \rangle & r_9 : \langle \text{not}_{p_{\perp}} \leftarrow_{\text{G}} \neg_1 p_{\perp} ; \top \rangle \\
r_5 : \langle t \leftarrow_{\text{G}} @_5[p, q, s, t] ; 0.8 \rangle &
\end{array}$$

where the aggregator operators $@_1, @_2, @_4, @_5 : [0, 1]^4 \rightarrow [0, 1]$ and $@_3 : [0, 1]^5 \rightarrow [0, 1]$ are defined as

$$\begin{aligned}
@_1[p, q, \text{not}_s, \text{not}_t] &= @_1^e[p, q; \neg_1(\text{not}_s), \neg_1(\text{not}_t)] = \min \left\{ \frac{q}{\neg_1(\text{not}_s) + \neg_1(\text{not}_t) + 0.1}, 1 \right\} \\
@_2[p, q, \text{not}_s, \text{not}_t] &= @_2^e[p, q; \neg_1(\text{not}_s), \neg_1(\text{not}_t)] = \max \left\{ \neg_1(\neg_1(\text{not}_s)), \neg_2(\neg_1(\text{not}_t)) \right\}
\end{aligned}$$

$$\begin{aligned}
@_3[p, \text{not}_q, \text{not}_{p_\perp}] &= f_\perp(\neg_1(\neg_1(\text{not}_{p_\perp}))) \&_G f_{0.7}(\neg_1(\neg_1(\text{not}_q))) \\
@_4[p, q, s, t] &= @_4^e[p, q, s, t] = 1 \\
@_5[p, q, s, t] &= @_5^e[p, q, s, t] = \max\{s, 0.7\}
\end{aligned}$$

Since $N^{\dot{e}} \equiv \{(p, 9/85), (q, 0.36), (s, 0.8), (t, 0.8), (p_\perp, 0)\}$ is a stable model of the constraint-free EMALP $\mathbb{P}^{\dot{e}}$, Theorem 23 leads us to conclude that the interpretation $N \equiv \{(p, 9/85), (q, 0.36), (s, 0.8), (t, 0.8), (p_\perp, 0), (\text{not}_q, 0.64), (\text{not}_s, 0.2), (\text{not}_t, 0.2), (\text{not}_{p_\perp}, 1)\}$ is a stable model of the MANLP \mathbb{P} . \square

Due to the fact that the semantics of an extended multi-adjoint logic program is defined in terms of the stable models of the program, ensuring the existence of stable models becomes a crucial task in order to define its semantics.

According to Corollary 25, we obtain that any result related to the existence (resp. unicity) of stable models for MANLPs is likely to be used in order to guarantee the existence (resp. unicity) of stable models of the original EMALP. Indeed, Theorem 8 leads us to infer the following result.

Theorem 27. *Let $(K, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, @_1^e, \dots, @_k^e)$ be an extended multi-adjoint lattice where K is a non-empty convex compact set in an euclidean space and $\mathbb{P}^{\dot{e}}$ a finite constraint-free EMALP defined on this lattice. If $\dot{\&}_1, \dots, \dot{\&}_n$ and the extended aggregator operators in the body of the rules of $\mathbb{P}^{\dot{e}}$ are continuous operators, and there exists a continuous involutive negation $\dot{\neg}: K \rightarrow K$ then $\mathbb{P}^{\dot{e}}$ has at least a stable model.*

PROOF. Let \mathbb{P} be the corresponding MANLP of $\mathbb{P}^{\dot{e}}$ given in Definition 22 by means of the negation operator \neg . The MANLP \mathbb{P} is then defined on the multi-adjoint normal lattice $(K, \preceq, \leftarrow_1, \&_1, \dots, \leftarrow_n, \&_n, \neg)$, being $\dot{\&}_1, \dots, \dot{\&}_n$ and $\dot{\neg}$ continuous operators. Clearly, since $\dot{\neg}$ is by hypothesis a continuous mapping, then the aggregator operator in the body of the rule

$$\langle \text{not}_q \leftarrow_j \neg q; \top \rangle$$

is a continuous operator, for each $q \in \mathcal{N}_{\mathbb{P}^{\dot{e}}}$.

Furthermore, for each rule

$$\langle p \leftarrow_i @^e[p_1, \dots, p_m; p_{m+1}, \dots, p_n]; \vartheta \rangle$$

in $\mathbb{P}^{\dot{e}}$, taking into account that $@^e$ and $\dot{\neg}$ are continuous mappings, we obtain that the aggregator $\dot{@}: L^n \rightarrow L$ defined as

$$\dot{@}[\vartheta_1, \dots, \vartheta_m, \vartheta_{m+1}, \dots, \vartheta_n] = @^e[\vartheta_1, \dots, \vartheta_m; \dot{\neg} \vartheta_{m+1}, \dots, \dot{\neg} \vartheta_n]$$

for all $\vartheta_1, \dots, \vartheta_n \in L$, is a continuous mapping. Therefore, the aggregator operator $\dot{\textcircled{a}}$ associated with the symbol \textcircled{a} in the body of the rule

$$\langle p \leftarrow_i \textcircled{a}[p_1, \dots, p_m, \text{not}_{p_{m+1}}, \dots, \text{not}_{p_n}]; \vartheta \rangle$$

is a continuous mapping.

Hence, we can apply Theorem 8 to the MANLP \mathbb{P} , from which we obtain that \mathbb{P} has at least a stable model M . Then, given a stable model M of \mathbb{P} , Proposition 24 allows us to assert that $M(\text{not}_q) = \hat{M}(\neg q)$, for each $q \in \mathcal{N}_{\mathbb{P}^{\hat{e}}}$. Finally, as stated by Theorem 23, we conclude that the interpretation $M^{\hat{e}}: \Pi_{\mathbb{P}^{\hat{e}}} \rightarrow L$ given by $M^{\hat{e}}(p) = M(p)$, for each $p \in \Pi_{\mathbb{P}^{\hat{e}}}$, is a stable model of $\mathbb{P}^{\hat{e}}$. Thus, there exists at least a stable model of $\mathbb{P}^{\hat{e}}$. \square

In the following, an illustrative example shows how Theorem 27 can be used to ensure the existence of stable models for a general EMALP.

Example 28. Suppose that we remove the rule r_3^e in the EMALP \mathbb{P}^e given in Example 16, constructing a new EMALP \mathbb{P}^{e*} . This may be interesting if, for instance, some features of the problem that is being simulated by the EMALP change and these changes imply that the condition of the models of \mathbb{P} satisfying that the evaluation of q is greater than 0.3 is not demanded anymore. The EMALP \mathbb{P}^{e*} is then defined on the multi-adjoint lattice $([0, 1], \leq, \leftarrow_G, \&_G, \leftarrow_P, \&_P, \leftarrow_L, \&_L)$ as the following four rules:

$$\begin{array}{ll} r_1^e: \langle p \leftarrow_P \textcircled{a}_1^e[p, q, s, t]; 0.5 \rangle & r_4^e: \langle s \leftarrow_G \textcircled{a}_4^e[p, q, s, t]; 0.8 \rangle \\ r_2^e: \langle q \leftarrow_P \textcircled{a}_2^e[p, q, s, t]; 0.6 \rangle & r_5^e: \langle t \leftarrow_G \textcircled{a}_5^e[p, q, s, t]; 0.8 \rangle \end{array}$$

being $\textcircled{a}_1^e, \textcircled{a}_2^e, \textcircled{a}_4^e, \textcircled{a}_5^e: [0, 1]^4 \rightarrow [0, 1]$ defined in Example 16.

Observe that $[0, 1]$ is a non-empty convex compact set in the euclidean space $([0, 1], +, *, \mathbb{R})$, being $+$ and $*$ the usual sum and product in \mathbb{R} , respectively. Furthermore, the conjunctions $\&_G$ and $\&_P$ and the extended aggregators $\textcircled{a}_1^e, \textcircled{a}_2^e, \textcircled{a}_4^e$ and \textcircled{a}_5^e are continuous operators in $[0, 1]^4$. Taking into account that, for instance, the mapping $\neg: [0, 1] \rightarrow [0, 1]$ given by $\neg(x) = 1 - x$ is a continuous involutive negation, Theorem 27 leads us to conclude that there exists at least a stable model of \mathbb{P}^{e*} . For instance, the interpretation N^e given by $N^e \equiv \{(p, 9/85), (q, 0.36), (s, 0.8), (t, 0.8)\}$ is a stable model of \mathbb{P}^{e*} . \square

Therefore, the introduced results in this section have completed the transformation from a general EMALP to a semantically equivalent MANLP. Specifically, each stable model of the transformed MANLP provides another stable model of the original EMALP.

6. Conclusions and future work

An extension of multi-adjoint normal logic programming has been presented. In this new kind of programs, special rules called constraints have been included and aggregator operators with order-reversing arguments are allowed to appear in the body of the rules. This consideration allows, for instance, to consider multi-adjoint normal logic programs with multiple negations.

Moreover, this extension generalizes the one given in [11, 15] and considerably increases the flexibility of MANLPs. After presenting the syntax and the semantics of EMALPs, a procedure in order to translate an EMALP into a constraint-free EMALP, preserving the semantic given by the stable models, has been provided. Then, a method to simulate a constraint-free EMALP by means of a semantically equivalent MANLP has been presented. These two procedures make possible, for example, that a user can consider results associated with stable models of MANLPs in order to obtain information about the stable models of an EMALP. For instance, we have shown that Theorem 8, presented in [6], provides sufficient conditions under which the existence of stable models of a constraint-free EMALP can be ensured. The uniqueness results given in [6] can also be applied.

Since the auxiliary mappings f_c considered in the transformation from EMALPs to constraint-free EMALPs are non-continuous, for each $c \in L$, Theorem 6 cannot be used in order to guarantee the existence of stable models for an EMALP with constraints. Therefore, new results need to be studied in the future. For example, Madrid proposes in [15] a feasible procedure to come to this aim for residuated logic programs on the unit interval with constraints which can be generalized to the case of EMALPs. This extension is one of the proposals for further research. Moreover, we have other two feasible alternatives to be considered as a future work:

- Obtaining a transformation from EMALPs to EMALPs without constraints by means of continuous operations, which would allow us to make use of Theorem 8.
- Providing sufficient conditions that ensure the existence of stable models of MANLPs and do not require the continuity of the operators in the body of the rules.

Finally, the study of inconsistency and incoherent information [1, 3, 5, 16, 17, 21, 24] in EMALPs will be another important task in the future.

- [1] D. Buchman and D. Poole. Negative probabilities in probabilistic logic programs. *International Journal of Approximate Reasoning*, 83:43 – 59, 2017.
- [2] H. Bustince, N. Madrid, and M. Ojeda-Aciego. The notion of weak-contradiction: Definition and measures. *IEEE Transactions on Fuzzy Systems*, 23(4):1057–1069, aug 2015.
- [3] M. E. Cornejo, D. Lobo, and J. Medina. Selecting the coherence notion in multi-adjoint normal logic programming. *Lecture Notes in Computer Science*, 10305:447–457, 2017.
- [4] M. E. Cornejo, D. Lobo, and J. Medina. Characterizing fuzzy y-models in multi-adjoint normal logic programming. In J. Medina, M. Ojeda-Aciego, J. L. Verdegay, I. Perfilieva, B. Bouchon-Meunier, and R. R. Yager, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, pages 541–552, Cham, 2018. Springer International Publishing.
- [5] M. E. Cornejo, D. Lobo, and J. Medina. Measuring the incoherent information in multi-adjoint normal logic programs. *Advances in Intelligent Systems and Computing*, 641:521–533, 2018.
- [6] M. E. Cornejo, D. Lobo, and J. Medina. Syntax and semantics of multi-adjoint normal logic programming. *Fuzzy Sets and Systems*, 345:41 – 62, 2018.
- [7] 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.
- [8] C. V. Damásio and L. M. Pereira. Hybrid probabilistic logic programs as residuated logic programs. In *Logics in Artificial Intelligence, JELIA'00*, pages 57–73. Lecture Notes in Artificial Intelligence 1919, 2000.
- [9] 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.
- [10] D. Dubois, J. Lang, and H. Prade. Automated reasoning using possibilistic logic: semantics, belief revision, and variable certainty weights. *Knowledge and Data Engineering, IEEE Transactions on*, 6(1):64–71, feb 1994.
- [11] J. Janssen, S. Schockaert, D. Vermeir, and M. D. Cock. A core language for fuzzy answer set programming. *International Journal of Approximate Reasoning*, 53(4):660–692, jun 2012.
- [12] P. Julián, G. Moreno, and J. Penabad. An improved reductant calculus using fuzzy partial evaluation techniques. *Fuzzy Sets and Systems*, 160:162–181, 2009.
- [13] 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. Theme: Logic and Computer Science.
- [14] J. Lloyd. *Foundations of Logic Programming*. Springer Verlag, 1987.
- [15] N. Madrid. A sufficient condition to guarantee the existence of fuzzy stable models on residuated logic programming with constraints. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, jul 2017.
- [16] N. Madrid and M. Ojeda-Aciego. On coherence and consistence in fuzzy answer set semantics for residuated logic programs. *Lect. Notes in Computer Science*, pages 5571:60–67, 2009.
- [17] N. Madrid and M. Ojeda-Aciego. Measuring inconsistency in fuzzy answer set se-

- mantics. *IEEE Transactions on Fuzzy Systems*, 19(4):605–622, Aug. 2011.
- [18] N. Madrid and M. Ojeda-Aciego. On the existence and unicity of stable models in normal residuated logic programs. *International Journal of Computer Mathematics*, 89(3):310–324, 2012.
 - [19] N. Madrid and M. Ojeda-Aciego. On the measure of incoherent information in extended multi-adjoint logic programs. In *2013 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*. Institute of Electrical and Electronics Engineers (IEEE), apr 2013.
 - [20] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. In *Logic Programming and Non-Monotonic Reasoning, LP-NMR’01*, pages 351–364. Lecture Notes in Artificial Intelligence 2173, 2001.
 - [21] E. Merhej, S. Schockaert, and M. D. Cock. Repairing inconsistent answer set programs using rules of thumb: A gene regulatory networks case study. *International Journal of Approximate Reasoning*, 83:243 – 264, 2017.
 - [22] P. J. Morcillo and G. Moreno. *Efficient Unfolding of Fuzzy Connectives for Multi-adjoint Logic Programs*, pages 57–78. Springer International Publishing, Cham, 2018.
 - [23] G. Moreno, J. Penabad, and J. A. Riaza. *Symbolic Unfolding of Multi-adjoint Logic Programs*, pages 43–51. Springer International Publishing, Cham, 2019.
 - [24] C. Redl. Explaining inconsistency in answer set programs and extensions. In M. Balduccini and T. Janhunen, editors, *Logic Programming and Nonmonotonic Reasoning*, pages 176–190, Cham, 2017. Springer International Publishing.
 - [25] H. Strass, S. Muñoz-Hernández, and V. Pablos-Ceruelo. Operational semantics for a fuzzy logic programming system with defaults and constructive answers, 2009.
 - [26] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5(2):285–309, 1955.
 - [27] P. Vojtáš. Fuzzy logic programming. *Fuzzy sets and systems*, 124(3):361–370, 2001.