Proceedings of the 11th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2011 26-30 June 2011.

Dedekind-MacNeille Completion and Multi-adjoint Lattices

P.J. Morcillo, G. Moreno, J. Penabad and C. Vázquez[1]

[1] *University of Castilla-La Mancha, Faculty of Computer Science Engineering 02071, Albacete (Spain),*

emails: {pmorcillo,cvazquez}@dsi.uclm.es
{Gines.Moreno,Jaime.Penabad}@uclm.es

Abstract

Among other applications, *multi-adjoint lattices* have been successfully used for modeling flexible notions of truth-degrees in the fuzzy extension of logic programming called MALP (*Multi-Adjoint Logic Programming*). In this paper we focus in the completion of such mathematical construct by adapting the classical notion of *Dedekind-MacNeille* in order to relax this usual hypothesis on such kind of ordered sets. On the practical side, we show too the role played by multi-adjoint lattices into the *"Fuzzy LOgic Programming Environment for Research" FLOPER* that we have developed in our research group.

Key words: Lattice, Completion, Multi-adjoint Logic Programming.
MSC 2000: Lattice, Dedekind-MacNeille Completion.

1 Introduction

In essence, the notion of multi-adjoint lattice considers a carrier set L (whose elements verify a concrete ordering $\leq$) equipped with a set of connectives like implications, conjunctions, disjunctions and other *hybrid aggregators*, with the particularity that for each implication symbol there exists its *adjoint conjunction* used for modeling the *modus ponens* inference rule in a fuzzy setting. For instance, some adjoint pairs, i.e. conjunctors and implications, in the lattice $([0,1], \leq)$ are presented below, where labels L, G and P mean respectively *Łukasiewicz logic*, *Gödel intuitionistic logic* and *product logic* (with different capabilities for modeling *pessimist*, *optimist* and *realistic scenarios*, respectively):

$$\begin{array}{llllll}
\&_{\texttt{P}}(x,y) & \triangleq & x * y & \leftarrow_{\texttt{P}}(x,y) & \triangleq & \min(1, x/y) \quad \textit{Product} \\
\&_{\texttt{G}}(x,y) & \triangleq & \min(x,y) & \leftarrow_{\texttt{G}}(x,y) & \triangleq & \begin{cases} 1 & \text{if } y \leq x \\ x & \text{otherwise} \end{cases} \quad \textit{Gödel} \\
\&_{\texttt{L}}(x,y) & \triangleq & \max(0, x+y-1) & \leftarrow_{\texttt{L}}(x,y) & \triangleq & \min\{x-y+1, 1\} \quad \textit{Łukasiewicz}
\end{array}$$

Moreover, in the MALP framework [25, 23, 24], each program has its own associated multi-adjoint lattice and each program rule (very similar to a Prolog clause[1]) is "weighted" with an element of L, whereas the components in its body are *linked* with connectives of the lattice. For instance, in the following propositional MALP program (where obviously $@_{\texttt{aver}}$ refers to the classical average aggregator):

$$\begin{array}{llll} p & \leftarrow_{\texttt{P}} & @_{\texttt{aver}}(q, r) & \textit{with} \quad 0.9 \\ q & \leftarrow & & \textit{with} \quad 0.8 \\ r & \leftarrow & & \textit{with} \quad 0.6 \end{array}$$

the last two rules directly assign truth values 0.8 and 0.6 to propositional symbols q and r, respectively, and the execution of p using the first rule, simply consists in evaluating the expression "$\&_{\texttt{P}}(0.9, @_{\texttt{aver}}(0.8, 0.6))$", which returns the final truth degree 0.63.

Anyway, although the class of multi-adjoint lattices is wide enough to model real-world application written with the MALP language [2], in [28, 26] we have proposed some debugging/tracing techniques based on lattices (whose elements are strings of characters) which do not fully accomplish with the hypothesis of complete lattice required by multi-adjoint lattices.

Motivated by this fact, in Sections 2 and 3 of this paper, we give a first step in solving such problem, inspired by the Dedekind-MacNeille completion of an ordered set P (also known as the normal completion of P and the completion by cuts) which was originally proposed by M. MacNeille in 1937 (see [18]) as an extension of the famous definitions of real numbers conceived as cuts from rational ones due to Dedekind[2] in 1872 [8].

The Dedekind-MacNeille completion is directly related to the concept of the canonical extension that was firstly introduced, for Boolean algebras, in [13] and that arises from Stone's duality theorem. Although out of the scope of this paper, in the future we plan to analyze some canonical extensions for multi-adjoint lattices, formally introduced in [9] (see also [10, 29], which study the completion of an n-ordered set), that have associated monotone operators and analyze the results especially for the habitual domains in multi-adjoint logic programming of bilattices and trilattices [20, 21, 19, 4, 5, 3, 6].

On the other hand, the last part of this paper is concerned with implementation and practical developments achieved in our group. More exactly, in Section 4 we present the FLOPER tool [27, 28, 26], which currently is useful for compiling (to standard Prolog code), executing and debugging MALP programs in a safe way and it is ready for being extended in the near future with powerful transformation and optimization techniques designed in our research group in the recent past [14, 11]. In this paper, we will focus in the management of multi-adjoint lattices performed by FLOPER, where such constructs can be easily expressed by means of a set Prolog clauses. Moreover, for a given program and goal, we will see too that different solution could be achieved depending on the currently loaded lattice (which can be changed as much as wanted even in a single work session).

[1] We assume familiarity with pure Logic Programming and its most popular language Prolog [16].

[2] This German mathematician was pupil of Gauss in Gotinga and nowadays is considered one of the founders of modern algebra.

2 Dedekind-MacNeille Completion

We start this section by giving some basic definitions before addressing the concept of Dedekind-MacNeille completion.

Definition 2.1. *Let* $(P, \leq)$ *be an ordered set and* $Q \subset P$*.*

i) *Q is a* down-set *(also called decreasing set and order ideal) if whenever* $x \in Q, y \in P$ *with* $y \leq x$*, we have* $y \in Q$*.*

ii) *Dually, Q is an* up-set *(also called increasing set and order filter) if whenever* $x \in Q, y \in P$ *with* $y \geq x$*, we have* $y \in Q$*.*

In what follows we will use the set, read as "down" Q, $\downarrow Q = \{y \in P : \exists x \in Q \ / \ y \leq x\}$, in particular, $\downarrow x = \{y \in P : y \leq x\}$ that is called *principal down-set* and also *principal ideal generated by* x (obviously, $\downarrow \{x\} = \downarrow x$). The set of all down-sets of P is denoted by $\mathcal{O}(\mathcal{P})$ which is an ordered set under the usual inclusion ordering. Similary, "up" Q, $\uparrow Q = \{y \in P : \exists x \in Q \ / \ x \leq y\}$.

If P is an ordered set and $X = \{\downarrow x : x \in P\}$ (ordered by inclusion), $Y = \{\uparrow x : x \in P\}$ (ordered by reverse inclusion), then the maps $\triangleright : X \to Y$ given by $\downarrow x \mapsto \uparrow x$ and $\triangleleft : Y \to X$ given by $\uparrow x \mapsto \downarrow x$ forms a *Galois connection* between X and Y. This notion appears in [22, 19], where a fuzzy generalization of the formal concept analysis was presented. In particular, multi-adjoint concept lattices were introduced into the MALP framework for application to formal concept analysis.

Definition 2.2. *Let* $(P, \leq), (Q, \leq)$ *be ordered sets. A map* $\varphi : P \to Q$ *is said to be*

i) *order-preserving (or monotone) if* $x \leq y$ *in* P *implies* $\varphi(x) \leq \varphi(y)$ *in* Q*.*

ii) *order-embedding if* $x \leq y$ *in* P *if and only if* $\varphi(x) \leq \varphi(y)$ *in* Q*.*

ii) *order-isomorphism if it is an order-embedding which maps* P *in* Q*.*

Definition 2.3. *Given a partially ordered set* $(P, \leq)$*, we define for every subset* A *of* P*, two subsets of* P *as follows:* $A^u = \{x \in P : a \leq x, \forall a \in A\}$ *and* $A^l = \{x \in P : x \leq a, \forall a \in A\}$*.*

The sets A^u, A^l are called A *upper* and A *lower*, respectively. A^u is the set of all upper bounds[3] of A and A^l is the set of all lower bounds[4]. Moreover, A^u is an up-set and A^l is a down-set.

Definition 2.4. *Let* $(P, \leq)$ *be ordered set and* $Q \subset P$*. Q is join-dense (similary, meet-dense) in P if for all* $a \in P$ *exists* $A \subset Q$ *such that* $a = \bigvee A$[5] *(similary,* $a = \bigwedge A$*).*

[3] By definition, an element $x \in P$ is an upper bound of A if $a \leq x$ for all $a \in A$.

[4] By definition, an element $x \in P$ is an lower bound of A if $x \leq a$ for all $a \in A$.

[5] We also write $\bigvee A$ for the *joint* or *supremum* of A instead of $sup(A)$ and $\bigwedge A$ for the *met* or *infimum* of A instead of $inf(A)$ where these exist.

The following proposition is elementary, but in later Theorems 2.10, 2.11 y 3.2 we find interesting examples of isomorphisms guarantees.

Proposición 2.5. *All ordered set* $(P, \leq)$ *is isomorphic to a subset of set* $(2^P, \subset)$.

Proof. It suffices to note that the map $f : P \to 2^P$, given by $f(x) = \{y \in P : y \leq x\}$ is injective and is order-preserving. On the other hand, the image $f(x)$ is a down-set of ordered set $(2^P, \subset)$. □

We shall be interested in the ordered sets in which the infimum and the supremum exist for all subsets.

Definition 2.6. *Let* $(P, \leq)$ *be a non-empty ordered set. If* $inf(S)$ *and* $sup(S)$ *exist for all* $S \subset P$*, then* P *is called a* complete lattice.

It is straightforward to prove that a non-empty P is complete lattice if and only if $inf(S)$ exists in P for every subset S of P.

On the other hand, there are many options for the embedding of an ordered set into a complete lattice. We examine here one such embedding that generalizes Dedekind's construction of $\mathbb{R}$ by cuts of $\mathbb{Q}$, in order to apply it to the case of multi-adjoint lattice.

Definition 2.7. *Let* P *be an ordered set. If* C *is a complete lattice and* $\varphi : P \to C$ *is an order-embedding, then we say that* C *is a* completion of P *(via* φ*).*

Since the map $\varphi : x \longmapsto \ \downarrow x$ is trivially an order-embeding of P into the complete lattice $\mathcal{O}(\mathcal{P})$ of all down-sets of P (with the inclusion order), this one is a natural completion of P. However, it is unnecessarily large: it is sufficient to take into account that if P is a complete lattice then P is a completion of itself (via the identity map), while $\mathcal{O}(\mathcal{P})$ is much larger. Another completion of an ordered set is the ideal completion. In what follows, we consider the smallest complete lattice containing P, namely the Dedekind-MacNeille completion.

Definition 2.8. *[7] The* Dedekind-MacNeille completion *of an ordered set* P *is the set* $DM(P) = \{A \subset P : A^{ul} = A\}$.

Moreover, it is also known as the completion by cuts and the normal completion of P (see [10]). By means of the following theorem, we can give equivalent definitions of the above concept, in terms of principal ideals of the notion of cut.

Theorem 2.9. *Let* $DM(P)$ *be the the Dedekind-MacNeille completion of* P*. Then,*

i) $DM(P) = \{A \subset P : \triangle A \subset A\}$*, where* $\triangle A = \bigcap\{\downarrow x : x \in A^{\uparrow}\}$.

ii) $DM(P) = \{A \subset P : (A, B)$ *is a cut of* P*, for some* $B \subset P\}$.

$(DM(P), \subset)$ is a complete lattice and, moreover, the map $\varphi : x \longmapsto \downarrow x$ is an order-embedding of P into $DM(P)$. Then, it is easy to prove the following theorem.

Theorem 2.10. *Let* P *be an ordered set and let* $\varphi : P \to DM(P)$ *be such that* $\varphi(x) = \downarrow x$ *for all* $x \in P$*. Then,* $DM(P)$ *is a completion of* P *via the map* φ*.*

$DM(P)$ is known as the Dedekind-MacNeille completion of an ordered set P. In $DM(P)$, since $inf(A)$ and $sup(A)$ exist for any subset, $A \subset DM(P)$, is a complete lattice. Moreover, this process can be readily applied to any lattice, if we define a completion of a lattice. The fundamental theorem that follows can be used to characterize the Dedekind-MacNeille completion.

Theorem 2.11. *[7] Let P be an ordered set and let $\varphi : P \to DM(P)$ be the order-embedding of P into its Dedekind-MacNeille completion given by $\varphi(x) = \downarrow x$ for all $x \in P$. Then*

i) *$\varphi(P)$ is both join-dense and meet-dense in $DM(P)$.*

ii) *If C is a complete lattice and P is a subset of C which is both join-dense and meet-dense in C, then $C \approx DM(P)$ via an order-isomorphism which agrees with φ on P.*

3 Completion of a Quasimulti-adjoint Lattice

In this section, we analyze the specific properties of the Dedekind-MacNeille completion in the case of the lattices used by MALP programs, starting with their formal definition.

Definition 3.1. *Let $(L, \leq)$ be a lattice. A* multi-adjoint lattice *is $(L, \leq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ such that:*

i) *$(L, \leq)$ is a complete lattice, namely, $\forall S \subset L, \exists\ inf(S), sup(S)$*[6].

ii) *$\&_i$ is increasing in both arguments, for all i, $i = 1, \ldots, n$.*

iii) *$\leftarrow_i$ is increasing in the first argument and decreasing in the second, for all i.*

iv) *If $\langle \&_i, \leftarrow_i \rangle$ is an* adjoint pair *in $(L, \leq)$ then, for any $x, y, z \in L$, we have that: $x \leq (y \leftarrow_i z)$ if and only if $(x \&_i z) \leq y$.*

This last condition, called *adjoint property*, is the most important feature of the framework. Moreover, if $(L, \leq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ is bounded and satisfy only $ii), iii), iv)$, we call *quasimulti-adjoint* lattice.

The following theorem guarantees that the Dedekind-MacNeille completion of a quasimulti-joint lattice has a quasimulti-adjoint sublattice isomorphic to the initial one. Also, in this theorem it can be viewed in detail the particular properties of the embedding φ.

Theorem 3.2. *If $(L, \leq, \leftarrow_1, \&_1, \ldots, \leftarrow_n, \&_n)$ is a* quasimulti-adjoint lattice*, then $Im(\varphi)$ is a quasimulti-adjoint sublattice of the complete lattice $DM(P)$ (via φ) that is isomorphic to L. Moreover, the order-embedding φ is a lattice homomorphism of lattices; preserves all joins and meets which exist in P; for any adjoint pair $(\leftarrow, \&)$ in L, there exists an adjoint pair $(\leftarrow_{\bar{L}}, \&_{\bar{L}})$ in $Im(\varphi)$. Finally, for any connective in L there exists an associated connective in $\bar{L}$.*

[6] Then, it is a bounded lattice, i.e. it has bottom and top elements, denoted by $\bot$ and $\top$, respectively.

Proof. We will prove first that the map $\varphi : L \to DM(L)$, given by $\varphi(x) = \downarrow x$ is an order-embedding, homomorphism and preserves the indicated joins and meets. Indeed:

i) φ is a map since $\varphi(x) = \downarrow x \in DM(L)$ for all $x \in DM(L)$: one has only to consider $(\downarrow x)^{\uparrow\downarrow} = \downarrow x$.

ii) φ is injective: if we assume that $\varphi(x) = \downarrow x = \downarrow y = \varphi(y)$, then $x \in \downarrow x = \downarrow y$, so $x \le y$. Similarly, shows that $y \le x$, and we obtain $x = y$ by the antisymmetric property.

iii) φ is order-preserving: if $x \le y$ in L implies $\varphi(x) \le \varphi(y)$ in $DM(L)$, by definition of lower bound.

iv) Also, φ is an order-embedding: if $\varphi(x) \le \varphi(y)$, then $\downarrow x \subset \downarrow y$. Since $x \in \downarrow x$, $x \in \downarrow y$ and therefore $x \le y$.

iv) φ is a lattice homomorphism, i.e., $\varphi(x \wedge y) = \varphi(x) \cap \varphi(y)$, $\varphi(x \vee y) = \varphi(x) \cup \varphi(y)$. Certainly, we shall prove the equality of both sets. If $z \in \varphi(x \wedge y)$ it holds that $z \le x \wedge y$ and, by definition of greatest lower bound, $z \le x, z \le y$. Then, we have $z \in \varphi(x), z \in \varphi(y)$, that is, $z \in \varphi(x) \cap \varphi(y)$. Thus, we obtain, $\varphi(x \wedge y) \subset \varphi(x) \cap \varphi(y)$. The reverse inclusion, $\varphi(x) \cap \varphi(y) \subset \varphi(x \wedge y)$, is analogous, like the dual result $\varphi(x \vee y) = \varphi(x) \cup \varphi(y)$.

v) φ preserves all joins and meets wich exist in P. Let A be a subset of L and assume that $\bigvee A$ exists in L. We shall prove that $\varphi(\bigvee A) = \bigvee \varphi(A)$, namely, $\downarrow (\bigvee A) = \bigvee \{\downarrow a : a \in A\}$. It is easy to prove that $\downarrow (\bigvee A)$ is an upper bound for $\{\downarrow a : a \in A\}$. Moreover, if $B \in DM(L)$ is an upper for the set $\{\downarrow a : a \in A\}$, we have $a \in \downarrow a \subset B$ for all $a \in A$, and therefore $A \subset B$. On the other hand, if $\downarrow (\bigvee A)$ exists in L, $\downarrow (\bigvee A) =$ [7] A^{ul}, and so $\downarrow (\bigvee A) = A^{ul} \subset B^{ul} = B$.

Likewise, if $\bigwedge A$ exists in L, we shall prove that $\varphi(\bigwedge A) = \bigwedge \varphi(A)$, that is, $\downarrow (\bigwedge A) = \bigwedge \{\downarrow a : a \in A\}$. Since, $\bigwedge \{\downarrow a : a \in A\} = \bigcap \{\downarrow a : a \in A\}$, we have the intended result.

Furthermore, we shall show that for any adjoint pair $(\leftarrow, \&)$ in L, there exists an adjoint pair $(\leftarrow_{\bar{L}}, \&_{\bar{L}})$ in $Im(\varphi)$, set denoted by $\bar{L}$. First, let $A, B, C \in \bar{L}$, $A = \varphi(x), B = \varphi(y), C = \varphi(z)$ be, for $x, y, z \in L$; then, we define the conjunction $\&_{\bar{L}}$ and the implication $\leftarrow_{\bar{L}}$ as

$$A \&_{\bar{L}} B = \varphi(x) \&_{\bar{L}} \varphi(y) := \varphi(x \& y) \qquad B \leftarrow_{\bar{L}} C = \varphi(y) \leftarrow_{\bar{L}} \varphi(z) := \varphi(y \leftarrow z)$$

resulting the following properties:

i) $\&_{\bar{L}}$ is increasing in both arguments: we shall show that if $A_1 \subset A_2$, then $A_1 \&_{\bar{L}} B \subset A_2 \&_{\bar{L}} B$. Since $A_1 = \varphi(x_1), A_2 = \varphi(x_2), B = \varphi(y)$, with $x_1, x_2, y \in L$, we have that $A_1 \&_{\bar{L}} B = \varphi(x_1 \& y) \subset \varphi(x_2 \& y) = A_2 \&_{\bar{L}} B$ being as φ is order-preserving and $\&$ is increasing in the first argument. Likewise, the increase in the second component is obtained.

[7] By definition of least upper bound and since A^{ul} is a down-set.

ii) $\&_{\bar{L}}$ has identity element, in particular the identity of $\bar{L}$ denoted by $\top_{\bar{L}}$ and is the set $\top_{\bar{L}} =\downarrow \top = \{z \in L : z \leq \top\} = L$. We need to check that $\top_{\bar{L}} \&_{\bar{L}} A = A$, for all $A \in \bar{L}$. Certainly, if $A = \varphi(x), x \in L$, we have $\top_{\bar{L}} \&_{\bar{L}} A = L \&_{\bar{L}} A = \varphi(\top \& x) = \varphi(x) = A$, because & is a conjunction in $\bar{L}$ and $\top$ is the identity element of $\bar{L}$.

iii) $\leftarrow_{\bar{L}}$ is increasing in the first argument and decreasing in the second argument or, more accurately, $\leftarrow_{\bar{L}}$ is order-preserving in the consequent and order-reversing in the antecedent. Regarding the antecedent, we need to prove that if $C_1 \subset C_2$, then $B \leftarrow_{\bar{L}} C_1 \supset B \leftarrow_{\bar{L}} C_2$. Since $C_1 = \varphi(z_1), C_2 = \varphi(z_2), B = \varphi(y)$, with $z_1, z_2, y \in L$, we have that $B \leftarrow_{\bar{L}} C_1 = \varphi(y \leftarrow z_1) \supset \varphi(y \leftarrow z_2) = B \leftarrow_{\bar{L}} C_2$ because $\leftarrow_L$ is an implication and φ is order-preserving. Similarly, the behavior in the consequent is obtained.

iv) $(\leftarrow_{\bar{L}}, \&_{\bar{L}})$ is an adjoint pair: we need to check that for any $A, B, C \in \bar{L}$, $A \subset (B \leftarrow_{\bar{L}} C) \Leftrightarrow A \&_{\bar{L}} C \subset B$ is fulfilled. Given $A, B, C \in \bar{L}$, $A = \varphi(x), B = \varphi(y), C = \varphi(z)$, with $x, y, z \in L$. For the first expression, we have $A \subset (B \leftarrow_{\bar{L}} C) \Leftrightarrow \varphi(x) \subset \varphi(y \leftarrow z) \Leftrightarrow x \subset (y \leftarrow z)$, where we use in the last step that φ is an order-embedding. On the other hand, using the definition of $\&_{\bar{L}}$ and again the character of order-embedding of φ, $A \&_{\bar{L}} C \subset B \Leftrightarrow \varphi(x \& z) \subset \varphi(y) \Leftrightarrow x \& z \leq y$, and we have the indeed equality in virtue of the adjoint property of pair $(\leftarrow, \&)$ in lattice L.

Finally, each connective in L defines a connective in $\bar{L}$, more detailed:

a) if $\wedge$ is a conjunction in L, there exists an associated conjunction $\bar{\wedge}$ in $\bar{L}$. The commutative and associative properties of $\bar{\wedge}$ are derived from the respective of $\wedge$. Moreover, $\bar{\wedge}$ verifies claims *i)*, *ii)* before that we have shown for $\&_{\bar{L}}$. All is routine and we omit it.

b) if $\vee$ is a disjunction in L, there exists a disjunction $\bar{\vee}$ in $\bar{L}$ associated. Similarly *a)*.

c) if @ is a aggregator in L, there exists an associated aggregator $\bar{@}$ in $\bar{L}$..

It is easy to prove that φ is suryective if φ preserves all joins and meets. In this case, L is isomorphic to (complete) multi-adjoint lattice $DM(P)$. □

4 Multi-adjoint Lattices in Practice using FLOPER

From now, we proceed with more practical aspects regarding multi-adjoint lattices and implementation issues. The parser of our FLOPER tool [27, 28] has been implemented by using the Prolog language. Once the application is loaded inside a Prolog interpreter, it shows a menu which includes options for loading/compiling, parsing, listing and saving fuzzy programs, as well as for executing/debugging fuzzy goals. Moreover, in [27] we explain that FLOPER has been recently equipped with new options, called "lat" and "show", for allowing the possibility of respectively changing and displaying the multi-adjoint lattice associated to a given program, as we are going to explain.

When modeling a lattice to be loaded into FLOPER, all its relevant components must be encapsulated inside a Prolog file which must necessarily contain the definitions of a minimal set of predicates defining the set of valid elements (including special mentions to the "top" and "bottom" ones), the full or partial ordering established among them, as well as the repertoire of fuzzy connectives which can be used for their subsequent manipulation. In order to simplify our explanation, assume that file "bool.pl" refers to the simplest notion of (a binary) adjoint lattice, thus implementing the following set of predicates:

- `member/1` which is satisfied when being called with a parameter representing a valid truth degree. For instance, in the Boolean case, both predicates can be simply modeled by the Prolog facts: `member(0).`, `member(1).` and `members([0,1]).`
- `bot/1` and `top/1` obviously answer with the top and bottom element of the lattice, respectively. Both are implemented into "bool.pl" as `bot(0).` and `top(1).`
- `leq/2` models the ordering relation among all the possible pairs of truth degrees, and obviously it is only satisfied when it is invoked with two elements verifying that the first parameter is equal or smaller than the second one. So, in our example it suffices with including into "bool.pl" the facts: `leq(0,X).` and `leq(X,1).`
- Finally, if we have some fuzzy connectives of the form $\&_{label_1}$ (conjunction), $\vee_{label_2}$ (disjunction) or $@_{label_3}$ (aggregation) with arities n_1, n_2 and n_3 respectively, we must provide clauses defining the *connective predicates* "`and_`$label_1$`/(`n_1`+1)`", "`or_`$label_2$`/(`n_2`+1)`" and "`agr_`$label_3$`/(`n_3`+1)`", where the extra argument of each predicate is intended to contain the result achieved after the evaluation of the proper connective. For instance, in the Boolean case, the following two facts model in a very easy way the behaviour of the classical conjunction operation: `and_bool(0,_,0).  and_bool(1,X,X).`

The reader can easily check that the use of lattice "bool.pl" when working with MALP programs whose rules have the form: "$A \;\; \leftarrow_{bool} \;\; \&_{bool}(B_1, \ldots, B_n) \;\; with \;\; 1$", being A and B_i typical atoms, successfully mimics the behaviour of classical Prolog programs where clauses accomplish with the shape "$A \;\; :- \;\; B_1, \ldots, B_n$". As a novelty in the fuzzy setting, the outputs associated to the evaluation of goals will contain the corresponding Prolog's substitution (i.e., the *crisp* notion of computed answer obtained by means of classical SLD-resolution) together with the maximum truth degree 1.

On the other hand, and following the Prolog style regulated by the previous guidelines, in file "num.lat" we have included the clauses shown in Figure 1. Here, we have modeled the more flexible lattice which enables the possibility of working with truth degrees in the real interval $[0, 1]$, allowing too the possibility of using conjunction and disjunction operators recasted from the three typical fuzzy logics proposals described before (i.e., the *Lukasiewicz*, *Gödel* and *product* logics), as well as a useful description for the hybrid aggregator *average*.

```
member(X) :- number(X),0=<X,X=<1.

bot(0).               top(1).               leq(X,Y) :- X=<Y.

and_luka(X,Y,Z) :- pri_add(X,Y,U1),pri_sub(U1,1,U2),pri_max(0,U2,Z).
and_godel(X,Y,Z):- pri_min(X,Y,Z).
and_prod(X,Y,Z) :- pri_prod(X,Y,Z).

or_luka(X,Y,Z)  :- pri_add(X,Y,U1),pri_min(U1,1,Z).
or_godel(X,Y,Z) :- pri_max(X,Y,Z).
or_prod(X,Y,Z)  :- pri_prod(X,Y,U1),pri_add(X,Y,U2),pri_sub(U2,U1,Z).

agr_aver(X,Y,Z) :- pri_add(X,Y,U),pri_div(U,2,Z).

pri_add(X,Y,Z)  :- Z is X+Y.     pri_min(X,Y,Z) :- (X=<Y,Z=X;X>Y,Z=Y).
pri_sub(X,Y,Z)  :- Z is X-Y.     pri_max(X,Y,Z) :- (X=<Y,Z=Y;X>Y,Z=X).
pri_prod(X,Y,Z) :- Z is X * Y.   pri_div(X,Y,Z) :- Z is X/Y.
```

Figure 1: Multi-adjoint lattice modeling truth degrees in the real interval [0,1].

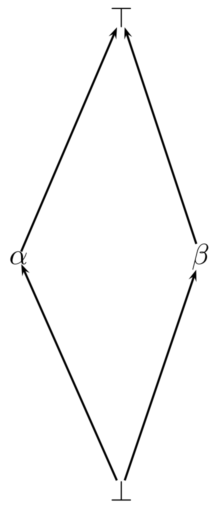
Note also that we have included definitions for auxiliary predicates, whose names always begin with the prefix "`pri_`". All of them are intended to describe primitive/arithmetic operators (in our case $+$, $-$, $*$, $/$, min and max) in a Prolog style, for being appropriately called from the bodies of clauses defining predicates with higher levels of expressivity (this is the case for instance, of the three kinds of fuzzy connectives we are considering: conjuntions, disjunctions and agreggations).

Assume that "new_num.pl" contains the same Prolog code than "num.pl" with the exception of the definition regarding the average aggregator. Now, instead of computing the average of two truth degrees, let us consider a new version which computes the average between the results achieved after applying to both elements the disjunctions operators described by Gödel and Łukasiewicz, that is: $@_{\texttt{aver}}(x_1, x_2) = (\vee_{\texttt{G}}(x_1, x_2) + \vee_{\texttt{L}}(x_1, x_2))/2$ (where $\vee_{\texttt{G}}(x_1, x_2) = max(x_1, x_2)$ and $\vee_{\texttt{L}}(x_1, x_2) = min(1, x_1, x_2)$. The corresponding Prolog clause modeling such definition into the "new_num.pl" file is:

```
agr_aver(X,Y,Z)  :-   or_godel(X,Y,Z1),or_luka(X,Y,Z2),
                      pri_add(Z1,Z2,Z3), pri_div(Z3,2,Z).
```

And now, if with the new lattice we execute goal p w.r.t. the same program seen in Section 1 (introduction), instead of obtaining 0.63, the new solution will be 0.81 since now $\&_{\texttt{P}}(0.9, @_{\texttt{aver}}(0.8, 0.6)) = \&_{\texttt{P}}(0.9, (\vee_{\texttt{G}}(0.8, 0.6) + \vee_{\texttt{L}}(0.8, 0.6))/2) = 0.9 * (max(0.8, 0.6) + min(1, 0.8 + 0.6))/2) = 0.9 * ((0.8 + 1)/2) = 0.81$.

To finish this section, in our last example we consider the partially ordered multi-adjoint lattice of Figure 2, for which the conjunction and implication connectives based on the *Gödel* intuistionistic logic conforms and adjoint pair (in the general case, the *Gödel*'s conjunctor is expressed in terms of "*inf*" instead of "*min*").




```
member(bottom).    member(alpha).
member(beta).      member(top).

leq(bottom,X). leq(alpha,alpha).  leq(alpha,top).
leq(beta,beta).  leq(beta,top).  leq(X,top).

and_godel(X,Y,Z) :- pri_inf(X,Y,Z).

pri_inf(bottom,X,bottom):-!.
pri_inf(alpha,X,alpha):-leq(alpha,X),!.
pri_inf(beta,X,beta):-leq(beta,X),!.
pri_inf(top,X,X):-!.
pri_inf(X,Y,bottom).
```

Figure 2: Partially-ordered multi-adjoint lattice.

Conclusions and Future Work

This paper has been mainly concerned with the Dedekind-MacNeille completion, a relevant and elegant mathematical concept which might help us to adapt some lattices for being safely used into the multi-adjoint logic programming framework. In particular, we have shown a technique which let us to "skip" in some cases the hypothesis of complete lattice usually required in multi-adjoint lattices, being this hypothesis mandatory, for instance, when describing the fix-point and model-theoretic declarative semantics of MALP [15]. In particular, the results achieved in this paper are useful to justify the safe use into FLOPER, according the methodology explained in the last part of this paper, of those lattices (composed by all finite strings whose elements are formed from an arbitrary alphabet of symbols) used in [28, 26] for documenting with declarative traces the execution of goals at a very low computational cost.

Since many standard completions of a lattice arise for suitable choices of the sets of up-subset and down-subset of P, in the future we plan to consider the concept of canonical extensions (i.e., dense and compact completions) of lattices with additional operations (introduced in [9]) and its application in the case of the quasimulti-adjoint lattices. Lattices with additional operations emerge from linear logics([1, 12, 17]). The results obtained can be applied for bilattices and trilattices (common MALP domains).

Acknowledgements

This work was supported by the EU (FEDER), and the Spanish Science and Innovation Ministry (MICINN) under grant TIN 2007-65749 as well as by the Castilla-La Mancha Administration under grant PII1I09-0117-4481.

References

[1] G. Allwein and J. M. Dunn. Kripke models for linear logic. *The Journal of Symbolic Logic*, 58(2):514–545, 1993.

[2] J.M. Almendros-Jiménez, A. Luna, and G. Moreno. A Flexible XPath-based Query Language Implemented with Fuzzy Logic Programming. In *Proc. of 5th Int. Symp. RuleML'11.* Springer LNCS (8 pages, in press), 2011.

[3] I. P. Cabrera, P. Cordero, and M. Ojeda-Aciego. Non-deterministic algebraic structures for soft computing. In *Proceedings of International Work-Conference on Artificial Neural Networks, IWANN'11*, 2011. Accepted.

[4] I.P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, and M. Ojeda-Aciego. A coalgebraic approach to non-determinism: applications to multilattices. *Information Sciences*, 180(22):4323–4335, 2010.

[5] I.P. Cabrera, P. Cordero, G. Gutiérrez, J. Martínez, and M. Ojeda-Aciego. Coalgebras and non-determinism: an application to multilattices. In *Proceedings of Physics and Computation*, 2010.

[6] C.V. Damásio, N. Madrid, and M. Ojeda-Aciego. On the notions of residuated-based coherence and bilattice-based consistence. In *Proceedings of International Workshop on Fuzzy Logic and Applications WILF'11*, 2011. Accepted.

[7] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order.* Cambrige, University Press, UK, 2002. Second Edition.

[8] R. Dedekind. Stetigkeit und irrationale Zahlen. *Braunscheweig*, 1872.

[9] M. Gehrke and J. Harding. Bounded Lattice Expansions. *Journal of Algebra*, 238:345–371, 2001.

[10] M. Gehrke, J. Harding, and Y. Venema. MacNeille completions and canonical extensions. *Transactions of the American Mathematical Soc.*, 358(2):573–590, 2005.

[11] J.A. Guerrero and G. Moreno. Optimizing fuzzy logic programs by unfolding, aggregation and folding. *Elec. Notes in Theoretical Comp. Sci.*, 219:19–34, 2008.

[12] C. Hartonas. Duality for lattice-ordered algebras and for normal algebraizable logics. *Estudia Logica*, 58:403–450, 1997.

[13] B. Jonnson and A. Tarski. Boolean algebras with operators I. *American Journal of Mathematics*, 73:891–939, 1951.

[14] P. Julián, G. Moreno, and J. Penabad. On Fuzzy Unfolding. A Multi-adjoint Approach. *Fuzzy Sets and Systems, Elsevier*, 154:16–33, 2005.

[15] P. Julián, G. Moreno, and J. Penabad. On the declarative semantics of multi-adjoint logic programs. In *Proceedings of the 10th Int. Work-Conference on Artificial Neural Networks, IWANN'09*, pages 253–260, 2009. Springer, LNCS 5517.

[16] J.W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, Berlin, 1987.

[17] W. MacCaull. Kripke semantics for logics with bck implication. *Bulletin of the Section of Logic of the University of Lódź*, 25(1):41–51, 1996.

[18] H. M. MacNeille. Partially ordered sets. *Transactions of the American Mathematical Society*, pages 416–460, 1937.

[19] J. Medina and M. Ojeda-Aciego. Towards attribute reduction in multi-adjoint concept lattices. In *Proceedings of Concept Lattices and Applications, CLA'10*, pages 92–103, 2010.

[20] J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. Fuzzy logic programming via multilattices. *Fuzzy Sets and Systems*, 158(6):674–688, 2007.

[21] J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. On reachability of minimal models of multilattice-based logic programming. *LNAI*, 4827:271–282, 2007.

[22] J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. Formal concept analysis via multi-adjoint concept lattices. *Fuzzy Sets and Systems*, 160(2):130–144, 2009.

[23] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. *Proceedings of Logic Programming and Non-Monotonic Reasoning, LPNMR'01, Springer, LNAI*, 2173:351–364, 2001.

[24] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. A procedural semantics for multi-adjoint logic programing. *Progress in Artificial Intelligence, EPIA'01, Springer-Verlag, LNAI*, 2258(1):290–297, 2001.

[25] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Similarity-based Unification: a multi-adjoint approach. *Fuzzy Sets and Systems*, 146:43–62, 2004.

[26] P. J. Morcillo, G. Moreno, J. Penabad, and C. Vázquez. Declarative Traces into Fuzzy Computed Answers. In *Proceedings of 5th Int. Symp. RuleML'11*, Springer, LNCS (16 pages, in press), 2011.

[27] P.J. Morcillo, G. Moreno, J. Penabad, and C. Vázquez. A Practical Management of Fuzzy Truth Degrees using FLOPER. In *Proceedings of 4th Intl Symp. RuleML'10*, pages 119–126. Springer, LNCS 6403, 2010.

[28] P.J. Morcillo, G. Moreno, J. Penabad, and C. Vázquez. Fuzzy Computed Answers Collecting Proof Information. In *Proceedings of the 11th Int. Work-Conference on Artificial Neural Networks, IWANN'11* LNCS (in press), page 8. Springer, 2011.

[29] G. Voutsadakis. Dedekind-Macneille Completion of n-ordered Sets. *Order*, 24(1):15–29, 2007.