Parameterised Simplification Logic: reasoning with implications in an automated way

Pablo Cordero, Manuel Enciso, Ángel Mora, and Vilem Vychodil

***Abstract*—In this sequel to our previous paper [1] on general inference systems for reasoning with if-then dependencies, we study transformations of if-then rules to semantically equivalent collections of if-then rules suitable to solve several problems related to reasoning with data dependencies. We work in a framework of general lattice-based if-then rules whose semantics is parameterised by systems of isotone Galois connections. This framework allows us to obtain theoretical insight as well as algorithms on a general level and observe their special cases by choosing various types of parameterisations. This way, we study methods for automated reasoning with different types of if-then rules in a single framework that covers existing as well as novel types of rules. Our approach supports a large family of if-then rules, including fuzzy if-then rules with various types of semantics. The main results in this paper include new observations on the syntactic inference of if-then rules, complete collections of rules, reduced normal forms of collections of rules, and automated reasoning methods. We demonstrate the generality of the framework and the results by examples of their particular cases focusing on fuzzy if-then rules.**

***Index Terms*—closure operator, lattice theory, completeness, data dependency, fuzzy if-then rules, approximate reasoning**

I. Introduction

Methods of automated reasoning play an important role in reasoning with if-then rules. In the case of the classic rules that include functional dependencies [2], [3], [4], attribute implications [5], [6], [7], and association rules [8], [9], efficient algorithms as well as the computational complexity of the problems involved are known. In the case of the fuzzy, vague, or graded if-then rules [10], [11], the situation is different mainly since there exist a vast number of types of rules that vary in their syntax and semantics. It would be a praiseworthy effort to review all of the existing approaches and, in each of them, focus on the issue of the basic entailment problem: "Does a given rule follow from a collection of other rules (and to what degree)?" One can imagine that as a result, it would be possible to make general conclusions about entailment problems of graded rules falling into several categories. However, such a review is hardly achievable due to the number of various systems of if-then rules that exist in the literature, cf. [12]. In this paper, we focus on this type of making general conclusions on properties of the entailment of graded/fuzzy if-then rules and related algorithms. Instead of going through a family of individual approaches, we work in a general framework that covers a large family of if-then rules that differ in their syntax and semantics. The framework is indeed general and captures possible semantics of if-then rules that are seemingly unrelated. For instance, as a special case of the approach, we are able to work with various types of fuzzy if-then rules as well as multiple types of temporal if-then rules, and in addition, the approach offers a way of their combination. In this paper, and mainly in the presented examples, we focus on the fuzzy rules, but it should be noted that the achieved results are not limited just to fuzzy rules.

The contribution of our paper is the following. Using a parametric approach [13] to if-then rules whose semantics is defined through special families of closure operators, we study a general entailment problem, investigate properties of complete inference systems and normal forms of collections of if-then rules that may represent knowledge bases specified by experts or rule bases inferred from graded/fuzzy input data. As we show in the paper; this ultimately gives us a description of a general algorithm that can be used to compute semantic closures that play a crucial role in deciding the entailment problem. In addition to the observations made on the general level, we show how the observations translate into particular theories of if-then rules that exist in the literature. As a consequence, for several of the theories, we obtain new theoretical insight into the entailment problem and new algorithms. Our investigation is based on and is an extension of the observations in our recent paper [1] where we have investigated parameterised simplification logic for general if-then rules.

This paper is organised as follows. Section II presents the preliminary notions of residuated lattices, isotone Galois connections, and general if-then rules. In Section III, we provide a characterisation of complete theories. In Section IV, we study an automated reasoning method for general if-then rules, and we present general algorithms of completion and computing closures. In Section V, we show an application of the general results in the case of automated reasoning with fuzzy if-then rules. Conclusions are presented in Section VI.

II. Preliminaries

As we mentioned in the Introduction, this article is a further step of the work presented in [1]. In this section, we summarise the basic notions introduced there to contextualise the results presented later.

We are going to work with complete dual residuated lattices, which is a structure $\mathbb{L} = \langle L, \leqslant, \oplus, \ominus, 0, 1\rangle$ fulfilling the following conditions:

Pablo Cordero, Manuel Enciso and Ángel Mora from the University of Málaga, Andalucía Tech, Spain e-mail: {pcordero,enciso, amora}@uma.es
Vilem Vychodil, Czech computer professor and researcher.
Manuscript received xx, 2021

- $\langle L, \leqslant, 0, 1\rangle$ is a complete lattice where 0 is the least element and 1 is the greatest element. As usual, we use the symbols $\vee$ and $\wedge$ to denote suprema (least upper bounds) and infima (greatest lower bounds), respectively;
- $\langle L, \oplus, 0\rangle$ is a commutative monoid;
- $\ominus$ is a binary operation so that the pair $\langle \oplus, \ominus\rangle$ satisfies the following adjointness property: For all $a, b, c \in L$, we have

$$a \leqslant b \oplus c \text{ if and only if } a \ominus b \leqslant c. \tag{1}$$

As a consequence, the operations $\oplus$ and $\ominus$, named *addition* and *residuated subtraction* respectively, fulfils the following properties: for any $a, b, c \in L$:

$$a \leqslant b \text{ if and only if } a \ominus b = 0, \tag{2}$$
$$a \ominus 0 = a, \tag{3}$$
$$a \ominus b \leqslant a \leqslant a \oplus b, \tag{4}$$
$$b \leqslant c \text{ implies } a \oplus b \leqslant a \oplus c,\ \ b \ominus a \leqslant c \ominus a \text{ and } a \ominus c \leqslant a \ominus b, \tag{5}$$
$$a \vee b \leqslant a \oplus (b \ominus a) \leqslant a \oplus b, \tag{6}$$
$$(a \oplus b) \ominus a \leqslant b \leqslant a \oplus (b \ominus a). \tag{7}$$

We use the usual terminology in lattice theory [14]: an element $k \in L$ is said to be *compact* if, for all $J \subseteq L$,

$$\text{if } k \leqslant \bigvee J, \text{there exists a finite } J' \subseteq J \text{ such that } k \leqslant \bigvee J'.$$

The set of all compact elements in $\mathbb{L}$ is denoted by K and we assume that $\mathbb{L}$ is *algebraic* (or compactly generated), i.e.,

$$\text{for all } a \in L \text{ there exists } X \subseteq K \text{ such that } a = \bigvee X.$$

Here, in the same way, that was presented in [1] we also assume that K is closed for $\oplus$ and $\ominus$:

$$a, b \in K \text{ implies } a \oplus b, a \ominus b \in K. \tag{8}$$

This structure is the basis for defining implications. We consider premises and conclusions of implications to be compact elements of the lattice $\mathbb{L}$. Thus, the language of the Logic is the following

$$\mathcal{L} = \{a \Rightarrow b \mid a, b \in K\}$$

As we mentioned in the introduction, the development of a framework as general as possible, requires the introduction of some kind of parameters. In our work, the role of such parameters are played by isotone Galois connections. Recall that a pair of mappings between lattices $\langle \boldsymbol{f}, \boldsymbol{g}\rangle$ is an isotone Galois connections if both mappings are isotone, $g \circ f$ is inflationary and $f \circ g$ is deflationary. As a consequence, for all $X \subseteq L$, we have that :

$$\boldsymbol{f}\big(\bigvee X\big) = \bigvee \boldsymbol{f}(X) \text{ and } \boldsymbol{g}\big(\bigwedge X\big) = \bigwedge \boldsymbol{g}(X). \tag{9}$$

An L-parameterization is a set S of isotone Galois connections in $\mathbb{L}$ such that $\langle \mathtt{I}, \mathtt{I}\rangle \in S$. If S is closed under composition, i.e., if $\langle S, \circ, \langle \mathtt{I}, \mathtt{I}\rangle\rangle$ is a monoid, we call it an $\mathbb{L}$-*parameterization*, see [13, Definition 1]. In addition, S is called *compact* (see [15, Definition 4]) if $\boldsymbol{f}(K) \subseteq K$ for all $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$.

The semantics of these implications was presented in terms of closure operators, which are compatible with the parameters.

An S-closure operator [15, Definition 3] is a mapping $\boldsymbol{c}\colon L \to L$ satisfying the following properties:

$$a \leqslant \boldsymbol{c}(a), \tag{10}$$
$$a \leqslant b \text{ implies } \boldsymbol{c}(a) \leqslant \boldsymbol{c}(b), \tag{11}$$
$$\boldsymbol{c}(\boldsymbol{g}(\boldsymbol{c}(a))) \leqslant \boldsymbol{g}(\boldsymbol{c}(a)), \tag{12}$$

are satisfied for all $a, b \in L$ and all $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$. A closure operator $\boldsymbol{c}$ is called additive whenever $a \oplus b \leqslant \boldsymbol{c}(a \vee b)$ for all $a, b \in L$. In addition, if S is an $\mathbb{L}$-parameterization, then $\boldsymbol{c}$ is called an $\mathbb{S}$-closure operator.

We now introduce the notions of a model for an implication. A mapping $\boldsymbol{c}\colon L \to L$ is said to be a model for $a \Rightarrow b \in \mathcal{L}$, written $\boldsymbol{c} \models a \Rightarrow b$, if the following condition holds:

$$\boldsymbol{c} \text{ is an additive } S\text{-closure operator in } \mathbb{L} \text{ and } b \leqslant \boldsymbol{c}(a) \tag{13}$$

As usual, a closure operator $\boldsymbol{c}$ is a model for a theory $\Sigma \subseteq \mathcal{L}$ if it is a model for all the implications in Σ and $\Sigma \models a \Rightarrow b$ if any model of Σ is a model of $a \Rightarrow b$.

Example 1. Consider the dual complete residuated lattice $\mathbb{L} = \langle L, \leqslant, \oplus, \ominus, 0, 1\rangle$where $L = \{0, 0.1, 0.2, \dots 1\}$, $\leqslant$ is the usual order and $\oplus$ and $\ominus$ are the following binary operations:

$$a \oplus b = \begin{cases} a + b, & \text{if } a + b \leqslant 0.5,\\ \max\{0.5, a, b\}, & \text{otherwise,}\end{cases}$$

$$a \ominus b = \begin{cases} 0, & \text{if } a \leqslant b,\\ 1 - b, & \text{if } 0 \leqslant b < a \leqslant 0.5,\\ a, & \text{otherwise.}\end{cases}$$

Over this structure, the dual complete residuated lattice of fuzzy sets in the universe $\mathbb{N}$ of natural numbers, $\mathbb{L}^{\mathbb{N}} = \langle L^{\mathbb{N}}, \subseteq, \oplus, \ominus, \emptyset, \mathbb{N}, \rangle$, is introduced in the usual way. Thus, given two fuzzy sets $A, B \in L^{\mathbb{N}}$, we have that $A \subseteq B$ iff $A(n) \leqslant B(n)$ for all $n \in \mathbb{N}$, and the addition and the residuated subtraction are defined componentwise as follows: $(A \oplus B)(n) = A(n) \oplus B(n)$ and $(A \ominus B)(n) = A(n) \ominus B(n)$.

The compact elements in $\mathbb{L}^{\mathbb{N}}$ are those sets having a finite support, i.e., $A \in L^{\mathbb{N}}$ such that $\{n \in \mathbb{N} \mid A(n) > 0\}$ is finite. In this case, we use the usual succinct notation of writing fuzzy sets, e.g. $A = \{1/_{0.3}, 3/_{0.6}\}$ denotes that $A(1) = 0.3$, $A(3) = 0.6$, and $A(n) = 0$ otherwise. Notice that $\mathbb{L}^{\mathbb{N}}$ is algebraic and is closed for $\oplus$ and $\ominus$ (8).

In addition, consider the compact $\mathbb{L}$-parameterization

$$S = \big\{\langle \boldsymbol{f}_a, \boldsymbol{g}_a\rangle \mid a \in \{0, 0.2, 0.4, \dots 1\}\big\}$$

where $\boldsymbol{f}_a, \boldsymbol{g}_a\colon L^{\mathbb{N}} \to L^{\mathbb{N}}$ are defined as follows:

$$(\boldsymbol{f}_a(A))(n) = \max\{0, A(n) - a\}$$
$$(\boldsymbol{g}_a(A)(n) = \min\{1, A(n) + a\}$$

In particular, for all $A \in L^{\mathbb{N}}$, $\boldsymbol{f}_1(A) = \emptyset$, $\boldsymbol{g}_1(A) = \mathbb{N}$, and $\boldsymbol{f}_0(A) = \boldsymbol{g}_0(A) = A$.

Finally, $\boldsymbol{c}\colon L^{\mathbb{N}} \to L^{\mathbb{N}}$ defined as follows:

$$\boldsymbol{c}(A)(n) = \begin{cases} 0.6 & \text{if } A(n) \leqslant 0.6,\\ 0.8 & \text{if } 0.6 < A(n) \leqslant 0.8,\\ 1 & \text{otherwise,}\end{cases}$$

for each $A \in L^{\mathbb{N}}$ and each $n \in \mathbb{N}$, which is an additive $\mathbb{S}$-closure operator, is a model for the following theory

$$\Sigma = \big\{\{2/_{0.9}\} \Rightarrow \{2/_{1}, 3/_{0.5}\}, \{3/_{0.9}, 4/_{0.7}\} \Rightarrow \{3/_{1}, 4/_{0.6}\}\big\}$$

□

As usual, reasoning based on models lacks efficiency, and a syntactic method is needed. This role is played by an axiomatic system, which is enriched in a further stage with an automated reasoning method. We now present the axiomatic system introduced in ,[1] which is the keystone of the automated method presented in this paper.

Definition 1. For all $a, b, c, d \in K$ and $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, the inference system considers the following axiom scheme:

$$\textit{Reflexivity} : \text{Infer } a \Rightarrow a, \qquad (\texttt{Ref})$$

together with the three following inference rules:

Composition :

$$\text{From } a \Rightarrow b \text{ and } a \Rightarrow c \text{ infer } a \Rightarrow b \oplus c, \qquad (\texttt{Comp})$$

Simplification :

$$\text{From } a \Rightarrow b \text{ and } c \Rightarrow d \text{ infer } a \oplus (c \ominus b) \Rightarrow d, \quad (\texttt{Simp})$$

Multiplication :

$$\text{From } a \Rightarrow b \text{ infer } \boldsymbol{f}(a) \Rightarrow \boldsymbol{f}(b). \qquad (\texttt{Mul})$$

An implication $a \Rightarrow b \in \mathcal{L}$ is said to be syntactically derived or inferred from a theory $\Sigma \subseteq \mathcal{L}$, denoted by $\Sigma \vdash a \Rightarrow b$, if there exists a sequence $\sigma_1, \ldots, \sigma_n \in \mathcal{L}$ such that σ_n is $a \Rightarrow b$ and, for all $1 \leqslant i \leqslant n$, at least one of the following conditions holds: $\sigma_i \in \Sigma$; or σ_i is an axiom (Ref); or σ_i can be obtained by applying one of the inference rules (Comp), (Simp), or (Mul) to formulas in $\{\sigma_j \mid 1 \leqslant j < i\}$.

The following rules are derived from the axiomatic system:

$$\textit{Generalized Reflexivity} : \vdash a \Rightarrow b \text{ when } b \leq a \qquad (\texttt{GRef})$$

$$\textit{Transitivity} : a \Rightarrow b, b \Rightarrow c \vdash a \Rightarrow c \qquad (\texttt{Tran})$$

Generalized Composition :

$$a \Rightarrow b, c \Rightarrow d \vdash a \vee c \Rightarrow b \oplus d \qquad (\texttt{GComp})$$

$$\textit{Augmentation} : a \Rightarrow b \vdash a \vee c \Rightarrow b \oplus c \qquad (\texttt{Augm})$$

Generalized Transitivity :

$$a \Rightarrow b, b \vee c \Rightarrow d \vdash a \vee c \Rightarrow d \qquad (\texttt{GTran})$$

Theorems 1 and 4 in [1] established soundness and correctness of the axiomatic system, respectively. That is, for all implication $a \Rightarrow b \in \mathcal{L}$ and all theory $\Sigma \subseteq \mathcal{L}$, the following equivalence holds

$$\Sigma \vdash a \Rightarrow b \text{ if and only if } \Sigma \models a \Rightarrow b. \qquad (14)$$

The above result allows us to define the twofold notion of equivalence in their syntactic and semantic ways. The second version of such notion is the following: two theories Σ_1 and Σ_2 are said to be equivalent, denoted $\Sigma_1 \equiv \Sigma_2$ if all their models coincide.

One outstanding characteristic of this axiomatic system is that each inference rule can be paired with an equivalence rule providing a way to transform theories into a more simple one, preserving its semantics:

$$\textit{Decomposition} : \{a \Rightarrow b\} \equiv \{a \Rightarrow b \ominus a\} \qquad (\texttt{DeEq})$$

$$\textit{Composition} : \{a \Rightarrow b, a \Rightarrow c\} \equiv \{a \Rightarrow b \oplus c\} \qquad (\texttt{CoEq})$$

Simplification : if $a \leqslant c$,

$$\{a \Rightarrow b, c \Rightarrow d\} \equiv \{a \Rightarrow b, c \ominus b \Rightarrow d \ominus b\} \quad (\texttt{SiEq})$$

Example 2. Let $\mathbb{L}^{\mathbb{N}}$ be the dual complete residuated lattice and S be the compact $\mathbb{L}$-parameterization defined in Example 1. Consider the theory

$$\Sigma = \big\{\{1/_{0.3}, 3/_{0.6}\}\Rightarrow\{2/_{0.7}\},\ \{3/_{0.7}\}\Rightarrow\{1/_{0.6}\}\big\}.$$

The following sequence shows that $\Sigma \vdash \{3/_{0.6}\}\Rightarrow\{2/_{0.7}\}$ holds.

(i) $\{1/_{0.3}, 3/_{0.6}\}\Rightarrow\{2/_{0.7}\}$ .... by hypothesis.
(ii) $\{3/_{0.7}\}\Rightarrow\{1/_{0.6}\}$ .... by hypothesis.
(iii) $\{3/_{0.5}\}\Rightarrow\{1/_{0.4}\}$ ... by (ii) and (Mul) with $\boldsymbol{f}_{0.2}$.
(iv) $\{3/_{0.6}\}\Rightarrow\{2/_{0.7}\}$ .... by (iii), (i) and (Simp).

On the other side, by (SiEq), we have that Σ can be simplified: $\Sigma \equiv \big\{\{3/_{0.6}\}\Rightarrow\{2/_{0.7}\},\ \{3/_{0.7}\}\Rightarrow\{1/_{0.6}\}\big\}$. □

In a natural way, theories are tied with a set of models. We have characterised a canonical model, named syntactic closure, in this way: Given a theory $\Sigma \subseteq \mathcal{L}$, we define $\boldsymbol{c}_\Sigma(a) : L \to L$ where for each $a \in L$, $\boldsymbol{c}_\Sigma(a) = \bigvee \mathcal{C}_\Sigma(a)$ and

$$\mathcal{C}_\Sigma(a) = \{b \in K \mid \Sigma \vdash c \Rightarrow b \text{ for some } c \in K \text{ with } c \leq a\}.$$

From Theorems 2 and 3 in [1], this operator is an additive $\mathbb{S}$-closure operator, which is also a characteristic model of the theory: for any $\Sigma \subseteq \mathcal{L}$ and $a \Rightarrow b \in \mathcal{L}$, we have:

$$\Sigma \vdash a \Rightarrow b \text{ if and only if } b \leq \boldsymbol{c}_\Sigma(a) \qquad (15)$$

This closure operator can be characterized by means of the compact elements as follows:

$$\boldsymbol{c}_\Sigma(a) = \bigvee\{\boldsymbol{c}_\Sigma(x) \mid x \in K, x \leq a\} \qquad (16)$$

and, if $a \in K$, then

$$\boldsymbol{c}_\Sigma(a) = \bigvee\{b \in K \mid \Sigma \vdash a \Rightarrow b\}. \qquad (17)$$

Observe that $a \in K$ does not imply $\boldsymbol{c}_\Sigma(a) \in K$.

Example 3. Let $\mathbb{L}^{\mathbb{N}}$ be the dual complete residuated lattice and S be the compact $\mathbb{L}$-parameterization defined in Example 1. Consider the compact set $X = \{2/_{0.8}\}$ and the theory $\Sigma = \big\{\{n/_{0.6}\}\Rightarrow\{(n+1)/_{0.7}\} \mid n \in \mathbb{N}\big\}$ Then, $\boldsymbol{c}_\Sigma(X) = \{1/_{0.5}, 2/_{0.8}\} \cup \{n/_{0.7} \mid n > 2\}$, which is not a compact element. We remark that this situation does not only hold for infinite theories. Consider now the finite theory $\Sigma = \{\{1/_{0.3}\} \Rightarrow \{1/_{0.7}\},\ \{2/_{0.6}\} \Rightarrow \{2/_{0.9}\}\}$. Then, $c_\Sigma(X) = \{1/_{0.7}, 2/_{0.9}\} \cup \{n/_{0.5} \mid n \in \mathbb{N}, n > 2\}$. The reason behind this situation is indeed the fact that the closure of the empty set is not a compact element: $\boldsymbol{c}_\emptyset(\emptyset) = \{n/_{0.5} \mid n \in \mathbb{N}\}$.

III. Complete theories for additive S-closure operators

In this section, we establish an isotone Galois connection between models and theories. We study when a theory characterises a given model and characterises the minimum model corresponding to the empty theory.

Definition 2. Let $\boldsymbol{c}\colon L \to L$ be an additive S-closure operator. A theory $\Sigma \subseteq \mathcal{L}$ is said to be *complete* for $\boldsymbol{c}$ if the following equivalence holds for all $a \Rightarrow b \in \mathcal{L}$:

$$\Sigma \vdash a \Rightarrow b \text{ if and only if } b \leq \boldsymbol{c}(a)$$

Obviously, the greatest theory (concerning the set inclusion) that is complete for an additive S-closure operator $\boldsymbol{c}$ is

$$\Sigma_{\boldsymbol{c}} = \{a \Rightarrow b \in \mathcal{L} \mid b \leq \boldsymbol{c}(a)\}$$

and, for any other complete theory Σ, we have that $\Sigma \equiv \Sigma_{\boldsymbol{c}}$.

Example 4. For the additive $\mathbb{S}$-closure operator $\boldsymbol{c}$ introduced in Example 1, we have that

$$\begin{aligned}\Sigma_{\boldsymbol{c}} =&\{\{n/_a\} \Rightarrow \{n/_b\} \mid n \in \mathbb{N}, a, b \in L, a, b \leq 0.6\}\\ &\cup\{\{n/_a\} \Rightarrow \{n/_b\} \mid n \in \mathbb{N}, a, b \in L, 0.6 < a, 0.8 \geq b\}\\ &\cup\{\{n/_a\} \Rightarrow \{n/_b\} \mid n \in \mathbb{N}, a, b \in L, 0.8 < a\}\end{aligned}$$

But it is not the unique complete theory for $\boldsymbol{c}$. For instance,

$$\Sigma = \{\emptyset \Rightarrow \{n/_{0.5}\},\ \{n/_{0.9}\} \Rightarrow \{n/_1\} \mid n \in \mathbb{N}\}$$

is also complete for $\boldsymbol{c}$. □

Theorem 1. *Let S be an $\mathbb{L}$-parameterization. If Σ is complete theory for $\boldsymbol{c}$, then $\boldsymbol{c}_\Sigma(a) = \boldsymbol{c}(a)$ for all $a \in K$.*

In addition, $\boldsymbol{c}_\Sigma$ is the greatest algebraic additive S-closure operator such that $\boldsymbol{c}_\Sigma \leq \boldsymbol{c}$.

Proof. First, since $\mathbb{L}$ is algebraic, Σ is complete for $\boldsymbol{c}$ and (17) holds, we have that, for all $a \in K$,

$$\begin{aligned}\boldsymbol{c}(a) &= \bigvee\{b \in K \mid b \leq \boldsymbol{c}(a)\}\\ &= \bigvee\{b \in K \mid \Sigma \vdash a \Rightarrow b\} = \boldsymbol{c}_\Sigma(a)\end{aligned}$$

Second, from (16), we have that, for all $a \in L$,

$$\begin{aligned}\boldsymbol{c}_\Sigma(a) &= \bigvee\{\boldsymbol{c}_\Sigma(x) \mid x \in K, x \leq a\}\\ &= \bigvee\{\boldsymbol{c}(x) \mid x \in K, x \leq a\} \leq \boldsymbol{c}(a)\end{aligned}$$

Finally, assume that there exists an algebraic additive S-closure operator $\boldsymbol{c}'\colon L \to L$ such that $\boldsymbol{c}'(a) \leq \boldsymbol{c}(a)$ for all $a \in L$. Then

$$\begin{aligned}\boldsymbol{c}'(a) &= \bigvee\{\boldsymbol{c}'(x) \mid x \in K, x \leq a\}\\ &\leq \bigvee\{\boldsymbol{c}(x) \mid x \in K, x \leq a\} = \boldsymbol{c}_\Sigma(a)\end{aligned}$$

□

Corollary 1. If $\boldsymbol{c}$ is an algebraic additive S-closure operator and Σ is complete for $\boldsymbol{c}$, then $\boldsymbol{c} = \boldsymbol{c}_\Sigma$.

The following theorem ensures that the pair of mappings $\boldsymbol{c} \rightsquigarrow \Sigma_{\boldsymbol{c}}$ and $\Sigma \rightsquigarrow \boldsymbol{c}_\Sigma$ is an isotone Galois connection between the set of additive S-closure operators with the induced relation $\leq$ (which is an order relation) and the set of theories with the preorder relation given by $\vdash$.

Theorem 2. *Let $\boldsymbol{c}$ be an additive S-closure operator and $\Sigma \subseteq \mathcal{L}$ be a theory. The following equivalences hold:*

$$\boldsymbol{c}_\Sigma \leq \boldsymbol{c} \textit{ if and only if } \Sigma \subseteq \Sigma_{\boldsymbol{c}} \textit{ (or, equivalently, iff } \Sigma_{\boldsymbol{c}} \vdash \Sigma).$$

Proof. On the one hand, we prove that, if $\boldsymbol{c}_\Sigma(x) \leq c(x)$ for all $x \in L$, then $\Sigma \subseteq \Sigma_c$. For all $a \Rightarrow b \in \Sigma$, we have that $\Sigma \vdash a \Rightarrow b$ and, by (15), $b \leq \boldsymbol{c}_\Sigma(a) \leq \boldsymbol{c}(a)$. Therefore, $a \Rightarrow b \in \Sigma_{\boldsymbol{c}}$.

On the other hand, if $\Sigma \subseteq \Sigma_c$ then $\Sigma_{\boldsymbol{c}} \vdash \Sigma$ and, for all $a \in L$,

$$\begin{aligned}\boldsymbol{c}_\Sigma(a) &= \bigvee\{b \in K \mid \Sigma \vdash c \Rightarrow b, c \in K, c \leq a\}\\ &\leq \bigvee\{b \in K \mid \Sigma_{\boldsymbol{c}} \vdash c \Rightarrow b, c \in K, c \leq a\}\\ &= \bigvee\{b \in K \mid b \leq \boldsymbol{c}(c), c \in K, c \leq a\} \leq \boldsymbol{c}(a)\end{aligned}$$

□

Corollary 2. Let $\Sigma_1, \Sigma_2 \subseteq \mathcal{L}$ and $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ be algebraic additive S-closure operators.

1) If $\Sigma_1 \subseteq \Sigma_2$ then $\boldsymbol{c}_{\Sigma_1} \leq \boldsymbol{c}_{\Sigma_2}$.
2) If $\boldsymbol{c}_1 \leq \boldsymbol{c}_2$ then $\Sigma_{\boldsymbol{c}_1} \subseteq \Sigma_{\boldsymbol{c}_2}$ and $\Sigma_{\boldsymbol{c}_2} \vdash \Sigma_{\boldsymbol{c}_1}$.

Corollary 3. Let $\Sigma \subseteq \mathcal{L}$ and $\boldsymbol{c}$ be an algebraic additive S-closure operator. Then,

$$\boldsymbol{c} = \boldsymbol{c}_{\Sigma_{\boldsymbol{c}}} \quad \text{and} \quad \Sigma \equiv \Sigma_{\boldsymbol{c}_\Sigma}.$$

Lemma 1. *Let $\boldsymbol{c}$ be an additive S-closure operator. For any $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$ and $a \in L$, we have $\boldsymbol{g}(\boldsymbol{c}(a)) \oplus \boldsymbol{g}(\boldsymbol{c}(a)) = \boldsymbol{g}(\boldsymbol{c}(a))$.*

Proof. From (4), we have $\boldsymbol{g}(\boldsymbol{c}(a)) \leq \boldsymbol{g}(\boldsymbol{c}(a)) \oplus \boldsymbol{g}(\boldsymbol{c}(a))$.

Conversely, since $\boldsymbol{c}$ is additive, we have

$$\boldsymbol{g}(\boldsymbol{c}(a)) \oplus \boldsymbol{g}(\boldsymbol{c}(a)) \leq \boldsymbol{c}\big(\boldsymbol{g}(\boldsymbol{c}(a)) \vee \boldsymbol{g}(\boldsymbol{c}(a))\big) = \boldsymbol{c}(\boldsymbol{g}(\boldsymbol{c}(a)))$$

Finally, since $\boldsymbol{c}$ is an S-closure operator, we have $\boldsymbol{g}(\boldsymbol{c}(a)) \oplus \boldsymbol{g}(\boldsymbol{c}(a)) \leq \boldsymbol{g}(\boldsymbol{c}(a))$. □

Theorem 3. *For any $a \in L$ and for all $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$ we have*

$$\boldsymbol{c}_\emptyset(a) = \bigwedge\{x \in L \mid a \leq x,\ \textit{and } \boldsymbol{g}(x) \oplus \boldsymbol{g}(x) = \boldsymbol{g}(x)\}.$$

Proof. We will prove that $\boldsymbol{c}\colon L \to L$, defined as

$$\boldsymbol{c}(a) = \bigwedge\{x \in L \mid a \leq x,\ \text{and } \boldsymbol{g}(x) \oplus \boldsymbol{g}(x) = \boldsymbol{g}(x)\}$$

is an additive S-closure operator, and then we will prove that $\boldsymbol{c}_\emptyset = \boldsymbol{c}$.

First, we prove that the set

$$\mathcal{C} = \{x \in L \mid \boldsymbol{g}(x) \oplus \boldsymbol{g}(x) = \boldsymbol{g}(x) \text{ for all } \langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S\}$$

is a closure system:

- $1 \in \mathcal{C}$ because $\boldsymbol{g}(1) \oplus \boldsymbol{g}(1) = 1 \oplus 1 = 1 = \boldsymbol{g}(1)$ for all $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$.

- For all $X \subseteq \mathcal{C}$ we have that $\bigwedge_{x\in X} x \in \mathcal{C}$ because

$$\begin{aligned}
&\boldsymbol{g}(\bigwedge_{x\in X} x) \oplus \boldsymbol{g}(\bigwedge_{x\in X} x)\\
=&\big(\bigwedge_{x\in X} \boldsymbol{g}(x)\big) \oplus \big(\bigwedge_{x\in X} \boldsymbol{g}(x)\big)\\
=&\bigwedge_{x\in X} \big(\boldsymbol{g}(x) \oplus \bigwedge_{y\in X} \boldsymbol{g}(y)\big)\\
=&\bigwedge_{x\in X}\bigwedge_{y\in X} \big(\boldsymbol{g}(x) \oplus \boldsymbol{g}(y)\big)\\
=&\bigwedge_{x\in X} \big(\boldsymbol{g}(x) \oplus \boldsymbol{g}(x)\big) \wedge \bigwedge_{x,y\in X, x\neq y} \big(\boldsymbol{g}(x) \oplus \boldsymbol{g}(y)\big)\\
=&\big(\bigwedge_{x\in X} \boldsymbol{g}(x)\big) \wedge \bigwedge_{x,y\in X, x\neq y} \big(\boldsymbol{g}(x) \oplus \boldsymbol{g}(y)\big)\\
=&\bigwedge_{x\in X} \boldsymbol{g}(x) = \boldsymbol{g}(\bigwedge_{x\in X} x)
\end{aligned}$$

  where, in the first and the last equalities, we have used that $\langle \boldsymbol{f}, \boldsymbol{g}\rangle$ is an isotone Galois connection between complete lattices, we have used (9) in the second and third equalities, the definition of $\mathcal{C}$ in the fifth one, and (4) in the sixth one.

As a consequence, the mapping $\boldsymbol{c}$ is a closure operator. In addition, for all $a \in L$ we have that $\boldsymbol{c}(a) \in \mathcal{C}$ because $\boldsymbol{c}(a) = \bigwedge\{x \in \mathcal{C} \mid a \le x\}$.

Second, we prove that $\boldsymbol{c}$ is an additive S-closure operator:

- $\boldsymbol{c}$ is additive because, for all $a, b \in L$, since $a, b \le \boldsymbol{c}(a \vee b)$, $\boldsymbol{c}(a \vee b) \in \mathcal{C}$ and $\langle I, I\rangle \in S$, we have that $a \oplus b \le \boldsymbol{c}(a \vee b) \oplus \boldsymbol{c}(a \vee b) = \boldsymbol{c}(a \vee b)$.
- Since S is closed for composition, we have that $\boldsymbol{g}(\boldsymbol{c}(a)) \in \mathcal{C}$ for all $a \in L$ and $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$ and, therefore, $\boldsymbol{c}(\boldsymbol{g}(\boldsymbol{c}(a))) \le \boldsymbol{g}(\boldsymbol{c}(a))$.

Now, from Theorem 2, we have that $\boldsymbol{c}_\emptyset \le \boldsymbol{c}$ and, from Lemma 1, we have $\boldsymbol{c}_\emptyset(a) \in \mathcal{C}$ for all $a \in L$, i.e, $\boldsymbol{c} \le \boldsymbol{c}_\emptyset$. □

IV. Automated reasoning method

Recall that the classic deduction theorem of propositional logic says that $\Sigma \vdash \varphi \Rightarrow \psi$ if and only if $\Sigma \cup \{\varphi\} \vdash \psi$. Using the fact that any propositional formula χ is equivalent to $\top \Rightarrow \chi$ where $\top$ denotes a tautology, the classic deduction theorem can be equivalently restated as $\Sigma \vdash \varphi \Rightarrow \psi$ if and only if $\Sigma \cup \{\top \Rightarrow \varphi\} \vdash \top \Rightarrow \psi$.

The automatic reasoning method we propose here is intended to answer the question of whether $a \Rightarrow b$ can be inferred from a theory Σ based on two pillars: one is a theorem of deduction reminiscent of propositional logic, and the other is a set of transformations that *simplify* the theory $\Sigma \cup \{0 \Rightarrow a\}$.

Theorem 4 (Deduction theorem)**.** *Let S be an L-parametrization. The following conditions are equivalent:*

1) For all $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$ and $a, b \in K$,

$$\boldsymbol{f}(a \oplus b) \le \boldsymbol{c}_\emptyset(a \oplus \boldsymbol{f}(b)).$$

2) For all $\Sigma \subseteq \mathcal{L}$ and $a, b, c \in K$,

$$\Sigma \cup \{0 \Rightarrow a\} \vdash b \Rightarrow c \;\; \textit{iff} \;\; \Sigma \vdash a \oplus b \Rightarrow c.$$

3) For all $\Sigma \subseteq \mathcal{L}$ and $a, c \in K$,

$$\Sigma \vdash a \Rightarrow c \;\; \textit{iff} \;\; \Sigma \cup \{0 \Rightarrow a\} \vdash 0 \Rightarrow c.$$

Proof. First, we prove that 1. implies 2.

Assume $\Sigma \vdash a \oplus b \Rightarrow c$. The following sequence proves $\Sigma \cup \{0 \Rightarrow a\} \vdash b \Rightarrow c$:

(i) $0 \Rightarrow a$ .... by hypothesis.
(ii) $a \oplus b \Rightarrow c$ .... by hypothesis.
(iii) $(a \oplus b) \ominus a \Rightarrow c$ .... by (i), (ii) and (`Simp`).
(iv) $b \Rightarrow (a \oplus b) \ominus a$ .... by (6) and (`GRef`).
(v) $b \Rightarrow c$ .... by (iv), (iii) and (`Tran`).

Conversely, we prove by induction that $\Sigma \cup \{0 \Rightarrow a\} \vdash b \Rightarrow c$ implies $\Sigma \vdash a \oplus b \Rightarrow c$:

Case 1: If $b \Rightarrow c \in \Sigma \cup \{0 \Rightarrow a\}$ we have two possibilities:

- If $b = 0$ and $c = a$, by (`Ref`), we obtain $\Sigma \vdash a \oplus b \Rightarrow c$.
- If $b \Rightarrow c \in \Sigma$, by (`GRef`), $\Sigma \vdash a \oplus b \Rightarrow b$ and, by (`Tran`), $\Sigma \vdash a \oplus b \Rightarrow c$.

Case 2: If $b \Rightarrow c$ is an axiom (i.e. $b = c$) then, by (`GRef`), $\Sigma \vdash a \oplus b \Rightarrow c$.

Case 3: If $b \Rightarrow c$ is the result of applying (`Comp`) to $b \Rightarrow d_1$ and $b \Rightarrow d_2$ with $c = d_1 \oplus d_2$ and, by induction hypothesis $\Sigma \vdash a \oplus b \Rightarrow d_1$ and $\Sigma \vdash a \oplus b \Rightarrow d_2$, then, by using also (`Comp`), we obtain $\Sigma \vdash a \oplus b \Rightarrow d_1 \oplus d_2$, i.e. $\Sigma \vdash a \oplus b \Rightarrow c$.

Case 4: If $b \Rightarrow c$ is the result of applying (`Simp`) to $u \Rightarrow v$ and $w \Rightarrow c$ (i.e. $b = u \oplus (w \ominus v)$), the following sequence proves $\Sigma \vdash a \oplus b \Rightarrow c$:

(i) $a \oplus u \Rightarrow v$ .... by induction hypothesis.
(ii) $a \oplus w \Rightarrow c$ .... by induction hypothesis.
(iii) $a \oplus u \oplus ((a \oplus w) \ominus v) \Rightarrow c$ .... by (i), (ii) and (`Simp`).
(iv) $a \oplus a \oplus b \Rightarrow a \oplus u \oplus ((a \oplus w) \ominus v)$ .. by (5), (7) and (`GRef`).
(v) $a \oplus a \oplus b \Rightarrow c$ .... by (iv), (iii) and (`Tran`).
(vi) $a \Rightarrow a$ .... by (`Ref`).
(vii) $a \Rightarrow a \oplus a$ .... by (vi) and (`Comp`).
(viii) $a \vee b \Rightarrow a \oplus a \oplus b$ .... by (vii) and (`Augm`).
(ix) $a \oplus b \Rightarrow a \vee b$ .... by (6) and (`GRef`).
(x) $a \oplus b \Rightarrow a \oplus a \oplus b$ ... by (ix), (viii) and (`Tran`).
(xi) $a \oplus b \Rightarrow c$ .... by (x), (v) and (`Tran`).

Case 5: If $b \Rightarrow c$ is the result of applying (`Mul`) to $u \Rightarrow v$ (i.e. $b = \boldsymbol{f}(u)$ and $c = \boldsymbol{f}(v)$ for some $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$) and, by induction hypothesis, $\Sigma \vdash a \oplus u \Rightarrow v$, then, by (`Mul`), $\Sigma \vdash \boldsymbol{f}(a \oplus u) \Rightarrow \boldsymbol{f}(v)$. From 1. and Corollary 2, we have that $\boldsymbol{f}(a \oplus u) \le \boldsymbol{c}_\emptyset(a \oplus \boldsymbol{f}(u)) \le \boldsymbol{c}_\Sigma(a \oplus \boldsymbol{f}(u))$ and, then, $\Sigma \models a \oplus b \Rightarrow \boldsymbol{f}(a \oplus u)$. Now, from (14), we have that $\Sigma \vdash a \oplus b \Rightarrow \boldsymbol{f}(a \oplus u)$ and, by (`Tran`), $\Sigma \vdash a \oplus b \Rightarrow c$.

Second, we prove that 2. implies 1.

For all $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$ and $a, b \in K$, the following sequence proves that $\{0 \Rightarrow a\} \vdash \boldsymbol{f}(b) \Rightarrow \boldsymbol{f}(a \oplus b)$:

(i) $0 \Rightarrow a$ .... by hypothesis.
(ii) $b \Rightarrow b$ .... by (`Ref`).
(iii) $b \Rightarrow a \oplus b$ .... by (i), (ii) and (`GComp`).
(iv) $\boldsymbol{f}(b) \Rightarrow \boldsymbol{f}(a \oplus b)$ .... by (iii) and (`Mul`).

Therefore, from condition 2., $\emptyset \vdash a \oplus \boldsymbol{f}(b) \Rightarrow \boldsymbol{f}(a \oplus b)$ and, from (14), we have that $\boldsymbol{f}(a \oplus b) \le \boldsymbol{c}_\emptyset(a \oplus \boldsymbol{f}(b))$.

Finally, it is straightforward that 2. implies 3. (consider $b = 0$), and the following sequence proves that 3. implies 2.

$$\begin{array}{rcll} \{0 \Rightarrow a\} \cup \Sigma & \vdash & b \Rightarrow c & \text{iff (by 3.)} \\ \{0 \Rightarrow b\} \cup \{0 \Rightarrow a\} \cup \Sigma & \vdash & 0 \Rightarrow c & \text{iff (by } (\texttt{CoEq})) \\ \{0 \Rightarrow a \oplus b\} \cup \Sigma & \vdash & 0 \Rightarrow c & \text{iff (by 3.)} \\ \Sigma & \vdash & a \oplus b \Rightarrow c & \end{array}$$

□

Notice that, when the addition $\oplus$ is idempotent (i.e. $\oplus$ and $\vee$ coincide), the first condition in Theorem 4 becomes to $\boldsymbol{f}(a \vee b) \le a \vee \boldsymbol{f}(b)$ for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$ and $a, b \in K$.

Example 5. Let $\mathbb{L}^\mathbb{N}$ be the dual complete residuated lattice and S be the compact $\mathbb{L}$-parameterisation defined in Example 1. For all $\langle \boldsymbol{f}_a, \boldsymbol{g}_a \rangle \in S$ and $x, y \in L$, we have that

$$\boldsymbol{f}_a(x \oplus y) = \begin{cases} \max\{0, x + y - a\} & \text{if } x + y \le 0.5, \\ \max\{0, 0.5 - a, x - a, y - a\} & \text{otherwise.} \end{cases}$$

and $\boldsymbol{f}_a(x \oplus y) \le x \oplus \boldsymbol{f}_a(y) = \max\{x, y - a\}$. As a consequence, for all $X, Y \in \mathbb{L}^\mathbb{N}$ being compact, we have that $\boldsymbol{f}(X \oplus Y) \le X \oplus \boldsymbol{f}(Y) \le \boldsymbol{c}_\emptyset(X \oplus \boldsymbol{f}(Y))$.

c The main goal of this paper is to design a general method to compute the closure $\boldsymbol{c}_\Sigma$, which leads to defining automated reasoning methods. A keystone in this direction is the Deduction Theorem, as Theorem 4 enunciates in condition 3. From the mentioned Theorem, this condition can be reduced to check if the S-parameterization fulfils condition 1. In the rest of the paper, we are going to work with L-parameterization fitting in the following definition:

Definition 3. A $\mathbb{L}$-parameterization $\mathbb{S} = \langle S, \circ, I \rangle$ is said to be *tractable* if it is finite, compact and, for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$ and $a, b \in K$, $\boldsymbol{f}(a \oplus b) \le \boldsymbol{c}_\emptyset(a \oplus \boldsymbol{f}(b))$.

As a first step to designing this general method, we built the closure of the base case $\boldsymbol{c}_\emptyset$.

Theorem 5. *For all $a, k \in K$ and $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, if $k \le \boldsymbol{g}(a)$, then $\{0 \Rightarrow a\} \equiv \{0 \Rightarrow a \oplus \boldsymbol{f}(k \oplus k)\}$. In addition, $\boldsymbol{c}_\emptyset(a) = a$ if and only if the following condition holds:*

$$a = a \oplus \boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a)) \quad \textit{for all} \quad \langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S. \tag{18}$$

Proof. By (`GRef`) and (`Tran`), we have $\{0 \Rightarrow a \oplus \boldsymbol{f}(k \oplus k)\} \vdash 0 \Rightarrow a$. The following sequence proves $\{0 \Rightarrow a\} \vdash 0 \Rightarrow a \oplus \boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a))$:

(i) $0 \Rightarrow a$ .... by hypothesis.
(ii) $k \Rightarrow k$ .... by (`Ref`).
(iii) $k \Rightarrow k \oplus k$ .... by (ii) and (`Comp`).
(iv) $\boldsymbol{f}(k) \Rightarrow \boldsymbol{f}(k \oplus k)$ .... by (iii) and (`Mul`).
(v) $a \Rightarrow \boldsymbol{f}(k)$ .... by (`GRef`) because $\boldsymbol{f}(k) \le \boldsymbol{f}(\boldsymbol{g}(a)) \le a$.
(vi) $0 \Rightarrow \boldsymbol{f}(k \oplus k)$ ...... by (i), (v), (iv) and (`Tran`).
(vii) $0 \Rightarrow a \oplus \boldsymbol{f}(k \oplus k)$ ...... by (i), (vi) and (`Comp`).

Assume now that (18) holds. Thus, for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, we have that $\boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a)) \le a \oplus \boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a)) = a$ and, since $\langle \boldsymbol{f}, \boldsymbol{g} \rangle$ is an isotone Galois connection, we have that $\boldsymbol{g}(a) \oplus \boldsymbol{g}(a) \le \boldsymbol{g}(a)$. Then, Theorem 5 ensures $\boldsymbol{c}_\emptyset(a) = a$.

Conversely, if $\boldsymbol{c}_\emptyset(a) = a$, by Theorem 5, for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, we have $\boldsymbol{g}(a) \oplus \boldsymbol{g}(a) \le \boldsymbol{g}(a)$ or, equivalently, $\boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a)) \le a$. By other side, since $\langle I, I \rangle \in S$, we have $a \oplus a = a$. Therefore, $a \le a \oplus \boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a)) \le a \oplus a = a$. □

Corollary 4. Let $a \in K$. Then $\boldsymbol{c}_\emptyset(a) = a$ if and only if $\boldsymbol{g}(a) \oplus \boldsymbol{g}(a) = \boldsymbol{g}(a)$ for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$.

If the lattice $\mathbb{L}$ satisfies the Ascending Chain Condition (ACC), c which is equivalent to ensure that any element in the lattice is compact (see [16, Pages 149-153], the above theorem provides an algorithm (Function EmptyClosure) to calculate $\boldsymbol{c}_\emptyset(a)$.

Proposition 1. *If the $\mathbb{L}$-parameterisation S is finite and $K = L$, for all input $a \in K$, Function EmptyClosure finishes. Moreover, if Function EmptyClosure finishes, it returns $\boldsymbol{c}_\emptyset(a)$.*

Proof. It is straightforward from $a \le a \oplus \boldsymbol{f}(\boldsymbol{g}(a) \oplus \boldsymbol{g}(a))$ for all $a \in L$ and Theorem 5. □

Notice that ACC is a sufficient condition but not necessary. In fact, depending on the nature of the $\oplus$ and the parameters, it could be possible that there is an expression to calculate $\boldsymbol{c}_\emptyset(a)$ directly. For instance, if $\oplus$ is idempotent (i.e. $\oplus$ and $\vee$ coincide) then $\boldsymbol{c}_\emptyset(a) = a$ for all $a \in K$.

Function EmptyClosure(a)

```
Input: a ∈ K
/* Assume S is a tractable 𝕃-parameterization.   */
Output: c_∅(a)
begin
    repeat
        a_old = a
        foreach ⟨f, g⟩ ∈ S do  a = a ⊕ f(g(a) ⊕ g(a))
    until a_old == a        /* A fix-point is reached */
    return a
end
c
```

Once we have analysed how compute $\boldsymbol{c}_\emptyset$, we look for a method to compute $\boldsymbol{c}_\Sigma$ where Σ is an arbitrary theory. To this aim, we use Theorem 4 and Corollary 2 that ensures $\boldsymbol{c}_\emptyset \le \boldsymbol{c}_\Sigma$.

The following theorem provides a set of equivalences to simplify $\Sigma \cup \{\emptyset \Rightarrow a\}$ having $0 \Rightarrow a$ as a guide to the simplification.

Theorem 6. *For all $a, b, c \in K$ and $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, the following equivalences hold:*

$$\{0 \Rightarrow a, 0 \Rightarrow b\} \equiv \{0 \Rightarrow a \oplus b\} \tag{19}$$

$$\{0 \Rightarrow a, b \Rightarrow c\} \equiv \{0 \Rightarrow a, b \ominus a \Rightarrow c \ominus a\} \tag{20}$$

$$\{0 \Rightarrow a, b \Rightarrow c\} \equiv \{0 \Rightarrow a \oplus \boldsymbol{f}(c), b \Rightarrow c\} \textit{ if } \boldsymbol{f}(b) \le a \tag{21}$$

Proof. Equivalences (19) and (20) are straightforward from (`CoEq`) and (`SiEq`) respectively. Finally, if $\boldsymbol{f}(b) \le a$, by using (`Mul`), (`GRef`), (`Tran`) and (`Comp`), we have that

$$\begin{aligned} \{0 \Rightarrow a, b \Rightarrow c\} &\equiv \{0 \Rightarrow a, b \Rightarrow c, a \Rightarrow \boldsymbol{f}(b), \boldsymbol{f}(b) \Rightarrow \boldsymbol{f}(c)\} \\ &\equiv \{0 \Rightarrow a, 0 \Rightarrow \boldsymbol{f}(c), b \Rightarrow c\} \\ &\equiv \{0 \Rightarrow a \oplus \boldsymbol{f}(c), b \Rightarrow c\} \end{aligned}$$

□

Function Complete(a, $b \Rightarrow c$)

Input: $a \in K$, $b \Rightarrow c \in \mathcal{L}$

/* Assume S is a tractable $\mathbb{L}$-parameterization. */

Output: $\langle a', b \ominus a' \Rightarrow c \ominus a' \rangle$ where $a' = \boldsymbol{c}_{\{b \Rightarrow c\}}(a)$

begin

Function Complete applies the equivalences given in Theorems 5 and 6 to $\{0 \Rightarrow a, b \Rightarrow c\}$ so that the successive values of a constitute an ascending chain. The stop condition is that this chain reaches a fix-point. If we can hail that these chains are always going to stabilise, the algorithm ends, and, as we will prove later, it gives us $\boldsymbol{c}_{b \Rightarrow c}(a)$.

The following straightforward proposition provides a sufficient condition to ensure the algorithm finishes.

Proposition 2. *If the $\mathbb{L}$-parameterization S is tractable and $K = L$, for all input $a \in K$ and $b \Rightarrow c \in \mathcal{L}$, Function Complete finishes.*

In addition, if Function Complete finishes and $\langle a', b' \Rightarrow c' \rangle$ is the output, then $\{0 \Rightarrow a, b \Rightarrow c\} \equiv \{0 \Rightarrow a', b' \Rightarrow c'\}$.

The closure method can be viewed as a transformation method for implication sets into a kind of normal form. In the following, we are going to introduce the notion of *reduced normal form* for sets of implications, which will lead to a closure-like definition for the parametrised system. First, we present a preliminary definition that will be used later on:

Definition 4. Given $a \in K$, we say that $b \Rightarrow c$ is bounded by a if for all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, if $\boldsymbol{f}(b) \leq a$ then $\boldsymbol{f}(c) \leq a$.

Definition 5. A theory $\Sigma \subseteq \mathcal{L}$ is said to be in *reduced normal form* if exists $0 \Rightarrow a \in \Sigma$ and the following conditions hold:

1) $\boldsymbol{c}_{\emptyset}(a) = a$.
2) any $b \Rightarrow c \in \Sigma$ is bounded by a.

The following lemma better characterises the shape of the reduced normal form.

Lemma 2. *Given $\Sigma \subseteq \mathcal{L}$, if Σ is in reduced normal form, then there exists a unique $a \Rightarrow b \in \Sigma$ such that $a = 0$.*

Proof. Given $\Sigma \subseteq \mathcal{L}$ in reduced normal form, let us suppose that there exist $a, b \in K$ where $a \neq b$ and $0 \Rightarrow a, 0 \Rightarrow b \in \Sigma$. From Definition 5, we have that $0 \Rightarrow b$ is bounded by a. Considering $\langle \boldsymbol{f}, \boldsymbol{g} \rangle = \langle \mathtt{I}, \mathtt{I} \rangle$ in Definition 4, we have that $b \leq a$. Conversely, following the same reasoning we also conclude that $a \leq b$ and the uniqueness of $0 \Rightarrow a$ in Σ is proved. □

Theorem 7. *Given $a \in K$ and $\Sigma \subseteq \mathcal{L}$. If Σ is in reduced normal form and $0 \Rightarrow a \in \Sigma$ then $\boldsymbol{c}_{\Sigma}(a) = a$*

Proof. First, we prove that bounding property can be extended to all implication derived from a Σ set in reduced normal form. Thus, we now prove by induction that all $b \Rightarrow c \in \mathcal{L}$ such that $\Sigma \vdash b \Rightarrow c$ fulfils the bounding property:

$$\text{For all } \langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S, \text{ if } \boldsymbol{f}(b) \leq a \text{ then } \boldsymbol{f}(c) \leq a \qquad (22)$$

CASE 1: If $b \Rightarrow c \in \Sigma$, by condition 4 in Definition 5, the formula $b \Rightarrow c$ satisfies (22).

CASE 2 (Ref): It is straightforward that any axiom $b \Rightarrow b$ satisfies (22).

CASE 3 (Comp): Assume, as induction hypothesis, that $b \Rightarrow c_1, b \Rightarrow c_2 \in \mathcal{L}$ such that $\Sigma \vdash b \Rightarrow c_1$ and $\Sigma \vdash b \Rightarrow c_2$ satisfy (22), and prove that $b \Rightarrow c_1 \oplus c_2$ also satisfies (22). For all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, if $\boldsymbol{f}(b) \leq a$, by induction hypothesis $\boldsymbol{f}(c_1) \leq a$ and $\boldsymbol{f}(c_2) \leq a$. Then $c_1 \leq \boldsymbol{g}(a)$ and $c_2 \leq \boldsymbol{g}(a)$. From 1 (Def. 5) and Lemma 1, we have that $c_1 \oplus c_2 \leq \boldsymbol{g}(a) \oplus \boldsymbol{g}(a) = \boldsymbol{g}(a)$ and, therefore, $\boldsymbol{f}(c_1 \oplus c_2) \leq a$.

CASE 4 (Simp): Assume, as induction hypothesis, that $b_1 \Rightarrow c_1, b_2 \Rightarrow c_2 \in \mathcal{L}$ such that $\Sigma \vdash b_1 \Rightarrow c_1$ and $\Sigma \vdash b_2 \Rightarrow c_2$ satisfy (22), and prove that $b \Rightarrow c_2$ with $b = b_1 \oplus (b_2 \ominus c_1)$ also satisfies (22). For all $\langle \boldsymbol{f}, \boldsymbol{g} \rangle \in S$, we prove that, if $\boldsymbol{f}(b) \leq a$ then $\boldsymbol{f}(c_2) \leq a$.

Since $b_1 \leq b_1 \oplus (b_2 \ominus c_1)$, we have $\boldsymbol{f}(b_1) \leq \boldsymbol{f}(b) \leq a$ and, by induction hypothesis, $\boldsymbol{f}(c_1) \leq a$ and $c_1 \leq \boldsymbol{g}(a)$.

In addition, since $b_2 \ominus c_1 \leq b_1 \oplus (b_2 \ominus c_1)$, we have that $\boldsymbol{f}(b_2 \ominus c_1) \leq \boldsymbol{f}(b) \leq a$ and then, we have $b_2 \ominus c_1 \leq \boldsymbol{g}(a)$. By the residuation property, $b_2 \leq g(a) \oplus c_1$.

From $c_1 \leq \boldsymbol{g}(a)$ we induce $b_2 \leq \boldsymbol{g}(a) \oplus \boldsymbol{g}(a) = \boldsymbol{g}(a)$ from 1 (Def. 5) and Lemma 1. We obtain $b_2 \leq \boldsymbol{g}(a)$ and, applying residuation, $\boldsymbol{f}(b_2) \leq a$ which, applying bounding property and the induction hypothesis, we have $\boldsymbol{f}(c_2) \leq a$.

CASE 5 (Mul): Assume as induction hypothesis that $b \Rightarrow c$ is bounded by a. We have to prove that for all $\langle \boldsymbol{f}_1, \boldsymbol{g}_1 \rangle \in S$, $\boldsymbol{f}_1(b) \Rightarrow \boldsymbol{f}_1(c)$ is bounded by a. That is, for all $\langle \boldsymbol{f}_2, \boldsymbol{g}_2 \rangle \in S$ if $\boldsymbol{f}_2(\boldsymbol{f}_1(b)) \leq a$ then $\boldsymbol{f}_2(\boldsymbol{f}_1(c)) \leq a$.

It is straightforward since $\langle \boldsymbol{f}_2 \boldsymbol{f}_1, \boldsymbol{g}_1 \boldsymbol{g}_2 \rangle \in S$ and induction hypothesis ensures it.

□

We can add an extra condition to the reduced normal form, introducing the following definition:

Definition 6. Given $\Sigma \subseteq \mathcal{L}$, we say that Σ is in *simplified normal form* is Σ is in reduced normal form, and the following condition holds:

$$\text{If } 0 \Rightarrow a, b \Rightarrow c \in \Sigma \text{ then } b = b \ominus a \text{ and } c = c \ominus a.$$

Simplified normal form definition induces a stronger notion than reduced normal form since it adds a more efficient management providing the inclusion of the notion of minimality to the implication sets.

Theorem 8. *If the $\mathbb{L}$-parameterization S is tractable, $K = L$ and Σ is finte, for all $a \in K$, Algorithm 1 finishes.*

In addition, if Algorithm 1 finishes, Σ_1 is the last computed theory and a' is the output, $\{0 \Rightarrow a'\} \cup \Sigma_1$ is in simplified normal form and is equivalent to $\{0 \Rightarrow a\} \cup \Sigma$.

Corollary 5. If Algorithm 1 finishes for an input a and Σ, the output is $\boldsymbol{c}_{\Sigma}(a)$.

The condition $K = L$ is easily assumed. Even if it is not satisfied, sometimes we can find an element $k \in K$ such that the interval $a, \boldsymbol{c}_{\Sigma}(a) \in [0, k] \subseteq K$ and reduce the problem to

Algorithm 1: S-closure algorithm

```
Input: Σ ⊆ ℒ finite, a ∈ K
/* Assume S is a tractable 𝕃-parameterization.   */
Output: c_Σ(a)
begin
    repeat
        Σ_1 = {b ⇒ c ⊖ b | b ⇒ c ∈ Σ}
        Σ = ∅
        foreach b ⇒ c ∈ Σ_1 do
            ⟨a, b ⇒ c⟩ = Complete(a, b ⇒ c)
            if c ≠ 0 then Σ = Σ ∪ {b ⇒ c}
    until Σ == Σ_1        /* A fix-point is reached */
    return a
```

an equivalent one in the lattice $[0, a]$. In other cases, we can consider a finite subalgebra of $\mathbb{L}$ (depending on the input) in the same way that it is usual to consider a finite discretisation when working with the unit interval.

V. Parameterised Simplification Logic for $\mathbb{L}$-fuzzy Sets

This section focuses on the particular case in which the complete dual residuated lattice elements are $\mathbb{L}$-fuzzy sets. Let $\mathbb{L}$ be an algebraic complete dual residuated lattice and $K(L)$ be the set of its compact elements. Consider an arbitrary set Y and the complete dual residuated lattice $\mathbb{L}^Y$ built pointwise in the standard way. Let $K(L^Y)$ be the set of compact elements in $\mathbb{L}^Y$. In addition, an $\mathbb{L}$-parameterization S is pointwise extended to the $\mathbb{L}$-power set as follows: if $\langle \boldsymbol{f}, \boldsymbol{g}\rangle \in S$, then $(\boldsymbol{f}(A))(y) = \boldsymbol{f}(A(y))$ and $(\boldsymbol{g}(A))(y) = \boldsymbol{g}(A(y))$ for all $A \in L^Y$ and $y \in Y$.

The support of $A \in L^Y$ will be denoted by Spp(A), i.e. $\mathrm{Spp}(A) = \{y \in Y \mid A(y) > 0\}$. It is extended to implications and theories as follows: $\mathrm{Spp}(A \Rightarrow B) = \mathrm{Spp}(A) \cup \mathrm{Spp}(B)$ and $\mathrm{Spp}(\Sigma) = \bigcup_{A\Rightarrow B\in\Sigma} \mathrm{Spp}(A \Rightarrow B)$.

Lemma 3. *Let $A \in L^Y$. Then $A \in K(L^Y)$ if and only if Spp(A) is finite and $A(y) \in K(L)$ for all $y \in Y$.*

Proof. Assume Spp(A) is finite and $A(y) \in K(L)$ for all $y \in Y$. Let $J \subseteq L^Y$ such that $A \leqslant \bigvee J$. Then, for all $y \in Y$, $A(y) \leqslant \bigvee_{j\in J} j(y)$ and there exists a finite subset $J_y \subseteq J$ such that $A(y) \leqslant \bigvee_{j\in J_y} j(y)$. Since Spp$(A)$ is finite, $J' = \bigcup_{y\in\mathrm{Spp}(A)} J_y$ is finite and $A \leqslant \bigvee J'$.

In the case of Spp(A) is infinite, consider the infinite set $J = \{j_y \in L^Y \mid y \in \mathrm{Spp}(A)\}$ where $j_y(y) = A(y)$ and, if $x \neq y$, $j_y(x) = 0$. It is straightforward that $A \leqslant \bigvee J$ and $A \nleqslant \bigvee J'$ for all $J' \subsetneq J$. Therefore, $A \notin K(L^Y)$.

Finally, if $A \in L^Y$ and there exists $y \in Y$ such that $A(y) \notin K(L)$, then there exists an infinite $D \subseteq L$ such that $A(y) \leqslant \bigvee D$ and $A(y) \nleqslant \bigvee D'$ for all $D' \subseteq D$ finite. Consider now the infinite set $J = \{j_d \in L^Y \mid d \in D\}$ where $j_d(y) = d$ and, if $x \neq y$, $j_d(x) = A(x)$. Then $A \leqslant \bigvee J$ but $A \nleqslant \bigvee J'$ for all $J' \subseteq J$ finite. □

Theorem 9. *The lattice $\mathbb{L}^Y$ is algebraic.*

Proof. Let $A \in L^Y$. Since L is algebraic, for all $y \in Y$, there exists $D_y \subseteq K(L)$ such that $A(y) = \bigvee D_y$. Define $J_y = \{j_{yd} \in L^Y \mid d \in D_y\}$ where $j_{yd}(y) = d$ and, if $x \neq y$, $j_{yd}(x) = 0$. Thus, if $J = \bigcup_{y\in Y} J_y$, then $A = \bigvee J$ and, by Lemma 3, $J \subseteq K(L^Y)$. □

Corollary 6. If Y is finite and $K(L) = L$ then $K(L^Y) = L^Y$.

As a consequence, in this framework, if $K(L) = L$ then, even the case of Y being infinite, for any finite theory Σ and any implication $A \Rightarrow B$, we can consider the finite set

$$Y_0 = \mathrm{Spp}(\Sigma) \cup \mathrm{Spp}(A \Rightarrow B)$$

and we have that $\Sigma \vdash A \Rightarrow B$ in $\mathbb{L}^Y$ if and only if $\Sigma \vdash A \Rightarrow B$ in $\mathbb{L}^{Y_0}$. Therefore, the algorithms described in this paper can be used to test it.

On the other hand, in order to compute $\boldsymbol{c}_\Sigma(A)$, we reduce the problem to $\mathbb{L}^{Y_0}$ where $Y_0 = \mathrm{Spp}(\Sigma) \cup \mathrm{Spp}(A)$ and, if Algorithm 1 returns A', then $\boldsymbol{c}_\Sigma(A) = A' \vee \boldsymbol{c}_\emptyset(\emptyset)$. Moreover, Algorithm 1 can be used to compute $\boldsymbol{c}_\emptyset(\emptyset)$ also: if $\boldsymbol{c}_\emptyset(0) = \widehat{0}$ in the framework of the lattice $\mathbb{L}$, then $\boldsymbol{c}_\emptyset(\emptyset)(y) = \widehat{0}$ for all $y \in Y$. To illustrate this situation, we introduce the following example:

Example 6. In our running example, $\widehat{0} = 0.5$.

For $X = \{2/_{0.8}\}$ and the finite theory $\Sigma = \{\{1/_{0.3}\} \Rightarrow \{1/_{0.7}\},\ \{2/_{0.6}\} \Rightarrow \{2/_{0.9}\}\}$, instead of $\mathbb{L}^{\mathbb{N}}$ we consider $\mathbb{L}^{Y_0}$ where $Y_0 = \{1, 2\}$. Algorithm 1 returns $\{1/_{0.7}, 2/_{0.9}\}$ and, therefore, $c_\Sigma(X) = \{1/_{0.7}, 2/_{0.9}\} \cup \{n/_{0.5} \mid n \in \mathbb{N}, n > 2\}$.

To end this work, we describe a complete example illustrating the execution of the reasoning method:

Example 7. Let $\mathbb{L}^{\mathbb{Y}}$ be the dual complete residuated lattice where $Y = \{a, b, c, d, e\}$ and S be the compact $\mathbb{L}$-parameterization defined in Example 1. Given the theory

$$\begin{aligned}\Sigma = \big\{\{a/_{0.4}, c/_{0.9}\} &\Rightarrow \{a/_{0.5}, d/_{0.7}, e/_{0.1}\},\\ \{a/_{0.9}, d/_{0.2}\} &\Rightarrow \{c/_{0.6}, e/_{0.7}\},\\ \{b/_{0.5}, e/_{0.7}\} &\Rightarrow \{c/_{0.8}, d/_{0.6}, e/_{0.6}\},\\ \{b/_{0.3}, c/_{0.9}\} &\Rightarrow \{d/_{0.3}, e/_{0.5}\},\\ \{c/_{0.5}, d/_{0.4}, e/_{0.8}\} &\Rightarrow \{a/_{0.8}, b/_{0.9}\},\\ \{d/_{0.6}, e/_{0.2}\} &\Rightarrow \{c/_{0.6}, d/_{0.5}, e/_{0.5}\}\big\}\end{aligned}$$

we compute the closure of $A = \{a/_{0.5}\}$. We build the implication $\emptyset \Rightarrow A$ and, after the first iteration of the *r*epeat loop, $A = \{a/_{0.5}, b/_{0.5}, c/_{0.5}, d/_{0.5}\}$ and the theory is reduced up to:

$$\begin{aligned}\Sigma = \big\{\{c/_{0.9}\} &\Rightarrow \{d/_{0.7}, e/_{0.1}\},\\ \{a/_{0.9}, d/_{0.2}\} &\Rightarrow \{c/_{0.6}, e/_{0.7}\},\\ \{b/_{0.5}, e/_{0.7}\} &\Rightarrow \{c/_{0.8}, d/_{0.6}, e/_{0.6}\},\\ \{b/_{0.3}, c/_{0.9}\} &\Rightarrow \{d/_{0.3}\},\\ \{d/_{0.4}, e/_{0.8}\} &\Rightarrow \{a/_{0.8}, b/_{0.9}\},\\ \{d/_{0.6}\} &\Rightarrow \{c/_{0.6}, d/_{0.5}\}\big\}\end{aligned}$$

After the second iteration, $A = \{a/_{0.5}, b/_{0.5}, c/_{0.5}, d/_{0.5}\}$ and the theory is reduced up to:

$$\begin{aligned}\Sigma = \{\{c/_{0.9}\} &\Rightarrow \{d/_{0.7}\},\\ \{a/_{0.9}\} &\Rightarrow \{c/_{0.6}, e/_{0.7}\},\\ \{e/_{0.7}\} &\Rightarrow \{c/_{0.8}, d/_{0.6}, e/_{0.6}\},\\ \{e/_{0.8}\} &\Rightarrow \{a/_{0.8}, b/_{0.9}\}\}\end{aligned}$$

After the third iteration, $A = \{a/_{0.5}, b/_{0.5}, c/_{0.6}, d/_{0.5}, e/_{0.5},\}$ and the theory is reduced up to:

$$\begin{aligned}\Sigma = \{\{c/_{0.9}\} &\Rightarrow \{d/_{0.7}\},\\ \{a/_{0.9}\} &\Rightarrow \{e/_{0.7}\},\\ \{e/_{0.7}\} &\Rightarrow \{c/_{0.8}, d/_{0.6}, e/_{0.6}\},\\ \{e/_{0.8}\} &\Rightarrow \{a/_{0.8}, b/_{0.9}\}\}\end{aligned}$$

And in the next iteration the algorithm reaches the fix point and finishes returning $A = \{a/_{0.5}, b/_{0.5}, c/_{0.6}, d/_{0.5}, e/_{0.5},\}$ and the following reduced theory:

$$\begin{aligned}\Sigma = \{\{c/_{0.9}\} &\Rightarrow \{d/_{0.7}\},\\ \{a/_{0.9}\} &\Rightarrow \{e/_{0.7}\},\\ \{e/_{0.7}\} &\Rightarrow \{c/_{0.8}, d/_{0.6}, e/_{0.6}\},\\ \{e/_{0.8}\} &\Rightarrow \{a/_{0.8}, b/_{0.9}\}\}\end{aligned}$$

VI. Conclusions

Parametrised Simplification Logic is a general framework. The main aim of this work is to develop such a general framework not only in the specification view but also in the executable view. Thus, different deduction methods are particular instances of the method presented in Algorithm 1. The way to be traversed is to describe each target logic as an instance of the Parameterized Simplification Logic, providing its complete dual residuated lattice and the set of parameters. In addition, to ensure that the method can be appropriately applied, we have to check that the conditions included in this paper hold. Now, we are going to illustrate this course of action with three different examples.

The classical simplification logic [17] is the instance of Parameterized Simplification Logic where the complete dual residuated lattice is $\mathbb{U} = \langle 2^U, \subseteq, \cup, \smallsetminus, \emptyset, U\rangle$ and the $\mathbb{U}$-parameterization is $S = \{\langle \mathtt{I}, \mathtt{I}\rangle\}$. This parameterization is tractable: it is finite, compact and $\boldsymbol{c}_\emptyset = \mathtt{I}$. Thus, the deduction method can be applied for this instance.

The Fuzzy Attribute Simplification Logic [18] is the instance of Parameterized Simplification Logic where, given the finite residuated lattice $\langle L, \leq, \otimes, \to\rangle$, its complete dual residuated lattice is $\mathbb{L} = \langle L, \leq, \oplus, \ominus\rangle$ being $\oplus = \vee$ and $\ominus$ the adjoint operation satisfying (1). This substraction operation is uniquely expressed as

$$a \ominus b = \bigwedge\{c \in L \,|\, a \leq b \vee c\}.$$

In particular, if $\mathbb{L}$ is linearly ordered, we have

$$a \ominus b = \begin{cases} a, & \text{if } a > b,\\ 0, & \text{otherwise.}\end{cases}$$

We work with the lattice of $\mathbb{L}$-fuzzy sets $\mathbb{L}^Y$ as described in Section V. Furthermore, given a hedge *, the $\mathbb{L}$-parameterization S consists of all $\langle \boldsymbol{f}_{c^*\otimes}, \boldsymbol{g}_{c^*\to}\rangle$ where $(\boldsymbol{f}_{c^*\otimes}(A))(y) = c^* \otimes A(y)$ and $(\boldsymbol{g}_{c^*\to}(A))(y) = c^* \to A(y)$ for any $A \in L^Y$, $c \in L$, and $y \in Y$.

This parameterization is tractable: it is finite, compact and $\boldsymbol{c}_\emptyset = \mathtt{I}$. This is a direct consequence of the following property, fulfilling in all residuated lattice: for all $a, b, c \in L$

$$c \otimes (a \vee b) = (c \otimes a) \vee (c \otimes b) \leq a \vee (c \otimes b)$$

Having described a pair of parameterised instances, we conclude this illustrative motivation of our work by providing a logic belonging to the simplification family, where the closure method is not applicable.

The Logic of Temporal Attribute Implciation [19] is the instance of Parameterized Simplification Logic where, given the complete dual residuated lattice is $\mathbb{L} = \langle 2^{Y\times\mathbb{Z}}, \subseteq, \cup, \smallsetminus\rangle$ being $\mathbb{Z}$ the discrete flow of time, $\cap$ and $\smallsetminus$ are the standard set operators. Given $A \in 2^{Y\times\mathbb{Z}}$, a pair $\langle y, i\rangle \in A$ is interpreted as "the attribute/feature y will be present after i days or was present before $-i$ days ago, depending on whether i is positive or negative.

Its $\mathbb{L}$-parameterization is $S = \{\langle \boldsymbol{f}_i, \boldsymbol{g}_i\rangle \mid i \in \mathbb{Z}\}$ where $\boldsymbol{f}_i(A) = \{\langle y, i+j\rangle \mid \langle y, j\rangle \in A\}$ and $\boldsymbol{g}_i(A) = \{\langle y, j-i\rangle \mid \langle y, j\rangle \in A\}$ for all $A \in 2^{Y\times\mathbb{Z}}$.

This parameterization is not finite, and therefore it is non-tractable. In addition, the first condition of Theorem 4 does not hold, since, for instance, $\boldsymbol{f}_3(\{\langle y_1, 1\rangle\} \cup \{\langle y_1, 2\rangle\}) = \{\langle y_1, 4\rangle, \langle y_1, 5\rangle\}$ and, since $\boldsymbol{c}_\emptyset = \mathtt{I}$, we have that $\boldsymbol{c}_\emptyset(\{\langle y_1, 1\rangle\} \cup \boldsymbol{f}_3(\{\langle y_1, 2\rangle\}) = \{\langle y_1, 1\rangle, \langle y_1, 5\rangle\}$. The reason behind this situation is that Deduction Theorem, which is the key point of the method, cannot be used since given $\langle y, i\rangle$, we have that $\emptyset \Rightarrow \langle y, i\rangle \vdash \langle y, j\rangle$ for all $j \in \mathbb{Z}$.

Acknowledgment

Supported by Grant TIN2017-89023-P of Spain's Science and Innovation Ministry, co-financed by the European Regional Development Fund (ERDF).

References

[1] P. Cordero, M. Enciso, A. Mora, and V. Vychodil, "Parameterized simplification logic I: reasoning with implications and classes of closure operators," *Int. J. Gen. Syst.*, vol. 49, no. 7, pp. 724–746, 2020.
[2] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 26, pp. 64–69, 1983.
[3] D. Maier, *Theory of Relational Databases*. Rockville, MD, USA: Computer Science Pr, 1983.
[4] C. Beeri and P. A. Bernstein, "Computational problems related to the design of normal form relational schemas," *ACM Trans. Database Syst.*, vol. 4, pp. 30–59, March 1979.
[5] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997.
[6] B. Ganter, "Two basic algorithms in concept analysis," in *Proceedings of the 8th International Conference on Formal Concept Analysis*, ser. ICFCA'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 312–340.
[7] J.-L. Guigues and V. Duquenne, "Familles minimales d'implications informatives resultant d'un tableau de données binaires," *Math. Sci. Humaines*, vol. 95, pp. 5–18, 1986.
[8] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, Jun. 1993.

[9] M. J. Zaki, “Mining non-redundant association rules,” *Data Mining and Knowledge Discovery*, vol. 9, pp. 223–248, 2004.
[10] R. Belohlavek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*. New York: Oxford University Press, 2017.
[11] J. M. Mendel, *Uncertin Rules-Based Fuzzy Systems: Introduction and New Directions*, 2nd ed. Springer International Publishing, 2017.
[12] L. Ježková, P. Cordero, and M. Enciso, “Fuzzy functional dependencies: A comparative survey,” *Fuzzy Sets and Systems*, vol. 317, pp. 88–120, 2017.
[13] V. Vychodil, “Parameterizing the semantics of fuzzy attribute implications by systems of isotone Galois connections,” *IEEE Trans. on Fuzzy Systems*, vol. 24, pp. 645–660, 2016.
[14] G. Birkhoff, *Lattice theory*, 1st ed. Providence: American Mathematical Society, 1940.
[15] V. Vychodil, “Closure structures parameterized by systems of isotone Galois connections,” *International Journal of Approximate Reasoning*, vol. 91, pp. 1–21, 2017.
[16] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, 2nd ed. Cambridge University Press, 2002.
[17] A. Mora, P. Cordero, M. Enciso, I. Fortes, and G. Aguilera, “Closure via functional dependence simplification,” *International Journal of Computer Mathematics*, vol. 89, no. 4, pp. 510–526, 2012.
[18] R. Belohlavek, P. Cordero, M. Enciso, A. Mora, and V. Vychodil, “Automated prover for attribute dependencies in data with grades,” *International Journal of Approximate Reasoning*, vol. 70, pp. 51–67, 2016.
[19] J. Triska and V. Vychodil, “Logic of temporal attribute implications,” *Annals of Mathematics and Artificial Intelligence*, vol. 79, no. 4, pp. 307–335, 2017.

Pablo Cordero Pablo Cordero is a Full Professor of the Applied Mathematics Department at the University of Málaga (Spain). He received the MSc in Mathematics in 1992 from the Universidad Complutense de Madrid and the PhD in Computer Science in 1999 from the University of Málaga. His research is devoted to Mathematics applied to Computer Sciences. Specifically, his research area is the information and knowledge treatment via logic and its algebraic foundations: Logic and Automated Reasoning, Fuzzy Logic, Formal Methods in Databases, Formal Concept Analysis, Hyperstructures and Multilattice Theory, Galois Connections and Applications, etc. He is co-author of more than forty research articles in a variety of scientific journals (e.g. Fuzzy Set Syst., Inf. Sci., Appl. Math. Comput., Ann. Math. Artif. Intell., Logic Jnl IGPL, etc.) and seventy contributions to conferences and workshops (e.g. IJCAI, IPMU, CLA, ICFCA, ICSOFT, IWANN, etc.). He is a member of the editorial board of the International Journal of Algebraic Hyperstructures and Applications.

Manuel Enciso Manuel Enciso is an Associate Professor in the Languages and Computer Sciences Department at the University of Málaga. In this university received a PhD degree in Computer Science. His research is focused on Intelligent Information Systems, logic, fuzzy logic, methods of automated reasoning, Formal Concept Analysis, and Software Meta-models. He is a member of the research group SICUMA and has published his research articles in international journals and conferences.

Angel Mora Angel Mora is an Associate Professor at the University of Málaga and received a PhD degree in Computer Science at the University of Málaga. His research interests are in the areas of applied mathematica, logic, modal logic, fuzzy logic, automated reasoning, Lattice theory and generalisations, Formal methods in databases, Formal Concept Analysis. He has published research articles in reputed international journals of applied mathematical and engineering sciences. He is referee of mathematical and computer science journals.

Vilem Vychodil Vilem Vychodil, Czech computer professor, researcher. Achievements include research in fuzzy logic, model theory, relational data analysis, logical foundations of knowledge engineering. Member of Institute of Electrical and Electronics Engineers, Association Computing Machinery.