Fixpoints of fuzzy closure operators via ordinary algorithms

Radim Belohlavek and Jan Konecny
Palacký University, Olomouc, Czech Republic,
e-mail: radim.belohlavek@acm.org; jan.konecny@upol.cz

***Abstract*—We present a way to compute the set of fixpoints of a given fuzzy closure operator via algorithms for computing sets of fixpoints of ordinary closure operators. We assume that the fuzzy closure operator is given by a set of fuzzy sets generating this operator. The proposed way is based on certain reduction theorems which we provide and which relate fuzzy and ordinary closure operators and the sets of their fixpoints. We also present explicit description of selected algorithms which result using the presented approach.**

I. Introduction

Fuzzy closure operators and fuzzy closure systems are among the fundamental mathematical concepts that have been explored since the early days of fuzzy logic. Fuzzy topology and later on Pavelka-style fuzzy logic were the first areas within which these concepts have been investigated. Since the late 1990s, a more thorough investigations of fuzzy closure operators started, partly due to their close connections to formal concept analysis of data with fuzzy attributes. In particular, a number of investigations examined the problem of computing sets and lattices of fixpoints of fuzzy closure operators. The reason is that these lattices are—up to an isomorphism—just the so-called fuzzy concept lattices, i.e. the main structures in formal concept analysis. Since the computational aspects are most developed in the context of formal concept analysis and since this area is practically important, we regard this area as the primary source of motivation for our study. For details we refer to [4].

One possibility in computing the sets of fixpoints of fuzzy closure operators and computing fuzzy concept lattices is to generalize the existing algorithms for the ordinary closure operators and concept lattices. An overview of some approaches of this kind is provided in [6]. Note that such generalizations are not obvious because, as we shall discuss in more detail below, in the fuzzy setting there are two generating operations instead of a single operation, namely the minimum, on which ordinary closure operators are based, and an operation based on the residuum. Another possibility, which is our main concern in this paper, derives from a question of general importance in fuzzy logic, namely the question of a relationship between ordinary notions and their fuzzified counterparts. In particular, we are interested in whether and to what extent the notions of a fuzzy closure operator and related ones are reducible to their ordinary counterparts in that the results and algorithms available for ordinary closure operators be applicable to fuzzy closure operators. Some such relationships were obtained in [3], [20]; see also [4].

Our paper is organized as follows. In Section II, we provide the notions needed in our paper. The contributions of our paper are the subject of the the next sections and are the following. First, we present in Section III several theorems related to the above-mentioned problem of reduction. Second, we apply in Section IV the results from Section III to obtain as examples explicit descriptions of selected algorithms for computing the sets of fixpoints of fuzzy closure operators. In Section V, we present conclusions and discuss some further problems to be explored.

II. Preliminaries

In accordance with modern fuzzy logic, we use residuated lattices as structures of truth degrees and require completeness for reasons mentioned later. Recall that a complete residuated lattice [12] is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice with 0 and 1 being the least and greatest element of L, respectively; $\langle L, \otimes, 1\rangle$ is a commutative monoid (i.e. $\otimes$ is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$); $\otimes$ and $\rightarrow$ satisfy so-called adjointness property:

$$a \otimes b \leq c \quad \text{iff} \quad a \leq b \rightarrow c \tag{1}$$

for each $a, b, c \in L$. The elements a of L are called truth degrees. The operations $\otimes$ and $\rightarrow$ are (truth functions of) "fuzzy conjunction" and "fuzzy implication". A common choice of $\mathbf{L}$ is a structure with $L = [0, 1]$ (unit interval), $\wedge$ and $\vee$ being minimum and maximum, $\otimes$ being a left-continuous t-norm with the corresponding residuum $\rightarrow$. Three most important pairs of adjoint operations on the unit interval are:

$$\text{Łukasiewicz:} \quad \begin{aligned} a \otimes b &= \max(a + b - 1, 0), \\ a \rightarrow b &= \min(1 - a + b, 1), \end{aligned} \tag{2}$$

$$\text{Gödel:} \quad \begin{aligned} a \otimes b &= \min(a, b), \\ a \rightarrow b &= \begin{cases} 1 & \text{if } a \leq b, \\ b & \text{otherwise,} \end{cases} \end{aligned} \tag{3}$$

$$\text{Goguen (product):} \quad \begin{aligned} a \otimes b &= a \cdot b, \\ a \rightarrow b &= \begin{cases} 1 & \text{if } a \leq b, \\ \frac{b}{a} & \text{otherwise.} \end{cases} \end{aligned} \tag{4}$$

Another common choice is a finite linearly ordered $\mathbf{L}$. For instance, one can put $L = \{a_0 = 0, a_1, \ldots, a_n = 1\} \subseteq [0, 1]$

($a_0 < \cdots < a_n$) with $\otimes$ given by $a_k \otimes a_l = a_{\max(k+l-n,0)}$ and the corresponding $\rightarrow$ given by $a_k \rightarrow a_l = a_{\min(n-k+l,n)}$. Such an $\mathbf{L}$ is called a finite Łukasiewicz chain. Another possibility is a finite Gödel chain which consists of L and restrictions of the Gödel operations on $[0,1]$ to L.

Having $\mathbf{L}$, we define usual notions: an L-set (fuzzy set) A in universe U is a mapping $A\colon U \rightarrow L$, $A(u)$ being interpreted as "the degree to which u belongs to A." If $U = \{u_1, \ldots, u_n\}$ then A can be denoted by $A = \{{}^{a_1}\!/u_1, \ldots, {}^{a_n}\!/u_n\}$ meaning that $A(u_i)$ equals a_i for each $i = 1, \ldots, n$.

Let L^U denote the collection of all L-sets in U. The operations with L-sets are defined componentwise. For instance, the intersection of L-sets $A, B \in L^U$ is an L-set $A \cap B$ in U such that $(A \wedge B)(u) = A(u) \wedge B(u)$ for each $u \in U$, etc. Binary L-relations (binary fuzzy relations) between U and V can be thought of as L-sets in the universe $U \times V$. That is, a binary L-relation $R \in L^{U \times V}$ between a set U and a set V is a mapping assigning to each $u \in U$ and each $v \in V$ a truth degree $R(u,v) \in L$ (a degree to which u and v are related by R). An L-set $A \in L^U$ is called crisp if $A(u) \in \{0,1\}$ for each $u \in U$. Crisp L-sets can be identified with (characteristic functions of) ordinary sets: A crisp L-set $A \in L^U$ corresponds to the ordinary set $\{u \in U \mid A(u) = 1\}$. Therefore, for a crisp A, we also write $u \in A$ for $A(u) = 1$ and $u \notin A$ for $A(u) = 0$. An L-set $A \in L^U$ is called empty (denoted by $\emptyset$) if $A(u) = 0$ for each $u \in U$; $A \in L^U$ is called full (denoted by U) if $A(u) = 1$ for each $u \in U$.

Given $A, B \in L^U$, we define the subsethood degree

$$S(A,B) = \textstyle\bigwedge_{u \in U}(A(u) \rightarrow B(u)) \tag{5}$$

of A in B generalizing the classical subsethood relation $\subseteq$. Described verbally, $S(A,B)$ represents a degree to which A is a subset of B. In particular, we write $A \subseteq B$ iff $S(A,B) = 1$. As a consequence, $A \subseteq B$ iff $A(u) \leq B(u)$ for each $u \in U$. In the following we use well-known properties of residuated lattices and fuzzy structures which can be found in [4], [12].

Throughout the rest of the paper, $\mathbf{L}$ denotes an arbitrary complete residuated lattice. For our purposes, $\mathbf{L}$ needs to be equipped with operations which model the so-called intensifying linguistic hedges such as "very." In particular, we use the concept of a truth-stressing hedge which is very close to the one used in [12], [13]. By a truth-stressing hedge (shortly, a hedge) on $\mathbf{L}$, we mean a unary mapping * on L satisfying

$$1^* = 1, \tag{6}$$
$$a^* \leq a, \tag{7}$$
$$(a \rightarrow b)^* \leq a^* \rightarrow b^*, \tag{8}$$
$$a^{**} = a^*, \tag{9}$$

for each $a, b \in L$. Hedge * can indeed be seen as a (truth function of) unary logical connective "very", "extremely", etc., see [12], [13]. Note that as a consequence of (6) and (8) we get monotony: if $a \leq b$ then $a^* \leq b^*$.

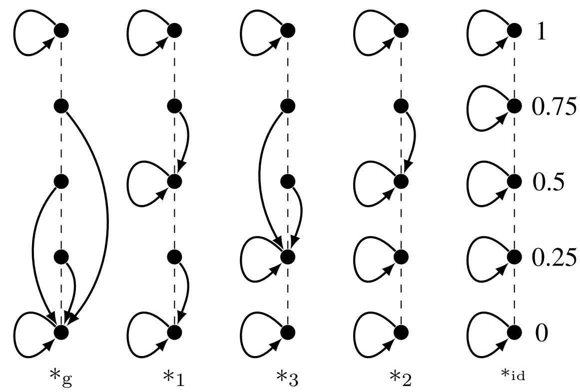

Fig. 1. Truth-stressing hedges on 5-element chain with Łukasiewicz operations $\mathbf{L} = \langle\{0, 0.25, 0.5, 0.75, 1\}, \min, \max, \otimes, \rightarrow, 0, 1\rangle$. The leftmost truth-stressing hedge $*_{\mathrm{g}}$ is the globalization, the rightmost truth-stressing denoted by $*_{\mathrm{id}}$ is the identity.

Two boundary cases of (truth-stressing) hedges are (i) identity, i.e. $a^* = a$ ($a \in L$); (ii) globalization [21]:

$$a^* = \begin{cases} 1 & \text{if } a = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

See Fig. 1 for examples of truth-stressing hedges.

A particularly important special case of a complete residuated lattice with hedge is a two-element Boolean algebra $\langle\{0,1\}, \wedge, \vee, \otimes, \rightarrow, {}^*, 0, 1\rangle$, denoted by $\mathbf{2}$, which is the structure of truth degrees of classical logic. That is, the operations $\wedge, \vee, \otimes, \rightarrow$ of $\mathbf{2}$ are the truth functions (interpretations) of the corresponding logical connectives of the classical logic and $0^* = 0$, $1^* = 1$.

III. Results

For reasons explained below, we are interested in the following concept of a fuzzy closure operator [2], [7]. For a complete residuated lattice $\mathbf{L}$ with a truth-stressing hedge $*$, an $\mathbf{L}^*$-closure operator in a set U is a mapping $C : L^U \rightarrow L^U$ satisfying

$$A \subseteq C(A),$$
$$S(A,B)^* \leq S(C(A), C(B)),$$
$$C(A) = C(C(A)),$$

for every $A, B \in L^U$. Notice in particular the second condition which expresses the requirement that C preserves graded subsethood and which employs the truth-stressing hedge $*$. An $\mathbf{L}^*$-closure system in U, which is the corresponding notion in terms of closure systems, is a system $\mathcal{S} \subseteq L^U$ of fuzzy sets in U satisfying:

$$a^* \rightarrow A \in \mathcal{S} \quad \text{whenever } a \in L \text{ and } A \in \mathcal{S} \text{ (closedness under } a^*\text{-shifts), and}$$

$$\textstyle\bigwedge_{j \in J} A_j \in \mathcal{S} \quad \text{whenever } A_j \in \mathcal{S}\ (j \in J) \text{ (closedness under intersections),}$$

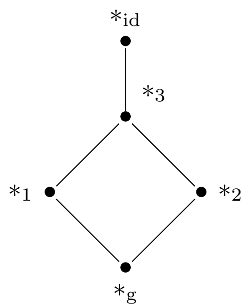

Fig. 2. Hasse diagram of $\leq$ on hedges from Fig. 1.

where $b \to A$ (called also a left $\to$-multiplication of A by b) and $\bigwedge_{j\in J} A_j$ are defined for fuzzy sets $A, A_j \in L^U$, $j \in J$, and $b \in L$ by $(\bigwedge_{j\in J} A_j)(u) = \bigwedge_{j\in J} A_j(u)$ and $(b \to A)(u) = b \to A(u)$. With these notions, one has the usual correspondence, i.e. $\mathbf{L}^*$-closure systems are just the sets of fixpoints of $\mathbf{L}^*$-closure operators.

$\mathbf{L}^*$-closure operators and systems arise in formal concept analysis in the following way. Given a formal fuzzy context $\langle X, Y, I\rangle$, i.e. X, Y, and I are a set of objects, a set of attributes, and a fuzzy relation between these sets, respectively, consider the mappings ${}^\uparrow : L^X \to L^Y$ and ${}^\downarrow : L^Y \to L^X$ defined by

$$\begin{aligned} A^\uparrow(y) &= \textstyle\bigwedge_{x\in X}(A(x) \to I(x,y)),\\ B^\downarrow(x) &= \textstyle\bigwedge_{y\in Y}(B(y)^* \to I(x,y)). \end{aligned} \tag{11}$$

The set

$$\mathcal{B}(X, Y^*, I) = \{\langle A, B\rangle \in L^X \times L^Y \mid A^\uparrow = B,\ B^\downarrow = A\}$$

of all fixpoints of $\langle{}^\uparrow, {}^\downarrow\rangle$ is called a (fuzzy) concept lattice of I. Note that this is an important particular case of so-called concept lattices with hedges; see [5], [10]. Two especially important cases result if $*$ is the identity, in which case we obtain the classical fuzzy concept lattices [5], [20], and the globalization in which case we obtain the so-called one-sided, or crisply generated, fuzzy concept lattices [10], [22], [15].

Some basic observations are as follows. It is easily shown that the set of extents of $\mathcal{B}(X, Y^*, I)$, i.e. the set

$$\begin{aligned} \mathrm{Ext}(X, Y^*, I) \;=\; \{A \in L^X \mid \langle A, B\rangle \in \mathcal{B}(X, Y^*, I) \\ \text{for some } B\}, \end{aligned}$$

forms an $\mathbf{L}^*$-closure system. Moreover, $\mathrm{Ext}(X, Y^*, I)$ is the set of all fixpoints of the $\mathbf{L}^*$-closure operator $C : L^X \to L^X$ defined by

$$C(A) = A^{\uparrow\downarrow}$$

for each $A \in L^X$.

A natural partial order of hedges on $\mathbf{L}$ is defined by $*_1 \leq *_2$ iff $a^{*_1} \leq a^{*_2}$ for every $a \in L$. Clearly, $*_{\rm g} \leq * \leq *_{\rm id}$ for every $*$ ($*_{\rm g}$ is the globalization, $*_{\rm id}$ is the identity). Fig. 2 illustrates $\leq$ on the set of hedges from Fig. 1.

One easily checks that $*_1 \leq *_2$ iff $\mathrm{fix}(*_1) \subseteq \mathrm{fix}(*_2)$ where

$$\mathrm{fix}(*) = \{a \in L \mid a^* = a\}.$$

One may also check that if $*_1 \leq *_2$ then $\mathcal{B}(X, Y^{*_1}, I) \subseteq \mathcal{B}(X, Y^{*_2}, I)$.

As was mentioned above, we assume that the $\mathbf{L}^*$-closure operator C is given by a system $\mathcal{S} \subseteq L^U$ of fuzzy sets in U: C is the operator generated by $\mathcal{S}$ in that the set of all fixpoints of C is just the $\mathbf{L}^*$-closure system generated by $\mathcal{S}$.

To analyze our problem further, we need the following concepts. For any system $\mathcal{S} \subseteq L^U$ of fuzzy sets in U, denote by

- $[\mathcal{S}]^*$ the least $\mathbf{L}^*$-closure system containing $\mathcal{S}$,
- $[\mathcal{S}]_\bigwedge$ the least system containing $\mathcal{S}$ that is closed under $\bigwedge$-intersections,
- $[\mathcal{S}]^*_\to$ the least system containing $\mathcal{S}$ that is closed under a^*-shifts for every $a \in L$.

From the present viewpoint, it is natural to consider a formal context $\langle X, Y, I\rangle$ as represented by the system $\mathcal{S} = \{I_y \in L^X \mid y \in Y\}$ of fuzzy sets I_y (columns of the context) defined by

$$I_y(x) = I(x, y)$$

for each $x \in X$. As shown by the following lemma, the $\mathbf{L}^*$-closure system generated by $\mathcal{S}$ is then just the set of all extents of the fuzzy concept lattice $\mathcal{B}(X, Y^*, I)$ of $\langle X, Y, I\rangle$:

Lemma 1. $[\mathcal{S}]^* = \mathrm{Ext}(X, Y^*, I)$.

Proof. Omitted due to lack of space. □

From now on we assume that $*$ satisfies

$$(a \otimes b)^* = a^* \otimes b^* \tag{12}$$

for every $a, b \in L$, i.e. $*$ is a member of a class of hedges employed in [9].

The following theorem provides an important insight into our problem.

Theorem 1. *For every system $\mathcal{S} = \{I_y \in L^Y \mid y \in Y\}$ we have*

$$[\mathcal{S}]_\bigwedge \;=\; \{\textstyle\bigwedge \mathcal{T} \mid \mathcal{T} \subseteq \mathcal{S}\}, \tag{13}$$

$$[\mathcal{S}]^*_\to \;=\; \{a^* \to I_y \mid a \in L,\ y \in Y\}, \tag{14}$$

$$[\mathcal{S}]^* \;=\; [[\mathcal{S}]^*_\to]_\bigwedge. \tag{15}$$

Proof. (13) is immediate to check.

(14): Since $1^* \to I_y = 1 \to I_y = I_y$, we have $\mathcal{S} \subseteq [\mathcal{S}]^*_\to$. Clearly, every system closed under a^*-shifts contains $[\mathcal{S}]^*_\to$. It is sufficient to show that $[\mathcal{S}]^*_\to$ itself is closed under a^*-shifts. This follows from the fact that due to (12), and the fact that $\alpha \to (\beta \to \gamma) = (\alpha \otimes \beta) \to \gamma$,

$$b^* \to (a^* \to I_y) = ((b^* \otimes a^*) \to I_y) = ((b \otimes a)^* \to I_y) \in [\mathcal{S}]^*_\to.$$

(15): It is easy to check that $[[\mathcal{S}]^*_\to]_\bigwedge$ contains $\mathcal{S}$ and is contained in every system closed under a^*-shifts and $\bigwedge$-intersections, i.e. in every $\mathbf{L}^*$-closure system, that contains $\mathcal{S}$. Therefore, it is sufficient to show that $[[\mathcal{S}]^*_\to]_\bigwedge$ is closed under $\bigwedge$-intersections and a^*-shifts. Closedness under $\bigwedge$-intersections is obvious. For a^*-shifts, let $a \in L$ and consider an arbitrary element of $[[\mathcal{S}]^*_{\rightarrow}]_{\bigwedge}$, i.e. an element of the form $\bigwedge_{y\in Z}(a_y^* \rightarrow I_y)$ for $Z \subseteq Y$. Then

$$\begin{aligned} a^* \rightarrow \textstyle\bigwedge_{y\in Z}(a_y^* \rightarrow I_y) &= \textstyle\bigwedge_{y\in Z}(a^* \rightarrow (a_y^* \rightarrow I_y)) \\ &= \textstyle\bigwedge_{y\in Z}(a^* \otimes a_y^*) \rightarrow I_y) \\ &= \textstyle\bigwedge_{y\in Z}((a \otimes a_y)^*) \rightarrow I_y) \in [[\mathcal{S}]^*_{\rightarrow}]_{\bigwedge}. \end{aligned}$$

□

In particular, (15) is an important decomposition result: To compute $[\mathcal{S}]^*$, i.e. $\mathbf{L}^*$-closure system generated by $\mathcal{S}$, it suffices to compute first the $[\ ]^*_{\rightarrow}$-closure of $\mathcal{S}$ and then apply to it the operator of $[\ ]_{\bigwedge}$-closure. Now, (14) shows that to obtain $[\mathcal{S}]^*_{\rightarrow}$, i.e. the $[\ ]^*_{\rightarrow}$-closure of $\mathcal{S}$, it suffices to add all the a^*-shifts of elements in $\mathcal{S}$. Computing the $[\ ]_{\bigwedge}$-closure is, due to (13), analogous to the problem of computing closure systems in the ordinary case.

The next result we need is:

Theorem 2. *If* $*_1 \leq *_2$ *then*

$$[\mathcal{S}]^{*_2} = [[\mathcal{S}]^{*_2}_{\rightarrow}]^{*_1}. \tag{16}$$

Proof. $[[\mathcal{S}]^{*_2}_{\rightarrow}]^{*_1} \subseteq [\mathcal{S}]^{*_2}$: Due to Theorem 1, every element of $[[\mathcal{S}]^{*_2}_{\rightarrow}]^{*_1}$ is of the form

$$\textstyle\bigwedge_{y\in Z}(a_y^{*_1} \rightarrow (b_y^{*_2} \rightarrow I_y)), \tag{17}$$

for some $Z \subseteq Y$ and $a_y, b_y \in L$. By assumption, $*_1 \leq *_2$ from which we easily obtain $\mathrm{fix}(*_1) \subseteq \mathrm{fix}(*_2)$. Therefore, since $a_y^{*_1} \in \mathrm{fix}(*_1)$, we have $a_y^{*_1} \in \mathrm{fix}(*_2)$, whence $a_y^{*_1} = (a_y^{*_1})^{*_2}$. We thus obtain

$$\begin{aligned} \textstyle\bigwedge_{y\in Z}(a_y^{*_1} \rightarrow (b_y^{*_2} \rightarrow I_y)) &= \textstyle\bigwedge_{y\in Z}((a_y^{*_1} \otimes b_y^{*_2}) \rightarrow I_y)) = \\ &= \textstyle\bigwedge_{y\in Z}(((a_y^{*_1})^{*_2} \otimes b_y^{*_2}) \rightarrow I_y)) = \\ &= \textstyle\bigwedge_{y\in Z}(((a_y^{*_1} \otimes b_y)^{*_2}) \rightarrow I_y)), \end{aligned}$$

which is an element of $[\mathcal{S}]^{*_2}$ on account of Theorem 1.

$[\mathcal{S}]^{*_2} \subseteq [[\mathcal{S}]^{*_2}_{\rightarrow}]^{*_1}$: Putting $a_y = 1$ in (17), which is the form of an arbitrary element in $[[\mathcal{S}]^{*_2}_{\rightarrow}]^{*_1}$, we obtain $\bigwedge_{y\in Z}(b_y^{*_2} \rightarrow I_y)$, which is the form of a general element in $[\mathcal{S}]^{*_2}$, proving the claim. □

Theorem 2 may be interpreted as follows. To obtain the set of all fixpoints of an $\mathbf{L}^{*_2}$-closure operator induced by $\mathcal{S}$, it is enough to have, for some $*_1 \leq *_2$, a method to obtain sets of fixpoints of $\mathbf{L}^{*_1}$-closure operators induced by systems of L-sets. Namely, it is enough to apply this method to the system $[\mathcal{S}]^{*_2}_{\rightarrow}$ of all a^{*_2}-shifts of the L-sets in $\mathcal{S}$.

Theorem 2 is particularly appealing if one puts $*_2 = *$ and $*_1$ is the globalization. Namely, we then get

$$\mathrm{Ext}(X, Y^*, I) = [\mathcal{S}]^* = [[\mathcal{S}]^*_{\rightarrow}]^{*_{\mathrm{g}}} = [\{a^* \rightarrow I_y \mid y \in Y\}]^{*_{\mathrm{g}}}.$$

Importantly, as the next theorem shows, for globalization, $[\ ]^{*_{\mathrm{g}}}$ may be computed using the existing algorithms for computing sets of fixpoints of ordinary closure operators.

Theorem 3. *For a system* $\mathcal{T} = \{J_z \in L^X \mid z \in Z\}$, *the operator* $cl_{\mathcal{T}} : 2^Z \rightarrow 2^Z$ *defined by*

$$cl_{\mathcal{T}}(D) = \{z \in Z \mid J_z(x) \geq \textstyle\bigwedge_{z'\in D} J_{z'}(x) \textit{ for all } x \in X\}$$

is an ordinary closure operator in Z *such that*

$$\mathrm{Ext}(X, Z^{*_{\mathrm{g}}}, J) = [\mathcal{T}]^{*_{\mathrm{g}}} = \{\textstyle\bigwedge_{z\in D} J_z \mid D \in \mathrm{fix}(cl_{\mathcal{T}})\}.$$

Proof. The proof is technically involved and thus omitted due to lack of space. □

We therefore arrive at the final result in this section:

Theorem 4. *For* $\mathcal{S} = \{I_y \in L^X \mid y \in Y\}$, *put* $Z = Y \times \mathrm{fix}(*)$, $\mathcal{T} = \{a \rightarrow I_y \mid a \in \mathrm{fix}(*), y \in Y\}$ *and consider the ordinary closure operator* $cl_{\mathcal{T}}$ *from Theorem 3. Then*

$$\mathrm{Ext}(X, Y^*, I) = [\mathcal{S}]^* = \{\textstyle\bigwedge_{\langle y,a\rangle\in D}(a \rightarrow J_y) \mid D \in \mathrm{fix}(cl_{\mathcal{T}})\}.$$

Proof. Follows directly from the previous results in this section. □

IV. Application: Algorithms computing fuzzy concept lattices

Theorem 4 can be used to formulate a principle allowing a direct generalization of algorithms for computing concept lattices, such as NextClosure [11] or CbO [16] (see [17] for a survey), to algorithms for computing fuzzy concept lattices $\mathcal{B}(X, Y^*, I)$. The principle is presented by a particular "wrap" around a given ordinary algorithm.

The correctness of our wrapping procedure follows directly from the results in the previous section. For convenience, we identify the intents in $\mathrm{Int}(X, Z^{*_g}, J)$ with their 1-cuts. That is, whenever we mention concepts $\langle A, B\rangle \in \mathcal{B}(X, Z^{*_g}, J)$ (which are in fact the crisply-generated fuzzy concepts), we actually mean pairs $\langle A, {}^1B\rangle$ (which are the corresponding one-sided fuzzy concepts) such that $A \in \mathrm{Ext}(X, Z^{*_g}, J)$ and ${}^1B$ is the 1-cut of B. This identification is justified by the existing results on fuzzy concept lattices.

The description of our wrapping procedure follows.

Step 1 (scaling): Transform the input fuzzy context $\langle X, Y, I\rangle$ to the fuzzy context $\langle X, Z, J\rangle$ where $Z = Y \times \mathrm{fix}(*)$ and $J \in L^{X\times Z}$ is given by

$$J(x, \langle y, a\rangle) = a \rightarrow I(x, y).$$

Note that we need not explicitly form a new fuzzy context which would be considerably larger than the original one. Instead, it suffices to obtain the concept-forming operators of J, namely the operators defined by

$$\begin{aligned} A^{\Uparrow} &= \{\langle y, a\rangle \mid A \subseteq a \rightarrow I_y\}, \\ B^{\Downarrow} &= \bigwedge_{\langle y,a\rangle\in B} a \rightarrow I_y. \end{aligned} \tag{18}$$

$\langle \Uparrow, \Downarrow\rangle$ are indeed the operators induced by J provided one identifies the intents B with their 1-cuts ${}^1B$ as described above.

Step 2 (ordinary computation): Compute $\mathcal{B}(X, Z^{*_g}, J)$ using an algorithm for computing fixpoints of an ordinary closure operator (apply this algorithm to the intents of $\mathcal{B}(X, Z^{*_g}, J)$, which are ordinary sets due to our identification of intents with their 1-cuts).

Step 3 (transformation of results): Transform $\mathcal{B}(X, Z^{*_g}, J)$ to the required fuzzy concept lattice $\mathcal{B}(X, Y^*, I)$.

Due to the above results, we just transform each $\langle A, B\rangle \in \mathcal{B}(X, Z^{*_g}, J)$ to its corresponding concept in $\mathcal{B}(X, Y^*, I)$. This transformation is simple since the two concept lattices are isomorphic and since the corresponding fuzzy concepts have the same extents. It is therefore enough to transform each $\langle A, B\rangle$ to $\langle A, A^{\uparrow}\rangle$. Computing $A^{\uparrow}$ is simplified by observing that $A^{\uparrow} = \lceil B \rceil$ where

$$\lceil B \rceil(y) = \bigvee\{a \mid \langle y, a\rangle \in B\}.$$

We now present two applications of our wrapping procedure. The first is a generalization of the CbO algorithm to compute the collection of all fuzzy concepts in $\mathcal{B}(X, Y^*, I)$. The second one is a generalization of Lindig's algorithm to compute a fuzzy concept lattice $\mathcal{B}(X, Y^*, I)$. In the descriptions, we assume that the elements of Z are represented by integers $\{1, \ldots, |Z|\}$.

A. Generalized Close by One

In this section, we demonstrate our approach on the CbO algorithm [16]. It is sufficient for our purpose to use a simplified version of CbO introduced in [14]. The generalizing wrap can be applied the same way to its advancements, like fCbO [19] and InClose family of algorithms [1].

The principal part of the CbO algorithm is the procedure GENERATEFROM (see Algorithm 1). The procedure is called with a fuzzy concept $\langle A, B\rangle$ (initial fuzzy concept) from $\mathcal{B}(X, Z^*, J)$ and an attribute $z \in Z$ (first attribute to be processed) as its arguments. The procedure recursively descends through the space of fuzzy concepts, starting with $\langle A, B\rangle$. When invoked with $\langle A, B\rangle$ and $y \in Y$, GENERATEFROM first outputs $\langle A, \lceil B \rceil\rangle$ (line 1) and then checks its halting condition (lines 2–4). The computation stops if $\langle A, B\rangle$ equals $\langle Z^{\Downarrow}, Z\rangle$ or if $y > n$. Otherwise, it goes through all attributes $j \in Z$ with $j \geq z$ which are not present in the intent B (lines 5 and 6). For each such attribute, a new formal concept

$$\langle C, D\rangle = \langle A \cap \{j\}^{\Downarrow}, (A \cap \{j\}^{\Downarrow})^{\Uparrow}\rangle \tag{19}$$

is formed (lines 7 and 8). Then the algorithm makes the canonicity test on $\langle C, D\rangle$ (line 9) to check whether it should continue with $\langle C, D\rangle$ and $j+1$ by recursively calling GENERATEFROM or not. The canonicity test is based on comparing $B \cap Z_j = D \cap Z_j$ where $Z_j \subseteq Z$ is defined by

$$Z_j = \{z \in Z \mid z < j\}.$$

The loop in lines 5–13 continues with the next value of j.

Note that the only changes in the procedure GENERATEFROM in comparison with its basic form are the following:

- We use the operators (18) in lines 7 an 8 to form concepts (19) of $\mathcal{B}(X, Z^{*_g}, J)$. This represents Step 1 of our procedure.
- In line 1, the fuzzy concept $\langle A, B\rangle$ of $\mathcal{B}(X, Z^{*_g}, J)$ is transformed to the fuzzy concept $\langle A, \lceil B \rceil\rangle$ of $\mathcal{B}(X, Y^*, I)$ before it is output. This represents Step 3 of our procedure.

Algorithm 1: GENERATEFROM($\langle A, B\rangle, z$)

1 **output** $\langle A, \lceil B \rceil\rangle$
2 **if** $B = Z$ **or** $z > n$ **then**
3   **return**
4 **end**
5 **for** j **from** z **upto** n **do**
6   **if** $j \notin B$ **then**
7         $C := A \cap \{j\}^{\Downarrow}$
8         $D := C^{\Uparrow}$
9   **if** $B \cap Z_j = D \cap Z_j$ **then**
10   GENERATEFROM($\langle C, D\rangle, j+1$)
11   **end**
12   **end**
13 **end**
14 **return**

B. Generalized Lindig's algorithm

The second algorithm is the Lindig's algorithm [18]. For convenience, we switch the roles of objects and attributes; thus we generate the concepts in a top-down manner instead of a bottom-up manner.

The core part of the algorithm is the procedure NEIGHBORS (see Algorithm 2) which generates for a fuzzy concept $\langle A, B\rangle$ of $\mathcal{B}(X, Z^{*_g}, J)$ the set N of its lower neighbors. It first initializes N to be empty (line 2). Then it goes trough all attributes not present in B (the loop in lines 3 and 11). A new fuzzy concept $\langle C, D\rangle = \langle (B \cup \{z\})^{\Downarrow}, (B \cup \{z\})^{\Downarrow\Uparrow}\rangle$ is then formed (lines 7 and 8). To avoid duplicates and generation of concepts which are not direct lower neighbors it makes a test using the set *min* of minimal possible generators. The set *min* is initialized with all attributes not present in B (line 1). If the new intent D contains a possible generator different from z (line 6) then z is removed from the set *min* (line 9). Otherwise, the newly formed fuzzy concept $\langle C, D\rangle$ is the lower neighbor of $\langle A, B\rangle$ and is added to N (line 7).

Algorithm 2: NEIGHBORS($\langle A, B\rangle$)

Data: concept $\langle A, B\rangle$
Result: Set N of lower neighbors of $\langle A, B\rangle$
1 *min* $:= Z - B$
2 $N := \emptyset$
3 **for** $z \in Z - B$ **do**
4     $C := (B \cup \{z\})^{\Downarrow}$
5     $D := C^{\Uparrow}$
6   **if** *min* $\cap ((D - B) - \{z\}) = \emptyset$ **then**
7         $N := N \cup \{\langle C, D\rangle\}$
8   **else**
9   *min* $:=$ *min* $- \{z\}$
10   **end**
11 **end**
12 **return** N

The procedure LINDIG (Algorithm 3) generates the largest concept and then, for each concept that is generated for the first time, generates all its lower neighbors. Lindig uses a tree of concepts that allows one to check whether some concept has already been generated. This property is used to compute the concepts of $\mathcal{B}(X, Z^{*_{\mathrm{g}}}, J)$. After all concepts are processed this way, it runs an additional loop (lines 12-13) to transform the intents by application of $\lceil\,\rceil$.

Algorithm 3: LINDIG($\langle X, Y, I\rangle$)

Data: **L**-context $\langle X, Y, I\rangle$
Result: $\mathcal{B}(X, Y^{*}, I)$

1 $c := \langle X, X^{\Uparrow}\rangle$
2 $\mathcal{T}$ is a concept tree consisting of the root node c
3 **while** $c \neq \texttt{null}$ **do**
4   **for** $n \in$ NEIGHBORS(c) **do**
5   **if** $n \notin \mathcal{T}$ **then**
6   insert n to $\mathcal{T}$
7   **end**
8   set n to be a lower neighbor of c
9   **end**
10     $c := \mathrm{Next}(c, \mathcal{T})$
11 **end**
12 **for** $\langle A, B\rangle \in \mathcal{T}$ **do**
13   change intent of $\langle A, B\rangle$ to $\lceil B\rceil$
14 **end**
15 **return** $\mathcal{T}$

The changes in comparison with the original algorithm are the following:

– We use the operators (18) at lines 4 and 5 of procedure NEIGHBORS (Algorithm 2) and line 1 of procedure LINDIG (Algorithm 3) to form fuzzy concepts (19) of $\mathcal{B}(X, Z^{*_g}, J)$ via the implicit simple scaling. This represents Step 1 of our procedure.
– Procedure LINDIG contains an additional loop on lines 12-14 to transform the intents of fuzzy concepts stored in $\mathcal{T}$. This represents Step 3 of our procedure.

V. Conclusions and Further Research

We provided new insights into the theory of fuzzy closure operators motivated by the question of whether and to what extent is it possible to apply results on ordinary closure operators to general fuzzy closure operators. In particular, we presented reduction results that enable one to apply established results and algorithms for computing sets of fixpoints of ordinary closure operators to obtain the corresponding results and algorithms for fuzzy closure operators. The two core insights involved are a decomposition theorem for fuzzy closure operators and a theorem emphasizing an important role of globalization used as a truth-stressing hedge. As an example of applications of our results, we provided a wrapping procedure which enables one to take an algorithm for computing ordinary concept lattices and obtain a corresponding algorithm for computing fuzzy concept lattices.

The topics for future research include: Experimental evaluation of the algorithms obtained using our wrapping procedure; exploration of further applications of our two key results, the decomposition theorem and the theorem emphasizing the role of globalization; further investigation of relationships between fuzzy and ordinary closure structures.

Acknowledgment

Supported by Grant No. GA15-17899S of the Czech Science Foundation.

References

[1] S. Andrews, “A ‘best-of-breed’ approach for designing a fast algorithm for computing fixpoints of Galois connections,” *Information Sciences*, vol. 295, pp. 633–649, 2015.
[2] R. Belohlavek, “Fuzzy closure operators,” *Journal of Mathematical Analysis and Applications*, vol. 262, no. 2, pp. 473–489, Oct. 2001.
[3] ——, “Reduction and a simple proof of characterization of fuzzy concept lattices,” *Fundamenta Informaticae*, vol. 46, no. 4, pp. 277–285, 2001.
[4] ——, *Fuzzy Relational Systems: Foundations and Principles*. Norwell, USA: Kluwer Academic Publishers, 2002.
[5] ——, “Concept lattices and order in fuzzy logic,” *Ann. Pure Appl. Log.*, vol. 128, no. 1-3, pp. 277–298, 2004.
[6] R. Belohlavek, B. De Baets, J. Outrata, and V. Vychodil, “Computing the lattice of all fixpoints of a fuzzy closure operator,” *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 546–557, 2010.
[7] R. Belohlavek, T. Funioková, and V. Vychodil, “Fuzzy closure operators with truth stressers,” *Logic Journal of IGPL*, vol. 13, no. 5, pp. 503–513, 2005.
[8] R. Belohlavek, V. Sklenar, and J. Zacpal, “Crisply generated fuzzy concepts,” in *International Conference on Formal Concept Analysis*. Springer, 2005, pp. 269–284.
[9] R. Belohlavek and V. Vychodil, “Fuzzy Horn logic I,” *Archive for Mathematical Logic*, vol. 45, no. 1, pp. 3–51, 2006.
[10] ——, “Formal concept analysis and linguistic hedges,” *Int. J. General Systems*, vol. 41, no. 5, pp. 503–532, 2012.
[11] B. Ganter and R. Wille, *Formal Concept Analysis – Mathematical Foundations*. Springer, 1999.
[12] P. Hájek, *Metamathematics of Fuzzy Logic (Trends in Logic)*. Springer, November 2001.
[13] P. Hájek, “On very true,” *Fuzzy Sets and Systems*, vol. 124, no. 3, pp. 329–333, 2001.
[14] P. Krajca, J. Outrata, and V. Vychodil, “Parallel algorithm for computing fixpoints of Galois connections,” *Annals of Mathematics and Artificial Intelligence*, vol. 59, no. 2, pp. 257–272, 2010.
[15] S. Krajci, “A generalized concept lattice,” *Logic Journal of the IGPL*, vol. 13, no. 5, pp. 543–550, 2005.
[16] S. O. Kuznetsov, “A fast algorithm for computing all intersections of objects from an arbitrary semilattice,” *Nauchno-Tekhnicheskaya Informatsiya Seriya 2-Informatsionnye Protsessy i Sistemy*, no. 1, pp. 17–20, 1993.
[17] S. O. Kuznetsov and S. Obiedkov, “Comparing performance of algorithms for generating concept lattices,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 14, pp. 189–216, 2002.
[18] C. Lindig, “Fast concept analysis,” *Working with Conceptual Structures-Contributions to ICCS*, vol. 2000, pp. 152–161, 2000.
[19] J. Outrata and V. Vychodil, “Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data,” *Information Sciences*, vol. 185, no. 1, pp. 114–127, 2012.
[20] S. Pollandt, *Fuzzy Begriffe: Formale Begriffsanalyse von unscharfen Daten*. Berlin–Heidelberg: Springer–Verlag, 1997.
[21] G. Takeuti and S. Titani, “Globalization of intui tionistic set theory,” *Annals of Pure and Applied Logic*, vol. 33, pp. 195–211, 1987.
[22] S. B. Yahia and A. Jaoua, *Discovering knowledge from fuzzy concept lattice*. Heidelberg, Germany: Physica-Verlag GmbH, 2001, pp. 167–190.