Jan Outrata

Factorizing Fuzzy Concept Lattices by Similarity

∼ Dissertation Thesis ∼

Ghent & Olomouc, 2006

Address of the author

Jan Outrata
Department of Computer Science
Faculty of Science, Palacký University
Tomkova 40
CZ-779 00 Olomouc
Czech Republic
jan.outrata@upol.cz

Keywords: object-attribute data, clustering, formal concept analysis, concept lattice, fuzzy attributes, similarity, factorization, thresholds

Shared responsibility statement

Some parts of the thesis are outcomes of the joint scientific work of Radim Bělohlávek (radim.belohlavek@upol.cz), Jiří Dvořák (jiri.dvorak@upol.cz) and Jan Outrata (jan.outrata@upol.cz) (section 2.2) and Radim Bělohlávek (radim.belohlavek@upol.cz), Jan Outrata (jan.outrata@upol.cz) and Vilém Vychodil (vilem.vychodil@upol.cz) (sections 2.3 and 3.2). All authors have the even share in the results and findings contained in the respective parts.

Věnuji svým rodičům

Preface

The goal of the work presented in the thesis was development of new, or improvement of existing, algebraic methods of data mining (DM), namely clustering techniques. Data mining is an area of data analysis the subject of which is to unfold, reveal or discover (blankly "dig out" or "mine") the relatively smaller amount of unknown, essential information or knowledge and relationships hidden in (usually) large amount of data. Clustering techniques as methods of DM does so by grouping (somehow) similar records in data and forming so-called clusters together with relationships between them. Data mining is worldwide actual topic in research of analyzing data and artificial intelligence, since present and even future information systems are storing enormously large and for humans ungraspable amount of information in the form of data.

The work focuses on relatively new method of data mining called Formal Concept Analysis (FCA, [27]) and utilizes the (algebraic) clustering technique of factorization to reduce the amount of structured information on the output of FCA. Formal concept analysis is an algebraic method of data mining, particularly exploratory data analysis, which aims at extracting a hierarchical structure (so-called concept lattice) of clusters (so-called formal concepts) and a collection of implication rules from object-attribute data tables (tabular data) describing the relationship between the collection of objects and the collection of attributes. In the work only the hierarchical structure of clusters is considered. In basic setting, the attributes are binary presence/absence attributes and more general attributes are suitably transformed with respect of the meaning of attributes. However, the transformation almost always means losing certain nature of data. The work deals with graded (fuzzy) attributes, like big or cheap, which apply to objects to intermediate degrees, not necessarily false or true only. The transformation of such attributes would lead to loss of the uncertainty or vagueness expressed in relationships between objects and attributes. For dealing with data tables containing fuzzy attributes, various extensions of FCA have been proposed, e.g. Burusco & Fuentes-Gonzales [20], Pollandt [37], Bělohlávek et al. [2], Krajči [34], BenYahia et al. [43]. Fuzzy FCA extends the original ("classical", crisp) FCA in working with many-valued fuzzy attributes

directly.
FCA has been applied in various fields of research, for instance in software engineering (information for software redesign), regulation systems and civil engineering (system for checking dependencies in regulations), text classification and classification and systematizing in general, psychology (development, applying and classification of concepts by children), physiology; see [27] and [22] for references and a survey of applications.
One of the hottest problems in application of FCA is a possibly large number of clusters/concepts extracted from data. A direct user comprehension and interpretation of the hierarchical structure of formal concepts may be difficult. There have been several methods proposed to help to reduce or manage the size of the structure, for instance decomposition [27], applying some classical (heuristic) or algebraic (factorization) clustering technique [36], dealing with a relevant local part of the structure of formal concepts [21], selection of relevant attributes [23] or using additional information to data table [16].
Interesting two approaches to reduce the size of a concept lattice, further extending formal concept analysis (FCA) of data with fuzzy attributes, were recently proposed in the literature. Namely, the approach via hedges [17] and the approach via thresholds [26]. Both of the approaches present parameterized ways to reduce the size of a concept lattice of data with fuzzy attributes. In the work, we show basic relationships between the two approaches. Furthermore, we show that the approaches can be combined in a natural way, i.e. we present an approach in which one deals with both thresholds and hedges. An important role in this analysis is played by so-called shifts of fuzzy attributes which appeared earlier in the study of factorization of fuzzy concept lattices.
Factorization, is one of the methods trying to apply some sort of clustering to make the structure of formal concepts smaller. Instead of viewing the whole structure, it provides a granular view through a factor structure, which can be considered a suitable granularized version of the original structure. Its elements are collections of pairwise similar original concepts and the factor structure is smaller than the original one. The main purpose is thus to have a smaller lattice which can be seen as a reasonable approximation of the original, possibly large, fuzzy concept lattice. In order to compute the factor structure (directly by definition), we have to compute the whole structure and then compute all the collections of similar concepts. In [3], a parametrized method of factorization for data with fuzzy attributes was presented. The similarity relation is induced by a threshold (parameter of factorization, specified by a user) and computed from input data. Then, the size of factor lattice depends on the threshold. The most important part of the work is the presentation of an easy and fast way to compute the (parametrized) factor concept lattice directly from input data and a user-specified threshold. We also explore the use of factorization in fuzzy concept lattices with hedges. The interesting aspect is discussed: the

question of relationships between adjusting input data, modifying the formation of concepts and factorization of the structure of original concepts, since each of these approaches leads to the same result – factor lattice. Beside the theoretical insight, the indivisible and indispensable part constitute the extensive experiments on small and middle-sized data tables from different areas of human activity (demography, sociology).
The individual results contained in the thesis were presented at selected international conferences and forums on FCA and data mining (ICCS, ICFCA, IEEE, CLA, RASC) and published in significant and prestigious journals in the area of data analysis and artificial intelligence research (JCSS, LNAI).
The developed data mining methods involving factorization in FCA are supposed to be included as an important part of a slightly larger software, which will, in broader range, implement more various methods regarding formal concept analysis. The software should enable immediate usage and utilization of FCA in practice.

This thesis is a summary of the joint research of me and my colleagues at Department of Computer Science, Palacký University. First of all, my biggest thanks go to my supervisor, prof. Radim Bělohlávek, who had friendly introduced, has guided and still guides me through the fancy world of science and university environment. Without his great help and leadership my research would be hardly possible. Thank you for all your endless help and support![1]
At the same time I would like to thank to me colleague and friend Vilém Vychodil for the valuable help and assistance, peculiar to himself. Great part of my thanks clearly belong to all the people, who helped me to finish this thesis and without whose support and tolerance it would hardly be written up. To my parents and entire broad family for indispensable home ground providing so important family peace and for their mental support. The work is dedicated to my parents. I would like to thank my friends and fellows for their endless patience and tolerance when I could not devote my time to them and our common business as they deserve.

First pieces of the thesis began to arise nearly two and a half month ago in Ghent, Belgium (thanks to prof. Bernard De Baets from Ghent University) and now after quite hard work it is finished. But behind these few months there are three years of scientific work and I invite you to see what was within that time done!

Jan Outrata
Olomouc, October 2006

[1] And a bit of patience, I remember our first plans, back at Luxembourg, of finishing the work a year earlier, but how many new results since then are included!

Contents

Chapter 1

Problem setting

1.1 Introduction and motivation

Finding interesting and well-interpretable groups in data is a challenging goal which concerns the mankind for several centuries. *Formal concept analysis (FCA)* is a method of exploratory data analysis which aims at extracting a hierarchical structure of clusters from tabular data describing objects and their attributes [27]. The history of FCA goes back to Wille's paper [42], theoretical foundations are gathered together in [27], algorithms and and a survey of applications can be found in [22]. There appeared many interesting applications of FCA as a data analysis method, see e.g. [1, 24, 38, 39], as well as a method for data preprocessing for reduction of the search space, see e.g. [41, 45].

Clusters in FCA are particular pairs $\langle A, B\rangle$ consisting of a collection A of objects and a collection B of attributes which are maximal with respect to the property that each object from A has every attribute from B. Since this approach corresponds to Port-Royal idea of a concept consisting of its extent (objects covered by the concept) and its intent (attributes covered by the concept), clusters $\langle A, B\rangle$ are called formal concepts. Formal concepts can be partially ordered by a subconcept-superconcept hierarchy (narrower clusters are under larger ones). The resulting partially ordered set of clusters forms a complete lattice (so-called concept lattice) and can be visualized by a labelled Hasse diagram from which one can see various relationships valid in the data. Alternatively, formal concepts can be thought of as maximal rectangles contained in the object-attribute data table. In basic setting, the input data table contains bivalent attributes, i.e. each table entry contains either 0 or 1. More general attributes are handled by a so-called conceptual scaling, i.e. a suitable transformation of a general data table into a 0/1-data table which respects the meaning of attributes, see [27] for details. FCA was also extended to data tables with fuzzy attributes (i.e. graded attributes, like big or cheap, which apply to objects to intermediate degrees, not necessarily 0 or 1), where the table entries contain degrees to which a particular attribute applies to a particular object. Then, the components A and B of a formal concept $\langle A, B\rangle$ are fuzzy sets rather than bivalent sets, corresponding well to the fact that objects and attributes are covered by concepts to various degrees. There were proposed various extensions of original („classical") FCA, among the most known are approaches by Burusco & Fuentes-Gonzales [20], Pollandt [37], Bělohlávek et al. [2], Krajči [34], BenYahia et al. [43], . . . , see e.g. [19] for an overview. In this work we will use and utilize approach independently proposed by Bělohlávek and Pollandt, which we find the most appealing.

An important problem in applications of formal concept analysis is a possibly large number of clusters extracted from data. Factorization is one of the methods being used to cope with the number of clusters. In [3], a method of parameterized factorization of concept lattices computed from data tables with fuzzy attributes is presented. A user supplies a similarity threshold a (parameter) and the method outputs, instead of the whole concept lattice which might be large, its factor lattice. The elements of the factor lattice are maximal blocks of clusters from the whole concept lattice which are pairwise similar to degree at least a (where the similarity relation between clusters is computed from them alone). The factor lattice is smaller than the original concept lattice and its size depends on the similarity threshold. The elements of the factor lattice are collections of clusters which are pairwise similar to degree at least a. For a user, the factor lattice provides a coarser version of the whole concept lattice—the less the similarity threshold a, the coarser. The factor lattice thus represents a granularized view on the original concept lattice. We present two algorithms for computing a factor lattice of a concept lattice from the data and a user-specified similarity threshold a ([3, 10, 11]). The main purpose is that the factor lattice is computed directly from data, i.e. without the need to compute the whole concept lattice first. Hence the method the algorithms implements are called fast and direct factorization of fuzzy concept lattices of data with fuzzy attributes.

Recently, there have been proposed additional parametrized approaches to reduce the number of clusters extracted from data with fuzzy attributes. It seems that parameterized approaches are of interest, the parameters control the number of the extracted formal concepts. First of the approaches consists in introducing two additional parameters into FCA of data with fuzzy attributes, see e.g. [19, 12, 14, 17]. These parameters, called hedges, are particular unary functions ^{*X} and ^{*Y} on the scale of truth degrees. The hedges are used to modify the extent and intent forming operators associated to formal context (input data table). Then, instead of original concept lattice, one considers so-called concept lattice with hedges, which is defined to be the set of fixed points of the modified operators. The basic idea is that stronger hedges lead to smaller concept lattice with hedges. An interesting point here is that the approach via hedges subsumes some of the earlier approaches to

FCA of data with fuzzy attributes. First, if both hedges are identities, one obtains the original approach by Pollandt and Bělohlávek [2, 37]. Second, if only one of the hedges is identity and the other one is so-called globalization (see later), the resulting concept lattice with hedges is in fact the so-called one-sided fuzzy concept lattice considered independently in [15, 43, 34]. In this work, we extend the results of fast factorization by similarity to the case of fuzzy concept lattices with hedges.

The second of the additional parametrized approaches exploits the idea of thresholds in FCA of data with fuzzy attributes, proposed in [25] and further improved in [26]. The idea is basically the following. Instead of considering the collection of all attributes shared by a collection of objects, it is intuitively appealing to pick a threshold δ and to consider a set of all attributes shared to a degree greater than or equal to δ. This simple idea also leads to parametrized reduction of the size of fuzzy concept lattice, with δ being the controlling parameter. With $\delta = 1$, this approach is equivalent to approaches proposed independently in [34, 43] mentioned above. In fact, we will show, that the approach using hedges subsumes the approach using thresholds.

Of course, there have been several other methods (either parametrized or not) proposed to help to reduce/manage the size of a concept lattice. A variety of methods is presented in [27]. Except of the above-mentioned factorization, there are several methods of decomposition described in [27]. Decomposition methods aim at finding natural relationships between parts of the concept lattice and corresponding substructures of the input data table. For instance, substitution decomposition is a method based on folding/unfolding the concept lattice which can be visualized by a nested diagram. Other methods of decomposition include subdirect decomposition, atlas decomposition, and tensorial decomposition (we omit details and refer to [27]). A method applied right on the input side of FCA based on selection (by user) of relevant attributes and dealing with only the corresponding part of a concept lattice is presented in [23]. In [21], the authors face the problem of a large concept lattice by computing and visualizing only its relevant local part which is used for browsing based on a user query. In [16], the authors propose to use additional information which is often supplied with the data table. Then, one can extract only those formal concepts which are in a natural way compatible with the additional information. Finally, an interesting approach of clustering of objects and attributes of the input data table is presented in [36]. The clustering is done by removing similar objects and attributes and has effect of clustering of formal concepts, but, however, there are some glitches and unanswered questions.

In this work we take a look at two parametrized approaches to reduce the size of a concept lattice mentioned above. First, the chapter 2 is devoted to the parameterized factorization by similarity. The theory is supported by extensive experiments. Second, the approach exploiting the idea of thresholds is examined and compared to the factorization approach in chapter 3, which is therefore ment to be read after chapter 2. But before delving into the theoretical treatise, we should reiterate the basic notions of fuzzy sets and fuzzy logic and foundations of FCA of data with fuzzy attributes, at least for reasons of unifying the terminology. This material is the subject of the following section 1.2, as well as an introduction to factorization of fuzzy concept lattices by similarity (including illustrative example).

1.2 Preliminaries

1.2.1 Fuzzy sets and fuzzy logic

We first recall the necessary notions from fuzzy sets and fuzzy logic. Only the notions used further in the text will be recalled, for further details we refer to [7, 30]. From (intuitive) theory of sets we know that an element either belongs to a set or not. The concept of a fuzzy set generalizes that of an (ordinary) set in that an element may belong to a fuzzy set to an intermediate degree, not only 0 (does not belong) or 1 (does belong). This enables us to model vaguely (nonsharply) delineated collections: for instance, the collection corresponding to linguistic label "bald man" can be modelled by a fuzzy set to which a man having no hair at all belongs to degree 1 (i.e. is bald), a man with a bit of remaining hair at the back and around ears belongs to degree 0.8 (is almost bald), a man with a small circle area without hair on top of head belongs to degree 0.3 (is almost not bald), whilst a man with a fleece way long down to his waist certainly belongs to degree 0 (is not bald at all). Formally, a *fuzzy set* A in a universe U [33] is a mapping assigning to each $u \in U$ a truth degree $A(u) \in L$ where L is some partially ordered set of truth degrees containing at least 0 (full false) and 1 (full true), see below. Usually, L is the unit interval $[0, 1]$ of real numbers or some of its subsets. $A(U)$ is then interpreted as a degree to which u belongs to A.
To be able to work with fuzzy sets (similarly as with ordinary sets), the scale L of truth degrees needs to be equipped with suitable operations generalizing logical connectives of classical (bivalent, 0/1) logic. Particularly, we will need fuzzy conjunction $\otimes$ and fuzzy implication $\rightarrow$. By natural requirements on conjunction and implication [28], the two connectives should be related. The set of truth degrees equipped with the logical connectives forms a so-called structure of truth degrees. In our work we assume that the structure forms a so-called *complete residuated lattice* $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$, where

(i) $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice (with the least element 0 and the greatest element 1), i.e. a partially ordered set in which arbitrary infima ($\bigwedge$, serving as general quantifier in semantics) and suprema ($\bigvee$, existential quantifier in semantics) exist,

(ii) $\langle L, \otimes, 1\rangle$ is a commutative monoid with unit element 1, i.e. $\otimes$ satisfies

$x \otimes (y \otimes z) = (x \otimes y) \otimes z$ (associativity), $x \otimes y = y \otimes x$ (commutativity), and $x \otimes 1 = x$ (1 acts as a unit) and

(iii) $\otimes$ and $\to$ satisfy $x \otimes y \leq z$ if and only if $x \leq y \to z$, the property called *adjointness*, which comes from the rule of entailment "modus ponens").

Elements x of L are called truth degrees and $\otimes$ and $\to$ are called *multiplication* and *residuum* and play the role of fuzzy conjunction and fuzzy implication, respectively. By the adjointness property we have many important properties of $\otimes$ and $\to$ like monotonicity ($\otimes$ is monotone in both arguments and $\to$ is monotone in second argument and antitone in first argument) or characterization of $\leq$ ($a \leq b$ iff $a \to b = 1$). Thorough the work, we will take advantage of several other properties $\otimes$ and $\to$ (see [7, 30] for a survey).
As stated before, the most applied set L of truth degrees is the real unit interval $[0, 1]$, equipped with $a \wedge b = \min(a, b)$, $a \vee b = \max(a, b)$ and with one of the three important pairs of fuzzy conjunction and fuzzy implication:

- Łukasiewicz: $a \otimes b = \max(a + b - 1, 0)$, $a \to b = \min(1 - a + b, 1)$,
- minimum (Gödel): $a \otimes b = \min(a, b)$, $a \to b = 1$ if $a \leq b$ and $= b$ else and
- product (Goguen): $a \otimes b = a \cdot b$, $a \to b = 1$ if $a \leq b$ and $= b/a$ else.

A complete residuated lattice on $[0, 1]$ with $\wedge$ and $\vee$ being minimum and maximum and $\otimes$ and $\to$ defined by one of the pairs of fuzzy conjunction and fuzzy implication is called a standard Łukasiwicz, Goguen, and Gödel algebra, respectively.
In real-life computing and applications we usually work with finite structures (especially linearly ordered), from obvious reasons of computational tractability. So as a structure of truth degrees we take a finite chain $\{a_0 = 0, a_1, \ldots, a_n = 1\}$ ($a_0 < \cdots < a_n$) equipped with Łukasiewicz connectives ($a_k \otimes a_l = a_{\max(k+l-n,0)}$, $a_k \to a_l = a_{\min(n-k+l,n)}$) or minimum connectives ($a_k \otimes a_l = a_{\min(k,l)}$, $a_k \to a_l = a_n$ for $a_k \leq a_l$ and $a_k \to a_l = a_l$ otherwise). Such structures are then called finite Łukasiewicz and Gödel chains, respectively. More generally we can take a subset of $\{a_0 = 0, a_1, \ldots, a_n = 1\}$ with appropriately defined pair of fuzzy conjunction and fuzzy implication. Note that for $L = \{0, 1\}$, there exists exactly one complete residuated lattice $\mathbf{L}$ – the two-element Boolean algebra, which is the structure of truth degrees of the classical (Boolean) logic.
The structure of truth degrees plays a crucial role in fuzzy logic since it carries all information about truth degrees and fuzzy logical operations. Complete residuated lattices cover entire classes of structures including the most widely used structures with logical operations like the minimum-based,

Łukasiewicz-based and many other (nonlinear and/or finite) structures used in applications. Complete residuated lattices are thus basic general structures of truth degrees used in fuzzy logic, see [28, 30, 29].
In addition to fuzzy conjunction and fuzzy implication we will use also another fuzzy operation. For a complete residuated lattice $\mathbf{L}$, a *(truth-stressing) hedge* is a unary function * satisfying

(i) $1^* = 1$,

(ii) $a^* \leq a$,

(iii) $(a \to b)^* \leq a^* \to b^*$ and

(iv) $a^{**} = a^*$, for all $a, b \in L$.

These properties have natural interpretations leading to that a hedge * acts as a logical connective "very true" [31]. Hedges are monotone mappings, for general $\mathbf{L}$ there are several of them among which, by pointwise ordering, the largest hedge is identity ($a^* = a$ for each $a \in L$), the least hedge is globalization[40] which is defined by $a^* = 1$ for $a = 1$ and $a^* = 0$ for $a < 1$. However, for $L = \{0, 1\}$ (two-element Boolean algebra) there exists the only hedge and it is the identity and globalization in the same time. In the work we will use as the structures of truth degrees complete residuated lattices, possibly with hedge(s).
Now we recall basic notions of fuzzy sets and fuzzy relations. A *fuzzy set with truth degrees from* $\mathbf{L}$ *in a universe* U (which is an ordinary set) is a mapping $A : U \to L$ assigning to any element $u \in U$ a truth degree $A(u) \in L$ to which u belongs to A. The truth degree $A(u) \in L$ is usually called the membership degree of u in A. If $U = \{u_1, \ldots, u_n\}$ then A is denoted by $A = \{ {}^{a_1}\!/u_1, \ldots, {}^{a_n}\!/u_n\}$ meaning that $A(u_i)$ equals a_i. For brevity, we omit elements of U whose membership degree is zero and do not write down the truth degree of 1 (i.e. $\{x\} = \{ {}^{1}\!/x\}$). Fuzzy sets are sometimes called also $\mathbf{L}$-sets to emphasize the structure $\mathbf{L}$ of truth degrees. The set of all $\mathbf{L}$-sets in a universe U is denoted L^U (in accordance to denoting by 2^U the set of all ordinary subsets of U). The operations with fuzzy sets are defined componentwise as usual. For instance, the intersection of fuzzy sets A, B in U is a fuzzy set $A \cap B$ in U such that, for each $u \in U$, $(A \cap B)(u) = A(u) \wedge B(u)$. For a fuzzy set $A \in L^U$ and a truth degree $a \in L$ we denote by ${}^aA$ the (so-called) a-cut of A, i.e. ${}^aA = \{u \in U \mid A(u) \geq a\}$. Thus a-cut of $A \in L^U$ is an ordinary set of elements from U which belong to A to degree at least a. A fuzzy set $a \to A$ in U defined by $(a \to A)(u) = a \to A(u)$ is called an a-shift of A. a-shift of a fuzzy set A has an effect of „shifting" (the membership degree of elements of) A towards the unit 1. In particular the membership degree of all elements of U which belong to A to degree at least a is shifted to 1, producing a thresholded fuzzy set of A by a. Similarly we define a fuzzy set $a \otimes A$ by $(a \otimes A)(u) = a \otimes A(u)$, though without additional

meaning. A fuzzy set A is called crisp if $A(u) \in \{0,1\}$, i.e. an element u either (fully) belongs to A or (fully) not belongs to A, i.e. following common usage we will identify crisp fuzzy sets in U with ordinary subsets of U. The subsethood of fuzzy sets can be defined either the similar way as for ordinary sets (crisp subsethood), i.e. for fuzzy sets A, B in U we put $A \subseteq B$ (A is a subset of B) if for each $u \in U$ we have $A(u) \leq B(u)$; or more generally in fuzzy fashion (graded subsethood, the degree of subsethood): the degree $S(A, B)$ to which A is a subset of B is defined by

$$S(A,B) = \bigwedge_{u \in U} \big(A(u) \to B(u)\big).$$

Then, $A \subseteq B$ is equivalent to $S(A, B) = 1$ which is the case when $A(u) \leq B(u)$ for each $u \in U$ (A is fully contained in B). A *(binary) fuzzy (***L***-) relation* I between (ordinary) sets X and Y is a fuzzy set in universe $U = X \times Y$, i.e. a mapping $I : X \times Y \to L$ assigning to any two elements $x \in X$ and $y \in Y$ a truth degree $I(x, y) \in L$ to which x and y are related under I. As fuzzy relations are but fuzzy sets, all previous notions for fuzzy sets apply also to fuzzy relations.

1.2.2 Formal concept analysis of data with fuzzy attributes

Let X be a non-empty finite set of objects, Y be a non-empty finite set of attributes and I be a (binary) $\mathbf{L}$-relation between X and Y, i.e. I is a mapping $I : X \times Y \to L$. *Formal fuzzy context* (in terms of FCA) is a triplet $\langle X, Y, I\rangle$ and represents a table which assigns to each object $x \in X$ and each attribute $y \in Y$ a truth degree $I(x, y) \in L$ to which object x has attribute y. A formal fuzzy context $\langle X, Y, I\rangle$ can be seen as a data table with fuzzy attributes with rows corresponding to objects, columns corresponding to attributes, and table entries filled with truth degrees $I(x, y)$ for corresponding row x and column y. For $L = \{0, 1\}$, formal fuzzy contexts can be identified in an obvious way with ordinary (two-valued) formal contexts (see [27] for foundations and applications of "classical" FCA).
For $\mathbf{L}$-set $A \in L^X$ of objects, $\mathbf{L}$-set $B \in L^Y$ of attributes we define $\mathbf{L}$-sets $A^{\Uparrow} \in L^Y$ ($\mathbf{L}$-set of attributes) and $B^{\Downarrow} \in L^X$ ($\mathbf{L}$-set of objects) by

$$A^{\Uparrow}(y) = \bigwedge_{x \in X} \big(A(x) \to I(x,y)\big), \tag{1.1}$$

$$B^{\Downarrow}(x) = \bigwedge_{y \in Y} \big(B(y) \to I(x,y)\big) \tag{1.2}$$

(sometimes denoted also $A^{\Uparrow_I}$ and $B^{\Downarrow_I}$ to make I explicit). Using basic rules of predicate fuzzy logic, $A^{\Uparrow}$ can be read as "a fuzzy set of all attributes common to all objects from A", and similarly $B^{\Downarrow}$ as "a fuzzy set of all objects sharing all attributes from B". Operators ${}^{\Downarrow}, {}^{\Uparrow}$ induced by $\langle X, Y, I\rangle$

form a fuzzy Galois connection [7] and were extensively studied [6, 8, 37]. The set of all fixed points of ${}^{\Downarrow}, {}^{\Uparrow}$

$$\mathcal{B}(X, Y, I) = \{\langle A, B\rangle \in \mathbf{L}^X \times \mathbf{L}^Y \mid A^{\Uparrow} = B, B^{\Downarrow} = A\}, \tag{1.3}$$

together with a binary ordering relation $\leq$ defined for $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathcal{B}(X, Y, I)$ by

$$\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \text{ iff } A_1 \subseteq A_2 \text{ or equivalently, iff } B_2 \subseteq B_1$$

is called a *fuzzy concept lattice* induced by $\langle X, Y, I\rangle$, denoted $\langle \mathcal{B}(X, Y, I), \leq\rangle$. Elements $\langle A, B\rangle$ of $\mathcal{B}(X, Y, I)$ are called *(formal) fuzzy concepts* and are interpreted as (interesting) clusters hidden in the input data table. Namely, the conditions $A^{\Uparrow} = B$ and $B^{\Downarrow} = A$ from the definition of formal concept say that "B is the collection of all attributes shared by all objects from A, and A is the collection of all objects sharing all attributes from B". Note that these conditions represent exactly the definition of a concept as developed in Port-Royal logic; A and B are called *extent* and *intent* of the concept $\langle A, B\rangle$, respectively, and represent the collection of all objects and all attributes covered by the particular concept. The ordering relation $\leq$ on concepts is the subconcept-superconcept hierarchy, where concept $\langle A_1, B_1\rangle$ is a subconcept of $\langle A_2, B_2\rangle$ iff each object from A_1 belongs to A_2 and, dually for attributes, iff each attribute from B_2 belongs to B_1. In other words, a subconcept has fewer objects and more attributes than the superconcept. Note also that for $L = \{0, 1\}$ (two truth degrees; bivalent case), the notions of formal fuzzy context, fuzzy concept, and fuzzy concept lattice coincide with the ordinary (crisp) notions in "classical" FCA [27].
As a set of fixed points of ${}^{\Downarrow}, {}^{\Uparrow}$ ordered under $\leq$, $\langle \mathcal{B}(X, Y, I), \leq\rangle$ is a complete lattice, see [3, 8] for further information on its basic properties and study of its structure. The following is the characterization of the structure (so-called *main theorem of fuzzy concept lattices*):

Theorem 1 *The set $\mathcal{B}(X, Y, I)$ is under $\leq$ a complete lattice where infima and suprema are given by*

$$\bigwedge_{j\in J} \langle A_j, B_j\rangle = \langle \bigcap_{j\in J} A_j, (\bigcup_{j\in J} B_j)^{\Downarrow\Uparrow}\rangle, \tag{1.4}$$

$$\bigvee_{j\in J} \langle A_j, B_j\rangle = \langle (\bigcup_{j\in J} A_j)^{\Uparrow\Downarrow}, \bigcap_{j\in J} B_j\rangle. \tag{1.5}$$

Moreover, an arbitrary complete lattice $\mathbf{V} = \langle V, \wedge, \vee\rangle$ is isomorphic to some $\mathcal{B}(X, Y, I)$ iff there are mappings $\gamma : X \times L \to V$, $\mu : Y \times L \to V$ such that

(i) $\gamma(X, L)$ is $\bigwedge$-dense in V,

(ii) $\mu(Y, L)$ is $\bigvee$-dense in V and

(iii) $a \otimes b \leq I(x, y)$ *iff* $\gamma(x, a) \leq \mu(y, b)$.

Recall that K is $\bigwedge$-dense in V if each $v \in V$ is a infimum of some subset of K (and dually for $\bigvee$-density). The lattice structure of $\mathcal{B}(X, Y, I)$ says that each set of concepts has its direct generalization (supremum) and its direct specialization (infimum) in $\mathcal{B}(X, Y, I)$. In addition to $\mathcal{B}(X, Y, I)$, we denote by $\mathrm{Ext}(I) = \{A \mid \langle A, B\rangle \in \mathcal{B}(X, Y, I)$ for some $B\}$ the set of extents of concepts and by $\mathrm{Int}(I) = \{B \mid \langle A, B\rangle \in \mathcal{B}(X, Y, I)$ for some $A\}$ the set of intents of concepts. Note that $\mathrm{Ext}(I)$ and $\mathrm{Int}(I)$ are complete lattices isomorphic or dually isomorphic, respectively, to $\mathcal{B}(X, Y, I)$.
Now, let ^{*_X} and ^{*_Y} be (truth-stressing) hedges. For **L**-sets $A \in L^X$ and $B \in L^Y$, consider **L**-sets $A^{\uparrow} \in L^Y$ and $B^{\downarrow} \in L^X$ defined by

$$A^{\uparrow}(y) = \bigwedge_{x \in X} (A^{*_X}(x) \to I(x, y)), \tag{1.6}$$

$$B^{\downarrow}(x) = \bigwedge_{y \in Y} (B^{*_Y}(y) \to I(x, y)). \tag{1.7}$$

Then, $A^{\uparrow}$ is "a fuzzy set of all attributes common to all objects for which it is very true that they are from A", and $B^{\downarrow}$ is "a fuzzy set of all objects sharing all attributes for which it is very true that they are from B". Operators ${}^{\uparrow}$ and ${}^{\downarrow}$ were introduced in [12, 17] as a parameterization (by hedges) of operators 1.1 and 1.2. Clearly, if both ^{*_X} and ^{*_Y} are identities on L, ${}^{\uparrow}$ and ${}^{\downarrow}$ coincide with ${}^{\Uparrow}$ and ${}^{\Downarrow}$, respectively. To simplify writing and emphasizing this special case, if ^{*_X} or ^{*_Y} is the identity on L, we omit ^{*_X} or ^{*_Y} in $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, e.g. we write just $\mathcal{B}(X^{*_X}, Y, I)$ if ${}^{*_Y} = \mathrm{id}_L$.
The set of all fixed points of ${}^{\downarrow}, {}^{\uparrow}$

$$\mathcal{B}(X^{*_X}, Y^{*_Y}, I) = \{\langle A, B\rangle \in \mathbf{L}^X \times \mathbf{L}^Y \mid A^{\uparrow} = B, B^{\downarrow} = A\}, \tag{1.8}$$

together with analogical binary ordering relation $\leq$ as above defined for $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathcal{B}(X^{*_X}, Y^{*_X}, I)$ by

$$\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \text{ iff } A_1 \subseteq A_2 \text{ or equivalently, iff } B_2 \subseteq B_1$$

is again called a *fuzzy concept lattice (with hedges)* of $\langle X, Y, I\rangle$ and the elements $\langle A, B\rangle \in \mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ *(formal) fuzzy concepts*. For the sake of brevity, we will sometimes write also $\mathcal{B}(X^*, Y^*, I)$ instead of $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$. $\langle \mathcal{B}(X^{*_X}, Y^{*_Y}, I), \leq\rangle$ also happens to be a complete lattice and we refer to [17] for results describing its structure (*main theorem of fuzzy concept lattices with hedges*). $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ is the basic structure used for formal concept analysis of the data table (with fuzzy attributes) represented by $\langle X, Y, I\rangle$.

1.2.3 Factorization by similarity

Finally, we recall the *parametrized method of factorization*of fuzzy concept lattice introduced in [3] to which we refer for details. Briefly, given a data table (formal fuzzy context) $\langle X, Y, I\rangle$, introduce a binary fuzzy relation $\approx$ on the set $\mathcal{B}(X, Y, I)$ of all formal concepts of $\langle X, Y, I\rangle$ by

$$(\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) \;=\; \bigwedge_{x\in X}(A_1(x) \leftrightarrow A_2(x)) \tag{1.9}$$

for $\langle A_i, B_i\rangle \in \mathcal{B}(X, Y, I)$, $i = 1, 2$. Here, $\leftrightarrow$ is a connective of fuzzy equivalence (so-called biresiduum) defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$. It is known (and is easily seen) that $\approx$ is a fuzzy equivalence relation, i.e. we have $(A \approx A) = 1$ (reflexivity), $(A_1 \approx A_2) = (A_2 \approx A_1)$ (symmetry), and $(A_1 \approx A_2) \otimes (A_2 \approx A_3) \leq (A_1 \approx A_3)$ (transitivity). $\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle$ is called the degree of similarity of $\langle A_1, B_1\rangle$ and $\langle A_2, B_2\rangle$. It is easily seen (again, using basic rules of predicate fuzzy logic) that it is the truth degree of "for each object $x \in X$: x is covered by A_1 iff x is covered by A_2". One can show [3] that defining $\approx$ over objects is equivalent to defining over attributes, i.e. $(\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) = \bigwedge_{y\in Y}(B_1(y) \leftrightarrow B_2(y))$. As a consequence, we don't have to distinguish the definitions of $\approx$ over objects and over attributes and write just $\approx$. The equivalence says that measuring similarity of formal concepts via extents A_i coincides with measuring similarity via intents B_i, corresponding to the duality of the extent/intent view on concepts.

Given a truth degree $a \in L$ (threshold specified by a user), consider the thresholded relation ${}^a{\approx}$ on $\mathcal{B}(X, Y, I)$ defined by

$$(\langle A_1, B_1\rangle, \langle A_2, B_2\rangle) \in {}^a{\approx} \quad \text{iff} \quad (\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) \geq a. \tag{1.10}$$

That is, ${}^a{\approx}$ is the (crisp) relation "being similar to degree at least a" and we thereby call it simply *similarity (relation)*. ${}^a{\approx}$ is reflexive and symmetric (i.e., a tolerance relation), but need not be transitive (it is transitive if $a\otimes b = a \wedge b$ holds true in $\mathbf{L}$). A similarity ${}^a{\approx}$ on $\mathcal{B}(X, Y, I)$ is said to be *compatible* if it is preserved under arbitrary suprema and infima in $\mathcal{B}(X, Y, I)$, i.e. if $c_j{}^a{\approx}c'_j$, implies both $(\bigwedge_{j\in J} c_j){}^a{\approx}(\bigwedge_{j\in J} c'_j)$ and $(\bigvee_{j\in J} c_j){}^a{\approx}(\bigvee_{j\in J} c'_j)$ for any $c_j, c'_j \in \mathcal{B}(X, Y, I)$, $j \in J$. We call $\approx$ compatible if ${}^a{\approx}$ is compatible for each $a \in L$.

Call a subset B of $\mathcal{B}(X, Y, I)$ a ${}^a{\approx}$*-block* if it is a maximal subset of $\mathcal{B}(X, Y, I)$ such that each two formal concepts from B are similar to degree at least a. Thus, the notion of a ${}^a{\approx}$-block generalizes that of an equivalence class: if ${}^a{\approx}$ is an equivalence relation (i.e. ${}^a{\approx}$ is transitive), ${}^a{\approx}$-blocks are exactly the equivalence classes. Denote by $\mathcal{B}(X, Y, I)/{}^a{\approx}$ the collection of all ${}^a{\approx}$-blocks. It can be shown that, if ${}^a{\approx}$ is compatible, then ${}^a{\approx}$-blocks are special intervals in the concept lattice $\mathcal{B}(X, Y, I)$ [3, 27]. In detail, for a formal concept

$\langle A, B\rangle \in \mathcal{B}(X, Y, I)$, put

$$\langle A, B\rangle_a \;:=\; \bigwedge\{\langle A', B'\rangle \mid (\langle A, B\rangle, \langle A', B'\rangle) \in {}^a\!\approx\}, \tag{1.11}$$

$$\langle A, B\rangle^a \;:=\; \bigvee\{\langle A', B'\rangle \mid (\langle A, B\rangle, \langle A', B'\rangle) \in {}^a\!\approx\}. \tag{1.12}$$

That is, $\langle A, B\rangle_a$ and $\langle A, B\rangle^a$ are the infimum and the supremum of the set of all formal concepts which are similar to $\langle A, B\rangle$ to degree at least a. Operators $\ldots_a$ and $\ldots^a$ are important in description of ${}^a\!\approx$-blocks [27]:

Lemma 2 *${}^a\!\approx$-blocks are exactly intervals of $\mathcal{B}(X, Y, I)$ of the form $[\langle A, B\rangle_a, (\langle A, B\rangle_a)^a]$, i.e.*

$$\mathcal{B}(X, Y, I)\,/{}^a\!\approx \;=\; \{[\langle A, B\rangle_a, (\langle A, B\rangle_a)^a] \mid \langle A, B\rangle \in \mathcal{B}(X, Y, I)\}.$$

Note that an interval with lower bound $\langle A_1, B_1\rangle$ and upper bound $\langle A_2, B_2\rangle$ is the subset

$$[\langle A_1, B_1\rangle, \langle A_2, B_2\rangle] = \{\langle A, B\rangle \mid \langle A_1, B_1\rangle \leq \langle A, B\rangle \leq \langle A_2, B_2\rangle\}.$$

Now, define a partial order $\preceq$ on blocks of $\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ by

$$[c_1, c_2] \preceq [d_1, d_2] \quad \text{iff} \quad c_1 \leq d_1 \text{ (iff } c_2 \leq d_2)$$

where $[c_1, c_2], [d_1, d_2] \in \mathcal{B}(X, Y, I)\,/{}^a\!\approx$ ($c_i \leq d_i$ denotes that in $\mathcal{B}(X, Y, I)$, c_i is a subconcept of d_i). Then we have

Theorem 3 *$\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ equipped with $\preceq$ is a partially ordered set which is a complete lattice, the so-called factor lattice of $\mathcal{B}(X, Y, I)$ by similarity $\approx$ and a threshold a.*

Generally, $\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ is a *factor lattice* of $\mathcal{B}(X, Y, I)$ by a compatible tolerance relation ${}^a\!\approx$ on $\mathcal{B}(X, Y, I)$ (see e.g. [27] for the notion of a factor lattice by a tolerance). Elements of $\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ can be seen as similarity-based granules of formal concepts from $\mathcal{B}(X, Y, I)$. $\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ thus provides a granular view on (a possibly large) $\mathcal{B}(X, Y, I)$. Note also that if ${}^a\!\approx$ is transitive then it is a congruence relation on $\mathcal{B}(X, Y, I)$ and $\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ is the usual factor lattice modulo a congruence. For further details and properties of $\mathcal{B}(X, Y, I)\,/{}^a\!\approx$ we refer to [3].
We now present an illustrative example. Consider the data table depicted in Tab. 1.1. The data table describes 25 countries of EU (objects from X) by some of their demographic and economic characteristics (attributes from Y). The data was obtained from the study "Czech Republic in European Union: benefits and costs" of Consortium for study of international relations (2004, in Czech, downloadable from `http://www.evropska-unie.cz`). The original values of the characteristics are scaled to interval $[0, 1]$ so that the

Table 1.1: Data table of EU countries, 3 truth degrees.

	a	b	c	d	e
1 Austria	0	0	$\frac{1}{2}$	1	1
2 Belgium	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$
3 Cyprus	0	0	$\frac{1}{2}$	1	1
4 Czech rep.	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
5 Denmark	0	0	$\frac{1}{2}$	1	1
6 Estonia	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
7 Finland	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
8 France	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$
9 Germany	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
10 Greece	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
11 Hungary	0	0	0	0	1
12 Ireland	0	0	$\frac{1}{2}$	$\frac{1}{2}$	1
13 Italy	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
14 Latvia	0	0	0	1	$\frac{1}{2}$
15 Lithuania	0	0	0	1	0
16 Luxembourg	0	0	1	1	1
17 Malta	0	0	0	1	$\frac{1}{2}$
18 Netherlands	0	0	$\frac{1}{2}$	$\frac{1}{2}$	1
19 Poland	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0
20 Portugal	0	0	0	$\frac{1}{2}$	1
21 Slovakia	0	0	0	0	0
22 Slovenia	0	0	$\frac{1}{2}$	0	1
23 Spain	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$
24 Sweden	0	1	$\frac{1}{2}$	1	1
25 UK	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1

attributes: a – many habitants (millions), b – large area (thousands km^2), c – high GDP (EUR), d – low inflation (%), e – low unemployment (%)

characteristics can be considered as fuzzy attributes with truth degrees from three element chain $L = \{0, \frac{1}{2}, 1\}$.
The corresponding concept lattice computed using Łukasiewicz fuzzy logical operations is depicted in Fig. 1.1 and all formal fuzzy concepts of it are listed in 1.2.
For $a = \frac{1}{2}$ there are ten ${}^{\frac{1}{2}}\!\approx$-blocks and they are depicted in Fig. 1.2 (blocks are highlighted by solid lines) together with the corresponding factor lattice $\mathcal{B}(X, Y, I) / {}^{\frac{1}{2}}\!\approx$.

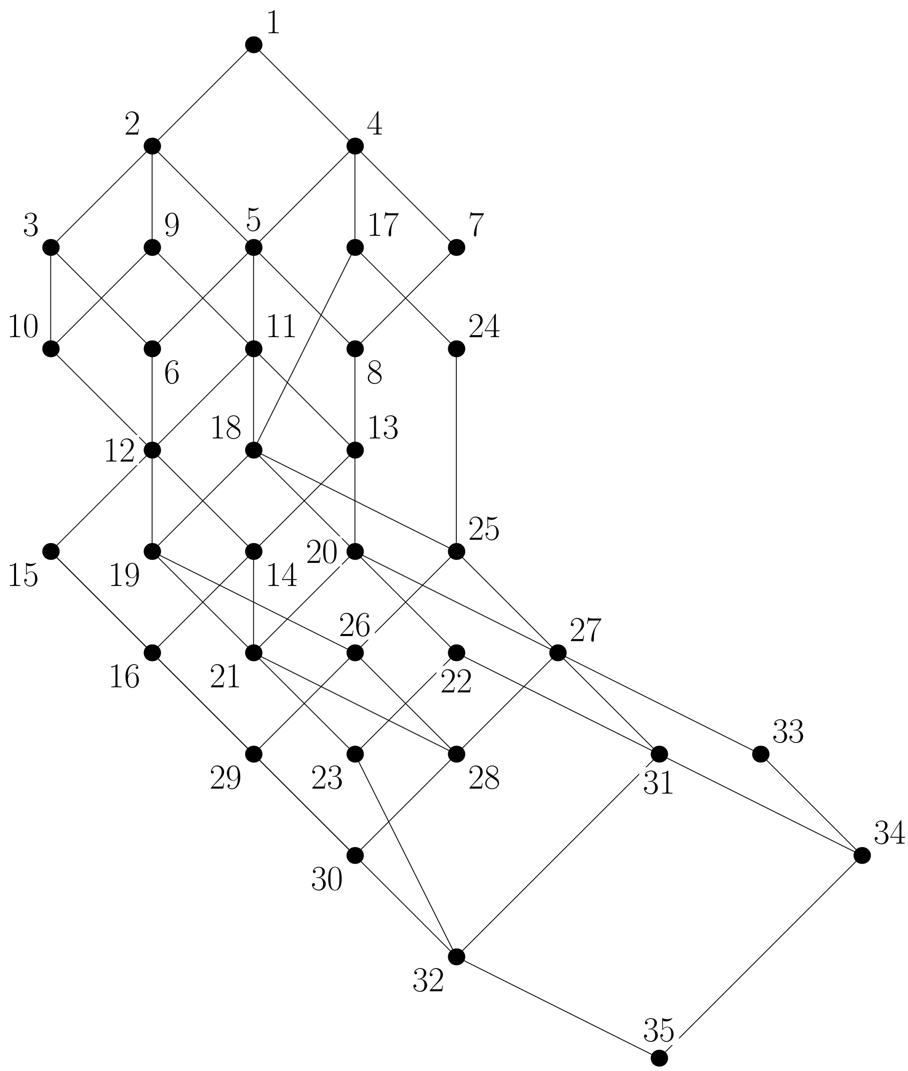


Figure 1.1: Concept lattice $\mathcal{B}(X, Y, I)$ of data table from Tab. 1.1.

Table 1.2: Concepts of data table from Tab. 1.1.

	extent																									intent				
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1	1	0	0	0	0	$\frac{1}{2}$
3	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	0	1	0	1	$\frac{1}{2}$	1	1	0	0	0	0	1
4	1	1	1	1	1	1	1	1	1	1	$\frac{1}{2}$	1	1	1	1	1	1	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	0	0	0	$\frac{1}{2}$	0
5	1	1	1	1	1	1	1	1	1	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
6	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1	1	0	0	0	$\frac{1}{2}$	1
7	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	1	1	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	1	1	0	0	0	1	0
8	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	1	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	1	1	0	0	0	1	$\frac{1}{2}$
9	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	1	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$
10	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	0	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	1	0	0	$\frac{1}{2}$	0	1
11	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
12	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	1
13	1	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	1	1	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$
14	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	1	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	1	1	0	0	$\frac{1}{2}$	1	1
15	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	$\frac{1}{2}$	1
16	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	1	1
17	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0
18	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
19	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	1	1	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
20	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	1	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
21	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	1	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1	1
22	0	0	0	0	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	0	0	0	1	1	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	$\frac{1}{2}$
23	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0	1	$\frac{1}{2}$	1	1
24	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0
25	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
26	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
27	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
28	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1
29	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1
30	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1
31	0	0	0	0	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	0	0	0	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$
32	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	1	1
33	0	0	0	0	0	0	0	$\frac{1}{2}$	1	0	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$
34	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	1	1	$\frac{1}{2}$	1	$\frac{1}{2}$
35	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	1	1	1	1	1

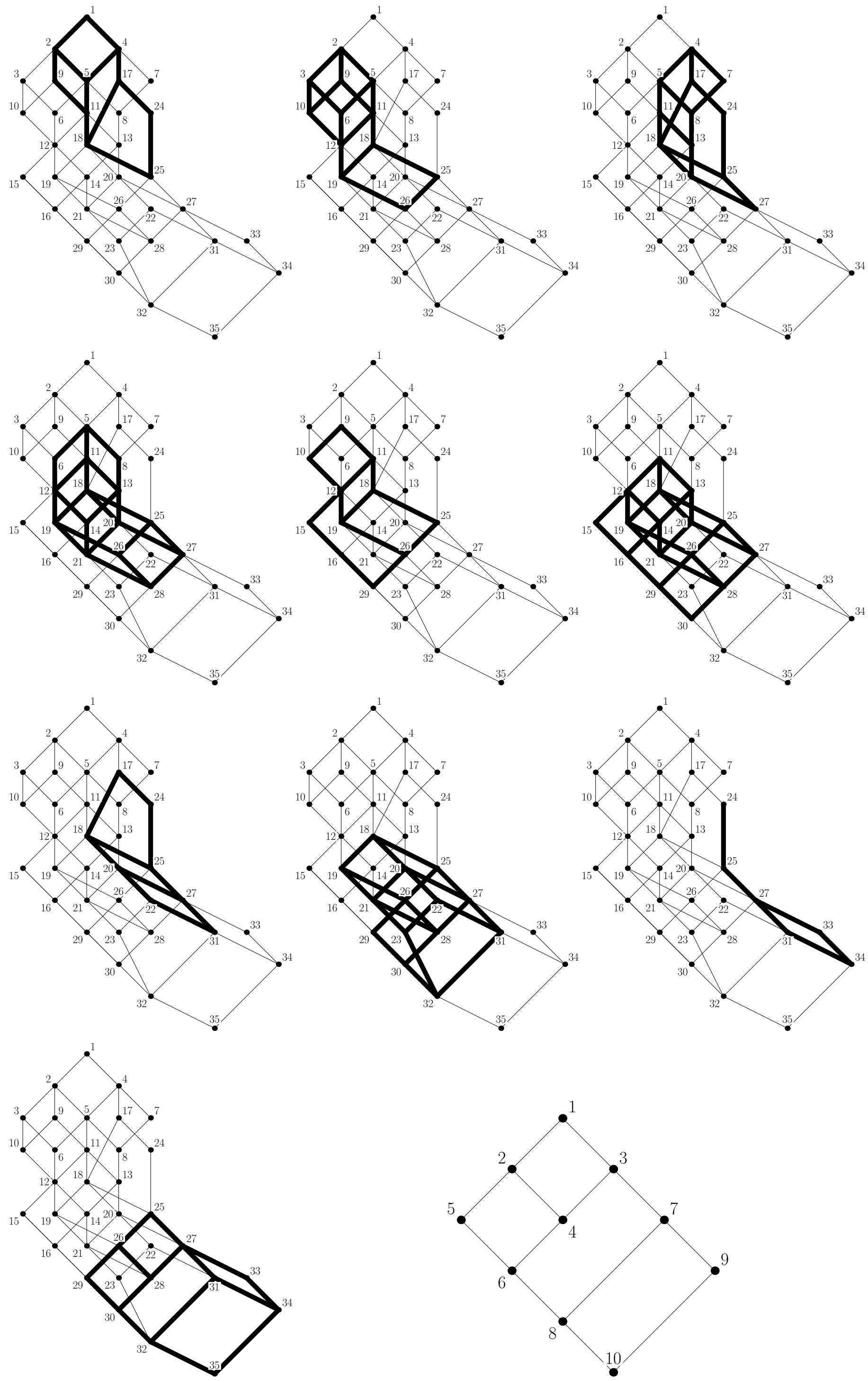


Figure 1.2: $\frac{1}{2}\!\approx$-blocks of concept lattice of Fig. 1.1 and the corresponding factor lattice $\mathcal{B}(X,Y,I)\,/\,\frac{1}{2}\!\approx$ (bottom right).

Chapter 2

Factorizing fuzzy concept lattices by ${}^a\!\approx$

2.1 Introduction

In section 1.1 there was introduced a method of parameterized factorization of concept lattices from data tables with fuzzy attributes. The factorization is by the similarity relation ${}^a\!\approx$ determined by a user-specified threshold a and was in short described in section 1.2.3. In order to compute the factor lattice (quite easily) directly by definition presented there, we have to (1) compute the whole concept lattice and then (2) to compute all the similarity blocks, i.e. elements of the factor lattice. For computation of an ordinary concept lattice from binary input data there exist dozens of algorithms with quite varying, but polynomial time delay, see [35] for survey and comparison. Without surprise, a fuzzy concept lattice can be computed by an algorithm with a polynomial time delay as well, see [9] for an algorithm which we will use and which is described in the following section. Finding all the similarity blocks can of course also be accomplished by an algorithm with polynomial time delay, concluding that computing the factor lattice directly by definition we get it in time polynomial to the size of input data table. Well, although polynomial time delay is fine, the computation of the whole possibly large concept lattice and, especially, factorizing this lattice, can be quite time demanding. And, realizing that we compute the whole concept lattice only in order to obtain the smaller factor lattice, also somewhat frustrating. Fortunately, the following sections show that we can avoid it!
We present two ways to compute the factor lattice directly from input data, without first computing the whole concept lattice and then computing the collections (similarity blocks) of concepts. First way (sometimes called fast factorization, presented in section 2.2.1) goes through describing a new (fuzzy) closure operator the fixed points of which are just concepts which uniquely determine the blocks of similar concepts. Then any algorithm com-

puting all fixed points of a closure operator does the rest. The second way (called direct factorization, presented in section 2.2.2) is somehow similar. Instead of forming new closure operator on original input data, we appropriately modify the input data table so that the concepts computed from such a modified table, using any algorithm for computing a fuzzy concept lattice, again uniquely determine the blocks of similar concepts, i.e. the elements of factor lattice. The resulting algorithms are indeed significantly faster than computing first the whole concept lattice and then computing the similarity blocks.[1] Furthermore, the smaller the similarity threshold, the faster the computation of the factor lattice. The feature, which the two-step computation directly by definition does not guarantee (as we will see in experiments), but which straightly corresponds to a rule "the more tolerance to imprecision, the faster the result" which is so characteristic for human categorization.
The previous paragraphs apply to parametrized factorization of fuzzy concept lattices as defined by (1.3). However, in preliminaries (section 1.2.1), we introduced hedges, so in section 2.3 we look to what extent the idea of factorization by ${}^{a}\!\approx$ can be applied to fuzzy concept lattices with hedges, defined by (1.8).
The important part of the chapter constitutes the section 2.4, which summarizes the extensive examples and experiments supporting our methods of factorization and demonstrating the significant speed-up. Finally, section 2.5 concludes and presents an outline of a future research.

The chapter summarizes results contained in [10] (section 2.2.1), [11] (section 2.2.2) and recent [13] (section 2.3).

2.2 Computing the factor lattice $\mathcal{B}(X,Y,I)/{}^{a}\!\approx$ directly from input data

2.2.1 New fuzzy closure operator

In order to compute $\mathcal{B}(X,Y,I)/{}^{a}\!\approx$ using definition and Lemma 2, one has (1) to compute the whole concept lattice $\mathcal{B}(X,Y,I)$ and then (2) to compute ${}^{a}\!\approx$-blocks on $\mathcal{B}(X,Y,I)$, which can be quite time demanding. We are going to propose a way to compute $\mathcal{B}(X,Y,I)/{}^{a}\!\approx$ directly from input data $\langle X,Y,I\rangle$.

We need some auxiliary results. For basic properties of concept lattices in fuzzy setting we refer to [3, 8]. For fuzzy sets C, D in U, put

$$(C \approx D) = \bigwedge_{u\in U} \big(C(u) \leftrightarrow D(u)\big).$$

[1] This is all about, of course.

Furthermore, we call a fuzzy set A in X an extent if there is a fuzzy set B in Y such that $\langle A, B\rangle \in \mathcal{B}(X,Y,I)$ (similarly, B is an intent if there is A with $\langle A, B\rangle \in \mathcal{B}(X,Y,I)$).

Lemma 4 *If A is an extent then so is $a \to A$; similarly, if B is an intent then so is $a \to B$.*

PROOF. We prove the assertion for extents. Let A be an extent, i.e. $\langle A, B\rangle \in \mathcal{B}(X,Y,I)$ for some B. We have to show that $\langle a \to A, B'\rangle \in \mathcal{B}(X,Y,I)$. It suffices to show that $a \to A = (a \to A)^{\Uparrow\Downarrow}$ (since then $\langle a \to A, (a \to A)^{\Uparrow}\rangle$ is a formal concept). Since $a \to A \subseteq (a \to A)^{\Uparrow\Downarrow}$ is always the case, we have to show $(a \to A)^{\Uparrow\Downarrow} \subseteq a \to A$ which holds iff $(a \to A)^{\Uparrow\Downarrow}(x) \leq a \to A(x)$ for each $x \in X$. Using adjointness, the latter is equivalent to $a \leq (a \to A)^{\Uparrow\Downarrow}(x) \to A(x)$. Since

$$\begin{aligned}(a \to A)^{\Uparrow\Downarrow}(x) \to A(x) \geq \bigwedge_{x\in X}\big((a \to A)^{\Uparrow\Downarrow}(x) \leftrightarrow A(x)\big) = \\ = ((a \to A)^{\Uparrow\Downarrow} \approx A),\end{aligned}$$

it suffices to show $a \leq ((a \to A)^{\Uparrow\Downarrow} \approx A)$. First, we have $a \leq ((a \to A) \approx A)$. Indeed, from $a \leq ((a \to A(x)) \to A(x))$ and $a \leq (A(x) \to (a \to A(x)))$ for each $x \in X$ we have $a \leq ((a \to A(x)) \leftrightarrow A(x))$ for each $x \in X$, i.e. $a \leq \bigwedge_{x\in X}((a \to A(x)) \leftrightarrow A(x)) = ((a \to A) \approx A)$. Furthermore, since $(A_1 \approx A_2) \leq (A_1^{\Uparrow} \approx A_2^{\Uparrow})$ and $(B_1 \approx B_2) \leq (B_1^{\Downarrow} \approx B_2^{\Downarrow})$ for $A_1, A_2 \in L^X$ and $B_1, B_2 \in L^Y$ (see [3]), we have $(A_1 \approx A_2) \leq (A_1^{\Uparrow} \approx A_2^{\Uparrow}) \leq (A_1^{\Uparrow\Downarrow} \approx A_2^{\Uparrow\Downarrow})$. Putting this together, we get $a \leq ((a \to A) \approx A) \leq ((a \to A)^{\Uparrow} \approx A^{\Uparrow}) \leq ((a \to A)^{\Uparrow\Downarrow} \approx A^{\Uparrow\Downarrow})$, completing the proof. □

The next lemma shows that for a formal concept $\langle A, B\rangle$, $\langle A, B\rangle_a$ and $\langle A, B\rangle^a$, defined by (1.11) and (1.12) as infimum and supremum of all formal concepts similar to $\langle A, B\rangle$ to degree at least a, can be computed from $\langle A, B\rangle$ directly.

Lemma 5 *For $\langle A, B\rangle \in \mathcal{B}(X,Y,I)$, we have*

(a) $\langle A, B\rangle_a = \langle (a \otimes A)^{\Uparrow\Downarrow}, a \to B\rangle$ *and*

(b) $\langle A, B\rangle^a = \langle (a \to A), (a \otimes B)^{\Downarrow\Uparrow}\rangle$.

PROOF. Due to duality we verify only (a). The assertion follows from the following claims.

(a1) $(a \otimes A)^{\Uparrow\Downarrow}$ is an extent of a formal concept $\langle (a \otimes A)^{\Uparrow\Downarrow}, D\rangle$ which is similar to $\langle A, B\rangle$ to degree at least a;

(a2) if $\langle C, F\rangle$ is a formal concept similar to $\langle A, B\rangle$ to degree at least a then $\langle (a \otimes A)^{\Uparrow\Downarrow}, D\rangle \leq \langle C, F\rangle$;

(a3) $a \to B$ is an intent of a concept c which is similar to $\langle A, B\rangle$ to degree at least a;

(a4) if $\langle C, F\rangle$ is a concept similar to $\langle A, B\rangle$ to degree at least a then for c from (a3) we have $c \leq \langle C, F\rangle$.

Indeed, from (a1) and (a2) we get that $\langle (a \otimes A)^{\Uparrow\Downarrow}, D\rangle$ is the least formal concept similar to $\langle A, B\rangle$ to degree at least a. Therefore, $\langle A, B\rangle_a = \langle (a \otimes A)^{\Uparrow\Downarrow}, D\rangle$. Then, (a3) and (a4) yield that $a \to B$ is an intent of the least formal concept similar to $\langle A, B\rangle$ to degree at least a, i.e. $a \to B = D$. We now verify (a1)–(a4).
(a1): We have $a \leq ((a \otimes A) \approx A) \leq ((a \otimes A)^{\Uparrow} \approx A^{\Uparrow}) \leq ((a \otimes A)^{\Uparrow\Downarrow} \approx A^{\Uparrow\Downarrow}) = ((a \otimes A)^{\Uparrow\Downarrow} \approx A)$ since A is an extent.
(a2): If $a \leq (A \approx C)$ then using adjointness, we get $a \otimes A \subseteq C$ from which we have $(a \otimes A)^{\Uparrow\Downarrow} \subseteq C^{\Uparrow\Downarrow} = C$, proving (a2).
(a3): By Lemma 4, $a \to B$ is an intent. Using adjointness we easily get $a \leq (B \approx a \to B) = (\langle A, B\rangle \approx c)$.
(a4): We need to show $F \subseteq a \to B$. Since $a \leq (\langle A, B\rangle \approx \langle C, F\rangle) = (B \approx F)$, adjointness gives $a \otimes F \subseteq B$ and then $F \subseteq a \to B$. The proof is complete. □

Thus we have $(\langle A, B\rangle_a)^a = \langle a \to (a \otimes A)^{\Uparrow\Downarrow}, (a \otimes (a \to B))^{\Downarrow\Uparrow}\rangle$.

Lemma 6 *For $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ we have $\langle A, B\rangle_a = (((\langle A, B\rangle_a)^a)_a$.*

PROOF. First we show that for every $c, d \in \mathcal{B}(X, Y, I)$ we have (1) $c \leq d$ implies $c_a \leq d_a$, (2) $c \leq d$ implies $c^a \leq d^a$, (3) $c \leq (c_a)^a$, (4) $c \geq (c^a)_a$.
(1): Recall that $c_a = \bigwedge\{e \in \mathcal{B}(X, Y, I) \mid \langle c, e\rangle \in {}^a{\approx}\}$. We need to show that if $\langle d, f\rangle \in {}^a{\approx}$ then $c_a \leq f$. Thus suppose $\langle d, f\rangle \in {}^a{\approx}$. From $\langle c, c\rangle \in {}^a{\approx}$ and from the fact that ${}^a{\approx}$ is a tolerance relation compatible with lattice operations on $\mathcal{B}(X, Y, I)$ we get $\langle c, c \wedge f\rangle = \langle c \wedge d, c \wedge f\rangle \in {}^a{\approx}$. Now, since c_a is the infimum of all e such that $\langle c, e\rangle \in {}^a{\approx}$, we have $c_a \leq c \wedge f$ and since $c \wedge f \leq f$, we get $c_a \leq f$, proving (1).
(2) can be proved analogously. (3) and (4) are obvious.
Now, let $c = \langle A, B\rangle$. By (3), $c \leq (c_a)^a$ and so $c_a \leq ((c_a)^a)_a$ by (1). Applying (4) to c_a we get $c_a \geq ((c_a)^a)_a$, proving $c_a = ((c_a)^a)_a$. □

By Lemma 6, if a ${}^a{\approx}$-block $[c_1, c_2]$ is generated by $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$, i.e. $c_1 = \langle A, B\rangle_a$, $c_2 = (\langle A, B\rangle_a)^a$, then it is also generated by c_2, i.e. $c_1 = (c_2)_a$ and $c_2 = ((c_2)_a)^a$. Therefore, ${}^a{\approx}$-blocks $[c_1, c_2]$ are uniquely given by their suprema c_2. Moreover, since each formal concept $c_2 = \langle A, B\rangle$ is uniquely given by A (namely, $B = A^{\Uparrow}$), ${}^a{\approx}$-blocks are uniquely given by extents of their suprema. Denote the set of all extents of suprema of ${}^a{\approx}$-blocks by ESB(a), i.e.

$$\begin{aligned}\mathrm{ESB}(a) = \{A \in L^X \mid\ & \langle A, B\rangle \in \mathcal{B}(X, Y, I) \text{ and} \\ & [\langle A, B\rangle_a, \langle A, B\rangle] \in \mathcal{B}(X, Y, I)/{}^a{\approx}\}.\end{aligned}$$

Before presenting the main result, let us recall that a fuzzy (**L**-)closure operator in a set U [4] is a mapping $C : A \to C(A)$ satisfying

(1) $A \subseteq C(A)$,

(2) $S(A_1, A_2) \leq S(C(A_1), C(A_2))$ and

(3) $C(A) = C(C(A))$,

for any $A, A_1, A_2 \in L^U$. A fixed point of C is any fuzzy set A in U such that $A = C(A)$. Denote by fix(C) the set of all fixed points of C, i.e.

$$\mathrm{fix}(C) = \{A \in L^X \mid A = C(A)\}.$$

Theorem 7 *Given input data $\langle X, Y, I\rangle$ and a threshold $a \in L$, a mapping C_a sending a fuzzy set A in X to a fuzzy set $a \to (a \otimes A)^{\Uparrow\Downarrow}$ in X is a fuzzy closure operator in X for which* $\mathrm{fix}(C_a) = \mathrm{ESB}(a)$.

PROOF. First, we verify that C_a is a fuzzy closure operator. $A \subseteq C_a(A)$ means $A \subseteq a \to (a \otimes A)^{\Uparrow\Downarrow}$ which is equivalent (by adjointness) to $a \otimes A \subseteq (a \otimes A)^{\Uparrow\Downarrow}$ which is true since $E \subseteq E^{\Uparrow\Downarrow}$ is always the case. We showed $A \subseteq C_a(A)$.
$S(A_1, A_2) \leq S(C_a(A_1), C_a(A_2))$: Since for $D_1, D_2 \in L^U$, $S(D_1, D_2) \leq S(a \otimes D_1, a \otimes D_2)$ and $S(D_1, D_2) \leq S(a \to D_1, a \to D_2)$, see [7], we have

$$\begin{aligned} &S(A_1, A_2) \leq S(a \otimes A_1, a \otimes A_2) \leq S((a \otimes A_1)^{\Uparrow\Downarrow}, (a \otimes A_2)^{\Uparrow\Downarrow}) \leq \\ &\leq S(a \to (a \otimes A_1)^{\Uparrow\Downarrow}, a \to (a \otimes A_2)^{\Uparrow\Downarrow}) = S(C_a(A_1), C_a(A_2)). \end{aligned}$$

To verify $C_a(A) = C_a(C_a(A))$, suppose first that A is an extent. Then, by Lemma 5, $C_a(A)$ is the extent of $(\langle A, A^{\Uparrow}\rangle_a)^a$. In order to show $C_a(A) = C_a(C_a(A))$, we thus have to check $(\langle A, A^{\Uparrow}\rangle_a)^a = ((((\langle A, A^{\Uparrow}\rangle_a)^a)_a)^a$ which is true due to Lemma 6. If A is not an extent, the assertion follows from the fact that $C_a(A) = C_a(A^{\Uparrow\Downarrow})$, the fact that $A^{\Uparrow\Downarrow}$ is an extent and the previous claim. We thus need to check $C_a(A) = C_a(A^{\Uparrow\Downarrow})$. We have $a \leq (A \approx a \otimes A) \leq (A^{\Uparrow\Downarrow} \approx (a \otimes A)^{\Uparrow\Downarrow})$. So, $A^{\Uparrow\Downarrow}$ is similar to $(a \otimes A)^{\Uparrow\Downarrow}$ to degree at least a, whence $a \to (a \otimes A)^{\Uparrow\Downarrow} \supseteq A^{\Uparrow\Downarrow}$ since by Lemma 5, $a \to (a \otimes A)^{\Uparrow\Downarrow}$ is the greatest one which is similar to $(a \otimes A)^{\Uparrow\Downarrow}$ to degree at least a. In fact, in order to apply Lemma 5, A needs to be an extent. However, going through the proof, one can see that $(a \otimes A)^{\Uparrow\Downarrow}$ is the extent of the least formal concept which is similar to A to degree at least a even for an arbitrary fuzzy set A (not necessarily an extent). Therefore, the claim of Lemma 5 can be safely used in our case. We therefore have $A \subseteq A^{\Uparrow\Downarrow} \subseteq a \to (a \otimes A)^{\Uparrow\Downarrow}$ and since $a \otimes (a \to b) \leq b$, we get

$$\begin{aligned} &(a \otimes A)^{\Uparrow\Downarrow} \subseteq (a \otimes A^{\Uparrow\Downarrow})^{\Uparrow\Downarrow} \subseteq (a \otimes (a \to (a \otimes A)^{\Uparrow\Downarrow}))^{\Uparrow\Downarrow} \subseteq \\ &\subseteq ((a \otimes A)^{\Uparrow\Downarrow})^{\Uparrow\Downarrow} = (a \otimes A)^{\Uparrow\Downarrow}. \end{aligned}$$

This proves $(a \otimes A)^{\Uparrow\Downarrow} = (a \otimes A^{\Uparrow\Downarrow})^{\Uparrow\Downarrow}$ and so $C_a(A) = a \to (a \otimes A)^{\Uparrow\Downarrow} = a \to (a \otimes A^{\Uparrow\Downarrow})^{\Uparrow\Downarrow} = C_a(A^{\Uparrow\Downarrow})$.
Second, we verify $\mathrm{fix}(C_a) = \mathrm{ESB}(a)$. Let $A \in \mathrm{fix}(C_a)$. By Lemma 2, the interval $[\langle A, A^{\Uparrow}\rangle_a, (\langle A, A^{\Uparrow}\rangle_a)^a]$ is a ${}^a\!\approx$-block, and by Lemma 5, $(\langle A, A^{\Uparrow}\rangle_a)^a = \langle a \to (a \otimes A)^{\Uparrow\Downarrow}, \dots\rangle$. Since $A = C_a(A) = a \to (a \otimes A)^{\Uparrow\Downarrow}$, A is the extent of a supremum of a block, i.e. $A \in \mathrm{ESB}(a)$. Conversely, let $A \in \mathrm{ESB}(a)$. Then $[\langle A, A^{\Uparrow}\rangle_a, \langle A, A^{\Uparrow}\rangle]$ is an ${}^a\!\approx$-block and so $(\langle A, A^{\Uparrow}\rangle_a)^a = \langle A, A^{\Uparrow}\rangle$. Lemma 5 now gives $A = a \to (a \otimes A)^{\Uparrow\Downarrow}$, i.e. $A = C_a(A)$ verifying $A \in \mathrm{fix}(C_a)$. $\square$

Therefore, A is the extent of some formal concept c_2 which is the supremum of some ${}^a\!\approx$-block $[c_1, c_2] \in \mathcal{B}(X, Y, I)/{}^a\!\approx$ if and only if A is a fixed point of C_a. By Theorem 7 and the above considerations, going through $\mathrm{fix}(C_a)$ and computing for each $A \in \mathrm{fix}(C_a)$ the corresponding $[\langle A, A^{\Uparrow}\rangle_a, \langle A, A^{\Uparrow}\rangle] = [\langle (a \otimes A)^{\Uparrow\Downarrow}, a \to A^{\Uparrow}\rangle, \langle A, A^{\Uparrow}\rangle]$ generates all ${}^a\!\approx$-blocks of $\mathcal{B}(X, Y, I)/{}^a\!\approx$. Strictly speaking, we do not generate the ${}^a\!\approx$-blocks $[c_1, c_2] \in \mathcal{B}(X, Y, I)/{}^a\!\approx$ but only their boundary formal concepts $c_1, c_2 \in \mathcal{B}(X, Y, I)$. This is, however, in accordance with the purpose of the factorization of $\mathcal{B}(X, Y, I)$: We are looking for a granular view which is more concise than $\mathcal{B}(X, Y, I)$ itself.

The problem of computing $\mathcal{B}(X, Y, I)/{}^a\!\approx$ thus reduces to the problem of computing $\mathrm{fix}(C_a)$. To this end, we can use the algorithm described in [9]. The algorithm is an extension of the Ganter's NextClosure algorithm generating all fixed points of an (ordinary) closure operator (see [27]) and generates all fixed points of a fuzzy closure operator C in a lexicographic order. Note that the algorithm in [9] is formulated in terms of the fuzzy closure operator ${}^{\Uparrow\Downarrow}$ (i.e. sending A to $A^{\Uparrow\Downarrow}$). But since each fuzzy closure operator is of the form of ${}^{\Uparrow\Downarrow}$ [4, 5], there is no loss of generality involved. We now briefly recall the algorithm from [9].
Suppose $X = \{1, 2, \dots, n\}$ and $L = \{0 = a_1, a_2, \dots, a_k = 1\}$ such that if $a_i \le a_j$ in $\mathbf{L}$ then $i \le j$ (i.e. the ordering of elements of L by indices extends their ordering in $\mathbf{L}$). For $i, r \in \{1, \dots, n\}$, $j, s \in \{1, \dots, k\}$, put

$$(i, j) \le (r, s) \text{ iff } i < r, \text{ or } i = r \text{ and } j \ge s.$$

For $A \in L^X$, $(i, j) \in X \times \{1, \dots, k\}$, put

$$A \oplus (i, j) := C_a((A \cap \{1, 2, \dots, i-1\}) \cup \{\, {}^{a_j}\!/i\}).$$

Here, $A \cap \{1, 2, \dots, i-1\}$ is the intersection of a fuzzy set A and the ordinary set $\{1, 2, \dots, i-1\}$, i.e. $(A \cap \{1, 2, \dots, i-1\})(x) = A(x)$ for $x < i$ and $(A \cap \{1, 2, \dots, i-1\})(x) = 0$ otherwise. Furthermore, for $A, B \in L^X$, put

$$\begin{aligned} &A <_{(i,j)} B \text{ iff} \\ &\quad A \cap \{1, \dots, i-1\} = B \cap \{1, \dots, i-1\} \text{ and } A(i) < B(i) = a_j. \end{aligned}$$

Finally, put

$$A < B \text{ iff } A <_{(i,j)} B \text{ for some } (i,j).$$

Then $<$ is a total order on L^X and for each $A \in L^X$, the least fixed point $A^+ \in \mathrm{fix}(C_a)$ which is greater (w.r.t. $<$) than A is given by $A^+ = A \oplus (i,j)$ where (i,j) is the greatest one with $A <_{(i,j)} A \oplus (i,j)$ (see [9]). The algorithm for generating ${}^a\!\approx$-blocks which is based on this description of the successor operator $^+$ follows.

INPUT: $\langle X,Y,I\rangle$ (data table with fuzzy attributes), $a \in L$ (similarity threshold)
OUTPUT: $\mathcal{B}(X,Y,I)\,/^a\!\approx$ (${}^a\!\approx$-blocks $[c_1,c_2]$)

```
/* Algorithm */
A := ∅
while A ≠ X do
  A := A⁺
  store([⟨(a ⊗ A)^⇑⇓, a → A^⇑⟩, ⟨A, A^⇑⟩])
```

As argued in [9], generating $\mathrm{fix}(C_a)$ has polynomial time delay complexity (i.e., given a fixed point, the next one is generated in time polynomial in terms of size of the input $\langle X,Y,I\rangle$ [32]). Since generating a ${}^a\!\approx$-block $[\langle (a\otimes A)^{\Uparrow\Downarrow}, a \rightarrow A^{\Uparrow}\rangle, \langle A, A^{\Uparrow}\rangle]$ from A takes a polynomial time, our algorithm is of polynomial time delay complexity as well.

2.2.2 Factorized context

Now we are going to propose another way to compute $\mathcal{B}(X,Y,I)\,/^a\!\approx$ directly from input data, without computing first the whole $\mathcal{B}(X,Y^{*_Y},I)$ and then computing the similarity blocks. First, we propose a construction of a similarity-based factorization assigning to $\langle X,Y,I\rangle$ a "factorized data" $\langle X,Y,I\rangle/a$. Then we show that $\mathcal{B}(X,Y,I)\,/^a\!\approx$ is isomorphic to $\mathcal{B}(\langle X,Y,I\rangle/a)$. This reduces the computation of $\mathcal{B}(X,Y,I)\,/^a\!\approx$ to the computation of an ordinary fuzzy concept lattice $\mathcal{B}(\langle X,Y,I\rangle/a)$ for which we have an algorithm (see [9] or previous section) with a polynomial time delay complexity (see [32]).

For a formal fuzzy context $\langle X,Y,I\rangle$ and a (user-specified) threshold $a \in L$, introduce a formal fuzzy context $\langle X,Y,I\rangle/a$ by

$$\langle X,Y,I\rangle/a \;:=\; \langle X,Y,a\rightarrow I\rangle.$$

$\langle X,Y,I\rangle/a$ will be called the factorized context of $\langle X,Y,I\rangle$ by threshold a. That is, $\langle X,Y,I\rangle/a$ has the same objects and attributes as $\langle X,Y,I\rangle$, and the incidence relation of $\langle X,Y,I\rangle/a$ is $a \rightarrow I$. Since

$$(a\rightarrow I)(x,y) \;=\; a\rightarrow I(x,y),$$

which makes computing $\langle X,Y,I\rangle/a$ from $\langle X,Y,I\rangle$ easy. Note that objects and attributes are more similar in $\langle X,Y,I\rangle/a$ than in the original context $\langle X,Y,I\rangle$. Indeed, for any $x_1,x_2 \in X$ and $y_1,y_2 \in Y$ one can easily verify that

$$I(x_1,y_1) \leftrightarrow I(x_1,y_1) \le (a \to I)(x_1,y_1) \leftrightarrow (a \to I)(x_2,y_2)$$

which intuitively says that in the factorized context, the table entries are more similar (closer) than in the original one.
The following is our main theorem.

Theorem 8 *For a formal fuzzy context $\langle X,Y,I\rangle$ and a threshold $a \in L$ we have*

$$\mathcal{B}\left(X,Y,I\right)/{}^a\!\approx \;\cong\; \mathcal{B}(\langle X,Y,I\rangle/a).$$

In words, $\mathcal{B}\left(X,Y,I\right)/{}^a\!\approx$ is isomorphic to $\mathcal{B}(\langle X,Y,I\rangle/a)$. Moreover, under the isomorphism, $[\langle A_1,B_1\rangle,\langle A_2,B_2\rangle] \in \mathcal{B}\left(X,Y,I\right)/{}^a\!\approx$ corresponds to $\langle A_2,B_1\rangle \in \mathcal{B}(\langle X,Y,I\rangle/a)$.

PROOF. Let ${}^\Uparrow$ and ${}^\Downarrow$ denote the operators induced by I (see 1.1 and 1.2) and ${}^{\Uparrow_a}$ and ${}^{\Downarrow_a}$ denote the operators induced by $a \to I$, that is, for $A \in L^X$ and $B \in L^Y$ we have

$$A^{\Uparrow_a}(y) = \bigwedge_{x\in X} A(x) \to (a \to I)(x,y),$$
$$B^{\Downarrow_a}(y) = \bigwedge_{y\in Y} B(y) \to (a \to I)(x,y).$$

Take any $A \in L^X$. Then $A^{\Uparrow_a}(y) = \bigwedge_{x\in X}(A(x) \to (a \to I(x,y))) = \bigwedge_{x\in X}(a \to (A(x) \to I(x,y))) = a \to \bigwedge_{x\in X}(A(x) \to I(x,y)) = a \to A^{\Uparrow}(x)$, and $A^{\Uparrow_a\Downarrow_a}(x) = \bigwedge_{y\in Y}(A^{\Uparrow_a}(y) \to (a \to I(x,y))) = \bigwedge_{y\in Y}(a \to (A^{\Uparrow_a}(y) \to I(x,y))) = a \to \bigwedge_{y\in Y}(A^{\Uparrow_a}(y) \to I(x,y)) = a \to \bigwedge_{y\in Y}([\bigwedge_{x\in X}(a \to (A(x) \to I(x,y)))] \to I(x,y)) = a \to \bigwedge_{y\in Y}([\bigwedge_{x\in X}((a\otimes A(x)) \to I(x,y))] \to I(x,y)) = a \to \bigwedge_{y\in Y}((a\otimes A)^{\Uparrow}(x) \to I(x,y)) = a \to (a\otimes A)^{\Uparrow\Downarrow}(x)$, i.e.

$$A^{\Uparrow_a} = a \to A^{\Uparrow} \quad \text{and} \quad A^{\Uparrow_a\Downarrow_a} = a \to (a\otimes A)^{\Uparrow\Downarrow}. \tag{2.1}$$

Now, let $[\langle A_1,B_1\rangle,\langle A_2,B_2\rangle] \in \mathcal{B}\left(X,Y,I\right)/{}^a\!\approx$. Using Lemma 2 and Lemma 5, there is $\langle A,B\rangle \in \mathcal{B}\left(X,Y,I\right)$ such that $\langle A_1,B_1\rangle = \langle A,B\rangle_a = \langle (a\otimes A)^{\Uparrow\Downarrow}, a \to B\rangle$ and $\langle A_2,B_2\rangle = (\langle A,B\rangle_a)^a = \langle a \to (a\otimes A)^{\Uparrow\Downarrow}, (a\otimes(a \to B))^{\Downarrow\Uparrow}\rangle$. Since $\langle A,B\rangle = \langle A,A^{\Uparrow}\rangle$, (2.1) yields $A_2 = a \to (a\otimes A)^{\Uparrow\Downarrow} = A^{\Uparrow_a\Downarrow_a}$ and $B_1 = a \to B = a \to A^{\Uparrow} = A^{\Uparrow_a}$. This shows $\langle A_2,B_1\rangle \in \mathcal{B}\left(X,Y,a \to I\right) = \mathcal{B}(\langle X,Y,I\rangle/a)$.
Conversely, if $\langle A_2,B_1\rangle \in \mathcal{B}(\langle X,Y,I\rangle/a)$ then using (2.1), $B_1 = A_2^{\Uparrow_a} = a \to A_2^{\Uparrow}$ and $A_2 = A_2^{\Uparrow_a\Downarrow_a} = a \to (a\otimes A_2)^{\Uparrow\Downarrow}$. By Lemma 2 and Lemma 5, $[\langle B_1^{\Downarrow},B_1\rangle,\langle A_2,A_2^{\Uparrow}\rangle] \in \mathcal{B}\left(X,Y,I\right)/{}^a\!\approx$. The proof is complete. □

Remark 1 *As we have seen, the blocks of $\mathcal{B}(X,Y,I)\,/^a\!\approx$ can be reconstructed from the formal concepts of $\mathcal{B}(\langle X,Y,I\rangle/a)$:*
If $\langle A,B\rangle \in \mathcal{B}(\langle X,Y,I\rangle/a)$ then $[\langle B^{\Downarrow}, B\rangle, \langle A, A^{\Uparrow}\rangle] \in \mathcal{B}(X,Y,I)\,/^a\!\approx$.

Computing $\mathcal{B}(\langle X,Y,I\rangle/a)$ means computing of the ordinary fuzzy concept lattice. This can be done by an algorithm of polynomial time delay complexity, see [9] or previous section.
Finally, lets look at the connection to the approach by the closure operator presented in previous section. From Theorem 8 and the previous remark we can see that the extents of concepts of $\mathcal{B}(\langle X,Y,I\rangle/a)$ are precisely the extents of suprema of $^a\!\approx$-blocks of $\mathcal{B}(X,Y,I)\,/^a\!\approx$. In previous section we have discovered that these extents in turn are precisely the fixed points of the fuzzy closure operator C_a. The connection lies in the correspondence of the composite mapping of the operators induced by $a \to I$ (denoted $^{\Uparrow_a}$ and $^{\Downarrow_a}$ in the proof of Theorem 8) and the fuzzy closure operator C_a; i.e. for any $A \in L^X$ we have

$$A^{\Uparrow_a \Downarrow_a} = a \to (a \otimes A)^{\Uparrow\Downarrow} = C_a(A).$$

2.3 Factorization of $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)$

Denote the binary fuzzy relation $\approx$ on $\mathcal{B}(X,Y,I)$ defined over objects by $\approx_{\mathrm{Ext}}$ and defined over attributes by $\approx_{\mathrm{Int}}$, i.e.

$$(\langle A_1,B_1\rangle \approx_{\mathrm{Ext}} \langle A_2,B_2\rangle) = \bigwedge_{x\in X} (A_1(x) \leftrightarrow A_2(x)) \tag{2.2}$$

$$(\langle A_1,B_1\rangle \approx_{\mathrm{Int}} \langle A_2,B_2\rangle) = \bigwedge_{y\in Y} (B_1(y) \leftrightarrow B_2(y)). \tag{2.3}$$

From section 1.2.3 we already know that $\approx$ is a fuzzy equivalence relation on $\mathcal{B}(X,Y,I)$ and that measuring similarity of formal concepts of $\mathcal{B}(X,Y,I)$ via intents B_i coincides with measuring similarity via extents A_i. As a consequence, we write also just $\approx$ instead of $\approx_{\mathrm{Ext}}$ and $\approx_{\mathrm{Int}}$.
This all said, applies to $\approx$ on $\mathcal{B}(X,Y,I)$. In this section we explore the use of $\approx$ (or the appropriate modification) with $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)$. We are questioning whether concept lattices with hedges can also be factorized by a similarity relation.

2.3.1 Similarity compatible with a hedge

Note first that one cannot directly apply the approach which works for $\mathcal{B}(X,Y,I)$. Namely, due to employing hedges, some important properties are no longer available (for instance, the composite mappings $^{\uparrow\downarrow}$ and $^{\downarrow\uparrow}$ are not fuzzy closure, nor even closure, operators in general, or $\approx$ is not compatible any more, see the following remark). Nevertheless, we propose a feasible approach to factorization of concept lattices with hedges. In some cases, however, we restrict ourselves to the case when one of the hedges is identity and leave the fully general case to future investigation. Note that in $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ corresponding to both "one-sided" fuzzy concept lattices, see [43] and [34], one of the hedges is globalization and the other is identity.

Remark 2 *If one would define $\approx_{\rm Ext}$ or $\approx_{\rm Int}$ on $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ simply by (2.2) or (2.3), compatibility would be lost. This is still true even if one of the hedges is identity. Consider e.g. ${}^*_X = \mathrm{id}_L$. Then, $\approx_{\rm Ext}$ is compatible with $\bigwedge$. Namely, $\bigwedge_{j\in J} A_j = \bigcap_{j\in J} A_j$ for $A_j = A_j^{\Uparrow\Downarrow}$ [17]. However, $\approx_{\rm Ext}$ need not be compatible with $\bigvee$ as shown by the following example. The dual situation applies to $\approx_{\rm Int}$.*

Example 1 *Take a Łukasiewicz structure on $[0,1]$, let ^{*_X} be identity and ^{*_Y} be globalization, and consider the following data table*

I	y_1	y_2	y_3
x_1	1	0.5	0
x_2	0	0	1
x_3	0.5	1	0

One can check that for $A_1 = \{{}^{0.5}\!/x_1,\ {}^{0.5}\!/x_3\}$, $B_1 = \{{}^{1}\!/y_1,\ {}^{1}\!/y_2,\ {}^{0.5}\!/y_3\}$, $A_2 = \{{}^{0.5}\!/x_1,\ {}^{1}\!/x_3\}$, $B_2 = \{{}^{0.5}\!/y_1,\ {}^{1}\!/y_2\}$, $A_3 = \{{}^{1}\!/x_1,\ {}^{0.5}\!/x_3\}$ and $B_3 = \{{}^{1}\!/y_1,\ {}^{0.5}\!/y_2\}$,

(1) $\langle A_i, B_i\rangle \in \mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, $i = 1,2,3$,

(2) $\langle A_1, B_1\rangle {}^a\!\approx \langle A_2, B_2\rangle$ and $\langle A_1, B_1\rangle {}^a\!\approx \langle A_3, B_3\rangle$,

(3) $(\langle A_1, B_1\rangle \wedge \langle A_1, B_1\rangle) = \langle A_1, B_1\rangle {}^a\!\approx \langle A_1, B_1\rangle = (\langle A_2, B_2\rangle \wedge \langle A_3, B_3\rangle)$, but

(4) $a \not\leq (\langle A_1, B_1\rangle \vee \langle A_1, B_1\rangle) \approx (\langle A_2, B_2\rangle \vee \langle A_3, B_3\rangle)$.

In order to propose our way to factorize $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, we need the following notion. Let $\approx$ be a fuzzy relation in $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, $a \in L$ be a truth degree, and * be a hedge (particularly, * will be ^{*_X} or ^{*_Y}). We say that $\approx$ is compatible with * and a if for each $c_1, c_2 \in \mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ we have that

$$\text{if } a \leq (c_1 \approx c_2), \text{ then } a \leq (c_1 \approx c_2)^*. \tag{2.4}$$

Consider the following fuzzy relations on $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$: By $\approx_{\rm Ext}$ and $\approx_{\rm Int}$ we denote the fuzzy relations defined by (2.2) and (2.3), respectively, and by $\approx^{*_X}_{\rm Ext}$ and $\approx^{*_Y}_{\rm Int}$ we denote fuzzy relations defined by

$$(\langle A_1, B_1\rangle \approx^{*_X}_{\rm Ext} \langle A_2, B_2\rangle) = \left(\bigwedge_{x\in X} (A_1(x) \leftrightarrow A_2(x))\right)^{*_X} \tag{2.5}$$

$$(\langle A_1, B_1\rangle \approx^{*_Y}_{\rm Int} \langle A_2, B_2\rangle) = \left(\bigwedge_{y\in Y} (B_1(y) \leftrightarrow B_2(y))\right)^{*_Y}. \tag{2.6}$$

Occasionally, we write also $(A_1 \approx^{*_X}_{\mathrm{Ext}} A_2)$ instead of $(\langle A_1, B_1\rangle \approx^{*_X}_{\mathrm{Ext}} \langle A_2, B_2\rangle)$, etc.
The following assertion is easy to see.

Lemma 9 *(1) If $a \in L$ is a fixed point of ^{*_X}, i.e. $a^{*_X} = a$, then $\approx_{\mathrm{Ext}}$ is compatible with ^{*_X} and a; similarly for ^{*_Y} and $\approx_{\mathrm{Int}}$.*
(2) For any $a \in L$, $\approx^{*_X}_{\mathrm{Ext}}$ is compatible with ^{*_X} and a; similarly for ^{*_Y} and $\approx^{*_Y}_{\mathrm{Int}}$.

PROOF. Due to similarity, we prove only the case of ^{*_X} and $\approx_{\mathrm{Ext}}$.
(1) If, for any $c_1, c_2 \in \mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, it holds $a \leq (c_1 \approx_{\mathrm{Ext}} c_2)$, then from the monotonicity of hedges (easily provable from (iii) of the definition of a hedge in section 1.2.1) we directly get $a = a^{*_X} \leq (c_1 \approx_{\mathrm{Ext}} c_2)^{*_X}$;
(2) Again directly from the idempotency of hedges ((iv) of the definition of a hedge). □

We need the following two assertions (here, $\approx$ is defined by $(A_1 \approx A_2) = \bigwedge_{x \in X}(A_1(x) \leftrightarrow A_2(x))$).

Lemma 10 *Let $A_1, A_2 \in L^X$. Then $(A_1 \approx A_2)^{*_X} \leq (A_1^{*_X} \approx A_2^{*_X})$.*

PROOF. Denote ^{*_X} by *. We have $(A_1 \approx A_2)^* \leq (A_1^* \approx A_2^*) = \bigwedge_{x\in X}(A_1(x)^* \leftrightarrow A_2(x)^*)$ iff $(A_1 \approx A_2)^* \leq (A_1(x)^* \leftrightarrow A_2(x)^*)$ for all $x \in X$. Since $(A_1 \approx A_2)^* \leq (A_1(x) \leftrightarrow A_2(x))^*$ for all $x \in X$ it suffices to show $(A_1(x) \leftrightarrow A_2(x))^* \leq (A_1(x)^* \leftrightarrow A_2(x)^*)$, which is true. Indeed, $(A_1(x) \leftrightarrow A_2(x))^* \leq (A_1(x) \rightarrow A_2(x))^* \wedge (A_2(x) \rightarrow A_1(x))^* \leq (A_1(x)^* \rightarrow A_2(x)^*) \wedge (A_2(x)^* \rightarrow A_1(x)^*) = (A_1(x)^* \leftrightarrow A_2(x)^*)$. □

Lemma 11 *For $A_1, A_2 \in L^X$ we have $(A_1 \approx A_2)^{*_X} \leq (A_1^{\uparrow} \approx A_2^{\uparrow})$.*

PROOF. Follows directly from Lemma 10 and $(A_1 \approx A_2) \leq (A_1^{\Uparrow} \approx A_2^{\Uparrow})$ [3]. □

Suppose we have two fuzzy equivalence relations on $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, $\approx_X$ and $\approx_Y$ such that $\approx_X$ is compatible with ^{*_X} and a, and $\approx_Y$ is compatible with ^{*_Y} and a. Although, in general, $\approx_X$ may be different from $\approx_Y$, the following theorem shows that their a-cuts coincide.

Theorem 12 *Let $\approx_X$ and $\approx_Y$ be fuzzy equivalence relations on $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ compatible with ^{*_X} and a, and with ^{*_Y} and a, respectively. Then ${}^a{\approx_X} = {}^a{\approx_Y}$.*

PROOF. Using Lemma 11, the proof is similar to the proof of $\approx_{\mathrm{Ext}}=\approx_{\mathrm{Int}}$ in [3]. $\square$

We can therefore write ${}^{a}{\approx}$ instead of ${}^{a}{\approx}_X$ and ${}^{a}{\approx}_Y$. Note that Theorem 12 applies in particular to the fuzzy relations from Lemma 9. With the above notation, the following theorem shows a way to factorize $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)$.

Theorem 13 *If $\approx$ is compatible with ^{*_X} and a, and with ^{*_Y} and a, then ${}^{a}{\approx}$ is a compatible tolerance relation on $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)$.*

PROOF. Theorem can be proved by applying (2.4) and Lemma 11 twice at the end of the proof of compatibility of $\approx$ on $\mathcal{B}\,(X, Y, I)$ in [3]. $\square$

Therefore, we can consider the factor lattice $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)\,/{}^{a}{\approx}$ of lattice $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)$ by tolerance ${}^{a}{\approx}$. The construction is the same as for $\mathcal{B}\,(X, Y, I)$ described in section 1.2.3 or in detail in [3]. In what follows, we present a way to obtain $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)\,/{}^{a}{\approx}$ directly, without the need to compute $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)$ first and then to compute the blocks of ${}^{a}{\approx}$. Basically, we follow and appropriately modify the methods from sections 2.2.1 and 2.2.2.

2.3.2 The $\mathbf{L}_{\{1\}}$-closure operator

Actually, the method from section 2.2.1 makes use of the fact that for each fuzzy set A in U we have

$$\langle A, a \otimes A\rangle \in {}^{a}{\approx} \text{ and } \langle A, a \rightarrow A\rangle \in {}^{a}{\approx}. \tag{2.7}$$

Here, ${}^{a}{\approx}$ is defined by $\langle A_1, A_2\rangle \in {}^{a}{\approx}$ iff $a \leq \bigwedge_{u\in U}(A_1(x) \leftrightarrow A_2(x))$. If ${}^{a}{\approx}$ has this feature, we can proceed for fast factorization of $\mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)$ by ${}^{a}{\approx}$. Note that (2.7) is satisfied, for instance, for ${}^{a}{\approx}_{\mathrm{Ext}}$ or ${}^{a}{\approx}_{\mathrm{Int}}$ if a is a fixed point of ^{*_X} or ^{*_Y}, respectively, cf. Lemma 9. In the remainder of the section we will suppose that ${}^{a}{\approx}$ always satisfies (2.7). The following assertion shows that $\langle A, B\rangle_a$ (the least formal concept ${}^{a}{\approx}$-similar to $\langle A, B\rangle$) and $\langle A, B\rangle^a$ (the greatest formal concept ${}^{a}{\approx}$-similar to $\langle A, B\rangle$) can be computed from $\langle A, B\rangle$ directly, cf. Lemma 5.

Lemma 14 *For $\langle A, B\rangle \in \mathcal{B}\,(X^{*_X}, Y^{*_Y}, I)$, we have*

(a) $\langle A, B\rangle_a = \langle (a \otimes A)^{\uparrow\downarrow}, (a \rightarrow B)^{\downarrow\uparrow}\rangle$ *and*

(b) $\langle A, B\rangle^a = \langle (a \rightarrow A)^{\uparrow\downarrow}, (a \otimes B)^{\downarrow\uparrow}\rangle$.

PROOF. Due to duality we sketch only the proof of (a). We need to prove, that $(a \otimes A)^{\uparrow\downarrow}$ is an extent of the least formal concept similar to $\langle A, B\rangle$ to degree at least a and $(a \rightarrow B)^{\downarrow\uparrow}$ is the corresponding intent. That is (1)

$(a \otimes A)^{\uparrow\downarrow}$ is an extent of a formal concept $\langle (a \otimes A)^{\uparrow\downarrow}, D\rangle$ which is similar to $\langle A, B\rangle$ to degree at least a; (2) if $\langle C, F\rangle$ is a formal concept similar to $\langle A, B\rangle$ to degree at least a then $\langle (a \otimes A)^{\uparrow\downarrow}, D\rangle \leq \langle C, F\rangle$; and similarly for intent $(a \to B)^{\downarrow\uparrow}$. Both (1) and (2) can be easily proved using (2.7), (2.4), Lemma 11 and adjointness property, see the proof of Lemma 5 for details. $\square$

Remark 3 *Thus we have* $(\langle A, B\rangle_a)^a = \langle (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}, (a \otimes (a \to B)^{\downarrow\uparrow})^{\downarrow\uparrow}\rangle$.

Another property, analogous to the case of $\mathcal{B}(X, Y, I)$, is the following.

Lemma 15 *If* ^{*_X} *is identity on* L *and* A *is an extent then we have* $a \to A = (a \to A)^{\uparrow\downarrow}$*; similarly for* ^{*_Y} *and an intent* B*.*

PROOF. We sketch the proof for extents. The inequality $\subseteq$ follows directly from $A = A^{*_X} \subseteq A^{\uparrow\downarrow}$ and the converse inequality $\supseteq$ can be proved the same way as the corresponding inequality in the analogous Lemma 4, with application of (2.7), (2.4) and Lemma 11 at appropriate places. $\square$

One way to obtain the factor lattice directly is based on the following theorem. Recall that an $\mathbf{L}_{\{1\}}$-closure operator in a set U [4] is a mapping $C : A \to C(A)$ satisfying (1) $A \subseteq C(A)$, (2) $A_1 \subseteq A_2$ implies $C(A_1) \subseteq C(A_2)$, (3) $C(A) = C(C(A))$ for $A, A_1, A_2 \in L^U$. A fixed point of C is any fuzzy set A in U such that $A = C(A)$.

Theorem 16 *Let* ^{*_X} *be identity on* L*. Then the mapping* $C_a : A \mapsto (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}$ *is an* $\mathbf{L}_{\{1\}}$*-closure operator in* $\mathbf{L}^X$ *such that the fixed points of* C_a *are just the extents of suprema of* $^a\!\approx$*-blocks of* $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)/^a\!\approx$*.*

PROOF. The idea of the proof remains the same as in the proof of analogous Theorem 7 for $\mathcal{B}(X, Y, I)/^a\!\approx$. Briefly, we need to (1) verify that C_a is an $\mathbf{L}_{\{1\}}$-closure operator and (2) prove the equality of the set of fixed points of C_a and the set of extents of suprema of $^a\!\approx$-blocks.

First, we verify that C_a is an $\mathbf{L}_{\{1\}}$-closure operator. $A \subseteq C_a(A)$ means $A \subseteq (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}$ which simplifies to $A \subseteq a \to (a \otimes A)^{\Uparrow\downarrow}$ due to $^{*_X} = \mathrm{id}_L$ and Lemma 15. The latter is equivalent (by adjointness) to $a \otimes A \subseteq (a \otimes A)^{\Uparrow\downarrow}$ which is true since $E \subseteq E^{\Uparrow\Downarrow} \subseteq E^{\Uparrow\downarrow}$ is always the case. We showed $A \subseteq C_a(A)$.

$A_1 \subseteq A_2$ implies $C(A_1) \subseteq C(A_2)$: Since for $D_1, D_2 \in L^U$, $D_1 \subseteq D_2$ implies $(a \otimes D_1) \subseteq (a \otimes D_2)$, $D_1 \subseteq D_2$ implies $(a \to D_1) \subseteq (a \to D_2)$ and $D_1 \subseteq D_2$ implies $D_1^{\uparrow\downarrow} \subseteq D_2^{\uparrow\downarrow}$ [12], by chaining we have $A_1 \subseteq A_2$ implies $C(A_1) \subseteq C(A_2)$.

To verify $C_a(A) = C_a(C_a(A))$, we will make use of the previous two already proven conditions. We already know that if we consider ${}^{*_X} = \mathrm{id}_L$, $C_a(A) = a \to (a \otimes A)^{\Uparrow\Downarrow}$. Using $A \subseteq C_a(A)$ we have $(a \otimes A)^{\Uparrow\Downarrow} \subseteq (a \otimes (a \to (a \otimes A)^{\Uparrow\Downarrow}))^{\Uparrow\Downarrow}$ which implies $a \to (a \otimes A)^{\Uparrow\Downarrow} \subseteq a \to (a \otimes (a \to (a \otimes A)^{\Uparrow\Downarrow}))^{\Uparrow\Downarrow}$, and since $a \otimes (a \to b) \leq b$, we get

$$\begin{aligned} & a \to (a \otimes A)^{\Uparrow\Downarrow} \subseteq a \to (a \otimes (a \to (a \otimes A)^{\Uparrow\Downarrow}))^{\Uparrow\Downarrow} \subseteq \\ & \subseteq a \to ((a \otimes A)^{\Uparrow\Downarrow})^{\Uparrow\Downarrow} = a \to (a \otimes A)^{\Uparrow\Downarrow}. \end{aligned}$$

This proves $a \to (a \otimes A)^{\Uparrow\Downarrow} = a \to (a \otimes (a \to (a \otimes A)^{\Uparrow\Downarrow}))^{\Uparrow\Downarrow}$ and so $C_a(A) = C_a(C_a(A))$.
Second, we verify that the set of fixed points of C_a equals the set of extents of suprema of ${}^a\!\approx$-blocks. Let A be a fixed point of C_a. By Lemma 2, the interval $[\langle A, A^\uparrow\rangle_a, (\langle A, A^\uparrow\rangle_a)^a]$ is a ${}^a\!\approx$-block, and by Lemma 14, $(\langle A, A^\uparrow\rangle_a)^a = \langle (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}, \dots\rangle$. Since $A = C_a(A) = (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}$, A is the extent of a supremum of a ${}^a\!\approx$-block. Conversely, let A be a supremum of a ${}^a\!\approx$-block. Then $[\langle A, A^\uparrow\rangle_a, \langle A, A^\uparrow\rangle]$ is an ${}^a\!\approx$-block and so $(\langle A, A^\uparrow\rangle_a)^a = \langle A, A^\uparrow\rangle$. Lemma 14 now gives $A = (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}$, i.e. $A = C_a(A)$ verifying A being a fixed point of C_a, finishing the proof. $\square$

Remark 4 *$C_a : A \mapsto a \to (a \otimes A)^{\Uparrow\Downarrow}$ is an $\mathbf{L}_{\{1\}}$-closure operator, but not an $\mathbf{L}$-closure operator in general, since we do not have $S(A_1, A_2) \leq S(C_a(A_1), C_a(A_2))$ for all $A_1, A_2 \in L^X$ as the following example shows.*

Example 2 *Consider the setting and data table from Example 1. Take $A_1 = \{\,{}^{0.5}/x_1,\ {}^{1}/x_2,\ {}^{0.5}/x_3\}$ and $A_2 = \{\,{}^{1}/x_2\}$. One can check that given $a = 1$, $C_a(A_1) = A_1^{\Uparrow\Downarrow} = \{\,{}^{1}/x_1,\ {}^{1}/x_2,\ {}^{1}/x_3\}$ and $C_a(A_2) = A_2^{\Uparrow\Downarrow} = \{\,{}^{1}/x_2\}$, hence $0.5 = S(A_1, A_2) \not\leq S(C_a(A_1), C_a(A_2)) = 0$.*

Now, fixed points of $\mathbf{L}_{\{1\}}$-closure operators can be efficiently computed by an extension of Ganter's NextClosure algorithm, see [9]. The extension was recalled at the end of section 2.2.1. Note that the algorithm described there is formulated in terms of a fuzzy closure operator. But looking closely at the algorithm and its proof of correctness (see [9]), we reveal that it in fact computes the fixed points of an $\mathbf{L}_{\{1\}}$-closure operator, not only $\mathbf{L}$-closure operator. The reason is that for the algorithm to work, the property of graded subsethood from the definition of fuzzy closure operator is not necessary and crisp subsethood suffices. We only need a small modification which is due to the use of hedges. The only modification is in the definition of $<_{(i,j)}$ (see the description of the algorithm in section 2.2.1). For $A, B \in L^X$, put

$$\begin{aligned} & A <_{(i,j)} B \text{ iff} \\ & \quad A^{*_X} \cap \{1, \dots, i-1\} = B^{*_X} \cap \{1, \dots, i-1\} \text{ and} \\ & \quad\quad A^{*_X}(i) < B^{*_X}(i) = a_j. \end{aligned}$$

The whole rest of the algorithm remains untouched, including the complexity of the algorithm. Note that to compute the concept lattice $\mathcal{B}\left(X, Y^{*_Y}, I\right)/^a\!\approx$ (i.e. ^{*_X} is identity on L) we have to use the version of the algorithm iterating over attributes in order to utilize the hedge ^{*_Y}.

2.3.3 Factorized context

Another way to obtain the factor lattice directly (from input data, without computing first the whole concept lattice and then computing the similarity blocks) is based on the idea of factorized context proposed in section 2.2.2. Recall that the factorized context $\langle X, Y, I\rangle/a$ of $\langle X, Y, I\rangle$ by (user-specified) threshold a is defined by

$$\langle X, Y, I\rangle/a \;:=\; \langle X, Y, a \to I\rangle,$$

that is, $\langle X, Y, I\rangle/a$ has the same objects and attributes as $\langle X, Y, I\rangle$, and the incidence relation of $\langle X, Y, I\rangle/a$ is $a \to I$. The following is the generalization of Theorem 8 for the case of $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)/^a\!\approx$.

Theorem 17 *If ^{*_X} is identity on L then for any $\langle X, Y, I\rangle$ and a threshold $a \in L$ we have*

$$\mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)/^a\!\approx \;\cong\; \mathcal{B}\left(X^{*_X}, Y^{*_Y}, a \to I\right).$$

In words, $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)/^a\!\approx$ is isomorphic to $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, a \to I\right)$. Moreover, under the isomorphism, $[\langle A_1, B_1\rangle, \langle A_2, B_2\rangle] \in \mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)/^a\!\approx$ corresponds to $\langle A_2, B_1\rangle \in \mathcal{B}\left(X^{*_X}, Y^{*_Y}, a \to I\right)$.

PROOF. We proceed the same way as in the proof of Theorem 8.
Denote ^{*_X} by *. Let ${}^\uparrow$ and ${}^\downarrow$ denote the operators (1.6) and (1.7) induced by I and ${}^{\uparrow_a}$ and ${}^{\downarrow_a}$ denote the operators induced by $a \to I$. Take any $A \in L^X$. Then we have $A^{\uparrow_a}(y) = \bigwedge_{x\in X}(A^*(x) \to (a \to I(x,y))) = \bigwedge_{x\in X}(a \to (A^*(x) \to I(x,y))) = a \to \bigwedge_{x\in X}(A^*(x) \to I(x,y)) = a \to A^\uparrow(x)$, and $A^{\uparrow_a\downarrow_a}(x) = \bigwedge_{y\in Y}(A^{\uparrow_a *}(y) \to (a \to I(x,y))) = \bigwedge_{y\in Y}(a \to (A^{\uparrow_a *}(y) \to I(x,y))) = a \to \bigwedge_{y\in Y}(A^{\uparrow_a *}(y) \to I(x,y)) = a \to \bigwedge_{y\in Y}([\bigwedge_{x\in X}(a \to (A^*(x) \to I(x,y)))]^* \to I(x,y)) = a \to \bigwedge_{y\in Y}([\bigwedge_{x\in X}((a \otimes A^*(x)) \to I(x,y))]^* \to I(x,y)) = a \to \bigwedge_{y\in Y}((a \otimes A^*)^{\Uparrow *}(x) \to I(x,y)) = a \to (a \otimes A^*)^{\Uparrow\downarrow}(x)$, i.e.

$$A^{\uparrow_a} = a \to A^\uparrow \quad \text{and} \quad A^{\uparrow_a\downarrow_a} = a \to (a \otimes A^{*_X})^{\Uparrow\downarrow}. \tag{2.8}$$

Now, let $[\langle A_1, B_1\rangle, \langle A_2, B_2\rangle] \in \mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)/^a\!\approx$. By Lemmas 2 and 14, there is $\langle A, B\rangle \in \mathcal{B}\left(X^{*_X}, Y^{*_Y}, I\right)$ such that $\langle A_1, B_1\rangle = \langle A, B\rangle_a = \langle (a \otimes A)^{\uparrow\downarrow}, (a \to B)^{\downarrow\uparrow}\rangle$ and $\langle A_2, B_2\rangle = (\langle A, B\rangle_a)^a = \langle (a \to (a \otimes A)^{\uparrow\downarrow})^{\uparrow\downarrow}, (a \otimes (a \to B)^{\downarrow\uparrow})^{\downarrow\uparrow}\rangle$. If we further consider ${}^{*_X} = \mathrm{id}_L$, the expressions for the boundary concepts simplify to $\langle A_1, B_1\rangle = \langle A, B\rangle_a = \langle (a \otimes A)^{\Uparrow\downarrow}, a \to B\rangle$ and

$\langle A_2, B_2\rangle = ((\langle A, B\rangle_a)^a = \langle a \to (a\otimes A)^{\Uparrow\Downarrow}, (a\otimes(a \to B))^{\Downarrow\Uparrow}\rangle$, due to Lemma 15. Since $\langle A, B\rangle = \langle A, A^{\Uparrow}\rangle$, (2.8) yields $A_2 = a \to (a \otimes A)^{\Uparrow\Downarrow} = A^{\uparrow_a\downarrow_a}$ and $B_1 = a \to B = a \to A^{\Uparrow} = A^{\uparrow_a}$. This shows $\langle A_2, B_1\rangle \in \mathcal{B}(X, Y^{*_Y}, a \to I)$. Conversely, if $\langle A_2, B_1\rangle \in \mathcal{B}(X, Y^{*_Y}, a \to I)$ then using (2.1), $B_1 = A_2^{\uparrow_a} = a \to A_2^{\Uparrow}$ and $A_2 = A_2^{\uparrow_a\downarrow_a} = a \to (a \otimes A_2)^{\Uparrow\Downarrow}$. By Lemma 2 and Lemma 14, $[\langle B_1^{\Downarrow}, B_1\rangle, \langle A_2, A_2^{\Uparrow}\rangle] \in \mathcal{B}(X, Y^{*_Y}, I) /{}^a\!\approx$. The proof is complete.

□

Remark 5 *The blocks of $\mathcal{B}(X, Y^{*_Y}, I) /{}^a\!\approx$ can be reconstructed from the formal concepts of $\mathcal{B}(X, Y^{*_Y}, a \to I)$:*
If $\langle A, B\rangle \in \mathcal{B}(X, Y^{*_Y}, a \to I)$ then $[\langle B^{\Downarrow}, B\rangle, \langle A, A^{\Uparrow}\rangle] \in \mathcal{B}(X, Y^{*_Y}, I) /{}^a\!\approx$.

Computing $\mathcal{B}(X, Y^*_Y, a \to I)$ means computing the fuzzy concept lattice with hedges, where the hedge ^{*_X} is identity. This can be done by an algorithm of polynomial time delay complexity, see [9] or previous section.
Finally, we mention the connection to the approach by the closure operator presented in previous section, similarily as in the case without hedges in section 2.2.2. In both approaches we are restricted to the case $^{*_X} = \mathrm{id}_L$. From Theorem 17 and the previous remark we have that the extents of concepts of $\mathcal{B}(X, Y^{*_Y}, a \to I)$ are precisely the extents of suprema of ${}^a\!\approx$-blocks of $\mathcal{B}(X, Y^{*_Y}, I) /{}^a\!\approx$ and from previous section we have that these extents are precisely the fixed points of the $\mathbf{L}_{\{1\}}$-closure operator C_a. The connection, without surprise, is analogical – the correspondence of the composite mapping of the operators induced by $a \to I$ (denoted $^{\uparrow_a}$ and $^{\downarrow_a}$ in the proof of Theorem 17) and the $\mathbf{L}_{\{1\}}$-closure operator C_a; i.e. for any $A \in L^X$ we have

$$A^{\uparrow_a\downarrow_a} = a \to (a \otimes A)^{\Uparrow\Downarrow} = C_a(A).$$

2.4 Examples and experiments

The aim of this section is to demonstrate experimentally the effect of reduction of size of a fuzzy concept lattice (with hedges) by factorization by similarity, and, especially, the speed-up of our algorithms of fast and direct factorization. By reduction of size of a fuzzy concept lattice (with hedges) given by a data table $\langle X, Y, I\rangle$ with fuzzy attributes and a user-specified threshold a, we mean the ratio

$$\frac{|\mathcal{B}(X, Y, I) /{}^a\!\approx|}{|\mathcal{B}(X, Y, I)|} \qquad \left(\frac{|\mathcal{B}(X^{*_X}, Y^{*_Y}, I) /{}^a\!\approx|}{|\mathcal{B}(X^{*_X}, Y^{*_Y}, I)|}\right)$$

of the number $|\mathcal{B}(X, Y, I) /{}^a\!\approx|$ (resp. $|\mathcal{B}(X^{*_X}, Y^{*_Y}, I) /{}^a\!\approx|$) of elements of $\mathcal{B}(X, Y, I) /{}^a\!\approx$ (resp. $\mathcal{B}(X^{*_X}, Y^{*_Y}, I) /{}^a\!\approx$), i.e. the number of elements of the factor lattice, to the number $|\mathcal{B}(X, Y, I)|$ (resp. $|\mathcal{B}(X^{*_X}, Y^{*_Y}, I)|$) of

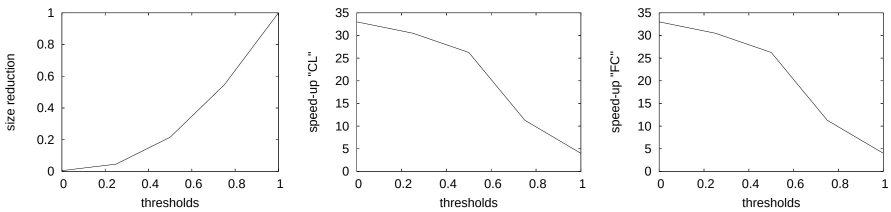


Figure 2.1: Size reduction and speed-ups from Tab. 2.2.

elements of $\mathcal{B}(X, Y, I)$ (resp. $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$), i.e. the number of elements of the original lattice. By a speed-up we mean the ratio of the time for computing the factor lattice by a naive algorithm to the time for computing the factor lattice by our algorithm. By "our algorithm" we mean the algorithms of fast and direct factorization described in the end of sections 2.2.1, 2.3.2 (fuzzy and $\mathbf{L}_{\{1\}}$-closure operator) and 2.2.2, 2.3.3 (factorized context without and with hedges). The algorithms exploiting a closure operator will be denoted "CL" in the following sections (tables, figures, etc.) presenting result of experiments; the algorithms computing a factor fuzzy concept lattice (with hedges) from a factorized context will be denoted "FC". By "naive algorithm" we mean computing the factor lattice by first generating the whole fuzzy concept lattice (with hedges) $\mathcal{B}(X, Y, I)$ (resp. $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$) by a polynomial time-delay algorithm (mentioned in previous sections; the same algorithm used for computing a factor lattice from a factorized context) and subsequently generating the ${}^a{\approx}$-blocks by producing the boundary concepts $[\langle A, B\rangle_a, (\langle A, B\rangle_a)^a]$.

2.4.1 Countries of EU

Consider the data table depicted in Tab. 2.1. The data table was introduced in section 1.2.3, there are 25 countries of EU (objects from X) described by some of their demographic and economic characteristics (attributes from Y). The original values of the characteristics are scaled to interval $[0, 1]$ so that the characteristics can be considered as fuzzy attributes with truth degrees from five element chain $L = \{0, 0.25, 0.5, 0.75, 1\}$.

Tab. 2.2 summarizes the results of factorization when using Łukasiewicz fuzzy logical operations, no hedges (identity) and threshold values $a = 0, 0.25, 0.5, 0.75, 1$ (recall that fator lattice for the case $a = 0$ is always an one-element trivial lattice and for the case $a = 1$ is always isomorphic to the whole concept lattice). Fig. 2.1 contains graphs depicting reduction $|\mathcal{B}(X, Y, I)\,/{}^a{\approx}|/|\mathcal{B}(X, Y, I)\,|$ and speed-ups from Tab. 2.2. Finally, for better illustration of size reduction, the factor lattices $\mathcal{B}(X, Y, I)\,/{}^a{\approx}$ are depicted in Fig. 2.2.

The example demonstrates that smaller thresholds lead to both larger size reduction and speed-up. We can see that the time needed for computing

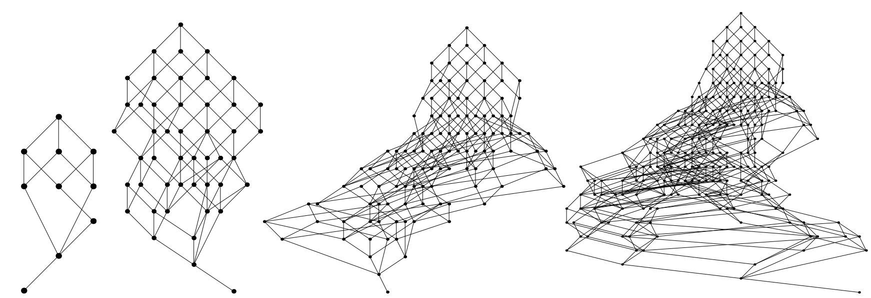
Table 2.1: Data table of EU countries, 5 truth degrees.

	a	b	c	d	e
1 Austria	0	0.25	0.5	1	1
2 Belgium	0	0	0.5	1	0.75
3 Cyprus	0	0	0.25	1	0.75
4 Czech rep.	0	0.25	0.25	0.75	0.75
5 Denmark	0	0	0.5	1	0.75
6 Estonia	0	0	0	0.5	0.5
7 Finland	0	0.5	0.5	0.75	0.5
8 France	0.75	1	0.5	1	0.75
9 Germany	1	0.75	0.5	1	0.75
10 Greece	0	0.25	0.25	0.75	0.5
11 Hungary	0	0.25	0.25	0.25	0.75
12 Ireland	0	0.25	0.5	0.75	1
13 Italy	0.75	0.5	0.5	1	0.5
14 Latvia	0	0	0	0.75	0.5
15 Lithuania	0	0	0	1	0.25
16 Luxembourg	0	0	1	1	1
17 Malta	0	0	0.25	0.75	0.75
18 Netherlands	0.25	0	0.5	0.5	1
19 Poland	0.5	0.5	0	0.5	0
20 Portugal	0	0.25	0.25	0.75	1
21 Slovakia	0	0	0	0	0
22 Slovenia	0	0	0.25	0.25	0.75
23 Spain	0.5	1	0.25	0.75	0.5
24 Sweden	0	0.75	0.5	0.75	0.75
25 UK	0.75	0.5	0.5	1	0.75

attributes: a – many habitants (millions), b – large area (thousands km^2), c – high GDP (EUR), d – low inflation (%), e – low unemployment (%)

Table 2.2: Results of factorization of $\mathcal{B}(X,Y,I)$ of data from Tab. 2.1, Łukasiewicz fuzzy logical connectives, no hedges (identity); $|\mathcal{B}(X,Y,I)| = 218$, time for computing $\mathcal{B}(X,Y,I) = 11$ ms.

thresholds	0	0.25	0.5	0.75	1
size $\lvert\mathcal{B}(X,Y,I)/^a\!\approx\rvert$	1	10	47	119	218
size reduction	0.005	0.046	0.216	0.546	1.000
naive algorithm (ms)	33	61	105	79	44
our algorithm “CL” (ms)	1	2	4	7	11
speed-up “CL”	33.00	30.50	26.25	11.29	4.00
our algorithm “FC” (ms)	1	2	4	7	11
speed-up “FC”	33.00	30.50	26.25	11.29	4.00



$a = 0.25$ $a = 0.5$ $a = 0.75$ $a = 1$

Figure 2.2: Factor lattices $\mathcal{B}(X,Y,I)/^a\!\approx$ of data from Tab. 2.1, Łukasiewicz fuzzy logical connectives, no hedges (identity).

Table 2.3: Results of factorization of $\mathcal{B}(X,Y,I)$ of data from Tab. 2.1, minimum-based fuzzy logical connectives, no hedges (identity); $|\mathcal{B}(X,Y,I)| = 166$, time for computing $\mathcal{B}(X,Y,I) = 7$ ms.

thresholds	0	0.25	0.5	0.75	1
size $\lvert\mathcal{B}(X,Y,I)/{}^a\!\approx\rvert$	1	10	27	79	166
size reduction	0.006	0.060	0.163	0.476	1.000
naive algorithm (ms)	18	17	17	19	23
our algorithm "CL" (ms)	1	2	2	4	7
speed-up "CL"	18.00	8.50	8.50	4.75	3.29
our algorithm "FC" (ms)	1	2	2	4	7
speed-up "FC"	18.00	8.50	8.50	4.75	3.29

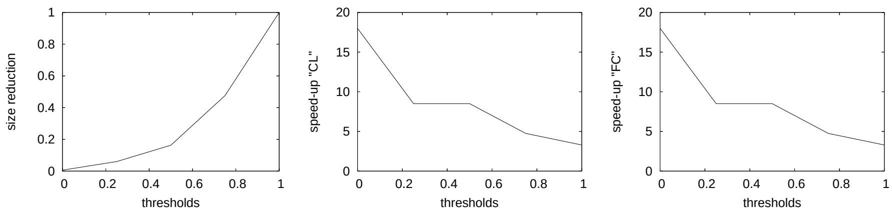
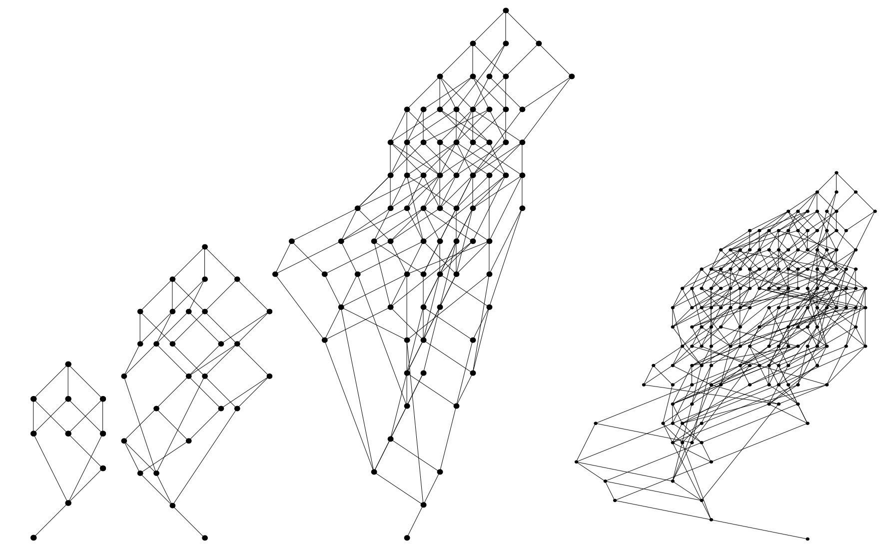


Figure 2.3: Size reduction and speed-ups from Tab. 2.3.

the factor lattice $\mathcal{B}(X,Y,I)/{}^a\!\approx$ by our algorithms is smaller than time needed for computing the original concept lattice $\mathcal{B}(X,Y,I)$ and, of course, much smaller than time needed for computing the factor lattice by the naive algorithm. For instance, for $a = 0.25$, time needed for computing $\mathcal{B}(X,Y,I)$ is 11 ms, computing the factor lattice $\mathcal{B}(X,Y,I)/{}^a\!\approx$ by the naive algorithm lasts long 61 ms, but our algorithms requires for the same task only 2 ms (both "CL" and "FC"), i.e. our algorithms are more than thirty times faster! The functionalities of size reduction and speed-up on the threshold values are roughly exponential. Note also that computing $\mathcal{B}(X,Y,I)/{}^a\!\approx$ using the naive algorithm, most of the time consumed is spent on factorization rather than computing $\mathcal{B}(X,Y,I)$: 61 ms is consumed in total of which 11 ms is spent on computing $\mathcal{B}(X,Y,I)$ and $50 = 61 - 11$ ms (82 %) is spent on factorization, i.e. on computing $\mathcal{B}(X,Y,I)/{}^a\!\approx$ from $\mathcal{B}(X,Y,I)$. Furthermore, we can see that both our algorithms are equally fast, from which we can conclude that the approaches of computing factor lattice either using new closure operator or first factorizing the context are of almost the same (or at least very similar) complexity (in the terms of time)[2].
Tab. 2.3, Fig. 2.3 and Fig. 2.4 show the same characteristics when using the minimum-based (Göedel) fuzzy logical operations instead of Łukasiewicz

[2]Formal time delay complexity is the polynomial time delay complexity of Ganter's NextClosure algorithm, see [9]



$a = 0.25$ $a = 0.5$ $a = 0.75$ $a = 1$

Figure 2.4: Factor lattices $\mathcal{B}(X, Y, I) / {}^{a}\!\approx$ of data from Tab. 2.1, minimum-based fuzzy logical connectives, no hedges (identity).

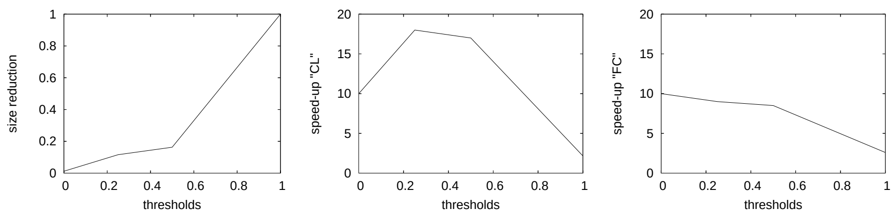


Figure 2.5: Size reduction and speed-ups from Tab. 2.4.

fuzzy logical operations. We can see the smaller speed-up, however, but still significant, our algorithms are almost twenty times faster than the naive algorithm. It seems factorizing fuzzy concept lattices built using minimum-based fuzzy logical operations is somehow ,,easier".

Now we look how the factorization works on fuzzy concept lattices with hedges. Recall from sections 2.3.2 and 2.3.3 that our algorithms are restricted on one hedge only (the other one has to be an identity). In all experiments involving hedges we are constraining attributes (constraining objects does not make much sense), thus we fix $*_X$ to be the identity on $\mathbf{L}$ and select non-identity truth-stressing hedges $*_Y$ on $\mathbf{L}$.

Let $*_{Y1}$ be a hedge defined as follows: for $a \in L$, $a^*_Y = 0.5$ if $a = 0.75$ and $a^*_Y = a$ otherwise. To measure the similarity of concepts we can use ${}^{a}\!\approx$, since $\approx$ is compatible with $*_Y$ and each fixed point $*_Y$ (Lemma 9) and ${}^{a}\!\approx$ satisfies (2.7). Thereby for threshold values we can use only the fixed points of $*_Y$, i.e. $a = 0, 0.25, 0.5, 1$. Tab. 2.4 summarizes the results

Table 2.4: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of data from Tab. 2.1, Łukasiewicz fuzzy logical connectives, hedge $*_{Y1}$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 86$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 5$ ms.

thresholds	0	0.25	0.5	1
size $\lvert\mathcal{B}(X, Y^{*_Y}, I)/{}^{a}\!\approx\rvert$	1	10	14	86
size reduction	0.012	0.116	0.163	1.000
naive algorithm (ms)	10	18	17	13
our algorithm “CL” (ms)	1	1	1	6
speed-up “CL”	10.00	18.00	17.00	2.17
our algorithm “FC” (ms)	1	2	2	5
speed-up “FC”	10.00	9.00	8.50	2.60

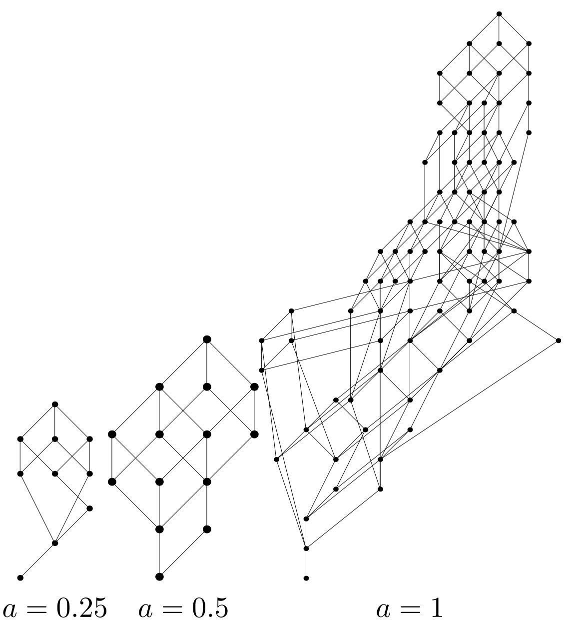


Figure 2.6: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)/{}^{a}\!\approx$ of data from Tab. 2.1, Łukasiewicz fuzzy logical connectives, hedge $*_{Y1}$.

Table 2.5: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of data from Tab. 2.1, minimum-based fuzzy logical connectives, hedge $*_{Y1}$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 95$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 5$ ms.

thresholds	0	0.25	0.5	1
size $\lvert\mathcal{B}(X, Y^{*_Y}, I) / {}^a\!\approx\rvert$	1	10	27	95
size reduction	0.011	0.105	0.284	1.000
naive algorithm (ms)	8	9	10	15
our algorithm "CL" (ms)	1	2	2	5
speed-up "CL"	8.00	4.50	5.00	3.00
our algorithm "FC" (ms)	1	2	3	5
speed-up "FC"	8.00	4.50	3.33	3.00

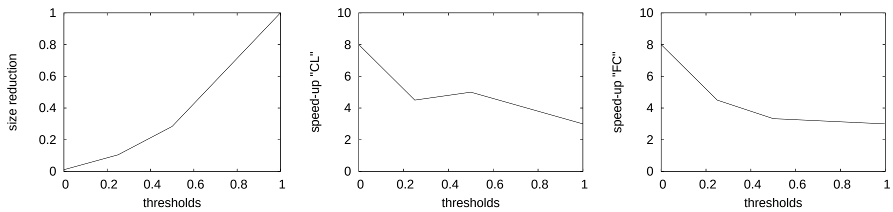
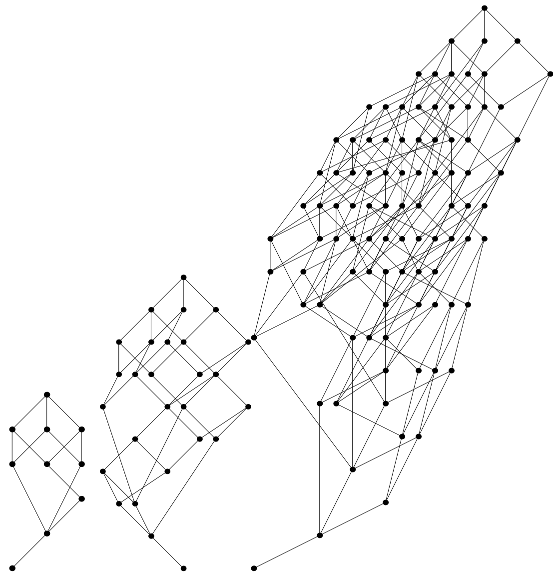


Figure 2.7: Size reduction and speed-ups from Tab. 2.5.

of factorization when using Łukasiewicz fuzzy logical operations, Fig. 2.5 contains graphs depicting size reduction and speed-up and the factor lattices $\mathcal{B}(X, Y^{*_Y}, I) / {}^a\!\approx$ are depicted in Fig. 2.6. We can see that the factorization by our algorithms works as well on lattices with a hedge as without it. Furthermore, we could conclude that the time consumed by the factorization part of the naive algorithm heavily depends on the structure of the whole lattice. Our algorithms, on the other side, compute the factor lattice just like any other concept lattice, no matter what the whole lattice looks like.
Tab. 2.5, Fig. 2.7 and Fig. 2.8 show the same characteristics when using the minimum-based (Göedel) fuzzy logical operations instead of Łukasiewicz fuzzy logical operations.
We also did a series of experiments for a more restrictive hedge $*_Y$: for $a \in L$, $a^*_Y = 0.25$ if $a = 0.5, 0.75$ and $a^*_Y = a$ otherwise. The possible threshold values in this case are (the fixed points of $*_Y$) $a = 0, 0.25, 1$, of which only $a = 0.25$ is non-trivial. Results are summarized and depicted in Tab. 2.6, Fig. 2.9 and Fig. 2.10 (Łukasiewicz fuzzy logical operations) and Tab. 2.7, Fig. 2.11 and Fig. 2.12 (minimum-based fuzzy logical operations).
Finally, we demonstrate the effects of factorization on an example of data table from Tab. 2.8 with a finer distribution of truth degrees, which are now from eleven element chain $L = \{0, 0.1, 0.2, \ldots, 0.9, 1\}$. We again use all possible values for threshold. of thresholds, $a = 0.1, 0.2, \ldots, 0.9$. The



$a = 0.25$ $\quad a = 0.5$ $\quad\quad a = 1$

Figure 2.8: Factor lattices $\mathcal{B}\left(X, Y^{*_Y}, I\right)/{}^{a}{\approx}$ of data from Tab. 2.1, minimum-based fuzzy logical connectives, hedge $*_{Y}1$.

Table 2.6: Results of factorization of $\mathcal{B}\left(X, Y^{*_Y}, I\right)$ of data from Tab. 2.1, Łukasiewicz fuzzy logical connectives, hedge $*_{Y}2$; $|\mathcal{B}\left(X, Y^{*_Y}, I\right)| = 33$, time for computing $\mathcal{B}\left(X, Y^{*_Y}, I\right) = 3$ ms.

thresholds	0	0.25	1
size $\lvert\mathcal{B}\left(X, Y^{*_Y}, I\right)/{}^{a}{\approx}\rvert$	1	10	33
size reduction	0.030	0.303	1.000
naive algorithm (ms)	4	8	6
our algorithm "CL" (ms)	1	2	3
speed-up "CL"	4.00	4.00	2.00
our algorithm "FC" (ms)	1	2	3
speed-up "FC"	4.00	4.00	2.00

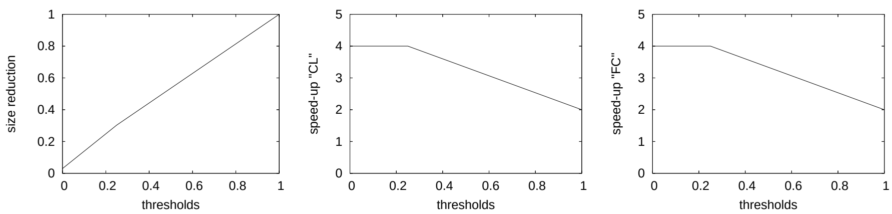


Figure 2.9: Size reduction and speed-ups from Tab. 2.6.

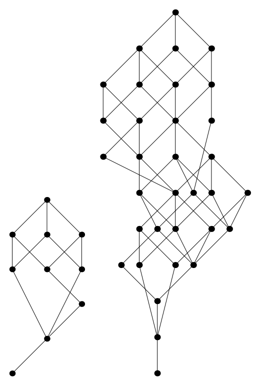


Figure 2.10: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)\,/^{a}\!\approx$ of data from Tab. 2.1, Łukasiewicz fuzzy logical connectives, hedge $*_{Y}2$.

Table 2.7: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of data from Tab. 2.1, minimum-based fuzzy logical connectives, hedge $*_{Y}2$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 50$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 3$ ms.

thresholds	0	0.25	1
size $\|\mathcal{B}(X, Y^{*_Y}, I)\,/^{a}\!\approx\|$	1	10	50
size reduction	0.020	0.200	1.000
naive algorithm (ms)	10	10	8
our algorithm “CL” (ms)	1	2	3
speed-up “CL”	10.00	5.00	2.67
our algorithm “FC” (ms)	1	1	3
speed-up “FC”	10.00	10.00	2.67

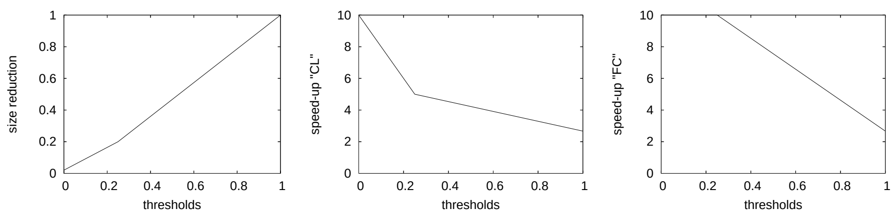
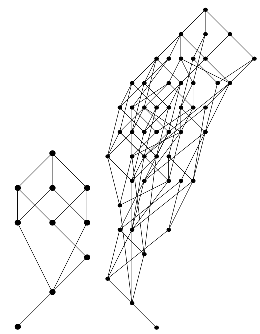


Figure 2.11: Size reduction and speed-ups from Tab. 2.7.

Table 2.8: Data table of EU countries, 11 truth degrees.

	a	b	c	d	e
1 Austria	0.1	0.2	0.5	0.9	0.9
2 Belgium	0.1	0.1	0.5	0.9	0.7
3 Cyprus	0	0	0.4	0.9	0.9
4 Czech rep.	0.1	0.1	0.2	0.7	0.7
5 Denmark	0.1	0.1	0.5	0.9	0.9
6 Estonia	0	0.1	0.1	0.5	0.4
7 Finland	0.1	0.6	0.4	0.8	0.6
8 France	0.7	1	0.4	0.9	0.6
9 Germany	1	0.7	0.5	0.9	0.7
10 Greece	0.1	0.2	0.2	0.7	0.5
11 Hungary	0.1	0.2	0.1	0.2	0.8
12 Ireland	0	0.1	0.5	0.7	0.9
13 Italy	0.7	0.6	0.4	0.9	0.6
14 Latvia	0	0.1	0	0.9	0.4
15 Lithuania	0	0.1	0	1	0.2
16 Luxembourg	0	0	1	0.9	1
17 Malta	0	0	0.1	0.9	0.7
18 Netherlands	0.2	0.1	0.4	0.6	1
19 Poland	0.5	0.6	0.1	0.6	0.1
20 Portugal	0.1	0.2	0.2	0.7	0.9
21 Slovakia	0.1	0.1	0.1	0	0
22 Slovenia	0	0	0.3	0.2	0.8
23 Spain	0.5	0.9	0.3	0.8	0.5
24 Sweden	0.1	0.8	0.4	0.8	0.8
25 UK	0.7	0.4	0.4	1	0.8

attributes: a – many habitants (millions), b – large area (thousands km^2), c – high GDP (EUR), d – low inflation (%), e – low unemployment (%)



$a = 0.25 \qquad a = 1$

Figure 2.12: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)/^a\!\approx$ of data from Tab. 2.1, minimum-based fuzzy logical connectives, hedge $*_{Y}2$.

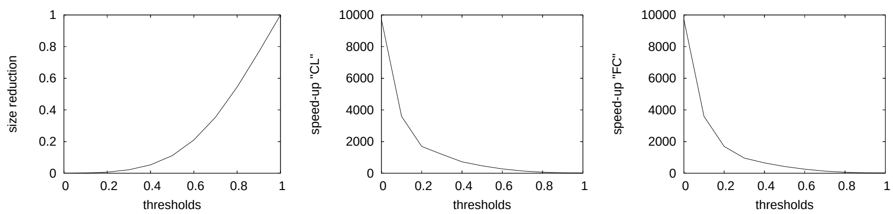


Figure 2.13: Size reduction and speed-ups from Tab. 2.9.

characteristics are the same as for the above case of five truth degrees. The two hedges (and possible threshold values) we use in this case are: (1) for $a \in L$, $a^*_Y = 0.5$ if $a = 0.6..0.9$ and $a^*_Y = a$ otherwise (thresholds $a = 0, 0.1, \ldots, 0.5, 1$); (2) for $a \in L$, $a^*_Y = 0.3$ if $a = 0.4..0.9$ and $a^*_Y = a$ otherwise (thresholds $a = 0, 0.1, 0.2, 0.3, 1$). You can see the results obtained when considering both Łukasiewicz and minimum-based fuzzy logical operations, no hedges and the two above described hedges in Tab. 2.9 to Tab. 2.14 (summarizing tables), Fig. 2.13 to Fig. 2.23 (size reduction and speed-up graphs) and Fig. 2.14 to Fig. 2.24 (some factor lattices, some are too big for showing, counting more than five hundreds of concepts).

Table 2.9: Results of factorization of $\mathcal{B}(X,Y,I)$ of data from Tab. 2.8, Łukasiewicz fuzzy logical connectives, no hedges (identity); $|\mathcal{B}(X,Y,I)| = 5435$, time for computing $\mathcal{B}(X,Y,I) = 473$ ms.

thresholds	0	0.1	0.2	0.3	0.4	0.5
size $\lvert\mathcal{B}(X,Y,I)/{}^{a}\!\approx\rvert$	1	10	39	117	286	605
size reduction	0.000	0.002	0.007	0.022	0.053	0.111
naive algorithm (ms)	9726	10786	13540	19133	26740	33161
our algorithm “CL” (ms)	1	3	8	16	37	71
speed-up “CL”	9726	3595	1692	1195	722	467
our algorithm “FC” (ms)	1	3	8	20	41	78
speed-up “FC”	9726	3595	1692	956	652	425

thresholds	0.6	0.7	0.8	0.9	1
size $\lvert\mathcal{B}(X,Y,I)/{}^{a}\!\approx\rvert$	1142	1925	2963	4171	5435
size reduction	0.210	0.354	0.545	0.767	1.000
naive algorithm (ms)	34541	28505	19046	12438	10704
our algorithm “CL” (ms)	125	200	292	393	483
speed-up “CL”	276	142	65	31	22
our algorithm “FC” (ms)	135	210	300	394	472
speed-up “FC”	255	135	63	31	22

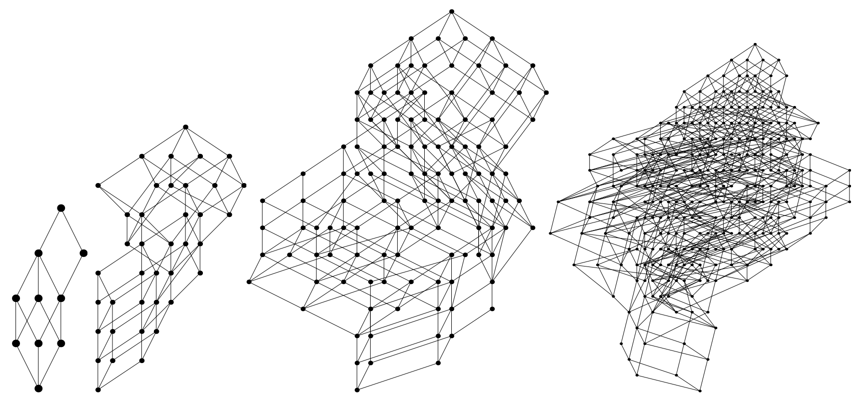


Figure 2.14: Factor lattices $\mathcal{B}(X,Y,I)/{}^{a}\!\approx$ of data from Tab. 2.8, Łukasiewicz fuzzy logical connectives, no hedges (identity).

Table 2.10: Results of factorization of $\mathcal{B}(X,Y,I)$ of data from Tab. 2.8, minimum-based fuzzy logical connectives, no hedges (identity); $|\mathcal{B}(X,Y,I)| = 1273$, time for computing $\mathcal{B}(X,Y,I) = 88$ ms.

thresholds	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
size $\lvert\mathcal{B}(X,Y,I)/{}^a\!\approx\rvert$	1	10	42	91	112	164	246	402	642	938	1273
size reduction	0.001	0.008	0.033	0.071	0.088	0.129	0.193	0.316	0.504	0.737	1.000
naive algorithm (ms)	516	479	468	474	478	482	490	509	539	575	620
our algorithm "CL" (ms)	1	3	6	11	14	18	25	34	51	68	90
speed-up "CL"	516.0	159.7	78.00	43.09	34.14	26.78	19.60	14.97	10.57	8.46	6.89
our algorithm "FC" (ms)	2	2	6	10	14	18	23	33	50	66	88
speed-up "FC"	258.0	239.5	78.00	47.40	34.14	26.78	21.30	15.42	10.78	8.71	7.05

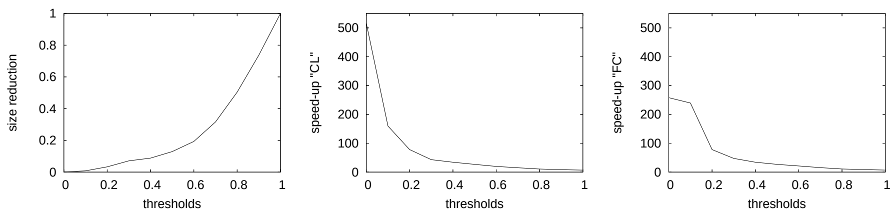


Figure 2.15: Size reduction and speed-ups from Tab. 2.10.

Table 2.11: Results of factorization of $\mathcal{B}(X,Y^{*_Y},I)$ of data from Tab. 2.8, Łukasiewicz fuzzy logical connectives, hedge $*_{Y}1$; $|\mathcal{B}(X,Y^{*_Y},I)| = 746$, time for computing $\mathcal{B}(X,Y^{*_Y},I) = 71$ ms.

thresholds	0	0.1	0.2	0.3	0.4	0.5	1
size $\lvert\mathcal{B}(X,Y^{*_Y},I)/{}^a\!\approx\rvert$	1	10	18	20	22	22	746
size reduction	0.001	0.013	0.024	0.027	0.029	0.029	1.000
naive algorithm (ms)	272	415	462	412	349	299	320
our algorithm "CL" (ms)	1	3	4	4	4	5	73
speed-up "CL"	272.0	138.3	115.5	103.0	87.25	59.8	4.38
our algorithm "FC" (ms)	1	3	4	5	5	5	71
speed-up "FC"	272.0	138.3	115.5	82.40	69.80	59.80	4.51

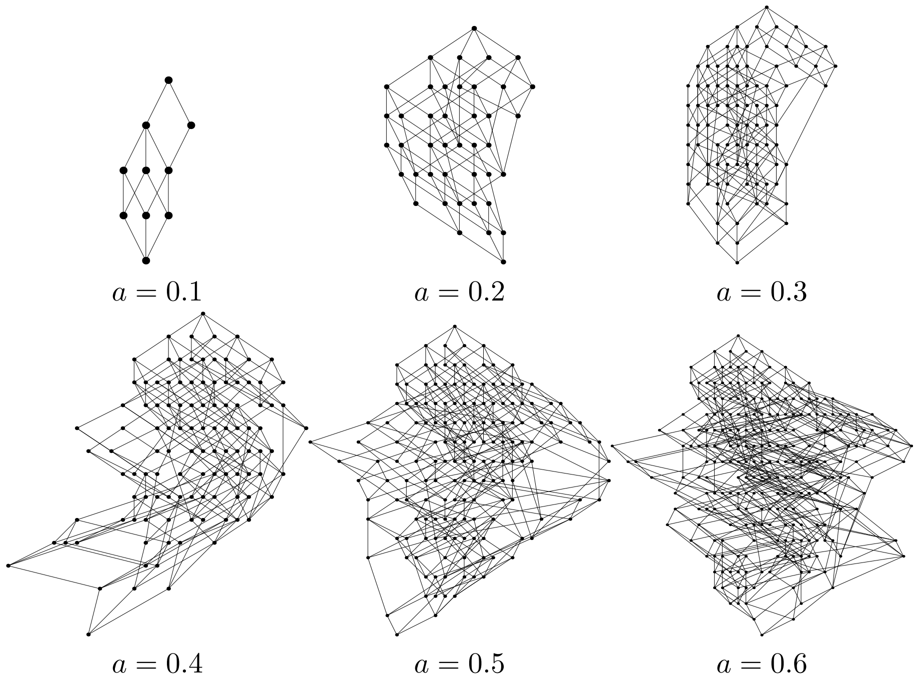


Figure 2.16: Factor lattices $\mathcal{B}(X, Y, I)/{}^a\!\approx$ of data from Tab. 2.8, minimum-based fuzzy logical connectives, no hedges (identity).

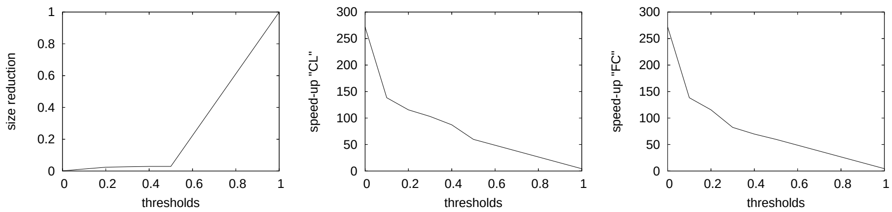


Figure 2.17: Size reduction and speed-ups from Tab. 2.11.

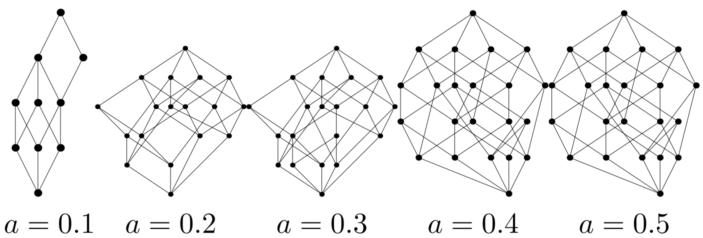


Figure 2.18: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)/{}^a\!\approx$ of data from Tab. 2.8, Łukasiewicz fuzzy logical connectives, hedge $*_{Y1}$.

Table 2.12: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of data from Tab. 2.8, minimum-based fuzzy logical connectives, hedge $*_{Y1}$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 431$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 40$ ms.

thresholds	0	0.1	0.2	0.3	0.4	0.5	1
size $\lvert\mathcal{B}(X, Y^{*_Y}, I)/^a\!\approx\rvert$	1	10	42	91	112	164	431
size reduction	0.002	0.023	0.097	0.211	0.260	0.381	1.000
naive algorithm (ms)	96	92	93	95	95	100	121
our algorithm "CL" (ms)	2	3	7	11	13	20	42
speed-up "CL"	48.00	30.67	13.29	8.64	7.31	5.00	2.88
our algorithm "FC" (ms)	1	3	6	11	13	18	40
speed-up "FC"	96.00	30.67	15.50	8.64	7.31	5.56	3.03

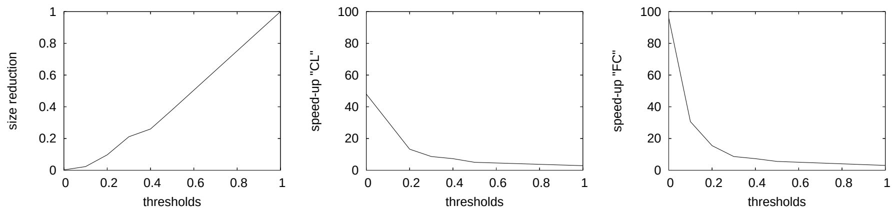


Figure 2.19: Size reduction and speed-ups from Tab. 2.12.

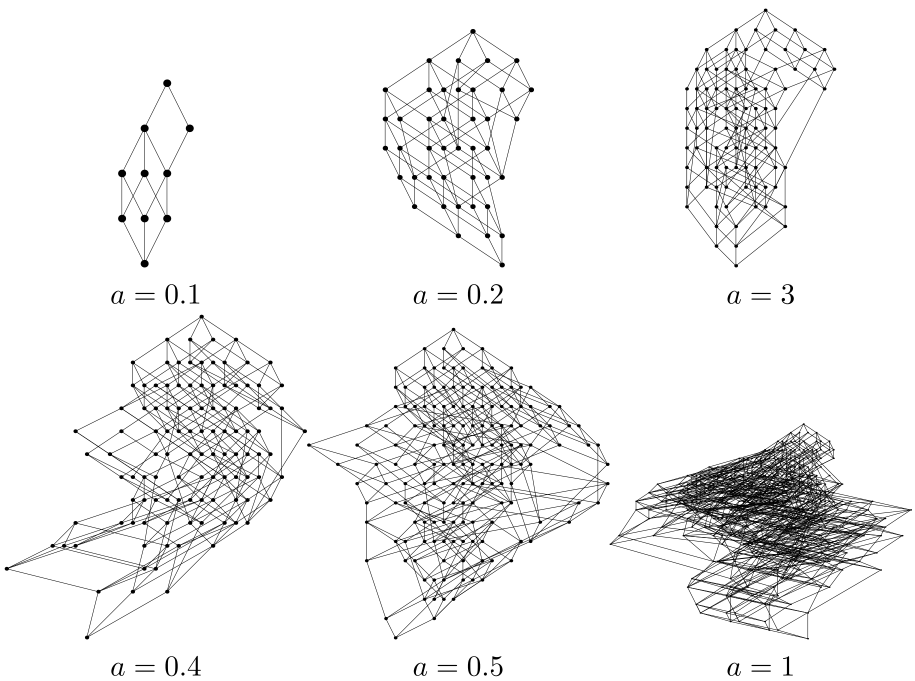


Figure 2.20: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)/^a\!\approx$ of data from Tab. 2.8, minimum-based fuzzy logical connectives, hedge $*_{Y1}$.

Table 2.13: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of data from Tab. 2.8, Łukasiewicz fuzzy logical connectives, hedge $*_Y2$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 165$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 19$ ms.

thresholds	0	0.1	0.2	0.3	1
size $\lvert\mathcal{B}(X, Y^{*_Y}, I) / {}^{a}\!\approx\rvert$	1	10	18	20	165
size reduction	0.006	0.061	0.109	0.121	1.000
naive algorithm (ms)	33	61	58	44	42
our algorithm “CL” (ms)	2	3	4	4	20
speed-up “CL”	16.50	20.33	14.50	11.00	2.1
our algorithm “FC” (ms)	1	3	5	5	19
speed-up “FC”	33.00	20.33	11.60	8.80	2.21

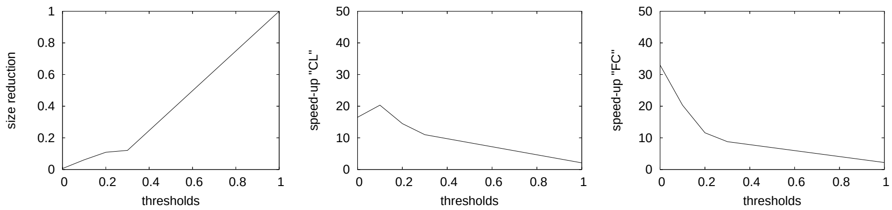
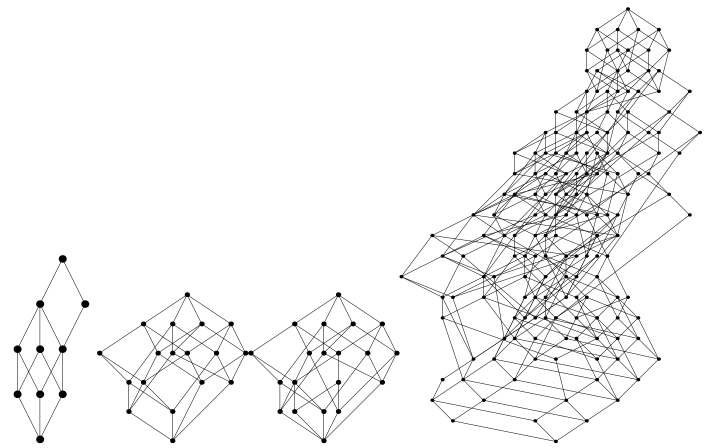


Figure 2.21: Size reduction and speed-ups from Tab. 2.13.

We can see much greater speed-up of our algorithms to the naive algorithm (even hundreds times faster!). The reason is that the factorization part of the naive algorithm is very time demanding – it can require even fifty times more time than the time required for computing the whole lattice (in the case of Łukasiewicz fuzzy logical operations). Furthermore, note that computing the whole lattice using the new closure operator (algorithm “CL”, threshold equal to 1) is slightly slower than computing the whole lattice using the original closure operator. Indeed, the new closure operator is a little bit more complex than the original one. On the other side, computing factor lattice from factorized context using original closure operator is never slower than computing whole lattice. This finding is also clear to explain.

2.4.2 IPAQ questionnaire

In the following example we process more larger (at least in the number of objects) and more real-life, however, input data table. The data table $\langle X, Y, I\rangle$ comes from samples of results of IPAQ questionnaire. The purpose of the IPAQ (International Physical Activity Questionnaire) is to monitor various attributes related to physical activity of a population. We used the full data set collected during a research program at the Faculty of Physical Culture, Palacký University, Olomouc. The objects from X are both men and women



$a = 0.1$ $a = 2$ $a = 0.3$ $a = 1$

Figure 2.22: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)\,/^a\!\approx$ of data from Tab. 2.8, Łukasiewicz fuzzy logical connectives, hedge $*_{Y2}$.

Table 2.14: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of data from Tab. 2.8, minimum-based fuzzy logical connectives, hedge $*_{Y2}$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 259$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 22$ ms.

thresholds	0	0.1	0.2	0.3	1
size $\|\mathcal{B}(X, Y^{*_Y}, I)\,/^a\!\approx\|$	1	10	42	91	259
size reduction	0.004	0.039	0.162	0.351	1.000
naive algorithm (ms)	45	46	47	50	60
our algorithm “CL” (ms)	1	3	6	11	23
speed-up “CL”	45.00	15.33	7.83	4.55	2.61
our algorithm “FC” (ms)	1	2	6	10	22
speed-up “FC”	45.00	23.00	7.83	5.00	2.73

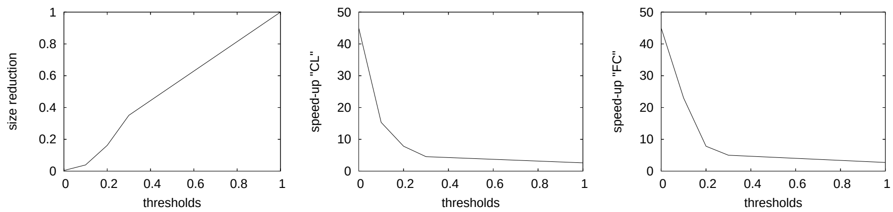


Figure 2.23: Size reduction and speed-ups from Tab. 2.14.

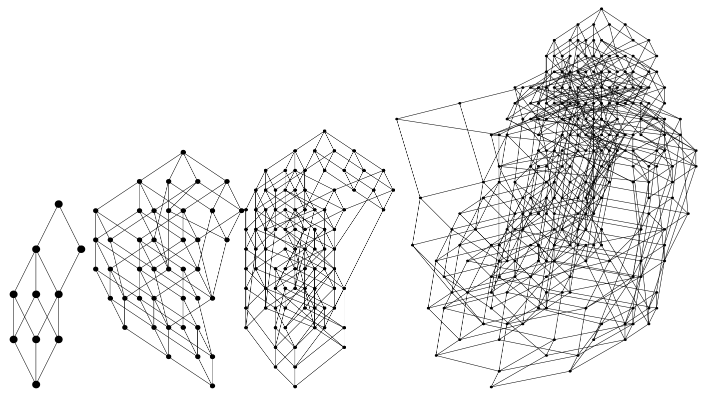


Figure 2.24: Factor lattices $\mathcal{B}(X, Y^{*_Y}, I)\,/{}^{a}\!\approx$ of data from Tab. 2.8, minimum-based fuzzy logical connectives, hedge $*_{Y}2$.

Table 2.15: Results of factorization of $\mathcal{B}(X, Y, I)$ of IPAQ data table, Łukasiewicz fuzzy logical connectives, no hedges (identity); $|\mathcal{B}(X, Y, I)| = 275990$, time for computing $\mathcal{B}(X, Y, I) = 1838.1$ s.

thresholds	0	0.25	0.5	0.75	1
size $\lvert\mathcal{B}(X, Y, I)\,/{}^{a}\!\approx\rvert$	1	128	4374	48492	275990
size reduction	0.000	0.000	0.016	0.176	1.000
naive algorithm (min)	590	572.3	2627	3350	877.5
our algorithm "CL" (s)	0.2	1.6	41.4	391.1	1840.9
speed-up "CL"	177000	21461	3807	514	28.6
our algorithm "FC" (s)	0.2	2.0	45.6	411.2	1835.2
speed-up "FC"	177000	17169	3457	489	28.7

in the Czech Republic who entered the questionnaire; counting 4318 objects. The attributes are selected IPAQ-attributes: "frequent intensive physical activity", "frequent medium-burdening physical activity", "frequent walking", "higher age", "high education", "excess hours in work", "living in large city" and "good BMI (Body Mass Index)"; total of 8 attributes. The attributes are scaled to $[0, 1]$ so that they can be considered as fuzzy attributes with truth degrees from five element chain $L = \{0, 0.25, 0.5, 0.75, 1\}$. We will refer to $\langle X, Y, I\rangle$ as "IPAQ data table" in the following.

The results for Łukasiewicz and minimum-based (Göedel) fuzzy logical operations, both structures without hedges and with the two hedges used in the previous example of countries of EU (for five element chain) are depicted in Tab. 2.15 to Tab. 2.20 (summarizing tables) and Fig. 2.25 to Fig. 2.30 (size reduction and speed-up graphs). We do not show the factor lattices, since they are too big counting tens thousand of formal concepts.

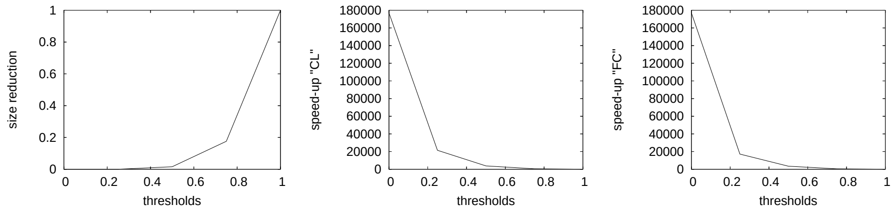


Figure 2.25: Size reduction and speed-ups from Tab. 2.15.

Table 2.16: Results of factorization of $\mathcal{B}(X,Y,I)$ of IPAQ data table, minimum-based fuzzy logical connectives, no hedges (identity); $|\mathcal{B}(X,Y,I)| = 178977$, time for computing $\mathcal{B}(X,Y,I) = 805$ s.

thresholds	0	0.25	0.5	0.75	1
size $\lvert\mathcal{B}(X,Y,I)/{}^{a}\!\approx\rvert$	1	128	2457	30326	178977
size reduction	0.000	0.001	0.014	0.169	1.000
naive algorithm (min)	107.3	62.4	50.9	207.7	479.6
our algorithm "CL" (s)	0.2	1.3	15.2	147.5	817.5
speed-up "CL"	32190	2880	201	84.5	35.2
our algorithm "FC" (s)	0.2	1.2	14.3	141.7	803.5
speed-up "FC"	32190	3120	214	87.9	35.8

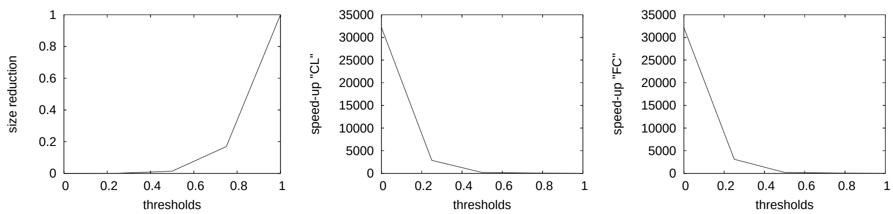


Figure 2.26: Size reduction and speed-ups from Tab. 2.16.

Table 2.17: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of IPAQ data table, Łukasiewicz fuzzy logical connectives, hedge $*_{Y}1$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 42686$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 286.6$ s.

thresholds	0	0.25	0.5	1
size $\lvert\mathcal{B}(X, Y^{*_Y}, I)/{}^{a}\!\approx\rvert$	1	128	256	42686
size reduction	0.000	0.003	0.006	1.000
naive algorithm (s)	1294.6	4114.5	2138.4	1638.5
our algorithm "CL" (s)	0.2	1.6	3.3	286.7
speed-up "CL"	6473	2572	648	5.72
our algorithm "FC" (s)	0.2	1.9	3.8	286.3
speed-up "FC"	6473	2166	563	5.72

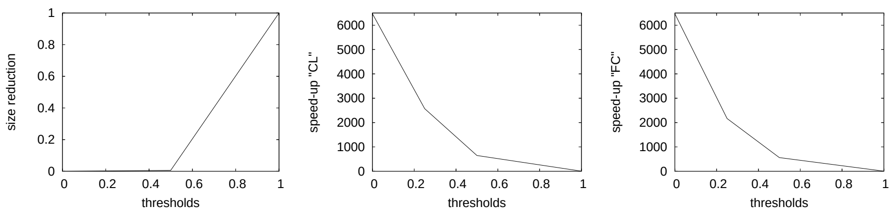


Figure 2.27: Size reduction and speed-ups from Tab. 2.17.

What can we see? First, a significant size reduction, especially in the case when hedges other than identity are used. The factor lattices don't grow above the fifth of the size of corresponding whole lattices even for threshold $a = 0.75$. Second, this great size reduction is accompanied by a giant speed-ups, our algorithms are many thousands times faster than the naive algorithm (see the case of Łukasiewicz fuzzy logical operations and no hedges, Tab. 2.15). Note that, in tables for cases without hedges, the values for the time needed for computing the factor lattice by the naive algorithm are denoted in minutes rather than seconds or microseconds, which means than the computation lasted for hours. For instance 3350 minutes $\approx$ 56 hours contrary to 1831 seconds $\approx$ 31 minutes for the whole lattice alone and contrary to roughly 400 seconds $\approx$ 7 minutes for our algorithms, see Tab. 2.15. This makes the naive algorithm absolutely impracticable! Lastly, take note of the very varying times of the naive algorithm across the threshold values and the different progress with Łukasiewicz and minimum-based fuzzy logical operations compared to the "smooth" exponential functionality of times of our algorithms.

Table 2.18: Results of factorization of $\mathcal{B}\,(X, Y^{*_Y}, I)$ of IPAQ data table, minimum-based fuzzy logical connectives, hedge $*_{Y1}$; $|\mathcal{B}\,(X, Y^{*_Y}, I)| = 30326$, time for computing $\mathcal{B}\,(X, Y^{*_Y}, I) = 144.8$ s.

thresholds	0	0.25	0.5	1
size $\lvert\mathcal{B}\,(X, Y^{*_Y}, I)\,/^a\!\approx\rvert$	1	128	2457	30326
size reduction	0.000	0.004	0.081	1.000
naive algorithm (s)	579.8	518.3	536.1	802.9
our algorithm "CL" (s)	0.2	1.2	15.2	146.6
speed-up "CL"	2899	432	35.27	5.47
our algorithm "FC" (s)	0.2	1.2	14.3	144.6
speed-up "FC"	2899	432	37.49	5.55

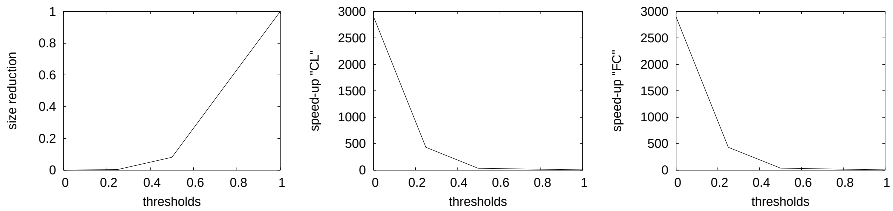


Figure 2.28: Size reduction and speed-ups from Tab. 2.18.

Table 2.19: Results of factorization of $\mathcal{B}\,(X, Y^{*_Y}, I)$ of IPAQ data table, Łukasiewicz fuzzy logical connectives, hedge $*_{Y2}$; $|\mathcal{B}\,(X, Y^{*_Y}, I)| = 3440$, time for computing $\mathcal{B}\,(X, Y^{*_Y}, I) = 27.8$ s.

thresholds	0	0.25	1
size $\lvert\mathcal{B}\,(X, Y^{*_Y}, I)\,/^a\!\approx\rvert$	1	128	3440
size reduction	0.000	0.037	1.000
naive algorithm (s)	41.7	138.0	67.5
our algorithm "CL" (s)	0.2	1.7	27.9
speed-up "CL"	208.5	81.18	2.42
our algorithm "FC" (s)	0.2	2.0	27.8
speed-up "FC"	208.5	69.00	2.43

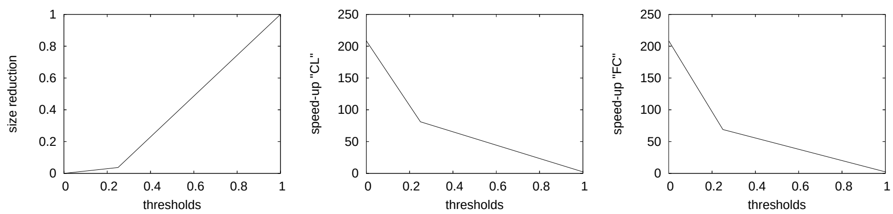


Figure 2.29: Size reduction and speed-ups from Tab. 2.19.

Table 2.20: Results of factorization of $\mathcal{B}(X, Y^{*_Y}, I)$ of IPAQ data table, minimum-based fuzzy logical connectives, hedge $*_{Y}2$; $|\mathcal{B}(X, Y^{*_Y}, I)| = 4374$, time for computing $\mathcal{B}(X, Y^{*_Y}, I) = 24.6$ s.

thresholds	0	0.25	1
size $\lvert\mathcal{B}(X, Y^{*_Y}, I)/{}^a\!\approx\rvert$	1	128	4374
size reduction	0.000	0.029	1.000
naive algorithm (s)	580.3	517.7	540.0
our algorithm "CL" (s)	0.2	1.4	24.7
speed-up "CL"	2902	369.8	21.86
our algorithm "FC" (s)	0.2	1.3	24.6
speed-up "FC"	2902	398.2	21.95

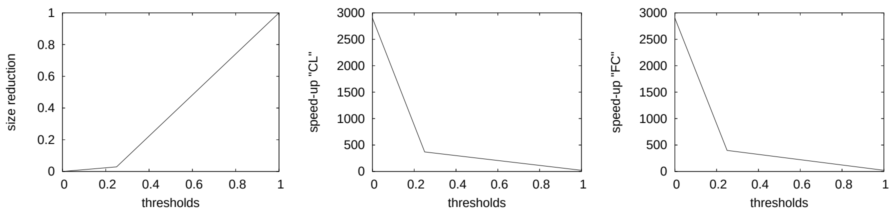


Figure 2.30: Size reduction and speed-ups from Tab. 2.20.

2.4.3 Randomly generated data tables

The final experiments on factorizing fuzzy concept lattices (with hedges) by similarity are done above randomly generated data tables. The aim is to demonstrate the effect of reduction of size of lattices and the speed-up of our algorithms on data stored in tables going from sparse ones, through medium-filled ones to quite dense tables. By sparse or dense data tables we mean formal context $\langle X, Y, I\rangle$ describing very few or very many, respectively, entries of the relation among objects and attributes. The data tables in the following experiments were generated using uniform random generation of truth degrees, keeping the specified pre-set fill ration. By fill ratio we mean the ratio

$$\frac{|I(x,y); I(x,y) > 0, \forall x \in X, \forall y \in Y|}{|X||Y|}$$

of the number $|I(x,y); I(x,y) > 0, \forall x \in X, \forall y \in Y|$ of relations $I(x,y)$ between an object and an attribute with non-zero truth degree, i.e. number of non-zero entries in the data table $\langle X, Y, I\rangle$, to the total number $|X||Y|$ of the number $|X|$ of objects multiply the number $|Y|$ of attributes, i.e. the size of the data table.

In the subsequent experiments we generated data tables of fixed size of 100 objects and 5 attributes with truth degrees from five element chain

$L = \{0, 0.25, 0.5, 0.75, 1\}$ and with fill ratios of 5 %, 25 %, 50 % and 75 %. Every experiment was repeated ten times with new randomly generated data and the final outcome of the experiment is the average of outcomes of particular runs. The size reduction and speed-up graphs only of results for Łukasiewicz and minimum-based (Göedel) fuzzy logical operations, both structures without hedges and with the first hedge (used in the previous examples of countries of EU and IPAQ questionnaire) are depicted in Fig. 2.31 to Fig. 2.34.

Note that while the size reduction has quite exponential functionality on thresholds for all values from 0 to 1, speed-ups has, however, exponential functionality mostly except threshold values close to 0. This is due to that for threshold values close to 0 there are very few similarity blocks and the factorization part of the naive algorithm discovers them relatively quickly. We can also see that size reduction as well as speed-ups are not much dependent on the fill ratio of data tables.

2.5 Summary and topics for future work

We have shown two ways to obtain $\mathcal{B}(X, Y, I) /^a\!\approx$ without computing first the whole $\mathcal{B}(X, Y, I)$ and then computing the factorization, the approach self-offered by the definition. First of the ways (fast factorization) lies in the fact that the extents (or intents) of suprema of blocks of $\mathcal{B}(X, Y, I) /^a\!\approx$ are fixed points of a certain fuzzy closure operator. By that, the factor lattice is isomorphic to the lattice of fixed points of the fuzzy closure operator. Compared to that, the second way (direct factorization) is due to interpretation of the blocks of $\mathcal{B}(X, Y, I) /^a\!\approx$ as formal concepts in a "factorized context" $\langle X, Y, I\rangle/a$, i.e. in a context in which objects and attributes are more similar than in the original context $\langle X, Y, I\rangle$. Both approaches are significantly faster than the "naive" two-step, from definition, approach, as we have seen in the experiments. In conclusion, it is worth mentioning that the method by fuzzy closure operator (and due to the equivalence also the other method) is subsumed by the more general approach imposing constraints (user-defined requirements) supplied along with the input data table and expressed by means of a fuzzy closure operator, which was very recently introduced in [18].

We also presented analogical methods of factorization of fuzzy concept lattices with hedges. The factor lattice can be computed directly from input data providing one of the hedges is identity. Future research could focus on eliminating this and other restrictions from the assumptions of the methods. Other topic for future research could be the development of a single method for direct factorization by *arbitrary* tolerance relation (interpreted as similarity on concepts), not only $^a\!\approx$. Then, the specification of the tolerance relation would be left to a user. Our presented methods results from the

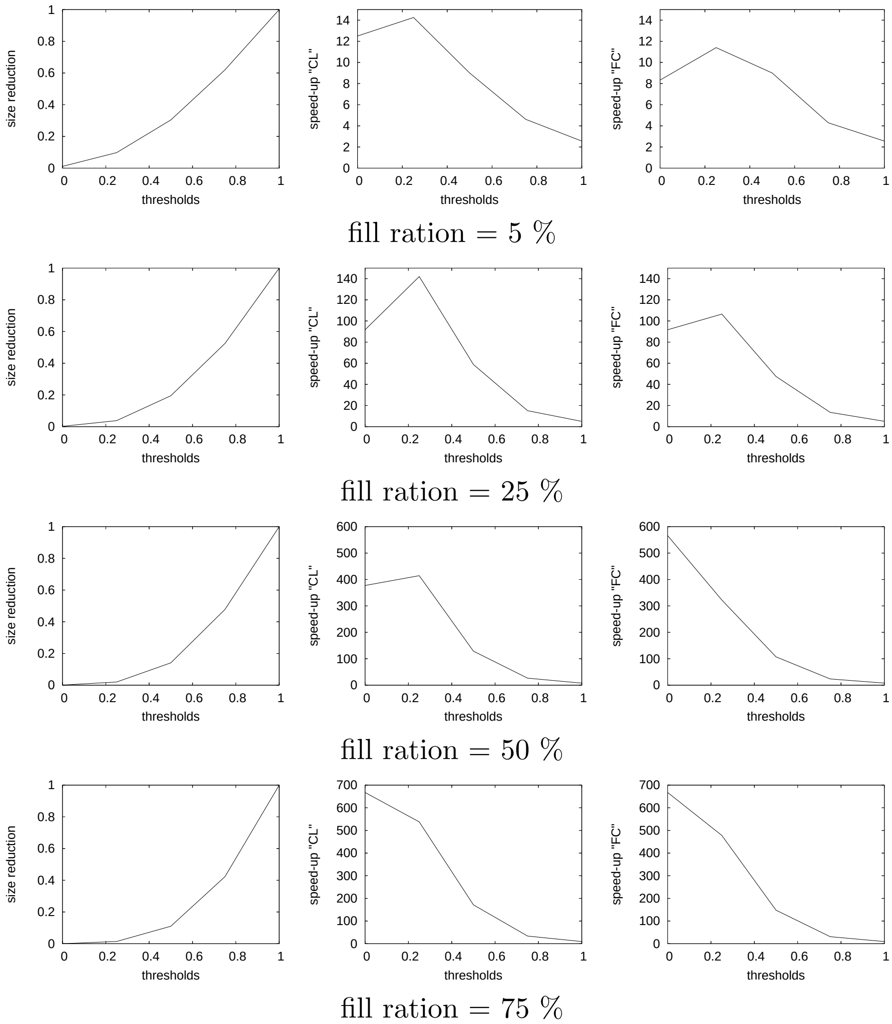


Figure 2.31: Size reduction and speed-ups of factorization of random contexts, Łukasiewicz fuzzy logical connectives, no hedges (identity).

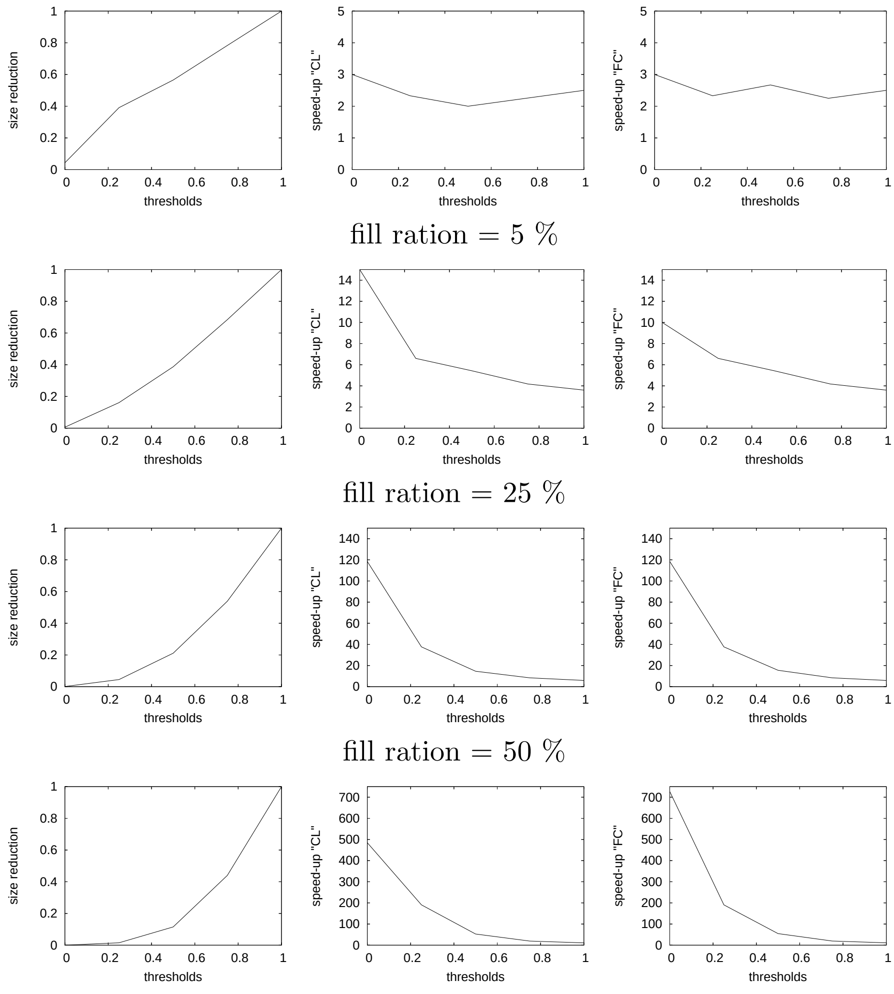


Figure 2.32: Size reduction and speed-ups of factorization of random contexts, minimum-based fuzzy logical connectives, no hedges (identity).

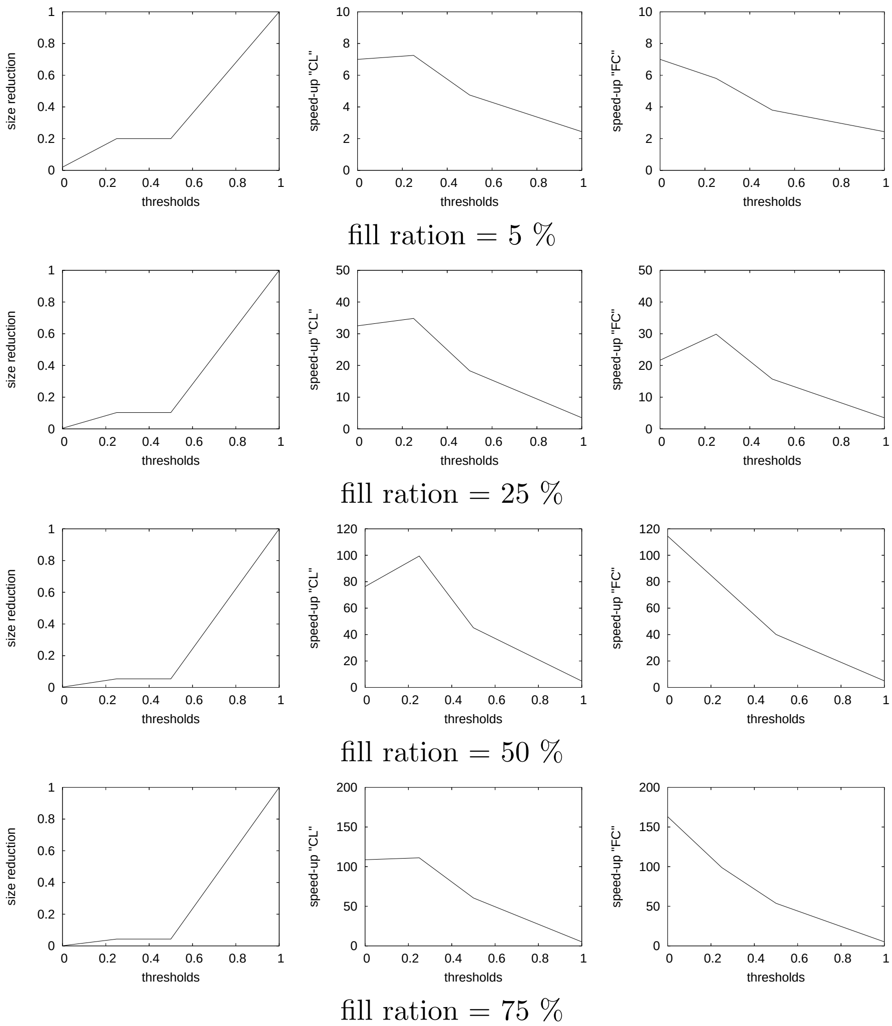
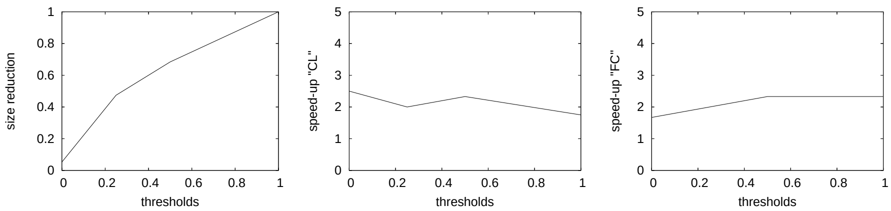
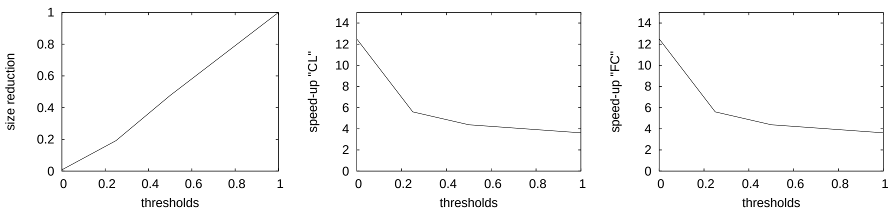


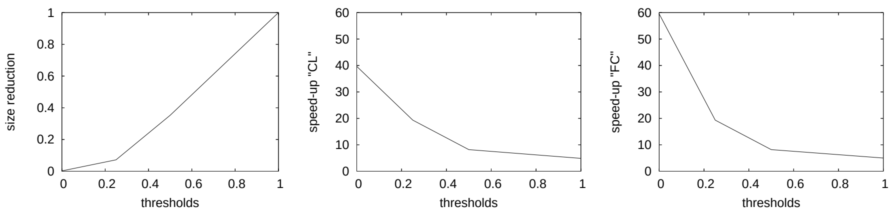
Figure 2.33: Size reduction and speed-ups of factorization of random contexts, Łukasiewicz fuzzy logical connectives, hedge $*_{Y}1$.



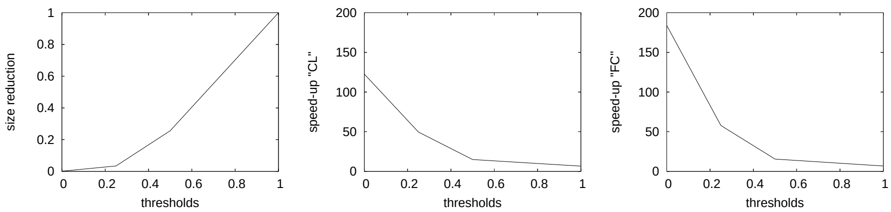
fill ration = 5 %



fill ration = 25 %



fill ration = 50 %



fill ration = 75 %

Figure 2.34: Size reduction and speed-ups of factorization of random contexts, minimum-based fuzzy logical connectives, hedge $*_{Y1}$.

properties of the similarity relation ${}^a\!\approx$ and are thus specific for this similarity. A method of factorization directly from input data by a tolerance relation on the ordinary concept lattice (from "classical" FCA, analyzing data tables with binary attributes) was described in [27]. A direct extension to data tables with graded (fuzzy) attributes is suggesting itself. The notion of similarity of concepts (under the tolerance relation) would then be naturally extended from similar/dissimilar to similar at least to some (truth) degree.

Acknowledgments The work summarized in this chapter was supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation and by grant No. 201/02/P076 of the GAČR, the principal investigator of which is R. Bělohlávek, and by institutional support of the Faculty of Science, Palacký University, research plan MSM 6198959214.

Chapter 3

Thresholds and shifted attributes

3.1 Introduction

This short chapter presents an introductory exploration of a new approach to reducing the number of formal concepts involving the idea of thresholds, which was introduced in section 1.1. Do you remember it? OK, we remind, and enter into more details. Given a collection A of objects, the collection $A^{\uparrow}$ of all attributes shared by all objects from A is in general a fuzzy set, i.e. attributes y belong to $A^{\uparrow}$ to various degrees $A^{\uparrow}(y) \in L$. Then we can pick a threshold δ and consider a set ${}^{\delta}A^{\uparrow} = \{y \mid A^{\uparrow}(y) \geq \delta\}$ of all attributes which belong to $A^{\uparrow}$ to a degree greater than or equal to δ. It is simple, and obviously it can be analogously considered for a collection of attributes. Or even both! We also reveal that with $\delta = 1$, this approach was proposed independently in [34, 43] (actually not as thresholds described, but rather as a way to eliminate, from certain point of view, unnatural fuzzy concepts). As lately as in [25], this was extended to arbitrary δ. However, the extent- and intent-forming operators defined in [25] do not form a Galois connection. This shortcoming was recognized and removed in [26] where the authors proposed new operators based on the idea of thresholds for general δ.

In section 3.2.1, we take a closer look at [26]. We show that while conceptually natural and appealing[1], the approach via thresholds, as proposed in [26], can be seen as a particular (thresholded) case of the approach via hedges. In particular, given a data table with fuzzy attributes, the fuzzy concept lattices induced by the operators of [26] are isomorphic (and in fact, almost the same) to fuzzy concept lattices with hedges induced from a data containing so-called shifts of the given fuzzy attributes (shifted attributes).

[1]The proposed extent- and intent-forming operator are not so appealing visually and practically, however.

In FCA, an attribute is conceived as a collection of objects to which it applies. Therefore, fuzzy attribute can be considered as a fuzzy set A such that a degree $A(x)$ to which an object x belongs to A is interpreted as the degree to which object x has attribute A. Given a fuzzy attribute A, i.e. a fuzzy set A of objects, a shifted attribute (shifted by δ) is a fuzzy set $\delta \to A$ where δ is a truth degree of the shift.
This observation suggests two things. First, a (just sketched) combination of the approach via hedges and shifted attributes and the approach via thresholds and, second, a relation to a factorization by similarity of a fuzzy concept lattice treated in previous chapter, since the shifts of fuzzy attributes play an important role there (in fast computation of factor lattice), as we have seen in sections 2.2.2 and 2.3.3. Of course, we will explore both of these fancy interconnections, in section 3.2.2.
The main purpose of section 3.3 is to sketch some of the promising ideas for future research.

Section 3.2 summarizes results recently presented in [14].

3.2 Fuzzy concept lattices defined by thresholds

3.2.1 New extent- and intent-forming operators

Take a look at the new operators proposed in [26]. In addition to the pair of operators ${}^{\Uparrow} : L^X \to L^Y$ (1.1) and ${}^{\Downarrow} : L^Y \to L^X$ (1.2), the authors in [26] define pairs of operators (we keep the notation of [26]) ${}^{\star} : 2^X \to 2^Y$ and ${}^{\star} : 2^Y \to 2^X$, ${}^{\square} : 2^X \to L^Y$ and ${}^{\square} : L^Y \to 2^X$, and ${}^{\diamondsuit} : L^X \to 2^Y$ and ${}^{\diamondsuit} : 2^Y \to L^X$, as follows. Let δ be an arbitrary truth degree from L (δ plays a role of a threshold). For $A \in L^X$, $C \in 2^X$, $B \in L^Y$, $D \in 2^Y$ define $C^{\star} \in 2^Y$ and $D^{\star} \in 2^X$ by

$$
\begin{aligned}
C^{\star} &= \{y \in Y \mid \textstyle\bigwedge_{x\in X}(C(x) \to I(x,y)) \geq \delta\}, &(3.1)\\
D^{\star} &= \{x \in X \mid \textstyle\bigwedge_{y\in Y}(D(y) \to I(x,y)) \geq \delta\}; &(3.2)
\end{aligned}
$$

$C^{\square} \in L^Y$ and $B^{\square} \in 2^X$ by

$$
\begin{aligned}
C^{\square}(y) &= \delta \to \textstyle\bigwedge_{x\in C} I(x,y), &(3.3)\\
B^{\square} &= \{x \in X \mid \textstyle\bigwedge_{y\in Y}(B(y) \to I(x,y)) \geq \delta\}; &(3.4)
\end{aligned}
$$

and $A^{\diamondsuit} \in 2^Y$ and $D^{\diamondsuit} \in L^X$ by

$$
\begin{aligned}
A^{\diamondsuit} &= \{y \in Y \mid \textstyle\bigwedge_{x\in X}(A(x) \to I(x,y)) \geq \delta\}, &(3.5)\\
D^{\diamondsuit}(x) &= \delta \to \textstyle\bigwedge_{y\in D} I(x,y), &(3.6)
\end{aligned}
$$

for each $x \in X$, $y \in Y$.

Denote the corresponding set of fixpoints of these pairs of operators by

$$\begin{aligned}
\mathcal{B}(X_\star, Y_\star, I) &= \{\langle A, B\rangle \in 2^X \times 2^Y \mid A^\star = B, B^\star = A\}, \\
\mathcal{B}(X_\Box, Y_\Box, I) &= \{\langle A, B\rangle \in 2^X \times L^Y \mid A^\Box = B, B^\Box = A\}, \\
\mathcal{B}(X_\Diamond, Y_\Diamond, I) &= \{\langle A, B\rangle \in L^X \times 2^Y \mid A^\Diamond = B, B^\Diamond = A\}, \\
\mathcal{B}(X_\Uparrow, Y_\Downarrow, I) &= \{\langle A, B\rangle \in L^X \times L^Y \mid A^\Uparrow = B, B^\Downarrow = A\} \quad (= \mathcal{B}(X, Y, I)).
\end{aligned} \tag{3.7}$$

Together with the operators $^\Uparrow$ and $^\Downarrow$, we can see the four pairs of operators (and four types of concept lattices). One pair forming formal concepts with both extent and intent being crisp sets, i.e. crisp formal concepts (constituting a ordinary concept lattice [15]), other two pairs forming formal concepts of which one part (extent or intent) is a crisp set while the other part is a fuzzy set (constituting a so-called one-sided fuzzy concept lattice [34, 43]) and finally, the pair forming formal concepts with both extent and intent being fuzzy sets, i.e. fuzzy formal concepts (constituting a fuzzy concept lattice). This is not very practical (among other aspects, for example, visual).

We now introduce a new pair of operators induced by a formal fuzzy context $\langle X, Y, I\rangle$. For $\delta, \varepsilon \in L$, fuzzy sets $A \in L^X$ and $B \in L^Y$, consider fuzzy sets $A^{\uparrow_{I,\delta}} \in L^Y$ and $B^{\downarrow_{I,\varepsilon}} \in L^X$ defined by

$$A^{\uparrow_{I,\delta}}(y) = \delta \to \textstyle\bigwedge_{x\in X}(A^{*_X}(x) \to I(x,y)), \tag{3.8}$$

$$B^{\downarrow_{I,\varepsilon}}(x) = \varepsilon \to \textstyle\bigwedge_{y\in Y}(B^{*_Y}(y) \to I(x,y)). \tag{3.9}$$

We will often write just $A^{\uparrow_\delta}$ and $B^{\downarrow_\varepsilon}$ or even $A^\uparrow$ and $B^\downarrow$ if I, δ, and ε are obvious, particularly if $\delta = \varepsilon$.

Remark 6 *Note that, due to the properties of $\to$, we have that*

$$A^{\uparrow_{I,\delta}}(y) = 1 \textit{ iff } \delta \leq \bigwedge_{x\in X} (A^{*_X}(x) \to I(x,y)),$$

i.e. iff the degree to which y is shared by all objects from A is at least δ and, analogously,

$$B^{\downarrow_{I,\varepsilon}}(x) = 1 \textit{ iff } \varepsilon \leq \bigwedge_{y\in Y} (B^{*_Y}(y) \to I(x,y)),$$

i.e. iff the degree to which x shares all attributes from B is at least ε.

In general, $A^{\uparrow_{I,\delta}}(y)$ can be thought of as a truth degree of "the degree to which y is shared by all objects (...) from A is at least δ and, analogously, $B^{\downarrow_{I,\varepsilon}}(x)$ can be thought of as a truth degree of "the degree to which x shares all attributes (...) from B is at least ε. As such, they can be considered

just another parametrization of the operators with and without hedges. We sometimes call them thresholded operators (with hedges). We will show that this general approach involving the idea of thresholds subsumes the proposals of [26] as special cases. Moreover, unlike formulas (3.3) and (3.4), and (3.5) and (3.6), formulas for operators ${}^{\uparrow_{I,\delta}}$ and ${}^{\downarrow_{I,\delta}}$ are symmetric and form (only) fuzzy formal concepts.
The set

$$\mathcal{B}\left(X_{\delta}^{*X}, Y_{\varepsilon}^{*Y}, I\right) = \{\langle A, B\rangle \mid A^{\uparrow} = B,\ B^{\downarrow} = A\}$$

of all fixpoints of $\langle {}^{\uparrow}, {}^{\downarrow} \rangle$ is called a (thresholded) fuzzy concept lattice (with hedges) of $\langle X, Y, I\rangle$; elements $\langle A, B\rangle \in \mathcal{B}\left(X_{\delta}^{*X}, Y_{\varepsilon}^{*Y}, I\right)$ will be called formal concepts of $\langle X, Y, I\rangle$; A and B are called the extent and intent of $\langle A, B\rangle$, respectively. However, describing the structure of $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\varepsilon}^{*Y}, I\right)$ (under a partial order $\leq$) remains an open problem to be studied. In the following, we will focus on the case $\delta = \varepsilon$ only.

Remark 7 *Since $1 \to \delta = \delta$ for each $\delta \in L$, we have $A^{\uparrow_{I,1}} = A^{\uparrow_I}$ and $B^{\downarrow_{I,1}} = B^{\downarrow_I}$ and, therefore, $\mathcal{B}\left(X_{1}^{*X}, Y_{1}^{*Y}, I\right) = \mathcal{B}\left(X^{*X}, Y^{*Y}, I\right)$.*

3.2.2 Reducing thresholds to shifted attributes and relationship to factorization

The following key theorem shows that from a mathematical point of view, $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ is, in fact, a fuzzy concept lattice with hedges (i.e. without thresholds) induced by a δ-shift $\delta \to I$ of I.

Theorem 18 *For any $\delta \in L$, ${}^{\uparrow_{I,\delta}}$ coincides with ${}^{\uparrow_{\delta\to I}}$, and ${}^{\downarrow_{I,\delta}}$ coincides with ${}^{\downarrow_{\delta\to I}}$. Therefore, $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right) = \mathcal{B}\left(X^{*X}, Y^{*Y}, \delta \to I\right)$.*

PROOF. Using $a \to (b \to c) = b \to (a \to c)$ and $a \to (\bigwedge_{j\in J} b_j) = \bigwedge_{j\in J}(a \to b_j)$ we get

$$\begin{aligned} A^{\uparrow_{I,\delta}}(y) &= \delta \to \textstyle\bigwedge_{x\in X}(A^{*X}(x) \to I(x,y)) = \\ &= \textstyle\bigwedge_{x\in X}(\delta \to (A^{*X}(x) \to I(x,y))) = \\ &= \textstyle\bigwedge_{x\in X}(A^{*X}(x) \to (\delta \to I(x,y))) = A^{\uparrow_{\delta\to I}}(y). \end{aligned}$$

One can proceed analogously to show that ${}^{\downarrow_{I,\delta}}$ coincides with ${}^{\downarrow_{\delta\to I}}$. Then the equality $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right) = \mathcal{B}\left(X^{*X}, Y^{*Y}, \delta \to I\right)$ follows immediately. □

Remark 8 *(1) Using [17], Theorem 18 yields that $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ is a complete lattice; we show a main theorem for $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ below.*
(2) In addition to $A^{\uparrow_{I,\delta}}(y) = A^{\uparrow_{\delta\to I}} = \delta \to A^{\uparrow_I}$ we also have $A^{\uparrow_{I,\delta}}(y) = (\delta \otimes A_X^{*})^{\Uparrow_I}$; similarly for $B^{\downarrow_{I,\delta}}$.

Recall, from previous chapter, that shifted fuzzy contexts $\langle X, Y, a \to I\rangle$ play an important role in direct factorization of a fuzzy concept lattices $\mathcal{B}(X, Y, I)$ and $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ by a similarity given by a parameter a, see sections 2.2.2 and 2.3.3. Briefly, $\mathcal{B}(X, Y, a \to I)$ is isomorphic to a factor lattice $\mathcal{B}(X, Y, I)\,/^a\!\approx$ (Theorem 8) where $^a\!\approx$ is an a-cut of a fuzzy equivalence relation $\approx$ defined on $\mathcal{B}(X, Y, I)$. Similarily, $\mathcal{B}(X^{*_X}, Y^{*_Y}, a \to I)$ is isomorphic to a factor lattice $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)\,/^a\!\approx$ (Theorem 17) where $^a\!\approx$ is an a-cut of a fuzzy equivalence relation $\approx$ defined on $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$, providing at least one of the hedges is identity and two technical conditions are satisfied (see section 2.3, $\approx$ has to be compatible with the other hedge and a and $^a\!\approx$ must have the feature 2.7; $\approx$ and $^a\!\approx$ satisfy the conditions if a is a fixed point of the (non-identity) hedge, cf. Lemma 9).
This means that the factor lattice by similarity $^a\!\approx$ and the thresholded lattice by threshold a of $\mathcal{B}(X, Y, I)$ are the same (up to isomorphism); similarily for $\mathcal{B}(X^{*_X}, Y^{*_Y}, I)$ (if at least one of the hedges is identity). In other words, the approaches to reducing the size of a fuzzy concept lattice (with hedges) via factorization by a similarity $^a\!\approx$ and via extent- and intent-forming operators thresholded by a threshold a lead to the same reduction.

The next theorem and Remark 9 show that the fuzzy concept lattices (3.7) defined in [26] are isomorphic, and in fact identical, to fuzzy concept lattices defined by (3.8) and (3.9) with appropriate choices of ^{*_X} and ^{*_Y}. As a consequence, the approach by thresholds proposed in [26] is reducible to our approach via hedges.

Theorem 19 *Let $\mathcal{B}(X_\star, Y_\star, I)$, $\mathcal{B}(X_\Box, Y_\Box, I)$, and $\mathcal{B}(X_\Diamond, Y_\Diamond, I)$ denote the concept lattices (3.7) defined by operators (3.1, 3.2, 3.3, 3.4, 3.5 and 3.6).*

(1) *$\mathcal{B}(X_\star, Y_\star, I)$ is isomorphic to $\mathcal{B}\left(X_\delta^{*_X}, Y_\delta^{*_Y}, I\right)$, and due to Theorem 18 also to $\mathcal{B}(X^{*_X}, Y^{*_Y}, \delta \to I)$, where both ^{*_X} and ^{*_Y} are globalizations on L.*

(2) *$\mathcal{B}(X_\Box, Y_\Box, I)$ is isomorphic to $\mathcal{B}\left(X_\delta^{*_X}, Y_\delta^{*_Y}, I\right)$, and due to Theorem 18 also to $\mathcal{B}(X^{*_X}, Y^{*_Y}, \delta \to I)$, where ^{*_X} is globalization and ^{*_Y} is the identity on L.*

(3) *$\mathcal{B}(X_\Diamond, Y_\Diamond, I)$ is isomorphic to $\mathcal{B}\left(X_\delta^{*_X}, Y_\delta^{*_Y}, I\right)$, and due to Theorem 18 also to $\mathcal{B}(X^{*_X}, Y^{*_Y}, \delta \to I)$, where ^{*_X} is the identity and ^{*_Y} is globalization on L.*

PROOF. We prove only (2); the proofs for (1) and (3) are similar. First, we show that for $\langle C, D\rangle \in \mathcal{B}\left(X_\delta^{*_X}, Y_\delta^{*_Y}, I\right)$ we have $\langle {}^1C, D\rangle \in \mathcal{B}(X_\Box, Y_\Box, I)$. Indeed, for ^{*_X} being globalization and ^{*_Y} being identity we have ${}^1C = C^{*_X}$ and $D = D^{*_Y}$ and thus

$$\begin{aligned}({}^1C)^\Box &= \delta \to \textstyle\bigwedge_{x \in {}^1C} I(x, y) = \delta \to \bigwedge_{x \in X}(({}^1C)(x) \to I(x, y)) = \\ &= \delta \to \textstyle\bigwedge_{x \in X}(C^{*_X}(x) \to I(x, y)) = C^{\uparrow_{I,\delta}},\end{aligned}$$

and

$$\begin{aligned} D^{\square} &= \{x \in X \mid \bigwedge_{y \in Y}(D(y) \to I(x,y)) \geq \delta\} = \\ &= \{x \in X \mid \delta \to \bigwedge_{y \in Y}(D(y) \to I(x,y)) = 1\} = \\ &= \{x \in X \mid D^{\downarrow_{I,\delta}}(x) = 1\} = {}^{1}(D^{\downarrow_{I,\delta}}) = {}^{1}C. \end{aligned}$$

Clearly, $\langle C, D\rangle \mapsto \langle {}^{1}C, D\rangle$ defines an injective mapping of $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ to $\mathcal{B}(X_{\square}, Y_{\square}, I)$. This mapping is also surjective. Namely, for $\langle A, B\rangle \in \mathcal{B}(X_{\square}, Y_{\square}, I)$ we have $\langle A^{\uparrow_{I,\delta}\downarrow_{I,\delta}}, B\rangle \in \mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ and $A = {}^{1}(A^{\uparrow_{I,\delta}\downarrow_{I,\delta}})$. Indeed, since $A = A^{*X}$, [12], ${}^{\uparrow_{I,\delta}} = {}^{\uparrow_{\delta\to I}}$, and ${}^{\downarrow_{I,\delta}} = {}^{\downarrow_{\delta\to I}}$ give $A^{\uparrow_{I,\delta}\downarrow_{I,\delta}\uparrow_{I,\delta}} = A^{\uparrow_{I,\delta}} = A^{\square} = B$. Furthermore, $B^{\downarrow_{I,\delta}} = A^{\uparrow_{I,\delta}\downarrow_{I,\delta}}$. This shows $\langle A^{\uparrow_{I,\delta}\downarrow_{I,\delta}}, B\rangle \in \mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$. Observing

$$B^{\square} = {}^{\delta}(B^{\downarrow_I}) = {}^{1}(B^{\downarrow_{\delta\to I}}) = {}^{1}(B^{\downarrow_{I,\delta}}) = {}^{1}(A^{\uparrow_{I,\delta}\downarrow_{I,\delta}})$$

finishes the proof. □

Remark 9 *(1) As one can see from the proof of Theorem 19, an isomorphism exists such that the corresponding elements* $\langle A, B\rangle \in \mathcal{B}(X_{\square}, Y_{\square}, I)$ *and* $\langle C, D\rangle \in \mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ *are almost the same, namely,* $\langle A, B\rangle = \langle {}^{1}C, D\rangle$. *A similar fact pertains to (1) and (3) of Theorem 19 as well.*
(2) Alternatively, Theorem 19 can be proved using results from [19]. Consider e.g. $\mathcal{B}(X_{\square}, Y_{\square}, I)$: *It can be shown that* $\mathcal{B}(X_{\square}, Y_{\square}, I)$ *coincides with one-sided fuzzy concept lattice of* $\langle X, Y, \delta \to I\rangle$ *(in the sense of [34]); therefore, by [19],* $\mathcal{B}(X_{\square}, Y_{\square}, I)$ *is isomorphic to a fuzzy concept lattice with hedges where* ^{*X} *is globalization and* ^{*Y} *is identity, i.e. to* $\mathcal{B}(X^{*X}, Y, \delta \to I)$.

From (3.1) and (3.2) one easily obtains the following assertion.

Corollary 20 *The concept lattice* $\mathcal{B}(X_{\star}, Y_{\star}, I)$ *coincides with an ordinary concept lattice* $\mathcal{B}\left(X, Y, {}^{\delta}I\right)$, *where* ${}^{\delta}I = \{\langle x, y\rangle \mid I(x,y) \geq \delta\}$ *is the* δ*-cut of* I.

PROOF. $\mathcal{B}(X_{\star}, Y_{\star}, I)$ is defined by operators * ((3.1) and (3.2)): $\mathcal{B}(X_{\star}, Y_{\star}, I) = \{\langle A, B\rangle \in 2^X \times 2^Y \mid A^{\star} = B, B^{\star} = A\}$. For $A \in 2^X$, $A^{\star} = \{y \in Y \mid \bigwedge_{x\in X}(A(x) \to I(x,y)) \geq \delta\} = \{y \in Y \mid \bigwedge_{x\in A} I(x,y) \geq \delta\} = \{y \in Y \mid \forall x \in A : I(x,y) \geq \delta\} = \{y \in Y \mid \forall x \in A : (x,y) \in {}^{\delta}I\}$; similarily for $B \in 2^Y$. Hence the operators define $\mathcal{B}\left(X, Y, {}^{\delta}I\right)$ too. □

Remark 10 *The foregoing results show that* $\mathcal{B}(X_{\star}, Y_{\star}, I)$, $\mathcal{B}(X_{\square}, Y_{\square}, I)$ *and* $\mathcal{B}(X_{\diamondsuit}, Y_{\diamondsuit}, I)$ *are isomorphic to* $\mathcal{B}\left(X_{\delta}^{*X}, Y_{\delta}^{*Y}, I\right)$ *(with appropriate choices of* ^{*X} *and* ^{*Y}*). Moreover, they are almost identical, but not equal. Alternatively, one can proceed so as to define our operators by*

$$A^{\uparrow_{I,\delta}}(y) = \left(\delta \to \bigwedge_{x\in X}(A(x) \to I(x,y))\right)^{*Y}, \tag{3.10}$$

$$B^{\downarrow_{I,\varepsilon}}(x) = \left(\varepsilon \to \bigwedge_{y\in Y}(B(y) \to I(x,y))\right)^{*X}. \tag{3.11}$$

Then, we even have the equality between the thresholded concept lattices and $\mathcal{B}\left(X_\delta^{*X}, Y_\delta^{*Y}, I\right)$ *(with the same choices of* ^{*X} *and* ^{*Y}*). We still prefer (3.8) and (3.9) to (3.10) and (3.11) for reasons like clearer interpretation, easier use and, last but not least, relation to other (earlier) approaches.*

Main theorem of fuzzy concept lattices defined by thresholds and hedges

Due to Theorem 18 and Theorem 19, using [17] we immediately obtain main theorems of concept lattices defined by thresholds (3.7) and, as well, for our general case of thresholded fuzzy concept lattice with hedges, $\mathcal{B}\left(X_\delta^{*X}, Y_\delta^{*Y}, I\right)$.

Theorem 21 (1) $\mathcal{B}\left(X_\delta^{*X}, Y_\delta^{*Y}, I\right)$ *is under* $\leq$ *a complete lattice where the infima and suprema are given by*

$$\bigwedge_{j\in J}\langle A_j, B_j\rangle = \langle(\bigcap_{j\in J} A_j)^{\uparrow_{I,\delta}\downarrow_{I,\delta}}, (\bigcup_{j\in J} B_j^{*Y})^{\downarrow_{I,\delta}\uparrow_{I,\delta}}\rangle, \quad (3.12)$$

$$\bigvee_{j\in J}\langle A_j, B_j\rangle = \langle(\bigcup_{j\in J} A_j^{*X})^{\uparrow_{I,\delta}\downarrow_{I,\delta}}, (\bigcap_{j\in J} B_j)^{\downarrow_{I,\delta}\uparrow_{I,\delta}}\rangle. \quad (3.13)$$

(2) *Moreover, an arbitrary complete lattice* $\mathbf{K} = \langle K, \leq\rangle$ *is isomorphic to* $\mathcal{B}\left(X_\delta^{*X}, Y_\delta^{*Y}, I\right)$ *iff there are mappings* $\gamma : X \times \mathrm{fix}(*_X) \to K$, $\mu : Y \times \mathrm{fix}(*_Y) \to K$ *such that*

(i) $\gamma(X \times \mathrm{fix}(*_X))$ *is* $\bigvee$*-dense in* K, $\mu(Y \times \mathrm{fix}(*_Y))$ *is* $\bigwedge$*-dense in* K*;*

(ii) $\gamma(x, a) \leq \mu(y, b)$ *iff* $a \otimes b \otimes \delta \leq I(x, y)$*,*

with $\mathrm{fix}(*) = \{a \mid a^* = a\}$ *denoting the set of all fixpoints of* **.*

3.3 Summary and topics for future work

We presented the approach of reducing the size of the fuzzy concept lattice based on the idea of thresholds. We showed that the thresholded extent- and intent-forming operators (with hedges) ${}^{\uparrow_\delta}$ and ${}^{\downarrow_\delta}$ based on this idea form, in fact, a particular case of operators (with hedges) defined on data table with shifted fuzzy attributes, ${}^{\uparrow_{\delta\to I}}$ and ${}^{\downarrow_{\delta\to I}}$. As a result, the results and algorithms developed within the framework of hedges are automatically available for the approaches via thresholds. Although the problems of operators with thresholds and their concept lattices can be reduced to problems of operators without thresholds and their concept lattices, the idea of thresholds is intuitively appealing, the thresholds being parameters which influence the size of the resulting concept lattices.
Furthermore, we revealed that the approach of reducing the size of a fuzzy concept lattice (with hedges) via threshold equals the approach of reducing the size of a fuzzy concept lattice via factorization by similarity ${}^a{\approx}$ presented in previous chapter. This result, with respect to which was said in the

previous paragraph, is indeed not surprising, since from the previous chapter we know that the factor lattice can be easily computed from data table modified by shifts of attributes, $a \to I$. Both approaches, via thresholds and via factorization by ${}^{a}\!\approx$ are thus but the variants of the approach of reducing the size of a fuzzy concept lattice (with hedges) via shifted attributes.
Shifted attributes thus evidently play an important role in fuzzy FCA, so it certainly deserve further concern in future research. For example, the topic of upcoming research is to answer the question on the relation between concept lattices constructed from a data table with shifted fuzzy attributes using different (comparable) values of the reduction, shifting/thresholding, parameter, $\delta_1 \leq \delta_2$? We already know that the smaller parameter we use the smaller lattice we get. But are e.g. concepts of the smaller lattice $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, \delta_1 \to I\right) = \mathcal{B}\left(X^{*_X}_{\delta_1}, Y^{*_Y}_{\delta_1}, I\right)$ contained in the larger lattice $\mathcal{B}\left(X^{*_X}, Y^{*_Y}, \delta_2 \to I\right) = \mathcal{B}\left(X^{*_X}_{\delta_2}, Y^{*_Y}_{\delta_2}, I\right)$?

Acknowledgments The work presented in this chapter was supported by grant No. 1ET101370417 of GA AV ČR and by grant No. 201/05/0079 of the Czech Science Foundation, the principal investigator of which is R. Bělohlávek, and by institutional support of the Faculty of Science, Palacký University, research plan MSM 6198959214.

Conclusion

The thesis summarizes and compares the results of two approaches to reduce the size of fuzzy concept lattices (with hedges). In both approaches, the amount of reduction is parametrized by the user. The first approach goes through factorizing (i.e. providing a granularized view on) the concept lattice by similarity parametrized by user-defined parameter and the two ways of computation of factor lattice directly from input data (i.e. without first computing the whole lattice and subsequent factorizing) were described. The theoretical insight is comped with illustrative examples and extensive experiments. The second approach uses the idea of thresholds, user-defined parameters, influencing the shape of formal concepts. We reveal that the two approaches lead to the same results (isomorphic smaller concept lattices) and that both relates the approach of fuzzy FCA with shifted attributes.

References

[1] Ammons G., Mandelin D., Bodik R., Larus J. R.: Debugging temporal specifications with concept analysis. In: *Proc. ACM SIGPLAN'03 Conference on Programming Language Design and Implementation*, pages 182–195, San Diego, CA, June 2003.

[2] R Bělohlávek. Fuzzy concepts and conceptual structures: induced similarities. In: *Proc. Joint Conf. Inf. Sci.'98*, Vol. I, pages 179–182, Durham, NC, 1998.

[3] Bělohlávek R.: Similarity relations in concept lattices. *J. Logic Comput.* **10**(6):823–845, 2000.

[4] Bělohlávek R.: Fuzzy closure operators. *J. Math. Anal. Appl.* **262**(2001), 473–489.

[5] Bělohlávek R.: Fuzzy closure operators II. *Soft Computing* **7**(2002) 1, 53-64.

[6] Bělohlávek R.: Fuzzy Galois connections. *Math. Logic Quarterly* **45**,4 (1999), 497–504.

[7] Bělohlávek R.: *Fuzzy Relational Systems: Foundations and Principles.* Kluwer Academic/Plenum Publishers, New York, 2002.

[8] Bělohlávek R.: Concept lattices and order in fuzzy logic. *Annals of Pure and Applied Logic* **128**(2004), 277–298.

[9] Bělohlávek R.: Algorithms for fuzzy concept lattices. *Proc. Fourth Int. Conf. on Recent Advances in Soft Computing, RASC 2002.* Nottingham, United Kingdom, 12-13 December, 2002, pp. 200–205.

[10] Bělohlávek R., Dvořák J., Outrata J.: Fast factorization of similarity in formal concept analysis. *J. Comp. Syst. Sciences* (submitted).

[11] Bělohlávek R., Dvořák J., Outrata J.: Direct factorization in formal concept analysis by factorization of input data. In: *Proc. 5th Int. Conf. on Recent Advances in Soft Computing, RASC 2004.* Nottingham, United Kingdom, 16–18 December, 2004, pp. 578–583.

[12] Bělohlávek R., Funioková T., Vychodil V.: Galois connections with hedges. In: Yingming Liu, Guoqing Chen, Mingsheng Ying (Eds.): *Fuzzy Logic, Soft Computing & Computational Intelligence: Eleventh International Fuzzy Systems Association World Congress* (Vol. II), 2005, pp. 1250–1255. Tsinghua University Press and Springer, ISBN 7–302–11377–7.

[13] Bělohlávek R., Outrata J., Vychodil V.: On factorization by similarity of fuzzy concept lattices with hedges. In: *CLA 2006 proceedings* (accepted), 13 pp., Hammamet, Tunisia, October–November 2006.

[14] Bělohlávek R., Outrata J., Vychodil V.: Thresholds and shifted attributes in formal concept analysis of data with fuzzy attributes. In: H. Schärfe, P. Hitzler, and P. Øhrstrøm (Eds.): Proc. ICCS 2006, *Lecture Notes in Artificial Intelligence* **4068**, pp. 117–130, Springer-Verlag, Berlin/Heidelberg, 2006.

[15] Bělohlávek R., Sklenář V., Zacpal J.: Crisply Generated Fuzzy Concepts. In: B. Ganter and R. Godin (Eds.): ICFCA 2005, *LNCS* **3403**, pp. 268–283, Springer-Verlag, Berlin/Heidelberg, 2005.

[16] R Bělohlávek, V. Sklenář, J. Zacpal. Formal concept analysis with hierarchically ordered attributes. *Int. J. General Systems* (to appear).

[17] Bělohlávek R., Vychodil V.: Reducing the size of fuzzy concept lattices by hedges. In: *FUZZ-IEEE 2005, The IEEE International Conference on Fuzzy Systems*, May 22–25, 2005, Reno (Nevada, USA), pp. 663–668.

[18] Bělohlávek R., Vychodil V.: Reducing the size of fuzzy concept lattices by fuzzy closure operators. In: *Proceedings of SCIS & ISIS 2006: Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems*, 2006, pp. 309–314.

[19] Bělohlávek R., Vychodil V.: What is a fuzzy concept lattice? In: *Proc. CLA 2005, 3rd Int. Conference on Concept Lattices and Their Applications*, September 7–9, 2005, Olomouc, Czech Republic, pp. 34–45, URL: *http://ceur-ws.org/Vol-162/*.

[20] A. Burusco, R. Fuentes-Gonzáles. The study of the L-fuzzy concept lattice. *Mathware & Soft Computing*, **3**:209–218, 1994.

[21] C. Carpineto, R. Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning* **24**:95–122, 1996.

[22] Carpineto C., Romano G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.

[23] R. Cole, P. Eklund. Scalability in formal context analysis: a case study using medical texts. *Computational Intelligence* **15**:11–27, 1999.

[24] Dekel U., Gill Y.: Visualizing class interfaces with formal concept analysis. In *OOPSLA'03*, pages 288–289, Anaheim, CA, October 2003.

[25] Elloumi S. et al.: A multi-level conceptual data reduction approach based in the Łukasiewicz implication. *Inf. Sci.* **163**(4)(2004), 253–264.

[26] Fan S. Q., Zhang W. X.: Variable threshold concept lattice. *Inf. Sci.* (submitted).

[27] Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.

[28] Goguen J. A.: The logic of inexact concepts. *Synthese* **18**(1968-9), 325–373.

[29] Gottwald S.: *A Treatise on Many-Valued Logics.* Research Studies Press, Baldock, Hertfordshire, England, 2001.

[30] Hájek P.: *Metamathematics of Fuzzy Logic.* Kluwer, Dordrecht, 1998.

[31] Hájek P.: On very true. *Fuzzy Sets and Systems* **124**(2001), 329–333.

[32] D. S. Johnson, M. Yannakakis, C. H. Papadimitrou. On generating all maximal independent sets. *Inf. Processing Letters* **15**:129–133, 1988.

[33] Klir G. J., Yuan B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall, Upper Saddle River, NJ, 1995.

[34] Krajči S.: Cluster based efficient generation of fuzzy concepts. *Neural Network World* **5**(2003), 521–530.

[35] O. S. Kuznetsov, S. A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intelligence* **14**(2/3):189–216, 2002.

[36] Outrata J.: Similarity clarification in formal concept analysis. *J. Electrical Engineering* **56**(12/s) (2005), pp. 41–45.

[37] Pollandt S.: *Fuzzy Begriffe.* Springer, Berlin, 1997.

[38] Snelting G.: Reengineering of configurations based on mathematical concept analysis. *ACM Trans. Software Eng. Method.* **5**(2):146–189, April 1996.

[39] Snelting G., Tip F.: Understanding class hierarchies using concept analysis. *ACM Trans. Program. Lang. Syst.* **22**(3):540–582, May 2000.

[40] Takeuti G., Titani S.: Globalization of intuitionistic set theory. *Annal of Pure and Applied Logic* **33**(1987), 195–211.

[41] P. Valtchev, R. Missaoui, R. Godin, M. Meridji. Generating frequent itemsets incrementally: two novel approaches based on Galois lattice theory. *J. Exp. Theor. Artif. Intelligence* **14**(2/3):115–142, 2002.

[42] Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival I.: *Ordered Sets.* Reidel, Dordrecht, Boston, 1982, 445–470.

[43] Yahia S., Jaoua A.: Discovering knowledge from fuzzy concept lattice. In: Kandel A., Last M., Bunke H.: *Data Mining and Computational Intelligence*, pp. 167–190, Physica-Verlag, 2001.

[44] Zadeh L. A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets and Systems* **90**(1997), 111–127.

[45] Zaki M.: Mining Non-Redundant Association Rules. *Data Mining and Knowledge Discovery* **9**(2004), 223–248.

Index

Jan Outrata, * December 14, 1978, Šternberk, Czech Republic
jan.outrata@upol.cz

Jan Outrata graduated at Faculty of Science, Palacký University in Olomouc (Czech Republic) with a MSc. (Mgr.) degree with distinction in Computer Science in 2003. He works at Department of Computer Science, Faculty of Science, Palacký University (`http://www.upol.cz`) since 2005, but he has led practices there since 2001.
He is the author or coauthor of several papers on formal concept analysis and has been a participant of several projects and grants in the area of data mining. His interests are data analysis (formal concept analysis, similarity, clustering) and data compression (both lossless and lossy). He is also interested in (free) software development, (free) operating systems (both theory and practice), computer networks and web technologies.