

Fuzzy Logic and Partial Order; First Attempts with the New PyHasse-Program L_eval

R. Bruggemann¹, A. Kerber²

1: *Leibniz Institute of Freshwater Ecology and Inland Fisheries*

brg_home@web.de

2: *Department of Mathematics, University of Bayreuth, Germany*

kerber@uni-bayreuth.de

(Received October 26, 2017)

Abstract

Often a decision, for instance concerning environmental pollution, can be based on an evaluation, in which several attributes are simultaneously considered. After performing the evaluation, the interest is directed to the attributes, which were selected for the evaluation. The corresponding analysis is called exploration.

When measured data are at hand, statistical tools such as correlation analysis can be applied. However, when the evaluation is done by applying methods of partial order theory, then it seems to be consequent to perform the exploration by concepts of partial order, too.

Here we present methods, derived from lattice theory, and apply them to an exploration of lead, cadmium, zinc, and sulfur and their possible interaction in geographical areas. We show that different logics can be selected, and hypothesize, for example, that the pollution by Cd implies a pollution by Zn. Other implications can be found when the set of regions is restricted. Then, for example, pollution by Pb implies for these regions a pollution by Cd.

1 Introduction

It is well known that Formal Concept Analysis allows an elegant access to a set of implications among sets of indicators describing complex objects [1, 2]. The starting point is a *context*, a matrix containing truth values of objects and their attributes. We assume that the entries of the

context are elements of a lattice L , for example, parameter values, say triples of real numbers between 0 and 1, where $L=[0,1]^3$. The crucial point is that in this case we can *select different set theories and corresponding logics which allow, for example, to argue more or less strictly*. An important step to derive the context of a matrix of real data, is a process called conceptual scaling. In [1, 3] several methods for the scaling are discussed. Here we suggest an alternative, which offers to select different set theories and corresponding logics in terms of a lattice-theoretical approach [4-6]. We show how implications on the presence of chemical substances which were polluting regions of Baden-Wuerttemberg (see e.g. [7-10]) can be obtained, using different logics.

Our paper is organized as follows:

In the methods section we demonstrate first the chemical problem, from which an evaluation can be derived. In the next methods section an example of Formal Concept Analysis is shown. As the new program `L_eval` is part of the software package PyHasse, we briefly explain PyHasse and then we show how `L_eval` applies the theoretical framework of [4-6]. Finally the methods section ends with an example from chemistry and implications generated under different logics, which can be interpreted as the exploration of the evaluation.

The discussion concludes the paper. Here we also present possible developments of `L_eval`.

2 Methods

2.1 The chemistry background

The environmental protection agency of Baden-Württemberg (south-western part of Germany) started many years ago a monitoring program to check the pollution state of different areas [7-9]. The areas were selected so that they can be considered as homogeneous in their environmental, ecological and economical aspects. Here we restrict ourselves to eight regions along the river Rhine from Basel to the tributary Neckar. Among the pollutants to be considered, the total concentrations of Pb, Cd, Zn and S were measured (mg/kg dry mass). Once the quadruple of concentrations was available, the question was which of the eight regions along the Rhine needed most attention. In other words, there is an evaluation problem, based on four attributes (syn. indicators). Often the application of methods, based on partial order theory can be useful, see for instance [12-21]. The graph, obtained by partial order theory, realizes and visualizes what we call the *evaluation*, and it can be helpful for the management of these areas. Once the evaluation problem has been established, clearly the role of the four chemical elements

is to be considered: Can we hypothesize that, for example, the presence of Pb implies the presence of Cd? Will, for instance, sulfur imply one of the heavy metals? We call this kind of consideration the *exploration* of the evaluation.

2.2 Application of formal concept analysis

2.2.1 Data matrix and its transformation

The data for the eight regions were available from the environmental protection agency of Baden-Württemberg. They are shown in Table 1:

Table 1. Regions, their labels and the total concentrations (mg/kg dry mass) of lead, Pb; cadmium, Cd; zinc, Zn and sulfur, S.

Region	Label	Pb	Cd	Zn	S
Lörrach	1	1	0.04	21	1540
westwards of Freiburg	24	1.7	0.18	39	1740
Heidelberg	52	2	0.23	36	4030
south of Mülheim	10	1	0.03	29	1780
Offenburg	31	1.1	0.15	28	1740
Mannheim	56	1	0.11	34	1970
westwards of Mülheim	19	0.8	0.01	18	4030
Rastatt	43	0.5	0.11	39	4030

Tables like Table 1 we call a data matrix: Columns contain the indicator values, rows the objects of interest, here the eight regions along the Rhine. An entry of this data matrix is denoted by $q(i,j)$. Subscript i refers to the object in the i -th row, subscript j refers to the attribute in the j -th column, i.e. to the attribute. We write $q(-,j)$, when the j -th attribute is of interest, without specifying its value for a certain object. As the relationships between the geographical locations and the degree of pollution were discussed elsewhere [22], we suppress further details here. It is convenient to introduce two sets: the set of objects, X , and the set of attributes, IB (**I**nformation **B**ase):

$$X := \{1, 24, 52, 10, 31, 56, 19, 43\}$$

IB:={Pb, Cd, Zn, S}

For the following, we coarsen the data into values 0 or 1 by the following rule (eq. 1):

Let $mw(j)$ be the arithmetic mean value of j -th chemical element (= Pb or Cd or Zn or S) taken over all eight regions, then:

$$q_n(i, j) = \begin{cases} 0, & \text{if } q(i, j) < mw(j), \\ 1, & \text{otherwise.} \end{cases} \quad (1)$$

where $q(i, j)$ means the total concentration of the j -th chemical element of region i , while $q_n(i, j)$ denotes its normalized value. It is completely clear that by eq. 1 the results are depending on outliers and the distribution of the data, because the mean value is statistically not a robust measure. We suppress the corresponding analysis, as these data only serve as a demonstration.

2.2.2 Formal Concept Analysis (FCA)

It is easy to deduce a context table (Table 2) by applying a classification by eq. 1. The mean values (mg/kg dry mass) are:

Pb: 1,1375 Cd: 0,1075 Zn: 30,5 S: 2607,5

Table 2. Context table as input for the program conexp (Yevtushenko [11]). Therein, we write (instead of $q_n(-, j)$) nPb , nCd , etc. for easier understanding and write 1 for x and 0 for the empty cell.

Regions\indicators	nPb	nCd	nZn	nS
1	0	0	0	0
10	0	0	0	0
24	1	1	1	0
31	1	1	0	0
19	0	0	0	1
43	0	1	1	1
52	1	1	1	1
56	1	1	1	0

Although Formal Concept Analysis is very well represented in Mathematical Chemistry, see for instance [17, 23, 24] we give some examples, how to read the context table (Table 2):

- 1) For region 24 it is true that nPb , nCd and nZn hold, i.e. that there is a value greater than or equal to the mean value. In other words, region 24 is highly polluted by these three elements.
- 2) High pollution with respect to lead, Pb, is found in the regions: 24, 31, 52 and 56.
- 3) The regions 24 and 56 are equivalent with respect to their pollution profile, i.e. whether or not they are polluted by nPb , nCd , nZn , nS .
- 4) Regions 1, 10 are equivalent too, their profile is (0, 0, 0, 0) within this crisp binary classification.

The lattice of concepts is derived from the context table (Table 2) by application of the central condition (eq. 3):

Let A be a subset of the set of all indicators used in this study, and B a subset of X . We introduce two mappings as follows:

$$A \rightarrow A' = \{x \in X: x \text{ "has" the indicator } q(-j) \text{ for all } q(-j) \in A \subseteq IB\} \quad (2a)$$

$$B \rightarrow B' = \{q(-j) \in A: \text{ all the } x : x \in B \subseteq X \text{ "have" the indicator } q(-j)\} \quad (2b)$$

The notion "has" and "have" resp. means that in the context table the entry for a certain object x and a certain indicator has the value 1. These two mappings are called derivations.

$$\text{If } A = A' \text{ or } B = B' \text{ then the pair } [A, B] \text{ is forming a "concept".} \quad (3)$$

The concept is the central notion in Formal Concept Analysis.

A'' , B'' are the result of repeated derivation.

The concepts (object subsets and attribute subsets) can be partially ordered by the application of the subset relation and it can be shown that this partial order is a lattice [1], *the lattice of concepts*. In our paper we only need the concept of derivations, eq's. 2a and 2b.

In Figure 1, the lattice of the concepts is shown. Note that the software conexp visualizes the lattice of concepts in a reduced form, see [1, 17].

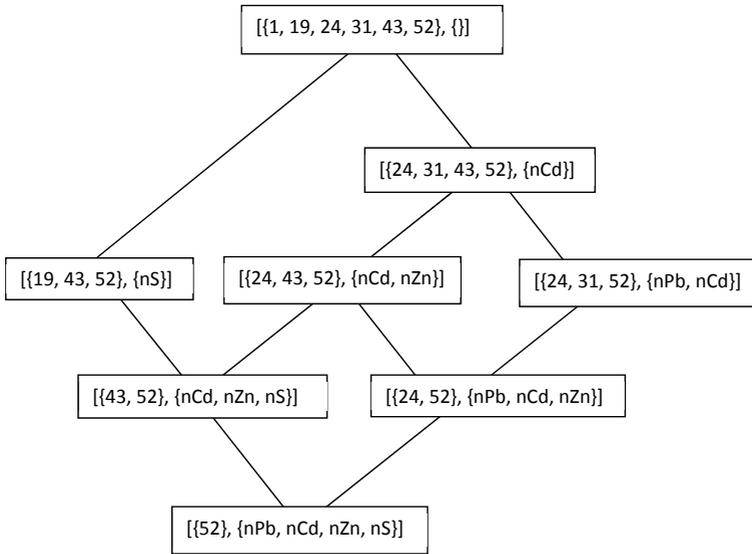


Figure 1. Lattice of concepts, i.e. of pairs of objects and attributes holding the condition of eq. 3. The parts of the pairs are subset-related. For example, upwards $\{52\} \subseteq \{43, 52\}$ and downwards: $\{nCd, nZn\} \subseteq \{nPb, nCd, nZn\}$ or $\{nCd, nZn\} \subseteq \{nCd, nZn, nS\}$.

Once again, for the sake of convenience of the reader, we give three examples:

- 1) For region 43 the pollution by S, Zn, Cd holds, but not that by Pb
- 2) A high pollution (in the sense of eq. 1) of Pb is found for regions 24, 31 and 52.
- 3) The only region, where all four attributes hold is region 52.

A conventional Hasse diagram would have region 52 at the top, because here all four chemical elements occur, whereas regions 1 and 10 would be at the bottom of the Hasse diagram, because with respect to eq. 1, they are not polluted. Figure 2 shows the Hasse diagram, as an example of an evaluation of the eight regions, based on the four indicators nPb , nCd , nZn and nS . Hasse diagrams such as that shown in Fig. 2 are constructed by means of the cover-relations which are derived from the order relations among the objects, evaluating eq. 4:

Let x, y be two objects, then:

$$x \leq y : \Leftrightarrow q(x, j) \leq q(y, j) \text{ for all } q(-, j) \in IB, x, y \in X \quad (4)$$

The case $x \leq y$ is denoted as “comparable” and the case when neither $x \leq y$ nor $y \leq x$ as “incomparable”.

Now, let x, y, z be three objects and assume that $x \leq z$ by eq. 4. If there is no y with $x \leq y \leq z$, $y \neq x, y \neq z$, then the relation $x \leq z$ is called a cover relation. By the transitivity axiom of partial order theory all other relations among the objects can be deduced from the set of cover-relations.

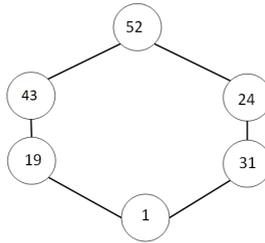


Figure 2. The Hasse diagram drawn in the conventional manner, representing the order of the objects. Note that region 10 is equivalent with region 1 and region 24 with region 56. These equivalent elements are suppressed in the drawing.

From the point of view of decision makers region 52 needs the highest priority, whereas region 1 and its equivalent region 10 are (in comparison to the other 6 regions) not highly polluted.

The program conexp [11] delivers some implications, which should be considered as geochemical hypotheses (Table 3):

Table 3. Implications as outcome of conexp. How often the premises are realized in the (transformed) data matrix is given by the numbers in <->-brackets.

Enumeration	<...>	Premise	implies	Conclusion
1	<4>	nPb	==>	nCd
2	<4>	nZn	==>	nCd
3	<2>	nCd, nS	==>	nZn

Within the transformation, and the selection of regions the hypotheses would be:

High pollution by lead implies a high pollution by cadmium. This is plausible as often by mining activities lead and cadmium are simultaneously found.

Similarly Zn also implies Cd, and finally, when the pollution by Cd and by S is high, then the hypothesis is that in this case Zn is also highly polluting. It should be realized that an implication of the form “S implies...” is not found.

These hypothetic implications can easily be verified by checking Table 2. Nevertheless the automatic hypothesis-generation is crucial, as the classification into two values (“white”: 0, “black”: 1) is very crude and in some cases the measured values are near (in the sense of a numerical difference) to the mean values so that the need of a finer classification is evident. Here is the starting point of the scaling process in FCA. We, however, will present an alternative and some basics will be explained in the next subsection.

2.3 L_eval as a module of PyHasse

2.3.1 PyHasse

PyHasse is a program package, written in Python 2.6 and developed by Bruggemann, see also [25]. The user interface is constructed, applying the TCL/TK-derived Tkinter software. The aim of PyHasse is the analysis of the following equation, defining a partial order relation, based on several indicator values $q(-,j)$, eq. 4.

Actually PyHasse contains 140 single programs, most of them equipped with analogous graphical user interface, help - and about - functions. The about-function informs about the aim of the program, whereas the help-function includes a step - by - step tutorial and some additional information (for example whether or not the program is still in a testing mode). The PyHasse software is steadily extended and modified, as it is the technical tool to perform actual scientific tasks. There is also an Internet-version available, with some basic tools and a detailed background information. Its development is done by a team of four scientists: P. Koppatz, V.Pratz, H. Richter and R. Bruggemann, see www.PyHasse.org and [26].

2.3.2 L_eval, general

Actually L_eval is still to be tested, its current version is 9_2. The task of L_eval09_2.py is as mentioned above, to practically apply the theoretical framework presented in [4-6]. Most of the PyHasse programs provide a Hasse diagram, based on the data matrix used. In Figure 3 the Hasse diagram is shown, directly based on the data of Table 1, applying eq. 4.

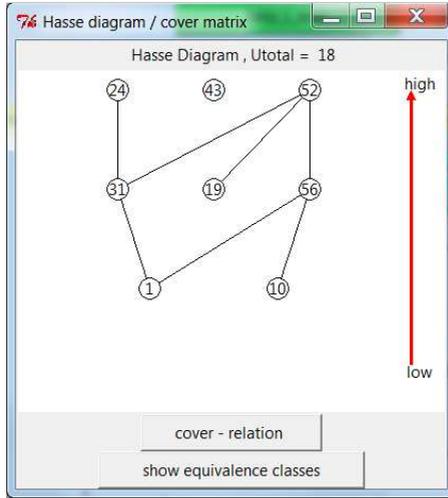


Figure 3. Hasse diagram, based on the measured data, as shown in Table 1.

The evaluation of the eight regions along the Rhine by the Hasse diagram shown in Figure 3 shows some remarkable differences to that, based on classified data in Figure 2. Region 43 is isolated, which is always a sign for a special data profile, worth to be studied in more detail. Inspecting Table 1 it can be seen that region 43 has the lowest value in Pb, but the largest values in Zn and S. The regions 24 and 56 (evaluated as equivalent when crisp $\{0,1\}$ -data are used) are now incomparable. There are also coincident order relations, for example $1 < 31$, $31 < 24$, $1 < 56$ and $19 < 52$ are reproduced in Figure 2.

The specific task in the current state of L_eval is to find answers to the following question:

Given a subset X' of the set X , what are the truth values, $tv \in [0, 1]$ for implications of the form

$$\{q(\cdot, j)\} \Rightarrow IB' \subseteq IB \tag{5}$$

For example what will be the truth value, tv , of the implication $Zn \Rightarrow Cd, S$? Due to eq. 5 we avoid to test implications of the form $Pb, S \Rightarrow Cd, Zn$, i.e. with a premise given by more than one indicator (see later).

Equation 5 must be evaluated, by assigning a certain truth value, tv , when the membership of elements is not crisp (answerable by “yes” or “no”). The starting point is to apply a certain t-norm (see [4-6]). The t-norm quantifies the intersection of two sets. Once the t-norm is selected,

i.e. once a certain logic is selected, then the central derivation operation must be evaluated by a truth value, a measure with values in $[0,1]$. The central concept to arrive at this goal is the residuum of the t-norm, which is a measure for a non-crisp subset-relation.

The program `L_eval09_2` can evaluate four different t-norms (see below).

In order to avoid too extensive outputs, it is recommended but not necessary, to

- a) check only premises, built by a single indicator (according to eq. 5),
- b) to insert a user-defined limiting value: only implications whose truth-values are greater than or equal to the limiting value are rendered. Note that once the t-norm is selected, the calculation can be done without any further manipulation by the user, in contrast to the scaling procedure in the conventional FCA.
- c) optionally, the user can select subsets of the set X of all objects and perform the analysis only for these. This corresponds to the fact that implications by `conexp` also are referring to the subset of objects which fulfill the premise.

Before `L_eval` is explained in more detail, it is needed to report the basic mathematics behind this program (see subsection 2.3.4).

2.3.3 Performance

It should be clear that `L_eval09_2` is by far not such a general program as `conexp`, as even the crucial Duquenne/Guigues-basis [4-6, 27] is still not available. This concept is extremely useful to obtain all attribute implications from a minimal set of implications. The new program `L_eval` simply checks all possible subsets of IB , which becomes quickly a computational hard problem. The performance of `L_eval09_2` was tested: With the WINDOWS 7 - operating system, and the interpreter programming language Python 2.6, the evaluation of a $7*7$ data matrix needed 20 minutes. A more detailed analysis of the performance is planned, when the full functionality of `L_eval` is reached.

2.3.4 Basic mathematics

As already mentioned, we do not repeat the theoretical background, cf. [4-6] of the generalized set theory and the corresponding logics. Here are just the main aspects.

Step 1: Whether or not an element is in the intersection of two sets is quantified by a t-norm, τ .

In the following we assume that all data are elements of [0,1]. (In `L_eval` a [0,1]-normalisation by eq. 6 is automatically performed.)

Let x be a real number, with $x \in [min, max]$, then the normalization procedure is defined as follows:

$$xn = (x - min)/(max - min) \tag{6}$$

Once again, this procedure is not statistically robust, but sufficient for our purpose.

Four t-norms can be selected in `L_eval`, with x,y as scalars taken from the closed interval [0,1], they are defined as follows:

standard norm: $s(x,y) := \min\{x,y\}$ with $x,y \in [0,1]$. (7)

drastic norm: $d(x,y) := \begin{cases} x, & \text{if } y = 1 \\ y, & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$ (8)

algebraic product (if $x,y \in [0,1]$): $a(x,y) := x*y$ (9)

Lukasiewicz-norm [28], sometimes called the “restricted difference”:

$$b(x,y) := \max\{0, x+y-1\} \tag{10}$$

In [4] an example is given, taken from elementary chemistry, how the selection of different t-norms leads to different results, i.e. “we are using some kind of semantic notion of truth”, based on the t-norm.

Step 2: The suitable measure for a generalized subset-set relation is the residuum of a t-norm. It can be derived from the four t-norms, mentioned above:

The residuum of standard norm:

$$\hat{s}(x, y) = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases} \quad (11)$$

The residuum of drastic norm:

$$\hat{d}(x, y) = \begin{cases} y & \text{if } x=1 \\ 1 & \text{otherwise} \end{cases} \quad (12)$$

In Table 4 three pairs of scalars are shown for a demonstration.

Table 4. Calculation of residua, expressing as to how far $x \leq y$ is true.

	(x,y) = (0.3, 0.7)	(x,y) = (1, 0.2)	(x,y) = (0.7,0.3)
\hat{s}	1	0.2	0.3
\hat{d}	1	0.2	1

The other residua - albeit realized in L_eval09_2 - will not be considered here, the interested reader should consult [4-6]. Generally we write τ^* for the residuum of a t-norm τ .

Step 3: In analogy of the usage of the context table we want to quantify the fact that an object “has” an attribute. In consequence of the approach proposed here, the question asks for a value in the range between 0 and 1. This question has two aspects:

3.1: We must select, from which attribute we want to quantify the “has”-relation, also called “holds” -relation.

3.2: the step 3.1 must be carried out for each object of interest.

In [4-6] the step 3.1 is very generally formulated, as we could think of a membership function over all attributes, which is formulated as the image of a mapping A .

$$A : IB \rightarrow [0,1]^m$$

The mapping A applied on IB could be for $m = 4$, i.e. for $(q(-,1), q(-,2), q(-,3), q(-,4))$ a quadruple like $(0.2, 0.6, 0.2, 0)$, expressing that we are interested in a fuzzy set, where $q(-,2)$ has the major contribution, whereas $q(-,1)$ and $q(-,3)$ have the same, but relatively small contribution and $q(-$

,4) does not contribute at all. This is the generalization, needed when there is no crisp subset-relation, and corresponds with eq. 2a or 2b.

This very general formulation is, however, for practical purposes not necessary. What is wanted, is to know, whether or not a certain attribute, say $q(-, 2)$ implies for instance $q(-, 1)$. In other words, an exploration should be based on *crisp* subsets of IB , i.e., we want to know whether or not for instance sulfur implies lead and not whether or not a fraction of sulfur implies some other fraction of lead. This means, the exploration has to be performed between well defined subsets of IB . Therefore the images of the mapping A (describing the premise of an implication) and of mapping B (describing the conclusion of an implication)

$$B: IB \rightarrow [0,1]^m$$

can be given by characteristic set theoretical functions, taking values in the set $\{0,1\}$. Whether or not $q(-, 2)$ implies $q(-, 1)$ is expressed by:

$(0,1,0,0)$ implies $(1,0,0,0)$ to a certain degree (a truth value),

where these quadruples are selecting the attributes from IB .

Whether or not the quadruple $(0,1,0,0)$ holds for example for region 31 with a certain truth value, tv , needs the application of a residuum (appropriately selected) of $(0,1,0,0)$ on the data values found for object 31:

$$tv(31 \text{ has Cd}) = \min\{ \tau^*(A(q(-, j)), q(31,1)), j=1,2,3,4 \} \quad (13)$$

Equation 13 written explicitly:

$$\begin{aligned} & tv(31 \text{ has } (0,1,0,0)), \text{ i.e. } tv(31 \text{ has only Cd}) = \\ & = \min\{ \tau^*(0, q(31,1)), \tau^*(1, q(31,2)), \tau^*(0, q(31,3)), \tau^*(0, q(31,4)) \} \end{aligned}$$

Let us assume that we are applying the standard norm, then $tv(31 \text{ has only Cd})$ is easily evaluated:

$0 < q(31, j)$, $j=1, 3, 4$ therefore we obtain three times a 1. The only exception is $\tau^*(1, q(31, 2))$. Here the residual norm based on the standard t-norm is $q(31, 2)$. Hence: $tv(31 \text{ has only Cd}) = \min\{ 1, 1, q(31, 2), 1 \} = q(31, 2)$.

The “hold” or “has” relation is usually and in accordance with the concepts of FCA called a derivation [1], denoted as A' with:

$$A'(x) = \min\{\tau^*(A(q(-, j)), q(x, j)), j=1, \dots, m\} \text{ for an object } x \in X \quad (14)$$

Equation 14 generates a non-crisp set, where each objects get a tv-value, being in general a number in $[0,1]$. I.e. eq. 14 starts from a crisp set but a fuzzy subset of X is the result.

Step 4: Implications. The second mapping B is introduced, as the task is: whether one subset of IB (image of A) implies another subset of IB (image of B). The implication is evaluated as a subset-set-relation as follows:

$$tv(A' \subseteq B') \quad (15)$$

Continuing the example: What is the truth value of the “has”-relation for all objects concerning the premise $(0, 1, 0, 0)$ and concerning the conclusion $(1, 0, 0, 0)$. In consequence of the generalized subset-set-relation, Kerber [4] shows that $tv(A' \subseteq B')$ is to be evaluated by the following expression:

$$tv(A' \subseteq B') = \min\{\tau^*(A''(x), B'(x))\}, \text{ taken over all } x \in X' \subseteq X \quad (16)$$

If for example X' is just $\{31, 52\}$, and the standard norm is assumed, recalling that the premise is $(0,1,0,0)$ and the conclusion is $(1,0,0,0)$, eq. 10 is analyzed by the following steps:

(i) In $\min\{\tau^*(A''(31), B'(31)), \tau^*(A''(52), B'(52))\}$ we start with the first term $\tau^*(A''(31), B'(31))$:

$$\tau^*(A''(31), B'(31)) = \tau^*(\min\{\tau^*(0, q(31,1)), \tau^*(1, q(31,2)), \tau^*(0, q(31,3)), \tau^*(0, q(31,4))\}, \min\{\tau^*(1, q(31,1)), \tau^*(0, q(31,2)), \tau^*(0, q(31,3)), \tau^*(0, q(31,4))\})$$

(ii) $\tau^*(A''(31), B'(31)) = \tau^*(\min\{1, q(31,2), 1, 1\}, \min\{q(31,1), 1, 1, 1\}) = \tau^*(q(31,2), q(31,1))$

analogously we obtain for the second term, $\tau^*(A''(52), B'(52))$:

(iii) $\tau^*(A''(52), B'(52)) = \tau^*(\min\{1, q(52,2), 1, 1\}, \min\{q(52,1), 1, 1, 1\}) = \tau^*(q(52,2), q(52,1))$

and finally:

$$(iv) tv(A' \subseteq B') = \min\{\tau^*(q(31, 2), q(31, 1)), \tau^*(q(52, 2), q(52, 1))\}$$

A further evaluation needs the knowledge of the data $q(i, j)$, with $i \in \{0, \dots, |X'|\}$, $X' \subseteq X$ and $j \in \{0, \dots, |B|\}$.

L_eval09_2 does these four (somewhat tedious and error-prone) steps automatically, whereby the user has several possibilities to control the calculations (section 2.4).

2.4 L_eval09_2 in detail

Figure 4 shows the user interface of L_eval09_2.

We describe the application of L_eval09_2 by starting from the top of Figure 4 and proceeding downwards.

The buttons: about and help inform about the aim of the program, the steps the user has to do and optional actions.

“*Select filename*”: Pressing this button a browser opens and the user can select his file, containing the data matrix of interest. A controlling output is immediately written below, confirming the selected filename and the path used in the individual file system of the user (see Figure 4).

“*Show data matrix*”: A list of the objects, the indicators and the data matrix itself can be obtained.

“*Draw HD (mandatory)*”: Even if one is not interested in the evaluative part, i.e. in the Hasse diagram, this button is to be used, to start internally all calculations, needed for the partial order framework.

The next four entry-boxes are very important to control the calculation:

1) The four t-norms as shown above can be selected, the residua will correspondingly be calculated. Label 1 stands for standard norm, label 2 for the drastic norm, label 3 for the Lukasiewicz-norm and label 4 for the product norm.

2) The truth values are contained in the interval $[0, 1]$. In order to reduce the output, those implications are suppressed whose truth-value are less than the limiting value, selected by the

user. In other words, checking Figure 4, we see that all implications will be presented in the output, because the limiting value was selected to be 0.

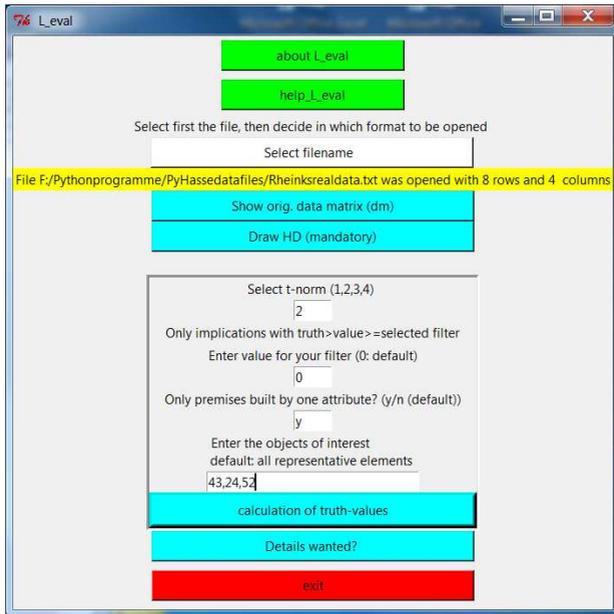


Figure 4. User interface of L_eval09_2, assuming the drastic norm, a limiting value for the truth value of the implications as 0, and the subset $X' = \{43, 24, 52\}$.

3) As explained above we are interested in premises built by only one attribute.

4) The next button “Calculation of truth-values” is the heart of this program.

After pressing this button the output is obtained, Figure 5.

File = F:/Pythonprogramme/PyHassedatafiles/Rheinksrealdata.txt				
implications filtered by 0.0				
Actually used data matrix				
0.33	0.14	0.14	0.0	
0.33	0.09	0.52	0.1	
0.8	0.77	1.0	0.08	
0.4	0.64	0.48	0.08	
0.2	0.0	0.0	1.0	

0.0	0.45	1.0	1.0
1.0	1.0	0.86	1.0
0.33	0.45	0.76	0.17

drastic norm

premises only by one attribute

Analyse

concerning the set of objects as follows

1, 10, 24, 31, 19, 43, 52, 56,

S, implies --empty--with truth-value 1.0

S, implies Zn, with truth-value 0.0

S, implies Zn, S, with truth-value 0.0

S, implies Cd, with truth-value 0.0

S, implies Cd, S, with truth-value 0.0

S, implies Cd, Zn, with truth-value 0.0

S, implies Cd, Zn, S, with truth-value 0.0

S, implies Pb, with truth-value 0.0

S, implies Pb, S, with truth-value 0.0

S, implies Pb, Zn, with truth-value 0.0

S, implies Pb, Zn, S, with truth-value 0.0

S, implies Pb, Cd, with truth-value 0.0

S, implies Pb, Cd, S, with truth-value 0.0

S, implies Pb, Cd, Zn, with truth-value 0.0

S, implies Pb, Cd, Zn, S, with truth-value 0.0

Zn, implies --empty--with truth-value 1.0

Zn, implies S, with truth-value 0.08

Zn, implies Zn, S, with truth-value 0.08

Zn, implies Cd, with truth-value 0.455

Zn, implies Cd, S, with truth-value 0.08

Zn, implies Cd, Zn, with truth-value 0.455
Zn, implies Cd, Zn, S, with truth-value 0.08
Zn, implies Pb, with truth-value 0.0
Zn, implies Pb, S, with truth-value 0.0
Zn, implies Pb, Zn, with truth-value 0.0
Zn, implies Pb, Zn, S, with truth-value 0.0
Zn, implies Pb, Cd, with truth-value 0.0
Zn, implies Pb, Cd, S, with truth-value 0.0
Zn, implies Pb, Cd, Zn, with truth-value 0.0
Zn, implies Pb, Cd, Zn, S, with truth-value 0.0
Cd, implies --empty--with truth-value 1.0
Cd, implies S, with truth-value 1.0
Cd, implies Zn, with truth-value 0.857
Cd, implies Zn, S, with truth-value 0.857
Cd, implies Cd, S, with truth-value 1.0
Cd, implies Cd, Zn, with truth-value 0.857
Cd, implies Cd, Zn, S, with truth-value 0.857
Cd, implies Pb, with truth-value 1.0
Cd, implies Pb, S, with truth-value 1.0
Cd, implies Pb, Zn, with truth-value 0.857
Cd, implies Pb, Zn, S, with truth-value 0.857
Cd, implies Pb, Cd, with truth-value 1.0
Cd, implies Pb, Cd, S, with truth-value 1.0
Cd, implies Pb, Cd, Zn, with truth-value 0.857
Cd, implies Pb, Cd, Zn, S, with truth-value 0.857
Pb, implies --empty--with truth-value 1.0
Pb, implies S, with truth-value 1.0
Pb, implies Zn, with truth-value 0.857
Pb, implies Zn, S, with truth-value 0.857
Pb, implies Cd, with truth-value 1.0

Pb, implies Cd, S, with truth-value 1.0
Pb, implies Cd, Zn, with truth-value 0.857
Pb, implies Cd, Zn, S, with truth-value 0.857
Pb, implies Pb, S, with truth-value 1.0
Pb, implies Pb, Zn, with truth-value 0.857
Pb, implies Pb, Zn, S, with truth-value 0.857
Pb, implies Pb, Cd, with truth-value 1.0
Pb, implies Pb, Cd, S, with truth-value 1.0
Pb, implies Pb, Cd, Zn, with truth-value 0.857
Pb, implies Pb, Cd, Zn, S, with truth-value 0.857

Figure 5. Output of *L_eval09_2*, see text.

In Figure 5 first the controlling conditions are repeated in order to obtain an unambiguous documentation of the calculation. The main result is the list of the implications. They are written in the form: <name of the attribute>, <names of the attributes of some subsets of *IB*>, “with truth-values”, <truth-value, tv>.

5) Pressing the button “Details”, further information can be obtained. Beside others, the derivations of any attribute subset can be checked. With respect to the actual state of programming, this button is mainly used for testing the program and will be the basis for further developments.

2.5 Exemplary exploration of the data by *L_eval*, using two different t-norms

The application of *L_eval*, concerning the full object set, two subsets and standard norm as well as drastic norm is condensed in Table 5. The basic assumptions are summarized in the following lines.

Target: herb layer in 8 regions: 1, 10, 19, 24, 31, 43, 52, 56

Chemicals: Pb, Cd, Zn and S

The conexp - analysis found with *crisp* data:

Pb \Rightarrow Cd (premise realized for regions 31, 24, 52) (here equivalent objects are suppressed)

Zn \Rightarrow Cd (premise realized for regions 24, 43, 52) (here equivalent objects are suppressed)

Summary: Couples (Pb,Cd) and (Zn,Cd) are motivated by the crisp {0,1} analysis performed by conexp.

The subset X_1 : 31, 24, 52 : relevant for $Pb \Rightarrow Cd$ in crisp analysis (conexp)

The subset X_2 : 43, 24, 52: relevant for $Zn \Rightarrow Cd$ in crisp analysis (conexp)

Table 5. Comparison of two different t-norms and their effect on the truth values of two couples of chemical elements (Pb, Cd) and (Zn, Cd).

tv-values	standard		standard		standard	
	drastic		drastic		drastic	
object set	all objects		set X_1		set X_2	
$Pb \Rightarrow Cd$	0	1	0.773	1	0.773	1
$Cd \Rightarrow Pb$	0	1	0.4	1	0	1
$Zn \Rightarrow Cd$	0.091	0.455	0.773	0.773	0.455	0.455
$Cd \Rightarrow Zn$	0.476	0.857	0.857	0.727	0.857	0.857

3 Results and Discussion

3.1 Results

Standard norm: The only implication among the four implications shown in Table 5 with a truth value $\gg 0$ is that of $Cd \Rightarrow Zn$. Nevertheless, the truth-value is less than 1, hence the hypothesis that

$Cd \Rightarrow Zn$ for all 8 regions is questionable.

The selection of proper subsets of X can enhance the values of the tv , because objects, whose values in the final evaluation are too low, could be eliminated. In Table 5 there are two implications for the regions of set X_1 , with the truth value 0.773: $Pb \Rightarrow Cd$ and $Zn \Rightarrow Cd$, whereas the truth value of $Cd \Rightarrow Zn$ is enhanced from 0.476 (all 8 regions) to 0.857 (regions of set X_1).

The drastic norm delivers in all cases values larger than or equal to those of the standard norm. Checking the definition makes this plausible, because the conditions when the drastic norm yields 1 are more often fulfilled.

3.2 Discussion

The program `L_eval` calculates the truth values of implications, by just inserting controlling inputs (most of them are even equipped with default values, so that the user may first play with the program without specifying the calculation's conditions). There are only few buttons to press, and the exploration in terms of implications is performed. The tedious manually performed procedure following the eq's: 13-16 which also is error-prone is automatically done by this new `PyHasse` program. As could be seen in section 2.5, the selection of different t-norms has a remarkable influence on the set of implications. A multitude of t-norms can be defined. Unfortunately, there are still missing rules of thumb, which t-norm should be selected according to a given evaluation/exploration. As the theory is based on the assumption that the data are real, conventional statistics may be helpful in exploring this problem. Nevertheless, a systematic work is planned but - clearly - results cannot be expected in the near future, because of the high amount of work, due to four different t-norms and a series of data matrices, where the correlation between indicators is systematically varied. The method presented here is suggested as an alternative, which could replace the conceptual scaling in conventional FCA. The evaluation of the scaling process is somewhat ambivalent: On the one hand the different methods for a scaling can be seen as modeling, which is the positive aspect, on the other hand there is some subjectivity in the scaling, which can be considered as a disadvantage, because the conceptual scaling requires user defined manipulations (see also [28]). In contrast to the conceptual scaling the fuzzy concept can be performed, once the t-norm is selected, without any user input. The mentioned controlling inputs, such as limiting value for truth values are introduced just to keep the outputs small. The user can easily disregard these controlling features. The arbitrariness of our method is the freedom to select a suitable set theory and the corresponding logic. Whereas by eq. 3 the conventional FCA renders defined subsets of attributes and objects, the `PyHasse` modul allows to select arbitrary subsets and to check the validity by means of the truth values.

Another important point is the programming of the Duquenne/Guigues basis, by which the list of outputs can be restricted to those minimal sets of implications from which all other implications can be deduced.

A pretty tricky problem is the representation of data: Rounding may affect the result as well as the internal representation of data in Python.

References

- [1] B. Ganter, R. Wille, *Formale Begriffsanalyse Mathematische Grundlagen*, Springer-Verlag, Berlin, 1996.
- [2] R. Wille, Formal concept analysis as mathematical theory of concepts and concept hierarchies, in: B. Ganter (Ed.), *Formal Concept Analysis*, Springer, Berlin, 2005, pp. 1–33.
- [3] B. Ganter, R. Wille, Conceptual scaling, in: F. Roberts (Ed.), *Applications of Combinatorics and Graph theory to the Biological and Social Sciences*, Springer, New York, 1989, pp. 139–167.
- [4] A. Kerber, Evaluation, considered as problem orientable mathematics over lattices, in: M. Fattore, R. Bruggemann (Eds.), *Partial Order Concepts in Applied Sciences*, Springer, Cham, 2017, pp. 87–103.
- [5] A. Kerber, Posets and lattices, contexts and concepts, *MATCH Commun. Math. Comput. Chem.* **54** (2005) 551–560.
- [6] A. Kerber, R. Bruggemann. Problem driven evaluation of chemical compounds and its exploration, *MATCH Commun. Math. Comput. Chem.* **73** (2015) 577–618.
- [7] K. Kreimes, Ansätze zur vergleichenden Bewertung von regionalen Belastungszuständen, in: Wasserforschung e.V., *Methodische Ansätze in der Umweltbewertung Dokumentation eines Workshops vom 20. November 1998 im Rahmen des Forschungsvorhabens "Nachhaltige Wasserwirtschaft - Entwicklung eines Bewertungs- und Prüfungssystems"*. IFV Wasserforschung e.V., Berlin. 1999, pp. 81–92.
- [8] K. Kreimes, Ökologisches Wirkungskataster Baden-Württemberg - Bewertung und zusammenfassende Darstellung von Untersuchungsergebnissen, in: U. Arndt, A. Fomin, S. Lorenz (Eds.), *Bio-Indikation; Neuere Entwicklungen - Nomenklatur - Synökologische Aspekte; Beiträge und Diskussion I. Hohenheimer Workshop zur Bioindikation am Kraftwerk Altbach-Deizisau, 1995*, Günter Heimbach, Ostfildern, 1996, pp. 160–169.
- [9] K. Kreimes, Bewertungs- und Darstellungsverfahren - Verfahren zur Bewertung und Darstellung von Bioindikationsdaten, *UWSF - Z. Umweltchem. Ökotox.* **8** (1996) 287–292.
- [10] K. De Loof, B. De Baets, H. De Meyer, R. Bruggemann, A lattice – theoretic approach to computing averaged ranks illustrated on pollution data in Baden-Württemberg, in: O. Hryniewicz, J. Studzinski, A. Szediw (Eds.), *Environmental Informatics and Systems Research, Volume 2: Workshop and Application papers*, Shaker Verlag, Aachen, 2007, pp. 177–179.

- [11] S. A. Yevtushenko, System of data analysis 'concept explorer'. Proceedings of the 7th National Conference on Artificial Intelligence, Russia, K II-2000, 2000, pp. 127–134. Software available from <http://sourceforge.net/projects/conexp>
- [12] S. Pudenz, R. Brüggemann, K. Voigt, G. Welzl. Nachhaltige Entwicklung von Managementstrategien - Multikriterielle Bewertungs- und Entscheidungshilfe-Instrumente, *UWSF - Z. Umweltchem. Ökotox.* **14** (2002) 52–57.
- [13] G. Restrepo, M. Weckert, R. Brüggemann, S. Gerstmann, H. Frank, Ranking of refrigerants. *Environ. Sci. & Techn.* **42** (2008) 2925–2930.
- [14] P. B. Sørensen, D. Lerche, M. Thomsen, Developing decision support based on field data and partial order theory, in: R. Brüggemann, L. Carlsen (Eds.), *Partial Order in Environmental Sciences and Chemistry*, Springer, Berlin, 2006, pp. 260–283.
- [15] K. Voigt, R. Brüggemann, H. Scherb, I. Cok, B. Mazmanci, M. A. Mazmanci, C. Turgut, K. W. Schramm, Evaluation of organochlorine pesticides in breast milk samples in Turkey applying features of the partial order technique, *Int. J. Environ. Health* **23** (2013) 226–246.
- [16] M. Fattore, R. Brüggemann, *Partial Order Concepts in Applied Sciences*, Springer, Cham, 2017.
- [17] R. Brüggemann, G. P. Patil, *Ranking and Prioritization for Multi-Indicator Systems – Introduction to Partial Order Applications*, Springer, New York, 2011.
- [18] L. Carlsen, R. Brüggemann, Partial order methodology: a valuable tool in chemometrics, *J. Chemometrics* **28** (2013) 226–234.
- [19] R. Brüggemann, H. G. Bartel, A theoretical concept to rank environmentally significant chemicals, *J. Chem. Inf. Comp. Sci.* **39** (1999) 211–217.
- [20] R. Brüggemann, S. Pudenz, K. Voigt, A. Kaune, K. Kreimes, An algebraic/graphical tool to compare ecosystems with respect to their pollution. IV: Comparative regional analysis by Boolean arithmetics, *Chemosphere* **38** (1999) 2263–2279.
- [21] P. Annoni, R. Brüggemann, L. Carlsen, Peculiarities in multidimensional regional poverty, in: M. Fattore, R. Brüggemann (Eds.), *Partial Order Concepts in Applied Sciences*, Springer, Cham, 2017, pp. 121–133.
- [22] R. Brüggemann, A. Kaune, D. Komossa, K. Kreimes, S. Pudenz, K. Voigt, Anwendungen der Theorie partiell geordneter Mengen in Bewertungsfragen, *DGM* **41** (1997) 205–209.
- [23] N. Y. Quintero, G. Restrepo, Formal concept analysis applications in chemistry: from radionuclides and molecular structure to toxicity and diagnosis, in: M. Fattore, R. Brüggemann (Eds.), *Partial Order Concepts in Applied Sciences*, Springer, Cham, 2017, pp. 207–217.

- [24] N. Y. Quintero, G. Restrepo, M. Cohen, Relating β^+ radionuclides' properties by order theory, *J. Radioanal. Nucl. Chem.* **298** (2013) 1937–1946.
- [25] R. Bruggemann, L. Carlsen, K. Voigt, R. Wieland, PyHasse software for partial order analysis, in: R. Bruggemann, L. Carlsen, J. Wittmann (Eds.), *Multi-Indicator Systems and Modelling in Partial Order*, Springer, New York, 2014, pp. 389–423.
- [26] P. Koppatz, R. Bruggemann, PyHasse and cloud computing, in: M. Fattore, R. Bruggemann (Eds.), *Partial Order Concepts in Applied Sciences*, Springer, Cham, 2017, pp. 291–300.
- [27] V. Duquenne, Contextual implications between attributes and some representation properties for finite lattices, in: B. Ganter, R. Wille, K. E. Wolff (Eds.), *Beiträge zur Begriffsanalyse*, BI Wissenschaftsverlag, Mannheim, 1987, pp. 213–239.
- [28] S. Pollandt, *Fuzzy-Begriffe - Formale Begriffsanalyse unscharfer Daten*, Springer-Verlag, Berlin, 1997.