Florent Domenach, Dmitry I. Ignatov, Jonas Poelmans (Eds.)

# ICFCA 2012 – International Conference on Formal Concept Analysis

Contributions to the 10th International Conference on Formal Concept Analysis (ICFCA 2012)
May 2012, Leuven, Belgium

**Volume Editors**

Florent Domenach
Department of Computer Science
University of Nicosia, Cyprus

Dmitry I. Ignatov
School of Applied Mathematics and Information Science
National Research University Higher School of Economics, Moscow, Russia

Jonas Poelmans
Faculty of Business and Economics
Katholieke Universiteit Leuven, Belgium

Printed in Belgium by the Katholieke Universiteit Leuven with ISBN 978-9-08-140995-7.

The proceedings are also published online on the CEUR-Workshop website in volume Vol-876 of a series with ISSN 1613-0073.

# Preface

This volume contains the papers presented at the 10th International Conference on Formal Concept Analysis (ICFCA 2012) held from May $7^{\text{th}}$ to May $10^{\text{th}}$, at the Katholieke Universiteit Leuven, Belgium.

There were 68 submissions by authors from 27 countries. Each submission was reviewed by at least three program committee members, and twenty regular papers (29%) were accepted for the Springer Proceedings. The program also included six invited talks on topical issues: Recent Advances in Machine Learning and Data Mining, Mining Terrorist Networks and Revealing Criminals, Concept-Based Process Mining, and Scalability Issues in FCA and Rough Sets. The corresponding abstracts are gathered in the first section of the Springer volume. Another fourteen papers were assessed as valuable for discussion at the conference and were therefore collected in this volume.

Formal Concept Analysis emerged in the 1980's from attempts to restructure lattice theory in order to promote better communication between lattice theorists and potential users of lattice theory. Since its early years, Formal Concept Analysis has developed into a research field in its own right with a thriving theoretical community and a rapidly expanding range of applications in information and knowledge processing including visualization, data analysis, and knowledge management.

The conference aims to bring together researchers and practitioners working on theoretical or applied aspects of Formal Concept Analysis within major related areas such as Mathematics, Computer and Information Sciences and their diverse applications to fields such as Software Engineering, Linguistics, Life and Social Sciences.

We would like to thank the authors and reviewers whose hard work ensured presentations of very high quality and scientific vigor. In addition, we express our deepest gratitude to all Program Committee and Editorial Board members as well as external reviewers, especially to Bernhard Ganter, Claudio Carpineto, Frithjof Dau, Sergei Kuznetsov, Sergei Obiedkov, Sebastian Rudolf and Stefan Schmidt for their advice and support.

We would like to acknowledge all sponsoring institutions and the local organization team who made this conference a success. In particular, we thank Amsterdam-Amstelland Police, IBM Belgium, OpenConnect Systems, Research Foundation Flanders, and Vlerick Management School.

We are also grateful to Katholieke Universiteit Leuven for publishing this volume and the developers of the EasyChair system which helped us during the reviewing process.

May, 2012

Florent Domenach
Dmitry I. Ignatov
Jonas Poelmans

# Organization

The International Conference on Formal Concept Analysis is the annual conference and principal research forum in the theory and practice of Formal Concept Analysis. The inaugural International Conference on Formal Concept Analysis was held at the Technische Universität Darmstadt, Germany, in 2003. Subsequent ICFCA conferences were held at the University of New South Wales in Sydney, Australia, 2004, Université d'Artois, Lens, France, 2005, Institut für Algebra, Technische Universität Dresden, Germany, 2006, Université de Clermont-Ferrand, France, 2007, Université du Québec à Montréal, Canada, 2008, Darmstadt University of Applied Sciences, Germany, 2009, Agadir, Morocco, 2010, and University of Nicosia, Cyprus, 2011. ICFCA 2012 was held at the Katholieke Universiteit Leuven, Belgium. Its committees are listed below.

## Conference Chair

| | |
|---|---|
| Jonas Poelmans | Katholieke Universiteit Leuven, Belgium |

## Conference Organization Committee

| | |
|---|---|
| Guido Dedene | Katholieke Universiteit Leuven, Belgium |
| Stijn Viaene | Vlerick Management School, Belgium |
| Aimé Heene | Ghent University, Belgium |
| Jasper Goyvaerts | Katholieke Universiteit Leuven, Belgium |
| Nicole Meesters | Katholieke Universiteit Leuven, Belgium |
| Elien Poelmans | Maastricht University, Netherlands |
| Gerda Verheyden | GZA Hospitals, Antwerpen, Belgium |

## Program Chairs

| | |
|---|---|
| Florent Domenach | University of Nicosia, Cyprus |
| Dmitry I. Ignatov | Higher School of Economics, Russia |

## Editorial Board

| | |
|---|---|
| Peter Eklund | University of Wollongong, Australia |
| Sébastien Ferré | Université de Rennes 1, France |
| Bernhard Ganter | Technische Universität Dresden, Germany |
| Robert Godin | Université du Québec à Montréal, Canada |
| Robert Jäschke | Universität Kassel, Germany |

| | |
|---|---|
| Sergei O. Kuznetsov | Higher School of Economics, Russia |
| Leonard Kwuida | Zurich University of Applied Sciences, Switzerland |
| Raoul Medina | Université de Clermont-Ferrand 2, France |
| Rokia Missaoui | Université du Québec en Outaouais, Canada |
| Sergei Obiedkov | Higher School of Economics, Russia |
| Uta Priss | Edinburgh Napier University, UK |
| Sebastian Rudolph | Karlsruhe Institute of Technology, Germany |
| Stefan Schmidt | Technische Universität Dresden, Germany |
| Bariş Sertkaya | SAP Research Center Dresden, Germany |
| Gerd Stumme | University of Kassel, Germany |
| Petko Valtchev | Université du Québec à Montréal, Canada |
| Rudolf Wille | Technische Universität Darmstadt, Germany |
| Karl Erich Wolff | University of Applied Sciences Darmstadt, Germany |

## Program Committee

| | |
|---|---|
| Simon Andrews | Sheffield Hallam University, UK |
| Michael Bain | University of New South Wales, Australia |
| Jaume Baixeries | Polytechnical University of Catalonia, Spain |
| Peter Becker | The University of Queensland, Australia |
| Radim Belohlavek | Palacky University, Czech Republic |
| Sadok Ben Yahia | Faculty of Sciences, Tunisia |
| Karell Bertet | Université de La Rochelle, France |
| Claudio Carpineto | Fondazione Ugo Bordoni, Italy |
| Nathalie Caspard | Université Paris 12, France |
| Frithjof Dau | SAP, Germany |
| Guido Dedene | Katholieke Universiteit Leuven, Belgium |
| Stephan Doerfel | University of Kassel, Germany |
| Vincent Duquenne | Université Paris 6, France |
| Alain Gély | LITA, Université Paul Verlaine, France |
| Joachim Hereth | DMC GmbH, Germany |
| Marianne Huchard | Université Montpellier 2 and CNRS, France |
| Tim Kaiser | SAP AG, Germany |
| Mehdi Kaytoue | LORIA Nancy, France |
| Markus Krötzsch | The University of Oxford, UK |
| Marzena Kryszkiewicz | Warsaw University of Technology, Poland |
| Yuri Kudryavcev | PMSquare, Australia |
| Lotfi Lakhal | LIF, Université Aix-Marseille, France |
| Wilfried Lex | TU Clausthal, Germany |
| Engelbert Mephu Nguifo | LIMOS, Université de Clermont-Ferrand 2, France |
| Amedeo Napoli | LORIA Nancy, France |
| Lhouari Nourine | LIMOS, France |
| Jan Outrata | Palacky University of Olomouc, Czech Republic |
| Jean-Marc Petit | LIRIS, INSA Lyon, France |

| | |
|---|---|
| Geert Poels | Ghent University, Belgium |
| Alex Pogel | New Mexico State University, USA |
| Sándor Radeleczki | University of Miskolc, Hungary |
| Olivier Raynaud | LIMOS, Université de Clermont-Ferrand 2, France |
| Camille Roth | CNRS/EHESS, France |
| Mohamed Rouane-Hacene | Université du Québec à Montréal, Canada |
| Dominik Ślęzak | University of Warsaw & Infobright, Poland |
| Laszlo Szathmary | University of Debrecen, Hungary |
| Andreja Tepavčević | University of Novi Sad, Serbia |
| Stijn Viaene | Katholieke Universiteit Leuven, Belgium |

## External Reviewers

Mikhail Babin, Russia
Philippe Fournier-Viger, Taiwan
Nathalie Girard, France
Tarek Hamrouni, France
Alice Hermann, France
Yury Katkov, Russia
Viet Phan Luong, France
Nikita Romashkin, Russia

## Sponsoring Institutions

Amsterdam-Amstelland Police, The Netherlands
IBM, Belgium
OpenConnect Systems, United States
Research Foundation Flanders, Belgium
Vlerick Management School, Belgium

## Table of Contents

# Composition of L-Fuzzy contexts

Cristina Alcalde[1], Ana Burusco[2], and Ramón Fuentes-González[2]

[1] Dpt. Matemática Aplicada. Escuela Universitaria Politécnica
UPV/EHU. Plaza de Europa, 1
20018 - San Sebastián (Spain)
c.alcalde@ehu.es
[2] Dpt. Automática y Computación. Universidad Pública de Navarra
Campus de Arrosadía
31006 - Pamplona (Spain)
{burusco,rfuentes}@unavarra.es

**Abstract.** In this work, we introduce and study the composition of two $L$-fuzzy contexts that share the same attribute set. Besides studying its properties, this composition allows to establish relations between the sets of objects associated to both $L$-fuzzy contexts.
We also define, as a particular case, the composition of an $L$-fuzzy context with itself.
In all the cases, we show some examples that illustrate the results.

**Key words:** Formal contexts theory, $L$-fuzzy contexts, Contexts associated with a fuzzy implication operator

## 1 Introduction

In some situations we have information that relates two sets $X$ and $Z$ to the same set $Y$ and we want to know if these relations allow us to establish connections between $X$ and $Z$. In the present work we will try to deal with the study of this problem using as tool the $L$-fuzzy Concepts Theory.

The Formal Concept Analysis developed by Wille ([13]) tries to extract some information from a binary table that represents a formal context $(X, Y, R)$ with $X$ and $Y$ being two finite sets (of objects and attributes, respectively) and $R \subseteq X \times Y$. This information is obtained by means of the formal concepts which are pairs $(A, B)$ with $A \subseteq X$, $B \subseteq Y$ fulfilling $A^* = B$ and $B^* = A$ (where $*$ is the derivation operator which associates to each object set $A$ the set $B$ of the attributes related to $A$, and vice versa). $A$ is the extension and $B$ the intension of the concept.

The set of the concepts derived from a context $(X, Y, R)$ is a complete lattice and it is usually represented by a line diagram.

In some previous works ([4],[5]) we defined the $L$-fuzzy context $(L, X, Y, R)$, where $L$ is a complete lattice, $X$ and $Y$ the sets of objects and attributes respectively and $R \in L^{X \times Y}$ an $L$-fuzzy relation between the objects and the attributes, as an extension to the fuzzy case of the Wille's formal contexts when

the relation between the objects and the attributes that we want to study takes values in a complete lattice $L$. When we work with these $L$-fuzzy contexts we use the derivation operators 1 and 2 defined by: For every $A \in L^X, B \in L^Y$

$$A_1(y) = \inf_{x \in X}\{I(A(x), R(x,y))\}, \quad B_2(x) = \inf_{y \in Y}\{I(B(y), R(x,y))\}$$

where $I$ is a fuzzy implication operator defined in $(L, \leq)$, $I : L \times L \longrightarrow L$, which is decreasing in its first argument, and, $A_1$ represents, as a fuzzy set, the attributes related to the objects of $A$ and $B_2$ the objects related to the attributes of $B$.

The information of the context is visualized by means of the $L$-fuzzy concepts which are pairs $(A, A_1) \in (L^X, L^Y)$ with $A \in \text{fix}(\varphi)$ the set of fixed points of the operator $\varphi$, being this one defined by the derivation operators 1 and 2 mentioned above as $\varphi(A) = (A_1)_2 = A_{12}$. These pairs, whose first and second components are the extension and the intension respectively, represent, as a fuzzy set, the set of objects that share some attributes.

The set $\mathcal{L} = \{(A, A_1) : A \in \text{fix}(\varphi)\}$ with the order relation $\leq$ defined as:

$$(A, A_1), (C, C_1) \in \mathcal{L}, \quad (A, A_1) \leq (C, C_1) \text{ if } A \leq C$$

(or equiv. $C_1 \leq A_1$) is a complete lattice that is said to be the $L$-fuzzy concept lattice ([4],[5]).

On the other hand, given $A \in L^X$, (or $B \in L^Y$) we can obtain the derived $L$-fuzzy concept applying the defined derivation operators. In the case of the use of a residuated implication operator (as it holds in this work), the associated $L$-fuzzy concept is $(A_{12}, A_1)$ (or $(B_2, B_{21})$).

Other extensions of the Formal Concept Analysis to the fuzzy area are in [14], [12], [3], [8], [10], [11] and [6].

## 2 Composed formal contexts

The composition of formal contexts allows to establish relations between the elements of two sets of objects that share the same attribute set.

**Definition 1.** *Let $(X, Y, R1)$ and $(Z, Y, R2)$ be two formal contexts, the composed formal context is defined as the context $(X, Z, R1 \star R2)$, where $\forall (x, z) \in X \times Z$:*

$$R1 \star R2(x,z) = \begin{cases} 1 & \text{if } R2(z,y) = 1,\ \forall y \text{ such that } R1(x,y) = 1 \\ 0 & \text{in other case} \end{cases}$$

*That is, the object $x$ is related to $z$ in the composed context if $z$ shares all the attributes of $x$ in the original contexts.*

**Proposition 1.** *The relation of the composed context, $R1 \star R2$, can also be defined as:*

$$R1 \star R2(x,z) = \min_{y \in Y}\{\max\{R1'(x,y), R2(z,y)\}\} \quad \forall (x,z) \in X \times Z$$

*where $R1'$ is the negation of the relation $R1$, that is, $R1'(x,y) = (R1(x,y))'$ $\forall (x,y) \in X \times Y$.*

This property will be helpful in the following sections.

*Remark 1.* Given the formal contexts $(X, Y, R1)$ and $(Z, Y, R2)$, the relation of the composed context $R1 \star R2$ is not necessarily the opposed of the relation $R2 \star R1$, that is, in general,

$$\text{There exists } (x,z) \in X \times Z \text{ such that } \; R1 \star R2(x,z) \neq R2 \star R1(z,x)$$

*Example 1.* Let us consider the formal contexts $(X, Y, R1)$ and $(Z, Y, R2)$, where $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3, y_4, y_5\}$, $Z = \{z_1, z_2, z_3, z_4\}$, and the respective relations are the following ones:

| $R1$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 0 | 1 | 1 | 0 | 1 |
| $x_2$ | 1 | 1 | 0 | 1 | 0 |
| $x_3$ | 0 | 0 | 1 | 0 | 1 |

| $R2$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $z_1$ | 1 | 1 | 0 | 1 | 0 |
| $z_2$ | 0 | 1 | 0 | 0 | 1 |
| $z_3$ | 1 | 1 | 0 | 1 | 1 |
| $z_4$ | 0 | 1 | 1 | 1 | 1 |

If we calculate the composition of the contexts defined above in the two possible orders, then the obtained relations are:

| $R1 \star R2$ | $z_1$ | $z_2$ | $z_3$ | $z_4$ |
|---|---|---|---|---|
| $x_1$ | 0 | 0 | 0 | 1 |
| $x_2$ | 1 | 0 | 1 | 0 |
| $x_3$ | 0 | 0 | 0 | 1 |

| $R2 \star R1$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $z_1$ | 0 | 1 | 0 |
| $z_2$ | 1 | 0 | 0 |
| $z_3$ | 0 | 0 | 0 |
| $z_4$ | 0 | 0 | 0 |

and, as can be seen, $(R1 \star R2)^{op} \neq R2 \star R1$.

This property will be helpful in the following sections.

### 2.1 Particular case: when a formal context is composed with itself

Let us analyze a particular case where some interesting results are obtained.

**Proposition 2.** *Let $(X, Y, R)$ be a formal context. If $(X, Y, R)$ is composed with itself, then the obtained context is $(X, X, R \star R)$ where the sets of objects and attributes are coincident and the relation $R \star R$ is a binary relation defined on $X$ as follows:*

$$R \star R(x_1, x_2) = \min_{y \in Y} \{\max\{R'(x_1, y), R(x_2, y)\}\} \; \forall (x_1, x_2) \in X \times X$$

*Remark 2.* The object $x_1$ is related to attribute $x_2$ in the composed context, if in the original context the object $x_2$ has at least the same attributes than the object $x_1$.

*Example 2.* Returning to the formal context $(X, Y, R)$ that we studied in the previous example, where the relation $R$ was:

| $R$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ |
|---|---|---|---|---|---|
| $x_1$ | 0 | 1 | 1 | 0 | 1 |
| $x_2$ | 1 | 1 | 0 | 1 | 0 |
| $x_3$ | 0 | 0 | 1 | 0 | 1 |

The composition of this context with itself is the context $(X, X, R \star R)$, and relation is given by the table:

| $R \star R$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $x_1$ | 1 | 0 | 0 |
| $x_2$ | 0 | 1 | 0 |
| $x_3$ | 1 | 0 | 1 |

**Proposition 3.** *The relation $R \star R$ obtained by the composition of the formal context $(X, Y, R)$ with itself is a preorder relation defined on the object set $X$.*

*Proof.* As a consequence of the definition, it is immediate to prove that:

1. The relation $R \star R$ is reflexive.
2. The relation $R \star R$ is transitive.

□

*Remark 3.* It is a simple verification to see that:

- The relation $R \star R$ is not, in general, a symmetric relation. To be symmetric it is necessary that whenever an object $x_2$ in the original context $(X, Y, R)$ has all the attributes of another object $x_1$, both objects have the same set of attributes.
- The relation $R \star R$ is not antisymmetric either. Therefore, $R \star R$ is not, in general, an order relation.

## 3 Extension to the *L*-fuzzy context case

The expression given in proposition 1 can be generalized to the fuzzy case substituting the maximum operator by a t-conorm $S$ and taking a strong negation $'$. In this way, we can define the compositions of two $L$-fuzzy contexts as follows:

**Definition 2.** *Let* $(L, X, Y, R1)$ *and* $(L, Z, Y, R2)$ *be two L-fuzzy contexts, we define the composed L-fuzzy context* $(L, X, Z, R1 \star R2)$*, where:*

$$R1 \star R2(x, z) = \inf_{y \in Y} \{S(R1'(x, y), R2(z, y))\} \quad \forall (x, z) \in X \times Z$$

*with* $S$ *being a t-conorm defined in the lattice* $L$*.*

If we remind the definition of a fuzzy S-implication, the previous one can be expressed in this way:

**Definition 3.** *Let* $(L, X, Y, R1)$ *and* $(L, Z, Y, R2)$ *be two L-fuzzy contexts, and* $I$ *an S-implication operator. We define the composed L-fuzzy context* $(L, X, Z,$ $R1 \star R2)$*, where:*

$$R1 \star R2(x, z) = \inf_{y \in Y} \{I(R1(x, y), R2(z, y))\} \quad \forall (x, z) \in X \times Z$$

We can generalize this definition to any fuzzy implication as we will see next.

### 3.1 Composition of *L*-fuzzy contexts associated with an implication operator

**Definition 4.** *Let* $(L, X, Y, R1)$ *and* $(L, Z, Y, R2)$ *be two L-fuzzy contexts, and let* $I$ *be a fuzzy implication operator, we define the* composed $L$-fuzzy context associated with the implication $I$ *as the L-fuzzy context* $(L, X, Z, R1 \star_I R2)$*, where:*

$$R1 \star_I R2(x, z) = \inf_{y \in Y} \{I(R1(x, y), R2(z, y))\} \quad \forall (x, z) \in X \times Z$$

*Remark 4.* If we remind the definition of the *triangle subproduct* operator $\triangleleft$ given by [9], one of the standard operators in the fuzzy relation theory which has been previously used in diverse works [1, 2], we can see that the composed relation defined here can be written as:

$$R1 \star_I R2 = R1 \triangleleft (R2)^{op}$$

As can be observed, also in this case a similar result to the crisp case is obtained.

**Proposition 4.** *Let* $(L, X, Y, R1)$ *and* $(L, Z, Y, R2)$ *be two L-fuzzy contexts. Then, the relation of the composed L-fuzzy context* $(L, X, Z, R1 \star_I R2)$ *is not, in general, the opposite of the relation of the composed L-fuzzy context* $(L, Z, X, R2 \star_I R1)$*.*

$$(R1 \star_I R2)^{op} \neq R2 \star_I R1$$

*That is, if we change the order of the composition, the obtained relation between the elements of* $X$ *and* $Z$ *is different.*

*Proof.* Given two $L$-fuzzy contexts $(L, X, Y, R1)$ and $(L, Z, Y, R2)$, and a fuzzy implication operator $I$, the relation of the composed $L$-fuzzy context $(L, X, Z, R1 \star_I R2)$ is:

$$R1 \star_I R2(x, z) = \inf_{y \in Y} \{I(R1(x, y), R2(z, y))\} \quad \forall (x, z) \in X \times Z$$

On the other hand, the relation of the composed $L$-fuzzy context $(L, Z, X, R2 \star_I R1)$ is defined as:

$$R2 \star_I R1(z, x) = \inf_{y \in Y} \{I(R2(z, y), R1(x, y))\} \quad \forall (z, x) \in Z \times X$$

As, in general, given a fuzzy implication $I(a, b) \neq I(b, a)$, then these relations are not opposed. □

*Example 3.* We have a company of temporary work in which we want to analyze the suitability of some candidates to obtain some offered employments. The company knows the requirements of knowledge to occupy each one of the positions, represented by means of the $L$-fuzzy context $(L, X, Y, R1)$, where the set of objects $X$ is the set of employments, the attributes $Y$ the necessary knowledge, and the relation among them appears in Table 1 with values in the chain $L$={0, 0.1, 0.2, . . . , 1}.

**Table 1.** The requirements of knowledge to obtain each one of the employments.

| $R1$ | computer science | accounting | mechanics | cooking |
|---|---|---|---|---|
| domestic helper | 0.1 | 0.3 | 0.1 | 1 |
| waiter | 0 | 0.4 | 0 | 0.7 |
| accountant | 0.9 | 1 | 0 | 0 |
| car salesman | 0.5 | 0.7 | 0.9 | 0 |

On the other hand, we have the knowledge of some candidates for these positions, represented by the $L$-fuzzy context $(L, Z, Y, R2)$ in which the objects are the different candidates to occupy the jobs, the attributes the necessary knowledge and the relation among them is given by Table 2.

A candidate will be suitable to obtain a job if he owns all the knowledge required in this position. Therefore, to analyze what candidate is adapted for each job, we would use the composed $L$-fuzzy context $(L, X, Z, R1 \star R2)$. The relation of this composed context, calculated using the Lukasiewicz implication operator, is the represented in Table 3.

To obtain the information of this $L$-fuzzy context we will use the ordinary tools of the $L$-fuzzy Concept Theory to analyze the associated $L$-fuzzy concepts. Thus, for example, if we want to find the best candidate to occupy the job of *waiter*, we take the set:

**Table 2.** Knowledge of the candidates.

| $R2$ | computer science | accounting | mechanics | cooking |
|---|---|---|---|---|
| C1 | 0.5 | 0.8 | 0.3 | 0.6 |
| C2 | 0.2 | 0.5 | 0.1 | 1 |
| C3 | 0 | 0.2 | 0 | 0.3 |
| C4 | 0.9 | 0.4 | 0.1 | 0.5 |
| C5 | 0.7 | 0.5 | 0.2 | 0.1 |

**Table 3.** Suitability of each candidate for each position.

| $R1 \star R2$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ |
|---|---|---|---|---|---|
| domestic helper | 0.6 | 1 | 0.3 | 0.5 | 0.1 |
| waiter | 0.9 | 1 | 0.6 | 0.8 | 0.4 |
| accountant | 0.6 | 0.3 | 0.1 | 0.4 | 0.5 |
| car salesman | 0.4 | 0.2 | 0.5 | 0.2 | 0.3 |

$$\{\text{domestic helper}/0, \text{waiter}/1, \text{accountant}/0, \text{car salesman}/0\}$$

and we obtain the derived $L$-fuzzy concept, whose intension is:

$$\{C_1/0.9, C_2/1, C_3/0.6, C_4/0.8, C_5/0.4\}$$

If we look at the attributes with the highest membership degree, we can deduce that the most suitable candidate for the job of *waiter* is $C_2$, followed by $C_1$ and $C_4$.

If, for instance, we want to find the best person to be *accountant* in a restaurant that also could work as a *waiter*, we take the set

$$\{\text{domestic helper}/0, \text{waiter}/1, \text{accountant}/1, \text{car salesman}/0\}$$

and the derived $L$-fuzzy concept is

$$\{C_1/0.6, C_2/0.3, C_3/0.1, C_4/0.4, C_5/0.4\}$$

where we can see that the most suitable candidate is $C_2$.

On the other hand, if our interest is to analyze which of the jobs is the most suitable for each candidate, we do the composition in the contrary order, obtaining the $L$-fuzzy context $(L, Z, X, R2 \star R1)$, where the composed relation is represented in Table 4.

We can see in this example that both compositions are different: A candidate can be the best to occupy a concrete job, but that job need not be the most appropriate for this candidate.

**Table 4.** Suitability of each employment for each candidate.

| $R2 \star R1$ | domestic helper | waiter | accountant | car salesman |
|---|---|---|---|---|
| $C_1$ | 0.5 | 0.5 | 0.4 | 0.4 |
| $C_2$ | 0.8 | 0.7 | 0 | 0 |
| $C_3$ | 0.3 | 0.2 | 0.2 | 0.7 |
| $C_4$ | 0.2 | 0.1 | 0.5 | 0.5 |
| $C_5$ | 0.4 | 0.3 | 0.8 | 0.8 |

The following result will be of interest to study the $L$-fuzzy concepts associated to the objects of the composed $L$-fuzzy context.

Before to proceed with the proposition, we are going to introduce a new notation: If the subscripts point out the derivation operators and the superscripts the $L$-fuzzy contexts where they are applied, then $A_1^{\circledast}$ is the derived set from $A$ obtained in the composed $L$-fuzzy context, $A_1^{①}$ is the derived set obtained in the $L$-fuzzy context $(L, X, Y, R1)$, and $(A_1^{①})_2^{②}$ the derived set of the last one in the $L$-fuzzy context $(L, Z, Y, R2)$

**Proposition 5.** *If the implication operator $I$ is residuated and we consider the set:*

$$A(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{in other case} \end{cases}$$

*then, the intension of the $L$-fuzzy concept obtained in the composed $L$-fuzzy context $(L, X, Z, R1 \star_I R2)$ from the set $A$, is equal to the extension of the $L$-fuzzy concept obtained in $(L, Z, Y, R2)$ from the intension of the $L$-fuzzy concept obtained in $(L, X, Y, R1)$ from $A$. That is, we obtain the same fuzzy set $Z$ applying the derivation operators twice (once in each one of the contexts that make up the composition), or once in the composed context.*

*Moreover, it is verify that:*

$$\forall z \in Z, \quad A_1^{\circledast}(z) = (A_1^{①})_2^{②}(z) = R1 \star_I R2(x_i, z)$$

*That is, the membership degrees obtained are the values of the row of $R1 \star_I R2$ that corresponds to the object $x_i$.*

*Proof.* Let be $A(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{in other case} \end{cases}$, the intension of the $L$-fuzzy concept obtained from $A$ in the context $(L, X, Y, R1)$ is the $L$-fuzzy subset of $Y$:

$$A_1^{①}(y) = \inf_{x \in X} \{I(A(x), R1(x, y))\}, \quad \forall y \in Y.$$

As the implication $I$ is residuated, $\forall a \in L$ it is verified that $I(0, a) = 1$ and $I(1, a) = a$, thus,

$$A_1^{①}(y) = R1(x_i, y), \quad \forall y \in Y.$$

Taking now the set $A_1^{①}$, we obtain the derived $L$-fuzzy concept in the $L$-fuzzy context $(L, Z, Y, R2)$, the extension of which is:

$$(A_1^{①})_2^{②}(z) = \inf_{y \in Y}\{I(A_1^{①}(y), R2(z,y))\} =$$
$$= \inf_{y \in Y}\{I(R1(x_i, y), R2(z,y))\} = R1 \star_I R2(x_i, z), \quad \forall z \in Z.$$

On the other hand, the intension of the obtained $L$-fuzzy concept in the composed $L$-fuzzy context from $A$ is:

$$A_1^{\circledast}(z) = \inf_{x \in X}\{I(A(x), R1 \star_I R2(x,y))\} = R1 \star_I R2(x_i, z), \quad \forall z \in Z.$$

□

*Example 4.* If we come back to example 3, we have analyzed which candidate is the most suitable for the job of *waiter*.

To do this, in the $L$-fuzzy context $(L, X, Z, R1 \star R2)$ (see Table 3) we have taken the set

$$A = \{\text{domestic helper}/0, \text{waiter}/1, \text{accountant}/0, \text{car salesman}/0\}$$

and we have calculated the closed $L$-fuzzy concept, where the fuzzy intension is:

$$A_1^{\circledast} = \{C_1/0.9, C_2/1, C_3/0.6, C_4/0.8, C_5/0.4\}$$

And here, if we look at those attributes whose membership degrees stand out from the others, we deduce that the most suitable candidates to be good *waiters* were, $C_2$, $C_1$ and $C_4$, in this order.

The same result is obtained if we take the $L$-fuzzy context $(L, X, Y, R1)$ (see Table1) and we calculate the $L$-fuzzy concept from $A$, which intension is:

$$A_1^{①} = \{\text{computer science}/0, \text{accounting}/0.4, \text{mechanics}/0, \text{cooking}/0.7\}$$

And, from this fuzzy set we obtain in the $L$-fuzzy context $(L, Z, Y, R2)$ (see Table2) the derived $L$-fuzzy concept the extension of which is:

$$(A_1^{①})_2^{②} = \{C_1/0.9, C_2/1, C_3/0.6, C_4/0.8, C_5/0.4\}$$

As can be seen, the result is the same that the obtained in the composed $L$-fuzzy context.

### 3.2 Composition of an *L*-fuzzy context with itself

The composition of an $L$-fuzzy context $(L, X, Y, R)$ with itself will allow us to set up some relationships between the elements of the object set $X$.

**Proposition 6.** *If $I$ is a residuated implication associated with a left continuous t-conorm $T$, then the relation $R \star_I R$ that results of the composition of $(L, X, Y, R)$ with itself, associated with the implication $I$, constitutes a fuzzy preorder relation defined in the object set $X$.*

*Proof.* 1. First, we prove that it is a reflexive relation, that is, the relation verifies:

$$\forall x \in X,\ R \star_I R(x,x) = 1.$$

By the definition of the composition associated with an implication operator, we have

$$\forall x \in X, \quad R \star_I R(x,x) = \inf_{y \in Y} \{I(R(x,y), R(x,y))\},$$

and, as any residuated implication verifies that $I(a,a) = 1,\ \forall a \in L$, then

$$\forall x \in X, \quad R \star_I R(x,x) = 1.$$

2. To see that $R \star_I R$ is a $T$-transitive relation, we have to prove that

$$\forall x,t,z \in X, \quad T(R \star_I R(x,t), R \star_I R(t,z)) \leq R \star_I R(x,z),$$

that is, the following inequality must be verified:

$$T\left(\inf_{\alpha \in Y} \{I(R(x,\alpha), R(t,\alpha))\}, \inf_{\beta \in Y} \{I(R(t,\beta), R(z,\beta))\}\right) \leq \inf_{\alpha \in Y} \{I(R(x,\alpha), R(z,\alpha))\}.$$

By the monotony of the t-norm, we have:

$$T\left(\inf_{\alpha \in Y} \{I(R(x,\alpha), R(t,\alpha))\}, \inf_{\beta \in Y} \{I(R(t,\beta), R(z,\beta))\}\right) \leq \inf_{\alpha \in Y} \left\{T\left(I(R(x,\alpha), R(t,\alpha)), \inf_{\beta \in Y} \{I(R(t,\beta), R(z,\beta))\}\right)\right\} \leq \inf_{\alpha \in Y} \{T\left(I(R(x,\alpha), R(t,\alpha)), I(R(t,\alpha), R(z,\alpha))\right)\}.$$

As the used t-norm $T$ is left-continuous, we know that [7]

$$\forall a,b,c \in [0,1],\ T(I(a,b), I(b,c)) \leq I(a,c),$$

and it is verified that:

$$T\left(\inf_{\alpha \in Y} \{I(R(x,\alpha), R(t,\alpha))\}, \inf_{\beta \in Y} \{I(R(t,\beta), R(z,\beta))\}\right) \leq \inf_{\alpha \in Y} \{I(R(x,\alpha), R(z,\alpha))\}.$$

□

*Remark 5.* The relation $R \star_I R$ is neither symmetric nor antisymmetric and then, is neither an equivalence nor an order relation. For instance, if we take the relation $R$ given by the table:

| $R$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ |
|---|---|---|---|---|
| $x_1$ | 0.1 | 0.3 | 0.5 | 0.1 |
| $x_2$ | 0.8 | 0.2 | 0.8 | 0.2 |
| $x_3$ | 0.4 | 0.7 | 0 | 0.1 |

then the relation $R \star_I R$ associated with the Lukasiewicz implication operator is:

| $R \star_I R$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $x_1$ | 1 | 0.9 | 0.5 |
| $x_2$ | 0.3 | 1 | 0.2 |
| $x_3$ | 0.5 | 0.5 | 1 |

and, as can be seen, is neither a symmetric nor an antisymmetric relation.

*Remark 6.* If we are using a non residuated implication operator, not always a fuzzy preorder relation is obtained. For instance, if we take the previous relation $R$ and we do the composition $R \star_I R$ associated with the Kleene-Dienes implication (that does not verify $I(x, x) = 1$), then we obtain the following relation:

| $R \star_I R$ | $x_1$ | $x_2$ | $x_3$ |
|---|---|---|---|
| $x_1$ | 0.5 | 0.7 | 0.5 |
| $x_2$ | 0.2 | 0.2 | 0.2 |
| $x_3$ | 0.3 | 0.3 | 0.3 |

that is neither a reflexive nor a fuzzy preorder relation.

The application of this composition can be very interesting in social or work relations as we can see next:

*Example 5.* There are four different manufacture processes in a factory and we want to organize the workers so that each of them is subordinate of another one if its capacity to carry out each one of the processes of manufacture is smaller.

To model this problem, we are going to take the $L$-fuzzy context $(L, X, Y, R)$, where the set of objects $X$ is formed by the workers $\{O_1, O_2, O_3, O_4, O_5\}$, the attributes are the different manufacture processes $\{P_1, P_2, P_3, P_4\}$, and the relation $R$ represents the capacity of each one of the workers to carry out each one of the processes, in a scale of 0 to 1 (See Table 5).

The $L$-fuzzy context that results of the composition of this context with itself allow us to define relations boss-subordinate between the workers so that the relation $R \star R(x, y)$ of the compound context (associated with the Lukasiewicz implication) gives the degree in which the worker $x$ is subordinate of the worker $y$. (See Table 6).

**Table 5.** Capacity of the workers to carry out each one of the manufacture processes

| $R$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|---|---|---|---|---|
| $O_1$ | 0.7 | 1 | 0.3 | 0 |
| $O_2$ | 0.3 | 0.8 | 0.9 | 0.4 |
| $O_3$ | 0.1 | 0.2 | 1 | 0.5 |
| $O_4$ | 0.5 | 0.3 | 0.2 | 0.4 |
| $O_5$ | 1 | 0.5 | 0.8 | 1 |

**Table 6.** Relation "be subordinate of".

| $R \star R$ | $O_1$ | $O_2$ | $O_3$ | $O_4$ | $O_5$ |
|---|---|---|---|---|---|
| $O_1$ | 1 | 0.6 | 0.2 | 0.3 | 0.5 |
| $O_2$ | 0.4 | 1 | 0.4 | 0.3 | 0.7 |
| $O_3$ | 0.3 | 0.9 | 1 | 0.2 | 0.8 |
| $O_4$ | 0.6 | 0.8 | 0.6 | 1 | 1 |
| $O_5$ | 0 | 0.3 | 0.1 | 0.4 | 1 |

This will allow us, for example, to choose bosses in the group watching the columns of the obtained relation: In this case, we could choose as bosses of the workers to $O_2$ and $O_5$ because both have as subordinate $O_3$ and $O_4$ and the subordination degrees are the biggest values of the columns.

## 4 Conclusions and future work

This work constitutes the first approach to the problem of composition of $L$-fuzzy contexts. In future works we will use these results in the resolution of other problems that seem interesting to us:

- First, this composition will be useful to study the chained $L$-fuzzy contexts, that is, to find relations between two defined contexts where the set of attributes of the first context is the same that the set of objects of the second one.

- On the other hand, we think that it will be useful to define the composition of $L$-fuzzy contexts in the interval-valued case in order to study certain situations.

## Acknowledgements

This work has been partially supported by the Research Group "Intelligent Systems and Energy (SI+E)" of the Basque Government, under Grant IT519-10.

## References

1. C. Alcalde, A. Burusco and R. Fuentes-González, "Analysis of certain L-Fuzzy relational equations and the study of its solutions by means of the L-Fuzzy Concept

Theory." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.* 20 No.1 (2012), pp. 21–40.
2. E. Bartl and R. Bělohlávek, "Sup-t-norm and inf-residuum are a single type of relational equations." *International Journal of General Systems.* 40 No.6 (2011), pp. 599–609.
3. R. Bělohlávek, "Fuzzy Galois connections and fuzzy concept lattices: from binary relations to conceptual structures", in: Novak V., Perfileva I. (eds.): *Discovering the World with Fuzzy Logic*, Physica-Verlag (2000), pp. 462–494.
4. A. Burusco and R. Fuentes-González, "The Study of the *L*-Fuzzy Concept Lattice." *Mathware and Soft Computing.* 1 No.3 (1994), pp. 209–218.
5. A. Burusco and R. Fuentes-González, "Construction of the *L*-Fuzzy Concept Lattice." *Fuzzy Sets and Systems.* 97 No.1 (1998), pp. 109–114.
6. Y. Djouadi, D. Dubois and H. Prade, "On the possible meanings of degrees when making formal concept analysis fuzzy." *EUROFUSE workshop. Preference Modelling and Decision Analysis.* Pamplona, Sep 2009, pp. 253–258.
7. J. Fodor, M. Roubens. *Fuzzy Preference Modelling and Multicriteria Decision Support.* Theory and Decision Library (Kluwer Academic Publishers), Dordrecht/Boston/London (1994).
8. A. Jaoua, F. Alvi, S. Elloumi, S. B. Yahia. "Galois Connection in Fuzzy Binary Relations." *Applications for Discovering Association Rules and Decision Making.* RelMiCS (2000), pp. 141–149.
9. L. J. Kohout, W. Bandler, Use of fuzzy relations in Knowledge representation, acquisition, and processing, in: *L. Zadeh, J. Kacprzyk (Eds.), Fuzzy Logic for Management of Uncertainty*, 1992, pp. 415-435.
10. S. Krajči. "A generalized concept lattice." *Logic J. IGPL* 13 (5) (2005) pp. 543–550.
11. J. Medina, M. Ojeda-Aciego, and J. Ruiz-Calviño. "On multi-adjoint concept lattices: definition and representation theorem." *Lect. Notes in Artificial Intelligence*, 4390,(2007), pp 197–209.
12. S. Pollandt, *Fuzzy Begriffe: Formale Begriffsanalyse unscharfer Daten*, Springer (1997).
13. R. Wille. "Restructuring lattice theory: an approach based on hierarchies of concepts",in: Rival I.(ed.),*Ordered Sets*, Reidel, Dordrecht-Boston (1982), pp. 445–470.
14. K.E. Wolff. "Conceptual interpretation of fuzzy theory", in: Proc. *6th European Congress on Intelligent techniques and Soft computing*, 1, (1998), pp. 555–562.

# Iterator-based Algorithms in Self-Tuning Discovery of Partial Implications

José L. Balcázar[1], Diego García-Sáiz[2], and Javier de la Dehesa[2]

[1] LSI Department, UPC, Campus Nord, Barcelona
jose.luis.balcazar@upc.edu
[2] Mathematics, Statistics and Computation Department, University of Cantabria
Avda. de los Castros s/n, Santander, Spain
garciasad@unican.es

**Abstract.** We describe the internal algorithmics of our recent implementation of a closure-based self-tuning associator: *yacaree*. This system is designed so as not to request the user to specify any threshold. In order to avoid the need of a support threshold, we introduce an algorithm that constructs closed sets in order of decreasing support; we are not aware of any similar previous algorithm. In order not to overwhelm the user with large quantities of partial implications, our system filters the output according to a recently studied lattice-closure-based notion of confidence boost, and self-adjusts the threshold for that rule quality measure as well. As a consequence, the necessary algorithmics interact in complicated ways. In order to control this interaction, we have resorted to a well-known, powerful conceptual tool, called Iterators: this notion allows us to distribute control among the various algorithms at play in a relatively simple manner, leading to a fully operative, open-source, efficient system for discovery of partial implications in relational data.

**Keywords:** Association mining, parameter-free mining, iterators, Python

.

## 1 Introduction

The task of identifying which implications hold in a given dataset has received already a great deal of attention [1]. Since [2], also the problem of identifying partial implications has been considered. Major impulse was received with the proposal of "mining association rules", a very closely related concept. A majority of existing association mining programs follow a well-established scheme [3], according to which the user provides a dataset, a support constraint, a confidence constraint, and, optionally, in most modern implementations, further constraints on other rule quality evaluation measures such as lift or leverage (a survey of quality evaluation measures for partial implications is [4]). A wealth of algorithms, of which the most famous is *apriori*, have been proposed to perform association mining.

Besides helping the algorithm to focus on hopefully useful partial implications, the support constraint has an additional role: by restricting the process to frequent (or frequent closed) itemsets, the antimonotonicity property of the support threshold defines a limited search space for exploration and avoids the often too wide space of the whole powerset of items.

Instead, however, the price becomes a burden on the user, who must supply thresholds on rule evaluation measures and on support. Rule measure thresholds may be difficult to set correctly, but at least they offer often a "semantic" interpretation that guides the choice; for instance, confidence is (the frequentist approximation to) the conditional probability of the consequent of the rule, given the antecedent, whereas lift and leverage refer to the (multiplicative or additive, respectively) deviation from independence of antecedent and consequent. But support thresholds are known to be very difficult to set right. Some smallish datasets are so dense that any exploration below 95% support, on our current technology, leads to a not always graceful breakdown of the associator program, whereas other, large but sparse datasets hardly yield any association rule unless the support is set at quantities as low as 0.1%, spanning a factor of almost one thousand; and, in order to set the "right" support threshold (whatever that means), no intuitive guidance is currently known, except for the rather trivial one of trying various supports and monitoring the number of resulting rules and the running time and memory needed.

The Weka *apriori* implementation automates partially the process, as follows: it explores repeatedly at several support levels, reducing the threshold from one run to the next by a "delta" parameter (to be set as well by the user), until a given number of rules has been gathered. Inspired by this idea, but keeping our focus in avoiding user-set parameters, we are developing an alternative association miner. It includes an algorithm that explores closed itemsets in order of decreasing support. This algorithm is similar in spirit to ChARM [5], except that some of the accelerations of that algorithm require ordering some itemsets by increasing support, which becomes inapplicable in our case. Additionally, our algorithm keeps adjusting automatically the support bound as necessary so as to be able to proceed with the exploration within the available memory resources. This is, of course, more expensive in computation time, compared to a traditional exploration with the "right" support threshold, as the number of closed frequent sets that can be filtered out right away is much smaller; on the other hand, no one can tell ahead of time which is the "right" support threshold, and our alternative spares the user the need of guessing it. To our knowledge, this is the first algorithm available for mining closed sets in order of descending support and without employing a user-fixed support threshold.

Similarly, in order to spare the user the choice of rule measure thresholds, we employ a somewhat complex (and slightly slower to evaluate) measure, the closure-based confidence boost, for which our previous work has led to useful, implementable bounds as well as to a specific form of self-tuning [6]. It can be proved that this quantity is bounded by a related, easy to compute quantity: namely, the closure-based confidence boost is always less than or equal to the

support ratio, introduced (with a less prononceable name) in [7], and defined below; this bound allows us to "push" into the closure mining process a constraint on the support ratio that spares computation of rules that will fail the rule measure threshold. We do this by postponing the consideration of the closed sets that, upon processing, would give rise only to partial implications below the confidence boost threshold.

As indicated, our algorithm self-tunes this threshold, which starts at a somewhat selective level, by lowering it in case the output rules show it appropriate. Then, the support ratio in the closure miner is to follow suit: the constraint is to be pushed into the closure mining process with the new value. This may mean that previously discarded closures are to be now considered. Therefore, we must reconcile four processes: one of mining closed frequent sets in order of decreasing support, filtering them according to their support ratio; two further ones that change, along the way, respectively, the support threshold and the support ratio threshold; and the one of obtaining the rules themselves from the closed itemsets. Unfortunately, these processes interfere very heavily with each other. Closed sets are the first objects to be mined from the dataset, and are to be processed in order of decreasing support to obtain rules from them, but they are to be processed only if they have both high enough support, and high enough support ratio. Closed sets of high support and low support ratio, however, cannot be simply discarded: a future decrease of the self-adjusting rule measure bound may require us to "fish" them back in, as a consequence of evaluations made "at the end" of the process upon evaluating rules; likewise, rules of low closure-based confidence boost need to be kept on hold instead of discarded, so as to be produced if, later, they turn out to clear the threshold after adjusting it to a lower level. The picture gains an additional complication from the fact that constructing partial implications requires not only the list of frequent closures, but also the Hasse edges that constitute the corresponding Formal Concept Lattice.

As a consequence, the varying thresholds make it difficult to organize the architecture of the software system in the traditional form of, first, mining the lattice of frequent closures and, then, extracting rules from them. We describe here how iterators offer a simple and efficient solution for the organization of our partial implication miner *yacaree*, available at SourceForge and shown at the demo track of a recent conference [8]. The details of the implementation are described here for the first time.

## 2 Concepts, Notation, and Overview

In our technological context (pure Python), "generators" constitute one of the ways of obtaining iterators. An iterator constructed in this way is a method (in the object-oriented sense) containing, anywere inside, the "yield" instruction; most often, this instruction is inside some loop. This instruction acts as a "return" instruction for the iterator, except that its whole status, including values of local variables and program counter, is stored, and put back into place at the next call to the method. Thus, we obtain a "lazy" method that gives us, one

by one, a sequence of values, but only computes one more value whenever it is called from the "consumer" that needs these values.

Generators as a comfortable way of constructing iterators are available only in a handful of platforms: several quite specialized lazy functional programming languages offer them, but, among the most common programming languages, only Python and C# include generators. Java or C++ offer a mere "iterator" interface that simply states that classes implementing iterators must offer, with specific names, the natural operations to iterate over them, but the notion of generators to program them easily is not available.

We move on to describe the essentials of our system, and the way iterators defined by means of generators allow us to organize, in a clear and simple way, the various processes involved.

A given set of available items $U$ is assumed; its subsets are called itemsets. We will denote itemsets by capital letters from the end of the alphabet, and use juxtaposition to denote union of itemsets, as in $XY$. The inclusion sign as in $X \subset Y$ denotes proper subset, whereas improper inclusion is denoted $X \subseteq Y$. For a given dataset $D$, consisting of $n$ transactions, each of which is an itemset labeled with a unique transaction identifier, we define the support $sup(X)$ of an itemset $X$ as the cardinality of the set of transactions that contain $X$. Sometimes, the support is measured "normalized" by dividing by the dataset size; then, it is an empirical approximation to the probability of the event that the itemset appears in a "random" transaction. Except where explicitly indicated, all our uses of support will take the form of ratios, and, therefore, it does not matter at all whether they come absolute or normalized.

An association rule is an ordered pair of itemsets, often written $X \to Y$. The confidence $c(X \to Y)$ of rule $X \to Y$ is $sup(XY)/sup(X)$. We will refer occasionally below to a popular measure of deviation from independence, often named *lift*: assuming $X \cap Y = \emptyset$, the lift of $X \to Y$ is

$$\frac{sup(XY)}{sup(X)\, sup(Y)}$$

where all three supports are assumed normalized (if they are not, then the dataset size must of course appear as an extra factor in the numerator).

An itemset X is called frequent if its support is greater than or equal to some user-defined threshold: $sup(X) > \tau$. We often assume that $\tau$ is known; no support bound is implemented by setting $\tau = 0$. Our algorithms will attempt at self-tuning $\tau$ to an appropriate value without concourse of the user. Given an itemset $X \subseteq U$, its closure $\overline{X}$ of $X$ is the maximal set (with respect to set inclusion) $Y \subseteq U$ such that $X \subseteq Y$ and $sup(X) = sup(Y)$. It is easy to see that $\overline{X}$ is unique. An itemset $X$ is closed if $\overline{X} = X$. Closure operators are characterized by the three properties of monotonicity, idempotency, and extensivity.

The support ratio was essentially employed first, to our knowledge, in [7], where, together with other similar quotients, it was introduced with the aim of providing a faster algorithm for computing representative rules. The support

ratio of an association rule $X \to Y$ is that of the itemset $XY$, defined as follows:

$$\sigma(X \to Y) = \sigma(XY) = \frac{sup(XY)}{\max\{sup(Z) \mid sup(Z) > \tau,\, XY \subset Z\}}.$$

For many quality measures for partial implications, including support, confidence, and closure-based confidence boost (to be defined momentarily), the relevant supports turn out to be the support of the antecedent and the support of the union of antecedent and consequent. As these are captured by the corresponding closures, we deem inequivalent two rules $X \to Y$ and $X' \to Y'$ exactly when they are not "mutually redundant" with respect to the closure space defined by the dataset: either $\overline{X} \neq \overline{X'}$, or $\overline{XY} \neq \overline{X'Y'}$. We denote that fact as $(X \to Y) \not\equiv (X' \to Y')$.

We now assume $sup(XY) > \tau$. As indicated, our system keeps a varying threshold on the following rule evaluation measure: $\beta(X \to Y) =$

$$\frac{c(X \to Y)}{\max\{c(X' \to Y') \mid (X \to Y) \not\equiv (X' \to Y'),\, sup(X'Y') > \tau,\, X' \subseteq \overline{X},\, Y \subseteq \overline{X'Y'}\}}.$$

This notion, known as "closure-based confidence boost", as well as the "plain confidence boost", which is a simpler variant where the closure operator reduces to the identity, are studied in depth in [6]. Intuitively, this is a relative, instead of absolute, form of confidence: we are less interested in a partial implication having very similar confidence to that of a simpler one. A related formula measures relative confidence with respect to logically stronger partial implications (confidence width, see [6]); the formula just given seems to work better in practice. For the value of this measure to be nontrivial, $XY$ must be a closed set; the following inequality holds:

**Proposition 1.** $\beta(X \to Y) \leq \sigma(X \to Y)$.

The threshold on $\beta(X \to Y)$ is self-adjusted along the mining process, on the basis of several properties such as coincidence with lift under certain conditions; all these details and properties are described in [6].

### 2.1 Architecture of *yacaree*

The diagram in Figure 1 shows the essentials of the class structure, for easier reference along the description of the iterators. For simplicity, a number of additional classes existing in the system are not shown. A couple of them, added recently, find minimal generators via standard means and implement a plain confidence boost version appropriate for full-confidence implications; their algorithmics are not novel, pose no challenge, and are omitted here. We are also omitting discussion of classes like the Dataset class, some heap-like auxiliary data structures, user interfaces, and a class capturing a few static values, as their role in our description is minor or easy to understand (or both).

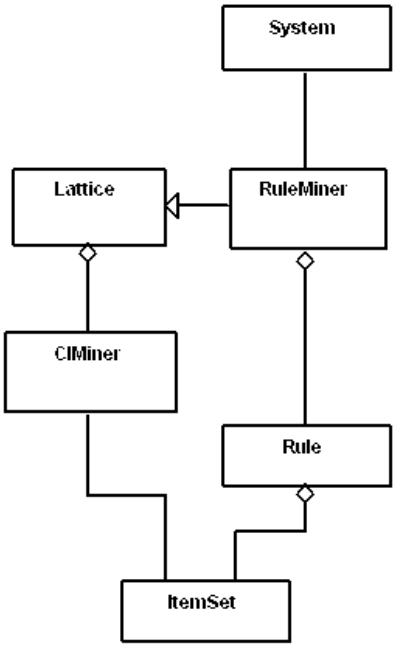



**Fig. 1.** Partial class diagram of the associator

### 2.2 Class Overview

We give a brief explanation of the roles of the classes given in the diagram. Details about their main methods (the corresponding iterators) come below.

Class ItemSet keeps the information and methods to prettyprint itemsets, including information such as support; it inherits from sets all set-theoretic operations. Class Rule keeps both antecedent and consequent (technically, it keeps the antecedent and the union of antecedent and consequent, as in this case the latter is always closed, which allows for more efficient processing), and is able to provide rule evaluation measures such as confidence or lift.

Class ClMiner runs the actual closure mining, with some auxiliary methods to handle all details. Its main method is the iterator mine_closures() (described below) which yields, one by one and upon being called, all closed sets having support above the threshold, in order of decreasing support. This "decreasing support" condition allows us to increase the support threshold, if necessary, to continue the exploration. As explained below, when the internal data structures of the closure miner are about to overflow, the support threshold is increased in such a way that half the closures found so far and still pending consideration are discarded.

Class Lattice runs its own iterator, candidate_closures(), which, in turn, calls mine_closures() as new closed sets become needed. Its main task is to call methods that implement the algorithms from [9] and actually build the lattice of closed sets, so that further iterators can expect to receive closures for which the immediate predecessors have been identified. Version 1.0 of *yacaree* employed the Border algorithm but in version 1.1 we have implemented the faster algorithm iPred and indeed obtained around a 10% acceleration. The fact that iPred could be employed in enumerations of closures by decreasing support was proved in [10]. See [11] for futher discussions.

Additionally, the support ratio of each closed set is also computed here, and the class offers yet another iterator that provides, for each closure, all the predecessor closures having support above a local, additional support threshold that can be specified at call time. In this way, we obtain all the candidate antecedents for a given closed set as potential consequent. This internal iterator amounts to a plain depth-first search, so that we do not discuss it further here.

Within class Lattice, two heap-structured lists keep, respectively, the closures that are ready to be passed on as they clear both the support and the support ratio thresholds (Lattice.ready) and the closures that clear the support threshold but fail the support ratio threshold (Lattice.freezer); these will be recovered in case a decrease of the confidence boost bound is to affect the support ratio pruning.

Finally, class RuleMiner is in charge of offering the system an iterator over all the association rules passing the current thresholds of support and closure-based confidence boost: mine_rules(). Its usage allows one to include easily further checks of confidence, lift, or any other such quantity.

## 3 Details

This section provides details of the main iterators and their combined use to attain our goals.

### 3.1 ClMiner.mine_closures()

The closure miner is the iterator that supports the whole scheme; it follows a "best-first" strategy, where here "best" means "highest support". We maintain a heap containing closures already generated, but which have not been expanded yet to generate further closures after them. The heap can provide the one of highest support in logarithmic time, as this is the closure that comes next. Then, as this closure is passed on to the lattice constructor, items (rather, closures of singletons) are added to it in all possible ways, and closure operations are applied in order to generate its closed successors, which are added to the heap unless they were already in it. The decreasing support condition ensures that they were never visited before. For simplicity, we omit discussing the particular case of the empty set, which, if closed, is to be traversed first, separately.

1: identify closures of singletons

```
2: organize them into a maxheap according to support
3: while heap nonempty do
4:   consider increasing the support threshold by monitoring the available
     memory
5:   if the support threshold must be raised then
6:     kill from the heap pending closures of support below new threshold,
       which is chosen so that the size of the heap halves
7:   end if
8:   pop from heap the max-support itemset
9:   yield it
10:  try to extend it with all singleton closures
11:  for such extensions with sufficient support do
12:    if their closure is new then
13:      add it to the heap
14:    end if
15:  end for
16: end while
```

In order to clarify how this algorithm works, we develop the following example. Consider a dataset with 24 transactions over universe $U = \{a, b, c, d, e\}$ of 5 items: $\{abcde, bcde \times 2, abe, cde, be, ae \times 3, ab \times 4, cd \times 6, b \times 2, a \times 3\}$. For this dataset, there are 12 closed sets, which we enumerate here with their corresponding supports: $\emptyset_{/24}$, $a_{/12}$, $b_{/11}$, $cd_{/10}$, $e_{/9}$, $ab_{/6}$, $ae_{/5}$, $be_{/5}$, $cde_{/4}$, $bcde_{/3}$, $abe_{/2}$, $abcde_{/1}$. The empty set is treated first, separately, as indicated. Then, the four next closures correspond to closures of singletons (the closures of $c$ and $d$ coincide) and form an initial heap, containing: $[a_{/12}, b_{/11}, cd_{/10}, e_{/9}]$.

The heap provides $a$ as next closure in descending support; it is passed on to further processing at the "yield" instruction, and it is expanded with singleton closures in all possible ways, enlarging the heap into containing six pending closures: $[b_{/11}, cd_{/10}, e_{/9}, ab_{/6}, ae_{/5}, abcde_{/1}]$: each of the new sets in the heap is obtained by adding to $a$ the closure of a singleton, and closing the result. The next closure is $b$, which goes into the "yield" instruction and, subsequently, generates two further closures to be added to the heap, which becomes: $[cd_{/10}, e_{/9}, ab_{/6}, ae_{/5}, be_{/5}, bcde_{/3}, abcde_{/1}]$. The closure $ab$ generated from $b$ is omitted, as it is repeated since it was already generated from $a$.

For illustration purposes, we assume now that the length of the heap, currently 7, is deemed too large. Of course, in a toy example like this one there is no need of moving up the support threshold, but let's do it anyway: assume that the test indicates that the heap is occupying too much memory, incurring in a risk of soon-coming overflow. Then, the support is raised as much as necessary so as to halve the length of the heap. Pending closures of support 5 or less would be discarded from the heap, the support threshold would be set at 6, and only three closures would remain in the heap: $[cd_{/10}, e_{/9}, ab_{/6}]$. Each of them would be processed in turn, given onwards by the "yield" instruction, and expanded with all closures of singletons; in all cases, we will find that expanding any of them with a singleton closure leads to a closure of support below 6, which is therefore

omitted as it does not clear the threshold. Eventually, these three closures in the heap are passed on, and the iterator will have traversed all closures of support 6 or higher.

As a different run, assume now that we accept the growth of the heap, so that it is not reduced. The traversal of closures would go on yielding $cd$, which would add $cde$ to the heap; adding either $a$ or $b$ to $cd$ leads to $abcde$ which is already in the heap. The next closure $e$ adds nothing new to the heap, and the next is $ab$ which adds $abe$; at this point the heap is $[ae_{/5}, be_{/5}, cde_{/4}, bcde_{/3}, abe_{/2}, abcde_{/1}]$. All further extensions only lead to repeated closures, hence nothing is further added to the heap, and, as it is emptied, the traversal of all the closures is completed.

The main property that has to be argued here is the following:

**Proposition 2.** *As the support threshold rises, all the closed sets delivered so far by the iterator to the next phase are still correct, that is, have support at least the new value of the threshold.*

*Proof.* We prove this statement by arguing the following invariants: first, the new threshold is bounded above by the highest support of a closure in the heap; second, all the closed sets provided so far up to any "yield" statement have support bounded below by all the supports currently in the heap. These invariants are maintained as we extract the closure $C$ of maximum support in the heap, and also when we add to it extensions of $C$: indeed, $C$ having been freshly taken off the heap, all previous deliveries have at least the same support, whereas all extensions that are to enter the heap are closed supersets of $C$ and must have lower support, because $C$ is closed.

Hence, all previous deliveries have support higher than the maximum support in the heap, which, in turn, is also higher than the new threshold; transitivity now proves the statement.

### 3.2 Lattice.candidate_closures()

In order to actually mine rules from the closures traversed by the loop described in the previous section, further information is necessary: data structures to allow for traversing predecessors, namely, the Hasse edges, that is, the immediate, nontransitive neighbors of each closure. These come from a second iterator that implements the iPred algorithm [9].

Additionally, we wish to push into the closure mining the confidence boost constraint. The way to do it is to compute the support ratio of each closure, and only pass it on to mine rules from it if this support ratio is above the confidence boost threshold; indeed, Proposition 1 tells us that, if the support ratio is below the threshold, the confidence boost will be too.

Due to the condition of decreasing support, we know that the closed superset that defines the support ratio is exactly the first successor to appear from the closure mining iterator. As soon as one successor of $C$ appears, if the support ratio is high enough, we can yield $C$, as the Hasse edges to its own predecessors

are guaranteed to have been set before. If the support ratio is not enough, it is kept on a "freezer" (again a heap-like structure) from where it might be "fished back in" if the confidence boost threshold decreases later on.

One has to be careful that the same closed set, say $C$, may be the first successor of more than one predecessor. As we set the predecessors $C'$ of $C$, we move to the "ready" heap those that have $C$ as first successor, if their support ratio is high enough; then we yield them all. Additionally, as we shall see, it may happen that RuleMiner.mine_rules() moves closures from "freezer" to "ready". We explain this below.

```
1: for each closed set C yielded by ClMiner.mine_closures() do
2:    apply a Hasse edges algorithm (namely iPred) to set up the lattice edges
      connecting C to its predecessors
3:    for each unprocessed predecessor C' do
4:       compute the support ratio of C'
5:       if support ratio is over the rule evaluation threshold then
6:          add C' to the "ready" heap
7:       else
8:          add C' to the "freezer" heap
9:       end if
10:   end for
11:   for each closure in the "ready" heap do
12:      yield it
13:   end for
14: end for
```

We observe here that we are not guaranteeing decreasing support order in this iterator, as the changes to the support ratio threshold may swap closures with respect to the order in which they were obtained. What we do need is that the most basic iterator, ClMiner.mine_closures(), does provide them in decreasing support order, first, to ensure that the support threshold can be raised if necessary, and, second, to make sure that the support ratio is correctly computed from the first successor found for each closure.

Along the same example as before, consider, for instance, what happens when ClMiner.mine_closures() yields the closure $ab$ to Lattice.candidate_closures(). The iPred algorithm identifies $a$ and $b$ as immediate predecessors of $ab$, and the corresponding Hasse edges are stored. Then, both $a$ and $b$ are identified as closures whose first successor (in decreasing support) has just appeared; indeed, other successors have less support than $ab$. The support ratios of $a$ and $b$, namely, $12/6 = 2$ and $11/6$, are seen to be higher than the confidence boost threshold (which starts at 1.15 by default) and both $a$ and $b$ are moved to the "ready" heap and yielded to the subsequent rule mining phase. On the other hand, if the confidence boost threshold was, say, at 1.4, upon processing $bcde$ we would find $4/3 < 1.4$ as support ratio of $cde$, and this closure would wait in the freezer heap, until (if at all) a revised lower value of the confidence boost threshold would let it through, by moving it from the freezer queue to the ready queue.

### 3.3 RuleMiner.mine_rules()

In the class RuleMiner, which inherits from Lattice, the iterator mine_rules() relies on the closures provided by the previous iterator in the pipeline:

```
1: reserved_rules = [ ]
2: for each closure from candidate_closures() do
3:   for each predecessor having high enough support so as to reach the confidence threshold do
4:     form a rule r with the predecessor as antecedent and the closure as consequent
5:     use it to revise the closure-based confidence boost threshold
6:     if threshold decreased then
7:       move from Lattice.freezer to Lattice.ready those closures whose support ratio now passes the new threshold
8:       for each rule in reserved_rules do
9:         if its closure-based confidence boost threshold passes the threshold then
10:          yield it
11:        else
12:          keep it in reserved_rules
13:        end if
14:      end for
15:    end if
16:    if the closure-based confidence boost of r passes the threshold then
17:      yield r
18:    else
19:      keep it in reserved_rules
20:    end if
21:  end for
22: end for
```

Each closure makes available an iterator over its predecessors in the closures lattice (closed proper subsets), up to a given support level that we can specify upon calling it. For instance, at the closure *bcde*, of support 3, and assuming a confidence threshold of 0.6, we would explore predecessors *be* and *cde*, which lead to rules $be \to cd$ and $cde \to b$. The confidence boost has to be checked, but part of the task is already made since the very fact that the closure *bcde* arrived here implies that its support ratio is over the confidence boost threshold. In this case, the support ratio of closure *bcde* is 3. We must test confidences with smaller antecedents (see [6]). As the confidences of $b \to cd$ and $e \to cd$ are low enough, the rule $be \to cd$ becomes indeed reported; $cde \to b$ does as well, after checking how low the confidences of $cd \to b$ and $e \to b$ are.

The revision of the closure-based confidence boost threshold can be done in a number of ways. The current implementation keeps computing the lift of those rules whose antecedent is a singleton, as the condition on support ratio ensures that, in this case, it will coincide with the confidence boost [6]; these lift values enter a weighted average with the current threshold, and, if the average is

sufficiently smaller, the threshold is decreased. Only a partial justification exists so far for this choice.

When the threshold for confidence boost decreases, closures whose support ratio was too low may become now high enough; thus, the freezer is explored and closures whose support ratio is now above the new confidence boost threshold are moved into the ready queue (lines 6 and 7), to be processed subsequently.

### 3.4 System

The main program simply traverses all rules, as obtained from the iterator mine_rules(), in the class RuleMiner:

```
1: for each rule in RuleMiner.mine_rules() do
2:     account for it
3: end for
```

What is to be done with each rule depends on the instructions from the user interface, but usually we count how many of them are obtained and we write them all on disk, maybe up to a fixed limit on the number of rules (that can be modified by editing the source code). In this case, we report those of highest closure-based confidence boost.

## 4 A Second Implementation

With a view to offering this system in a more widespread manner, we have developed a joint project with KNIME GmbH, a small company that develops the open source data mining suite KNIME. This data mining suite is implemented in Java. Hence, we have constructed a second implementation in Java.

However, the issue is not fully trivial because of two main reasons. The first is that the notion of iterator in Java is different from that in Python, and is not obtained from generators: the "yield" instruction, which saves the state of an iteration at the time it is invoked, does not exist in Java, which simply declares that hasNext() and next() methods must be made available: respectively, to know whether there are more elements to process and to get the next element. A second significative change is that the memory control to ensure that the list of pending closures does not overflow has to be made in terms of the memory management API of KNIME, and requires one extra loop to check whether the decrease in memory usage was sufficient.

Therefore, we have to use the Iterator class to "copy", to the extent possible, the "yield" behavior, saving all necessary information to continue in queues and lists. The three most affected methods for this issue are, of course, mine_rules(), candidate_closures() and mine_closures(). We describe here only mine_rules().

In this case, a queue, called ready_rules, is needed in order to store the rules that are built from the current closure among the candidates and have achieved the support, confidence, and confidence boost requirements. Rules that do not clear these thresholds are stored in another queue, reserved_rules, as in the Python implementation. The code is shown next:

```
reserved_rules = empty queue of rules
ready_rules = empty queue of rules
ready_rules_iterator = iterator for ready_rules
while !ready_rules_iterator.hasNext() do
  for each closure from candidate_closures() do
    for each predecessor having high enough support so as to reach the
    confidence threshold do
      form a rule r with the predecessor as antecedent and the closure as
      consequent
      use it to revise the threshold for the rule evaluation measure
      if threshold decreased then
        move from Lattice.freezer to Lattice.ready those closures whose
        support ratio now passes the new threshold
        for each rule in reserved_rules do
          if its rule measure passes the new threshold then
            store it in ready_rules
          else
            keep it in reserved_rules
          end if
        end for
      end if
      if the rule measure of r passes the threshold then
        store it in ready_rules
      else
        keep it in reserved_rules
      end if
    end for
  end for
end while
return ready_rules
```

In Lattices.candidate_closures(), the candidate closures are likewise stored in a list called cadidate_closures_list in order that mine_rules method can obtain them. The program is constructed in the same way as the one just described, and is omitted here.

The last method that needs a change in the translation from Python to Java and KNIME is clminer.mine_closures(), and it consists of storing in a list called max-support_itemset_list the candidate itemsets that obey the max-support requirement, and of returning this list at the end of the method. In this case iterators aren't needed beacuse in this method is only required to store and return the list, so next() and hastNext() methods are not used.

```
max-support_itemset_list = empty list of itemset
identify closures of singletons
organize them into a maxheap according to support
while heap nonempty do
```

```
5:    consider increasing the support threshold by monitoring the available
      memory
6:    if the support threshold must be raised then
7:       kill from the heap pending closures of support below new threshold,
         which is chosen so that the size of the heap halves
8:    end if
9:    pop from heap the max-support itemset and store it in max-support_itemset_list
10:   try to extend it with all singleton closures
11:   for such extensions with sufficient support do
12:      if their closure is new then
13:         add it to the heap
14:      end if
15:   end for
16:   return max-support_itemset_list
17: end while
```

## 5 Conclusion

We have studied a variant of the basic association mining process. In our variant, we try to avoid burdening the user with requests to fix threshold parameters. We keep an internal support threshold and adjust it upwards whenever the computation process shows that the system will be unable to run down to the current threshold value. We tackle the problem of limiting the number of rules through one specific rule measure, closure-based confidence boost, for which the threshold is self-adjusted along the mining process. A minor detail is that for full-confidence implications it is not difficult to see that closure-based confidence boost is inappropriate, and plain confidence boost is to be used. Further details about this issue will be given in future work.

The confidence boost constraint is pushed into the mining process through its connection to the support ratio. Therefore, the closure miner has to coordinate with processes that move upwards the support threshold, or downwards the support ratio threshold.

Further study on the basis of our implementation is underway, and further versions of our association miner, with hopefully faster algorithmics, will be provided in the coming months. Another line of activity is as follows: granted that our approach offers partial implications without user-defined parameters, to what extent users that are *not* experts in data analysis are satisfied with the results? Our research group explores that topic in a separate paper [12].

Additionally, we are aware of two independent works where an algorithm is proposed to traverse the closure space in linear time [13], [14]; these algorithms do *not* follow an order of decreasing support, and we find nontrivial to modify them so that they fulfill this condition. Our research group is attempting at it, as, if successful, faster implementations could be designed.

## References


1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag (1999)
2. Luxenburger, M.: Implications partielles dans un contexte. Mathématiques et Sciences Humaines **29** (1991) 35–55
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996) 307–328
4. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surv. **38**(3) (2006)
5. Zaki, M.J., Hsiao, C.J.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering **17**(4) (2005) 462–478
6. Balcázar, J.L.: Formal and computational properties of the confidence boost in association rules. Available at: [http://personales.unican.es/balcazarjl]. Extended abstract appeared as "Objective novelty of association rules: Measuring the confidence boost. In Yahia, S.B., Petit, J.M., eds.: EGC. Volume RNTI-E-19 of Revue des Nouvelles Technologies de lInformation., Cepadu'es-Editions (2010) 297-302" (2010)
7. Kryszkiewicz, M.: Closed set based discovery of representative association rules. In Hoffmann, F., Hand, D.J., Adams, N.M., Fisher, D.H., Guimarães, G., eds.: Proc. of the 4th International Symposium on Intelligent Data Analysis (IDA). Volume 2189 of Lecture Notes in Computer Science., Springer-Verlag (2001) 350–359
8. Balcázar, J.L.: Parameter-free association rule mining with yacaree. In Khenchaf, A., Poncelet, P., eds.: EGC. Volume RNTI-E-20 of Revue des Nouvelles Technologies de l'Information., Hermann-Éditions (2011) 251–254
9. Baixeries, J., Szathmary, L., Valtchev, P., Godin, R.: Yet a faster algorithm for building the Hasse diagram of a concept lattice. In Ferré, S., Rudolph, S., eds.: Proc. of the 7th International Conference on Formal Concept Analysis (ICFCA). Volume 5548 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2009) 162–177
10. Balcázar, J.L., Tîrnăucă, C.: Border algorithms for computing Hasse diagrams of arbitrary lattices. In Valtchev, P., Jäschke, R., eds.: ICFCA. Volume 6628 of Lecture Notes in Computer Science., Springer (2011) 49–64
11. Kuznetsov, S.O., Obiedkov, S.A.: Algorithms for the construction of concept lattices and their diagram graphs. In Raedt, L.D., Siebes, A., eds.: Proc. of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD). Volume 2168 of Lecture Notes in Artificial Intelligence., Springer-Verlag (2001) 289–300
12. García-Sáiz, D., Zorrilla, M., Balcázar, J.L.: Closures and partial implications in educational data mining. ICFCA, Supplementary proceedings (2012)
13. Ganter, B.: Two basic algorithms in concept analysis (preprint 1987). In Kwuida, L., Sertkaya, B., eds.: ICFCA. Volume 5986 of Lecture Notes in Computer Science., Springer (2010) 312–340
14. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In Suzuki, E., Arikawa, S., eds.: Discovery Science. Volume 3245 of Lecture Notes in Computer Science., Springer (2004) 16–31

# Completing Terminological Axioms with Formal Concept Analysis

Alexandre Bazin and Jean-Gabriel Ganascia

Université Pierre et Marie Curie, Laboratoire d'Informatique de Paris 6
Paris, France
Alexandre.Bazin@lip6.fr
Jean-Gabriel@Ganascia.name

**Abstract.** Description logics are a family of logic-based formalisms used to represent knowledge and reason on it. That knowledge, under the form of concepts and relationships between them called terminological axioms, is usually manually entered and used to describe objects in a given domain. That operation being tiresome, we would like to automatically learn those relationships from the set of instances using datamining techniques. In this paper, we study association rules mining in the description logic EL. First, we characterize the set of all possible concepts in a given EL language. Second, we use those characteristics to develop an algorithm using formal concept analysis to mine the rules more efficiently.

**Keywords:** Description Logic, Association Rules Mining, Ontology

## 1 Introduction

Ontologies are knowledge representation tools used in various domains of application. The semantic web, for example, makes an extensive use of them. They are essentially composed of a list of concepts relevant to a particular domain and relations (mainly inclusion and equivalence, i.e. hierarchical relations) existing between them. Description Logics (DL) are increasingly popular logical frameworks used to represent ontologies and on which is based the OWL[1] language for the semantic Web. They have a great representation power and allow powerful reasoning tools. However, the construction of ontologies, usually performed manually by knowledge engineers, is both a tedious and tricky operation. One of the difficulties is to ensure the consistency and the completeness of the set of relations between concepts. in order to facilitate this step, we propose to automatize, at least partially, the process of relation generation.

Based on the lattice theory, Formal Concept Analysis (FCA) is a mathematical framework that also deals with concepts and their hierarchical relationships. FCA provides solid theoretical foundations for association rule learning tools.

[1] OWL is an acronym for *Ontology Web Language*, which is a W3C standard

It therefore seems to be a good natural candidate for this task, i.e. for the automatic generation of relationships between concepts, from object descriptions, i.e. from concept instances.

Despite differences between the use of the notion of concept in these two formalisms, it would be interesting to combine them both and draw benefits from their mutual advantages. This combination has already been investigated and two main approaches exist. The first integrates operators of FCA to the DL framework in order to be able to apply learning algorithms directly to a knowledge base expressed in DL [4] [8], the second, which corresponds to our present work, translates data from DL to a form comprehensible by FCA, in other words, it interprets DL formalism within the lattice theory [2] [3] [7].

We claim that, by using the specific lattice structure of the set of concepts of description logics, we will be able to modify classical FCA algorithms in order to build complete and consistent sets of terminological axioms from object descriptions given as assertions. This work, which constitutes a first attempt in this direction, will make use of a simple description logic, which is $\mathcal{EL}$. But, the approach is not restricted to $\mathcal{EL}$; it will certainly be possible to generalize it to other DL, which will be investigated in further work.

Apart from the introduction and the conclusion, this paper is divided into four parts. The first briefly recalls the usual definitions in both Description Logics and Formal Concept Analysis, the second characterizes the structure of the set of $\mathcal{EL}$-concepts making use of the function $\Phi$ that is the set of subsets of incomparable elements of a language, the third describes a simple association rule learning algorithm that works within the set of $\mathcal{EL}$-concepts previously described. It then studies the properties of the set of terminological axioms that it generates. The last part is dedicated to a brief example, which illustrates the different notions presented in this paper.

## 2 Definitions and Recalls

### 2.1 Description Logics

Descriptions logics are decidable fragments of first-order logic used to represent and reason on knowledge. Syntactically, every description logic language makes use of a set of concept names $N_C$, a set of role names $N_R$ and a set of object names $N_O$ and combines them using constructors to build concept descriptions or, in short, concepts. The set of constructors used defines the language's expression power and the complexity of its reasoning procedures. In this paper, we will consider the logic $\mathcal{EL}$. In it, every concept name is a concept description and, for any concept descriptions $A$ and $B$ and any role $r$, $A \sqcap B$ and $\exists r.A$ are also concept descriptions. Having only two constructors, this logic is one of the simplest.

Semantics are defined by means of interpretations. An interpretation is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a set of objects called the domain and $.^{\mathcal{I}}$ a function mapping every concept name $C$ to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and every role name $r$

to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. As such, concepts are defined by the set of objects which belong to them.

An important notion in description logic systems is the *subsumption* relation between concept descriptions. Given two concept descriptions $C$ and $D$, we say that $D$ *subsumes* $C$ ($C \sqsubseteq D$) if the set of objects belonging to $C$ is included in the set of objects belonging to $D$ ($C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$) for all interpretations $\mathcal{I}$. For a given TBox $\mathcal{T}$, we say that $D$ *subsumes* $C$ *with respect to* $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model of $\mathcal{T}$. If $C \sqsubseteq D$ and $D \sqsubseteq C$, it gives the definition $C \equiv D$. Constructions such as $C \sqsubseteq D$ and $C \equiv D$ expressing subsumption relations are called *terminological axioms*.

For any given concept $C$, role $r$ and object names $o$ and $o'$, $o : C$ and $(o, o') : r$ are called *assertional sentences*. The constructions $o : C$ means that the object $o$ belongs to the concept $C$ and $(o, o') : r$ means that the object $o'$ fulfills the role $r$ for the object $o$.

A *knowledge base* consists of a *TBox* and an *ABox*. The TBox is constituted of terminological axioms, which we try to learn in this paper, and concept definitions. The ABox is a set of assertional sentences and can be viewed as a set of descriptions of objects.

### 2.2 Formal Concept Analysis

In formal concept analysis (FCA), we call *formal context* a triplet $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ where $\mathcal{O}$ is a set of objects, $\mathcal{A}$ a set of attributes and $\mathcal{R}$ a binary relation between objects and attributes. We say here that $(o, a) \in \mathcal{R}$ means that $a$ describes $o$.

We have at our disposal two functions $.'$ such as

$$.' : 2^{\mathcal{A}} \mapsto 2^{\mathcal{O}}$$

$$A' = \bigcap_{a \in A} \{o \in \mathcal{O} \mid (o, a) \in \mathcal{R}\} \tag{1}$$

and

$$.' : 2^{\mathcal{O}} \mapsto 2^{\mathcal{A}}$$

$$O' = \bigcap_{o \in O} \{a \in \mathcal{A} \mid (o, a) \in \mathcal{R}\} \tag{2}$$

$A'$ is then the set of objects described by every attribute of A and $O'$ is the set of attributes describing every object of O. If $A \subseteq B$, then $B' \subseteq A'$ and if $O \subseteq P$ then $P' \subseteq O'$. As such, those two functions form a *Galois Connection*.

A *formal concept* is defined as a pair $(E, I) \in A^{\mathcal{O}} \times 2^{\mathcal{A}}$ where $E = I'$ and $I = E'$. We say that $E$ and $I$ are closed. $E$ and $I$ are respectively called the *extent* and the *intent* of the concept. In order to prevent confusion, formal

concept will not be abbreviated and the term concept will be used exclusively for DL-concepts.

We call $FC(\mathcal{O}, \mathcal{A}, \mathcal{R})$ the set of formal concepts we can find in $(\mathcal{O}, \mathcal{A}, \mathcal{R})$. We can define an order $<$ (a relation "is more general than") on this set such as $(E, I) < (F, J) \Leftrightarrow (F \subset E \text{ and } I \subset J)$ and the pair $(FC(\mathcal{O}, \mathcal{A}, \mathcal{R}), <)$ satisfies the properties of a complete lattice. Such a lattice is called a concept (or Galois) lattice. For example, for a formal context in which $\mathcal{O} = \{a, b, c, d, e\}$, $\mathcal{A} = \{1, 2, 3, 4, 5\}$ and $\mathcal{R} = \{(a, 2), (a, 4), (b, 3), (b, 5), (c, 1), (c, 2), (c, 4), (d, 2), (d, 3), (e, 2), (e, 4)\}$ we obtain the following concept lattice.

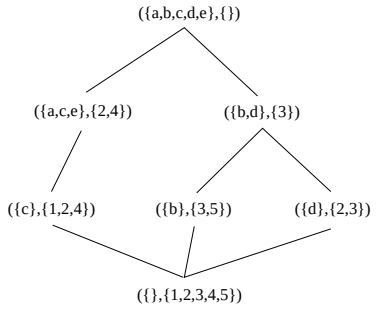



**Fig. 1.** A Concept Lattice

FCA allows us to find implications in the formal context which are ordered pairs $(B, C)$, often written $B \to C$. An implication $B \to C$ holds in a context if every object described by every attribute in $B$ is also described by every attribute in $C$.

**Definition 1.** *We say that a set $X \in A$ respects an implication $B \to C$ if $B \subseteq X$ implies $C \subseteq X$.*

*An implication $B \to C$ follows from a set of implications $\mathcal{L}$ if every $X \in A$ that respects every implication in $\mathcal{L}$ respects $B \to C$. A set $\mathcal{L}$ of implications is then called a basis if every implication in $\mathcal{L}$ holds in the context and every implication that holds in the context follows from $\mathcal{L}$.*

It is a known fact that $\{X \to X'' \mid X \subseteq A\}$ is an implicational basis which means that, in order to obtain a basis of minimal cardinality, we need only to find implications whose right-hand side are concept intents. Finding suitable left-hand side has thus been the subject of many works.

**Definition 2.** *A set $X \in A$ is a pseudo-intent of the context $(O, A, R)$ if $X$ is not a concept intent and, for all pseudo-intent $Y \subset X$, $Y'' \subseteq X$.*

**Definition 3.** *The set of implications* $\{X \rightarrow X'' \mid X$ *is a pseudo-intent*$\}$ *is called a Duquenne-Guigues Basis.*

The Duquenne-Guigues Basis is the minimal set of implication from which we can find every other implications that hold through inference.

## 3 The Set of Concept Descriptions

Before using an association rules learning algorithm, we will study the structure of the set of concepts one can build with the description logic $\mathcal{EL}$.

We will use $\Omega$ to denote the set of terminological axioms $A \sqsubseteq B$ in an acyclic TBox $\mathcal{T}$. $N_C = A_C \cup D_C$ will denote the set of concept names used in $\mathcal{T}$ with $A_C$ the set of atomic concepts and $D_C$ the set of defined concepts, appearing in the left hand side of definitions. The set of pairs $(C1, C2)$ such as $C1 \equiv C2$ will be called $Def(\mathcal{T})$. $\Omega$ induces a partial order on the set of equivalence classes of concepts, noted $N_{C_\equiv}$, used in axioms (if $(x \sqcap y \sqsubseteq z) \in \Omega$, we will consider there is some $d$ in $D_C$ such as $d \equiv x \sqcap y$). We will simply use $b \leq a$ for $[a]_\equiv \sqsubseteq [b]_\equiv$. $(N_{C_\equiv}, \leq)$ is then a partially ordered set such as, for all $x$ in $N_C$, $[\top]_\equiv \leq [x]_\equiv$. For clarity purposes, we will now use $CN^0$ to denote a set of concept names containing a unique representative of each equivalence class together with the order $\leq$. Obviously, $CN^0$ is isomorphic to $(N_{C_\equiv}, \leq)$.

We are interested in the set of every possible concept we can construct with $N_C$, $N_R$ and the constructors $\sqcap$ and $\exists$. Suppose there are two concepts $A$ and $B$ such as $A \sqsubseteq B$. This means that $A^I \subseteq B^I$ so $A \sqcap B \equiv A$. Those two concept descriptions being equivalent we consider they are the same and we do not want to include both of them in the set of possible concepts. As such, we want the set of concepts resulting from the conjunction of incomparable elements.

**Definition 4.** *We call* $\Phi(CN^0) = \{X \subseteq CN^0 \mid x \in X \wedge y \in X \Rightarrow x||y\}$ *the set of subsets of incomparable elements of* $CN^0$

We call $\Phi(CN^0)$ the set of subsets of incomparable elements of $CN^0$ and $\sqcap A$ the concept built from the conjunction of the elements of $A$. For any two elements $C, D \in \Phi(CN^0)$, we say that $C \leq D$ if and only if $\sqcap D \sqsubseteq \sqcap C$. That is, $C \leq D$ if and only if for every element $c \in C$ there is some $d \in D$ such as $c \leq d$. Evidently, $\Phi(CN^0)$ is isomorphic to the set of ideals of $CN^0$ ordered by inclusion and its elements are the sets of maximal elements of those ideals. $\Phi(CN^0)$ is then a distributive lattice.

**Proposition 1.** *For any two elements* $A, B \in \Phi(CN^0)$, $A \wedge B = Max(\{x \in CN^0 \mid \exists a \in A,\ \exists b \in B, x \leq a\ \&\ x \leq b\})$ *and* $A \vee B = Max(A \cup B)$.

$\sqcap(A \wedge B)$ corresponds to the least common subsumer of $\sqcap A$ and $\sqcap B$ and $\sqcap(A \vee B)$ to the most specific concept subsumed by $\sqcap A$ and $\sqcap B$. They can be easily computed from $CN^0$.

$\Phi(CN^0)$ being finite and distributive, for all $A$ and $B$ in $\Phi(CN^0)$, there is a least element $X$ such as $A \vee X \geq B$ called difference and noted $B \setminus A$. It is equal

to $Max(\downarrow B \setminus \downarrow A)$ where $\downarrow A$ is the set of elements lower or equal to elements of $A$ in $CN^0$.

**Proposition 2.** *For a given linear extension $\sigma$ of $CN^0$, the relation $A \rightsquigarrow B \Leftrightarrow B \setminus A = Max_\sigma(B)$ defines a spanning tree of the covering graph of $\Phi(CN^0)$.*

The spanning tree gives us, for every element $A \in \Phi(CN^0)$, a unique path from $\{\top\}$ to $A$ in which $A \rightsquigarrow B \Rightarrow A \leq B$.

$\Phi(CN^0)$ is the set of different conjunctions of concept names based on the subsumption relation. However, the TBox can also contain equivalences between elements of $\Phi(CN^0)$. If $(A, B \sqcap C) \in Def(\mathcal{T})$ then $B \leq A$ and $C \leq A$ in $CN^0$. In $\Phi(CN^0)$, $\{A\}$ is thus strictly greater than $\{B, C\}$. Those two concepts being equivalent, every element greater or equal to $\{B, C\}$ and lower than $\{A\}$ is considered redundant.

**Definition 5.** *$\Phi(CN^0)_{Def(\mathcal{T})} = \Phi(CN^0) \setminus \{B \mid (A, B) \in Def(\mathcal{T})\}$ is the set of subsets of incomparable elements of $CN^0$ without the elements corresponding to right-hand sides of definitions of $\Phi(C, \leq_X)_{Def(\mathcal{T})}$ the TBox.*

**Proposition 3.** *For any two elements $A, B \in \Phi(CN^0)_{Def(\mathcal{T})}$, $A \wedge B$ in $\Phi(CN^0)_{Def(\mathcal{T})}$ is equal to $A \wedge B$ in $\Phi(CN^0)$.*

**Proposition 4.** *For all $A$ and $B$ in $\Phi(CN^0)_{Def(\mathcal{T})}$, the difference $A \setminus B$ in $\Phi(CN^0)$ is an element of $\Phi(CN^0)_{Def(\mathcal{T})}$.*

These operations on $\Phi(CN^0)_{Def(\mathcal{T})}$ are thus the same than on $\Phi(CN^0)$. The differences appear when we try to compute the upper cover of an element $D$, i.e. elements immediately greater than $D$. We call $Cand$ – for candidate – the set of minimal elements not lower than elements of $D$ in $CN^0$. In $\Phi(CN^0)$, the upper cover of $D$ is then $\{Max(D \cup c) \mid c \in Cand\}$. In $\Phi(CN^0)_{Def(\mathcal{T})}$, if there is some $(L, R) \in Def(\mathcal{T})$ such as $L \geq Max(D \cup c) \geq R$, $c$ must be removed from the list of candidates and $L$ added if it is minimal in $Cand \setminus c$. In order to find the elements following $D$ in the spanning tree of $\Phi(CN^0)_{Def(\mathcal{T})}$ induced by some $\sigma$ it would then be sufficient to remove the candidates $c$ such as $c \leq_\sigma d$ for some $d$ in $D$. The algorithm is as follows :

**Algorithm 1**

**Require:** $CN^n$, $D$
1: $Cand = \{c \in CN^n \mid c \in Min(CN^0 \setminus \downarrow D)$ and $\forall d \in D, c \geq_\sigma d\}$
2: **for** each $c \in Cand$ **do**
3: **if** $\exists (L, R) \in Def(\mathcal{T})$ such as $L \geq Max(D \cup c) \geq R$ **then**
4: $Cand = Min((Cand \setminus c) \cup L)$
5: **end if**
6: **end for**
7: Return $\{Max(D \cup c) \mid c \in Cand\}$

Now, $\Phi(CN^0)_{Def(\mathcal{T})}$ is only the lattice of concepts built from a conjunction of concept name without roles. However, it gives us informations on the structure of the set of role-concepts. We know that, for a given role $r$, $A \sqsubseteq B \Rightarrow \exists r.A \sqsubseteq \exists r.B$. The partially ordered set of roles of a depth 1 is then isomorphic to $\Phi(CN^0)_{Def(\mathcal{T})}$. We use $CN^1_r$ to denote it. If $CN^1 = CN^0 \bigcup_{i=1}^{|N_R|} CN^1_{r_i}$ is the set of both concept names and roles of depth 1 together with the partial order induced by $\Omega$, then $\Phi(CN^1)_{Def(\mathcal{T})}$ is the lattice of concepts containing roles up to a depth 1. Recursively, $\Phi(CN^n)_{Def(\mathcal{T})}$ where $CN^n = CN^0 \bigcup_{i=1}^{|N_R|} CN^n_{r_i}$ with $CN^n_{r_i}$ isomorphic to $\Phi(CN^{n-1})_{Def(\mathcal{T})}$ is the set of every possible concept descriptions up to an arbitrary role depth n.

## 4 Learning Axioms with Formal Concept Analysis

As we said previously, we take the approach of creating a formal context corresponding to the DL-objects we want to manipulate. More precisely, we use the formal context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ where $\mathcal{O}$ is a set of objects, $\mathcal{A} = \Phi(CN^n)_{Def(\mathcal{T})}$ is the set of every possible concept descriptions defined in Section 2.2 and $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$ is the relation associating objects to the most specific concept to which they belong. In that respect, it is very similar to contexts from Logical Concept Analysis [5] or the work of Baader [1] which also deals with finding implications in $\mathcal{EL}$.

We re-define the following operators :

$$.' : \mathcal{A} \mapsto 2^{\mathcal{O}}$$

$$X' \;=\; \{o \in \mathcal{O} \mid oRa \Rightarrow X \leq a\} \tag{3}$$

and

$$.' : 2^{\mathcal{O}} \mapsto \mathcal{A}$$

$$O' \;=\; \bigwedge \{a \in \mathcal{A} \mid o \in O \Rightarrow oRa\} \tag{4}$$

The first operator maps a concept description to the set of objects belonging to it while the second is the generalization of the most specific concepts describing the objects, which corresponds to the infimum in the lattice.

Now, if we want to get implications of the form $X \rightarrow X'' \setminus X$, we cannot use the set-theoretic difference directly. The difference $B \setminus A$ in the distributive lattice defined in the previous section corresponds to the most general concept whose conjunction with $A$ would be more specific than $B$. It can also be seen intuitively as the part of $B$ not covered by $A$. Thus, in the remainder of this work, we will use this definition of the difference.

By using the structure of $\Phi(CN^n)_{Def(\mathcal{T})}$, we can enumerate concept descriptions and get a set of implications by using the following algorithm :

**Algorithm 2**

**Require:** $CN^n$, $\sigma$
1: $Open = \{\{\top\}\}$
2: **for** every minimal element $X$ of minimal role depth in $Open$ **do**
3: C = $X''$
4: **if** $C \neq X$ **then**
5: Update $CN^n$ with $X \rightarrow C \setminus X$
6: Add elements following $X$ in the spanning tree of $\Phi(CN^n)_{def(\mathcal{T})}$ to $Open$
7: **end if**
8: **end for**

Beginning with $\{\top\}$, the least element of $\Phi(CN^n)_{def(\mathcal{T})}$, we classically compute its closure. We then compute the closure of every element of $\Phi(CN^n)_{def(\mathcal{T})}$ immediately greater than $\{\top\}$ and so on. Of course, an element of the upper cover of $D$ should not be considered if it contains an element that does not subsume any description of elements of $D'$. As soon as $X''$ is different from $X$ a new implication is found and $CN^n$ is updated, adding a new element to $D_C$ if necessary, and $X$ becomes a closed set of the new $\Phi(CN^n)_{def(\mathcal{T})}$.

For any minimal element $X$ in $Open$, the elements of its lower cover are closed sets. As such, for any $Y \subset X$, $Y'' \subseteq X$. If $X \neq X''$, $X$ is a pseudo-intent. Thus, by considering a minimal element of $Open$ at every step of the algorithm, we make sure we obtain the Duquenne-Guigues Basis of the original context. As a new implication $A \rightarrow B$ changes the structure of the lattice for role concepts we must select the minimal elements in ascending role-depth order.

As an element is added to $N_C$ for every $A \sqcap B \sqsubseteq C$ found and $A \sqcap B$ becomes a closed element of the new $\Phi(CN^n)_{def(\mathcal{T})}$, the algorithm terminates with $CN^n$ isomorphic to the concept lattice of the formal context minus the maximal formal concept.

The method we propose in this paper is similar to the one presented by Rudolph in [6]. However, we feel some important differences must be pointed out. First, our algorithm immediately considers all concepts up to the maximum role depth instead of using a different learning phase for each depth. Second, new implications are immediately included in the background knowledge. We believe this is especially important for axioms of the form $A \sqsubseteq B \sqcap \exists r.C$ where $A \sqsubseteq B$ would be found a first time before the step including roles.

## 5 Example

In our example, $N_C$ = {Man, Woman, Father, Mother, Parent, GrandFather, GrandMother} and $N_R$ = {hasChild}. Moreover, we know that

Mother$\sqsubseteq$ Woman $\sqcap$ Parent

We consider the following set of objects described by concept descriptions

Bob : Man ⊓ Father ⊓ Parent ⊓ ∃hasChild.Man
Bill : Man
Benjamin : Man ⊓ GrandFather ⊓ Father ⊓ Parent ∃hasChild.(Man ⊓ Father ⊓ Parent)
Bertrand : Man ⊓ GrandFather ⊓ Father ⊓ Parent ⊓ ∃hasChild.(Mother ⊓ Parent)
Bernard : Man ⊓ Father ⊓ Parent ⊓ ∃hasChild.Woman
Clara : Mother ⊓ ∃hasChild.Woman
Coralie : Mother ⊓ GrandMother ⊓ ∃hasChild.(Man ⊓ Father ⊓ Parent)
Claire : Mother ⊓ GrandMother ⊓ ∃hasChild.(Mother ⊓ Parent)
Chloe : Woman

Initially, $Open = \{\top\}$ and $CN^n$ is as follows :

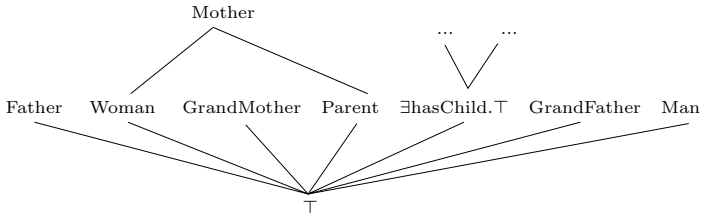



**Fig. 2.** $CN^n$ at Step 0 (irrelevant role-concepts omitted)

$\top'' = \emptyset$ so there is no new implication.

$Open = \{$Woman, Father, GrandMother, Parent, ∃hasChild.⊤, GrandFather, Man$\}$

Woman$'' = \emptyset$ so there is no new implication.

$Open = \{$Father, GrandMother, Parent, ∃hasChild.⊤, GrandFather, Man, Mother, {Woman, Father}, {Woman, GrandMother}, {Woman, Parent}, {Woman, ∃hasChild.⊤}, {Woman, GrandFather}, {Woman, Man}$\}$

Father$'' = \{$Father, Man, Parent, ∃hasChild.⊤$\}$ so The implication Father $\rightarrow$ {Man, Parent, ∃hasChild.⊤} is added.

$CN^n$ is then updated.

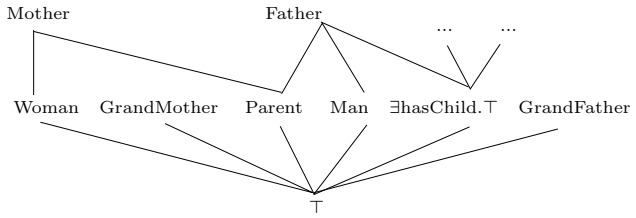



**Fig. 3.** $CN^n$ at Step 3 (irrelevant role-concepts omitted)

*Open* = {GrandMother, Parent, ∃hasChild.⊤, GrandFather, Man, Mother, {Woman, Father}, {Woman, GrandMother}, {Woman, Parent}, {Woman, ∃hasChild.⊤}, {Woman, GrandFather}, {Woman, Man}, {Father, GrandMother}, {Father, GrandFather}, {Father, ∃hasChild.Parent}, {Father, ∃hasChild.Woman}, {Father, ∃hasChild.Man}}

Others implications are then found for GrandMother, Parent, ∃hasChild.⊤, {Woman, Parent}, GrandFather, {Father, ∃hasChild.Parent}, {Man, Parent}, {Mother, ∃hasChild.Parent}, {Father, ∃hasChild.Parent}. The algorithm terminates with $CN^n$ in the following state.

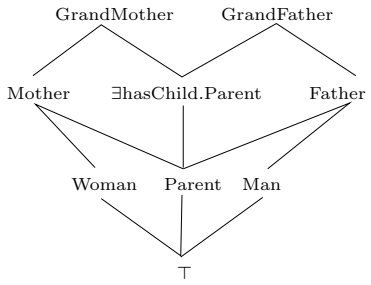



**Fig. 4.** $CN^n$ at the end of the algorithm (irrelevant role-concepts omitted)

Note that ∃hasChild.⊤ does not appear in $CN^n$ because it has been found equivalent to Parent.

The following terminological axioms have been found :

$$\text{Father} \equiv \text{Parent} \sqcap \text{Man}$$
$$\text{Mother} \equiv \text{Parent} \sqcap \text{Woman}$$
$$\text{GrandMother} \equiv \text{Mother} \sqcap \exists\text{hasChild.Parent}$$
$$\text{GrandFather} \equiv \text{Father} \sqcap \exists\text{hasChild.Parent}$$
$$\exists\text{hasChild.}\top \equiv \text{Parent}$$

## 6 Conclusion

As mentionned in the introduction, this research aims at completing the TBox with terminological axioms learned from assertions contained in an ABox. Our approach translates data from DL formalism, that is instances of the ABox, to a form homogeneous to FCA, i.e. to lattices. More precisely, by using the lattice structure of the set of concepts of description logics, we modify classical FCA algorithms in order to build complete and consistent sets of terminological axioms from object descriptions given as assertions.

In this paper, we have restricted our approach to $\mathcal{EL}$. We have shown that the set of $\mathcal{EL}$-concept descriptions, ordered by the subsumption relation, is isomorphic to a certain subset – that depends on the definitions of the TBox – of the lattice of ideals of the partially ordered set of equivalence classes built on the union of concept names and role concepts. We then proposed a simple algorithm exploiting this structure to learn terminological axioms from examples. Every implication found in the data is added to the TBox. We can easily make this algorithm interactive. More precisely, it is possible to change it into an attribute exploration-like algorithm in which experts are asked about each axiom and may give counterexamples. In this work, we dealt with description logic $\mathcal{EL}$ but the main idea of considering sets of incomparable concepts names is also valid for DLs with the concept $\bot$ or the constructor $\forall$. However, it no longer works with constructors such as the negation because it adds new constraints between concept names. More complex DL languages will be the subject of future investigations on our part.

## References

1. Franz Baader and Felix Distel. A finite basis for the set of el-implications holding in a finite model. In *In ICFCA, vol.4933 of LNAI*, pages 46–61. Springer Verlag, 2008.
2. Franz Baader, Bernhard Ganter, Baris Sertkaya, and Ulrike Sattler. Completing description logic knowledge bases using formal concept analysis. In *In Proc. of IJCAI 2007*, pages 230–235. AAAI Press, 2007.
3. Franz Baader and Baris Sertkaya. Applying formal concept analysis to description logics. In Peter Eklund, editor, *Concept Lattices*, volume 2961 of *Lecture Notes in Computer Science*, pages 593–594. Springer Berlin / Heidelberg, 2004.
4. Felix Distel. *Learning Description Logic Knowledge Bases from Data Using Methods from Formal Concept Analysis*. PhD thesis, Technische Universität Dresden, 2011.
5. Sébastien Ferré and Olivier Ridoux. A logical generalization of formal concept analysis. In *Int. Conf. Conceptual Structures, LNCS 1867*, pages 371–384. Springer, 2000.
6. Sebastian Rudolph. Exploring relational structures via fle. In *Conceptual Structures at Work: 12th International Conference on Conceptual Structures. Volume 3127 of LNCS*. Springer, 2004.
7. Sebastian Rudolph. *Relational Exploration - Combining Description Logics and Formal Concept Analysis for Knowledge Specification*. Universitätsverlag Karlsruhe, December 2006.

8. N. V. Shilov and S.-Y. Han. A proposal of description logic on concept lattices. In *Proceedings of the Fifth International Conference on Concept Lattices and their Applications*, 2007.

# Structural properties and algorithms on the lattice of Moore co-families

Laurent Beaudou[1] and Pierre Colomb[1] and Olivier Raynaud[1]

Université Blaise Pascal, Campus Universitaire des Cézeaux, 63173 Aubière, France

**Abstract.** A collection of sets on a ground set $U_n$ ($U_n$ denotes the set $\{1, 2, ..., n\}$) closed under intersection and containing $U_n$ is known as a Moore family. The set of Moore families for a fixed $n$ is in bijection with the set of Moore co-families (union-closed families containing the empty set) denoted $\mathbb{M}_n$. In this paper, we show that the set $\mathbb{M}_n$ can be endowed with the quotient partition associated with some operator $h$. This operator $h$ is the main concept underlying a recursive description of $\mathbb{M}_n$. By this way each class of the partition contains all the families which have the same image by $h$. Then we prove some structural results linking any Moore co-family to its image by $h$. From these results we derive an algorithm which computes efficiently the image by $h$ of any given Moore co-family.

**Key words:** Moore co-families, Formal Concept Analysis, lattices

## References

1. Barbut, M., Monjardet, B.: *Ordre et classification.* Hachette (1970)
2. Birkhoff, G.: *Lattice Theory.* Third edn. American Mathematical Society (1967)
3. Birkhoff, G.: Rings of sets. *Duke Mathematical Journal* **3** (1937) 443–454
4. Caspard, N., Monjardet, B.: The lattices of closure systems, closure operators, and implicational systems on a finite set: a survey. *Discrete Appl. Math.* **127** (2003) 241–269
5. Cohn, P.: *Universal Algebra.* Harper and Row, New York (1965)
6. Colomb, P., Irlande, A., Raynaud, O.: Counting of Moore families on n=7. In: *ICFCA, Lecture Notes in Artificial Intelligence 5986, Springer.* (2010)
7. Colomb, P., Irlande, A., Raynaud, O., Renaud, Y.: About the recursive décomposition of the lattice of moore co-families. In: *ICFCA.* (2011)
8. Davey, B.A., Priestley, H.A.: *Introduction to lattices and orders.* Second edn. Cambridge University Press (2002)
9. Demetrovics, J., Molnar, A., Thalheim, B.: Reasoning methods for designing and surveying relationships described by sets of functional constraints. *Serdica J. Computing* **3** (2009) 179–204
10. Demetrovics, J., Libkin, L., Muchnik, I.: Functional dependencies in relational databases: A lattice point of view. *Discrete Appl. Math.* **40(2)** (1992) 155–185
11. Doignon, J.P., Falmagne, J.C.: *Knowledge Spaces.* Springer, Berlin (1999)
12. Duquenne, V.: Latticial structure in data analysis. *Theoretical Computer Science* **217** (1999) 407–436

13. Ganter, B., Wille, R.: *Formal Concept Analysis.* mathematical foundations, Berlin-Heidelberg-NewYork, Springer (1999)
14. Habib, M., Nourine, L.: The number of Moore families on n=6. *Discrete Mathematics* **294** (2005) 291–296
15. Sierksma, G: Convexity on union of sets. *Compositio Mathematica* **volume 42** (1981) 391–400
16. van de Vel, M.L.J.: *Theory of convex structures.* North-Holland, Amsterdam (1993)

# A Tool-Based Set Theoretic Framework for Concept Approximation

Zoltán Csajbók[1] and Tamás Mihálydeák[2]

[1] Department of Health Informatics, Faculty of Health, University of Debrecen,
Sóstói út 2-4, H-4400 Nyíregyháza, Hungary
csajbok.zoltan@foh.unideb.hu
[2] Department of Computer Science, Faculty of Informatics, University of Debrecen
Egyetem tér 1, H-4032 Debrecen, Hungary
mihalydeak.tamas@inf.unideb.hu

**Abstract.** Modelling positive and negative knowledge has a long-standing tradition in Formal Concept Analysis. To approximate concepts we propose a tool-based set theoretic partial approximation framework in which positive features and their negative counterparts of observed objects can be approximated simultaneously.

**Key words:** Concept approximation, positive-negative knowledge, rough set theory, partial approximation framework

## 1 Introduction

Modelling of learning from positive and negative examples has a long-standing tradition in machine learning, for a brief historical survey see, e.g. [16]. A possible model in terms of Formal Concept Analysis was described in [10, 15].

The idea of knowing negatively was introduced explicitly by M. Minsky in [19]. Negative knowledge has a number of beneficial effects in professional contexts which are discussed in detailed, e.g. in [12, 19]. The adjectives 'positive' and 'negative' do not imply a valuation *per se*. 'Positive' knowledge is not good, advantageous or benign, whereas 'negative' knowledge is not bad, disadvantageous or malign in and of itself.

Both positive and negative knowledge have procedural [19] and declarative aspects [23]. A procedural aspect of positive and negative knowledge can be paraphrased as 'to know what to do' and 'to know what not to do' resp., whereas a declarative aspect as 'to know what one knows' and 'to know what one does not know' resp. In addition, both positive and negative knowledge have two different degrees of knowing or not-knowing. Positive knowledge is informed (uninformed) when one is (not) aware of his/her own relevant knowledge. Negative knowledge is informed (uninformed) when one is (not) aware of his/her own lack of relevant knowledge. Our discussion deals with the declarative aspect of positive and negative knowledge and informed way of knowing/not-knowing.

In our approach, first, we consider a class of objects which is modelled as an abstract set, called the universe of discourse. We assume that a concept is defined over the universe as a subset. In real life the concepts are usually expressed in natural language and so their exact definition cannot be given. The concept approximation is a fundamental problem in artificial intelligence in order to be able to solve real-world problems [17, 22, 31]. A possible way to approximate concepts is to induce their approximations from available experimental (observed, measured) data which is also modelled as subsets over the universe [29–31]. Concepts are generally rough, whereas measurements are always crisp.

It is also assumed that we have some well-defined, decidable features with which an observed object possesses or not. These features assign crisp subsets within the universe. In other words, we model an object of interest as a member of an abstract set, called the universe, and its property 'it possesses a feature' as 'it is the element of a crisp subset of the universe'.

In practice, a concept, of course, cannot be specified completely over the universe. Instead, two relevant sample groups of objects can be established determined by our currently available and necessarily constrained knowledge: a group of which members characteristically possess some features concerning the concept in question, and another group of which members do not substantially possess the same features. Both groups correspond two crisp subsets of the universe. They are disjoint, and, in general, the union of them does not add up to the whole universe. For obvious reasons, the former can be marked with the adjective positive and called the *positive sample set*, whereas the latter with negative and called the *negative sample set*.

Moreover, in real life, a feature of objects cannot be observed directly as well. We need *tools* at our disposal with which we are able to measure one or more constituents of a feature which are called *properties*. For instance, let us say that we observe velocity (feature) of cars (objects). Velocity is a vector quantity with *speed* and *direction*. They are two properties of velocity which can be measured simultaneously and both of them can be expressed numerically. And so, a car is modelled as a member of an abstract set, the universe, and its velocity as it is a member of intersection of two subsets of cars with given speed and given direction (tools) which were measured at the same time.

It is assumed that we are able to judge easily and unambiguously whether an object possesses a property ascertained by a tool or not. It is expected that tools can be used simply and quickly. The objects classified by a tool can be modelled as a crisp subset of the universe. With a slight abuse of terminology, this subset is also simply called tool.

Different tools form different subsets, but they are not necessarily disjoint. Intersections of not disjoint tools are also viewed as tools. The complement of a tool is not necessarily a tool at the same time. In practice, there are properties which can be measured but their counterparts cannot. For instance, a given disease can be diagnosed but the health cannot be measured. These significant facts confirm the partial nature of our approach.

Let us distinguish two types of tools: *positive* and *negative* ones. It is a natural assumption that a subset cannot be positive and negative tool simultaneously.

To manage the problem outlined above we need an *approximation framework*. It may be built on the *rough set theory* because it provides a powerful foundation to reveal and discover important structures in data and classify complex objects [27, 28]. The rough set theory was introduced by the Polish mathematician, Z. Pawlak in the early 1980s [24, 25]. It can be seen as a new mathematical approach to *vagueness* [14]. According to Pawlak's idea, the vagueness of a subset within the universe $U$ is defined by the difference of its upper and lower approximations with respect to a partition of $U$. Using partitions, however, is a very strict requirement. Our starting point is an *arbitrary* family of subsets of $U$ which does not cover the universe necessarily. The lower and upper approximations are straightforward point-free generalizations of Pawlak's ones [2–6]. We apply them to build a *set theoretic tool-based partial approximation framework* in which positive features and their negative counterparts of any clump of observed objects can be approximated *simultaneously*.

The rest of the paper is organized as follows. Section 2 sums up the most important features of rough set theory and partial approximation spaces. Classical rough set theory and formal concept analysis use similar structures to represent information which is briefly described in Section 3. In Section 4 we will propose a tool-based set theoretical framework for concept approximation based on partial approximation spaces. Its main notions are illustrated in Section 5. Finally, in Section 6, we conclude the paper.

## 2 Partial Approximation of Sets

First, we summarize the most important concepts and properties of rough set theory [13, 25]. Let $U$ be a nonempty set and $\varepsilon$ be an equivalence relation on $U$. Let $U/\varepsilon$ denote the partition of $U$ generated by $\varepsilon$. Members of the partition are called *$\varepsilon$-elementary sets*. $X \subseteq U$ is *$\varepsilon$-definable*, if it is a union of $\varepsilon$-elementary sets, otherwise *$\varepsilon$-undefinable*. By definition, the empty set is considered to be an $\varepsilon$-definable set.

The pair $\langle U, \varepsilon \rangle$ is called a Pawlakean approximation space. The lower and upper $\varepsilon$-approximations of $X \subseteq U$ can be defined as follows.

The *lower $\varepsilon$-approximation* of $X$ is[3]

$$\underline{\varepsilon}(X) = \bigcup \{Y \mid Y \in U/\varepsilon, Y \subseteq X\},$$

and the *upper $\varepsilon$-approximation* of $X$ is

$$\overline{\varepsilon}(X) = \bigcup \{Y \mid Y \in U/\varepsilon, Y \cap X \neq \emptyset\}.$$

The set $B_\varepsilon(X) = \overline{\varepsilon}(X) \setminus \underline{\varepsilon}(X)$ is the *$\varepsilon$-boundary* of $X$. $X$ is *$\varepsilon$-crisp*, if $B_\varepsilon(X) = \emptyset$, otherwise $X$ is *$\varepsilon$-rough*.

[3] If $\mathfrak{A} \subseteq 2^U$, we define $\bigcup \mathfrak{A} = \{x \mid \exists A \in \mathfrak{A}(x \in A)\}$, and $\bigcap \mathfrak{A} = \{x \mid \forall A \in \mathfrak{A}(x \in A)\}$. If $\mathfrak{A}$ is an empty family of sets, $\bigcup \emptyset = \emptyset$ and $\bigcap \emptyset = U$.

Let $\mathfrak{D}_{U/\varepsilon}$ denote the family of $\varepsilon$-definable subsets of $U$. Clearly, $\underline{\varepsilon}(X), \overline{\varepsilon}(X) \in \mathfrak{D}_{U/\varepsilon}$, and the maps $\underline{\varepsilon}, \overline{\varepsilon} : 2^U \to \mathfrak{D}_{U/\varepsilon}$ are monotone, total and many-to-one. It can easily be seen ([25], Proposition 2.2, points 1, 9, 10) that the map $\underline{\varepsilon}$ is *contractive* and $\overline{\varepsilon}$ is *extensive*, i.e. $\forall X \in 2^U(\underline{\varepsilon}(X) \subseteq X \subseteq \overline{\varepsilon}(X))$. In other words, $X$ is bounded by its lower and upper approximations.

Now, let us turn to the theory of partial approximation of sets [2, 4, 5]. Its most fundamental notion is the base system.

**Definition 1.** *Let $\mathfrak{B} \subseteq 2^U$ be a nonempty family of nonempty subsets of $U$.*
*$\mathfrak{B}$ is called the* base system*, its members are the* $\mathfrak{B}$-sets.

An extension of the base system is specified by the next definition.

**Definition 2.** *A nonempty subset $X \subseteq U$ is* $\mathfrak{B}$-definable *if there exists a family of sets $\mathfrak{D} \subseteq \mathfrak{B}$ in such a way that $X = \bigcup \mathfrak{D}$, otherwise $X$ is* $\mathfrak{B}$-undefinable.
*The empty set is considered to be a $\mathfrak{B}$-definable set.*
*Let $\mathfrak{D}_{\mathfrak{B}}$ denote the family of $\mathfrak{B}$-definable sets of $U$.*

Note that neither the base system $\mathfrak{B}$ nor $\mathfrak{D}_{\mathfrak{B}}$ covers the universe necessarily.

Let us define the lower and upper approximations of sets based on partial covering of the universe.

**Definition 3.** *Let $\mathfrak{B} \subseteq 2^U$ be a base system and $X$ be any subset of $U$.*
*The* lower $\mathfrak{B}$-approximation of $X$ *(Fig. 1) is*

$$\mathfrak{C}^{\flat}_{\mathfrak{B}}(X) = \bigcup\{Y \mid Y \in \mathfrak{B}, Y \cap X = Y\},$$

*the* upper $\mathfrak{B}$-approximation of $X$ *(Fig. 2) is*

$$\mathfrak{C}^{\sharp}_{\mathfrak{B}}(X) = \bigcup\{Y \mid Y \in \mathfrak{B}, Y \cap X \neq \emptyset\}.$$

*Remark 1.* In Definition 3, the members of the base system may be seen as the elements of the lattice $2^U$, and instead of set theoretic operations may be used lattice operations. In this way, point-free generalizations of Pawlakean lower and upper approximations can be obtained.

Clearly, $\mathfrak{C}^{\flat}_{\mathfrak{B}}(X), \mathfrak{C}^{\sharp}_{\mathfrak{B}}(X) \in \mathfrak{D}_{\mathfrak{B}}$, and the maps $\mathfrak{C}^{\flat}_{\mathfrak{B}}, \mathfrak{C}^{\sharp}_{\mathfrak{B}} : 2^U \to \mathfrak{D}_{\mathfrak{B}}$ are total, monotone and in general many-to-one.

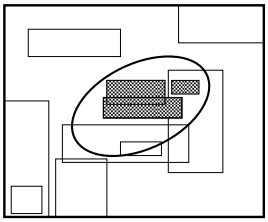


**Fig. 1.** Lower approximation

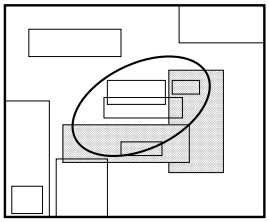


**Fig. 2.** Upper approximation

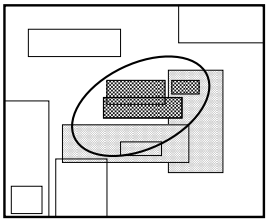


**Fig. 3.** Lower and upper approximations

**Proposition 1 ([6], Proposition 4.8).** *Let $\mathfrak{B} \subseteq 2^U$ be a base system. Then*

1. $\forall X \in 2^U(\mathfrak{C}^{\flat}_{\mathfrak{B}}(X) \subseteq \mathfrak{C}^{\sharp}_{\mathfrak{B}}(X))$;
2. $\forall X \in 2^U(\mathfrak{C}^{\flat}_{\mathfrak{B}}(X) \subseteq X)$—*that is,* $\mathfrak{C}^{\flat}_{\mathfrak{B}}$ *is contractive;*
3. $\forall X \in 2^U(X \subseteq \mathfrak{C}^{\sharp}_{\mathfrak{B}}(X))$ *if and only if* $\bigcup \mathfrak{B} = U$—*that is,* $\mathfrak{C}^{\sharp}_{\mathfrak{B}}$ *is extensive if and only if* $\mathfrak{B}$ *covers the universe.*

Using the previous notations, the notion of the partial approximation space can be introduced.

**Definition 4.** *The ordered quadruple* $\langle U, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{C}^{\flat}_{\mathfrak{B}}, \mathfrak{C}^{\sharp}_{\mathfrak{B}} \rangle$ *is called the* (weak) partial $\mathfrak{B}$-approximation space.

Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be two posets.

**Definition 5.** *The pair of maps* $f : P \to Q$ *and* $g : Q \to P$ *forms a* (regular) Galois connection *between* $P$ *and* $Q$, *in notation* $\mathbb{G}(P, f, g, Q)$, *if*

$$\forall p \in P \forall q \in Q \ (f(p) \leq_Q q \Leftrightarrow p \leq_P g(q)).$$

*If* $P = Q$, $\mathbb{G}(P, f, g, P)$ *is called a* Galois connection on $P$.

*Remark 2.* Here we adopted the definition of Galois connection in which the maps are monotone. It is also called monotone or covariant form. For more details on Galois connections, see, e.g. [8]. Note that since Galois connections are not necessarily symmetric, the order of the maps is important.

It is well known fact ([13], Proposition 138) that upper and lower $\varepsilon$-approximations form a Galois connection $\mathbb{G}(2^U, \overline{\varepsilon}, \underline{\varepsilon}, 2^U)$ on $(2^U, \subseteq)$. Next theorem shows the conditions under which upper and lower $\mathfrak{B}$-approximations also form a Galois connection.

**Theorem 1 ([6], Theorem 4.14).** *Let* $\langle U, \mathfrak{B}, \mathfrak{C}^{\flat}_{\mathfrak{B}}, \mathfrak{C}^{\sharp}_{\mathfrak{B}} \rangle$ *be a partial* $\mathfrak{B}$*-approximation space. The upper and lower* $\mathfrak{B}$*-approximations form a Galois connection* $\mathbb{G}(2^U, \mathfrak{C}^{\sharp}_{\mathfrak{B}}, \mathfrak{C}^{\flat}_{\mathfrak{B}}, 2^U)$ *on* $(2^U, \subseteq)$ *if and only if the base system* $\mathfrak{B}$ *is a partition of* $U$.

According to Proposition 1, point 3, $X \subseteq \mathfrak{C}^{\sharp}_{\mathfrak{B}}(X)$ if and only if the base system $\mathfrak{B}$ covers the universe.

**Definition 6.** *A subset* $X \subseteq U$ *is* $\mathfrak{B}$-approximatable *if* $X \subseteq \mathfrak{C}^{\sharp}_{\mathfrak{B}}(X)$, *otherwise it is said that* $X$ *has a* $\mathfrak{B}$-approximation gap.

A $\mathfrak{B}$-approximation gap may be interpreted so that our knowledge about the universe encoded in the base system is incomplete and not enough to approximate $X$.

**Definition 7.** *Let* $\langle 2^U, \mathfrak{D}_{\mathfrak{B}}, \mathfrak{C}^{\flat}_{\mathfrak{B}}, \mathfrak{C}^{\sharp}_{\mathfrak{B}} \rangle$ *be a partial* $\mathfrak{B}$*-approximation space, and* $X$ *be any subset of* $U$.

*The* partial upper $\mathfrak{B}$-approximation *of* $X$ *is*

$$\partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}(X) = \begin{cases} \mathfrak{C}^{\sharp}_{\mathfrak{B}}(X), & \text{if } X \text{ is } \mathfrak{B}\text{-approximatable;} \\ \text{undefined, otherwise.} \end{cases} \quad (1)$$

There exists at least one nonempty $B \in \mathfrak{B}$ $\mathfrak{B}$-set by Definition 2. Then $B \subseteq \mathfrak{C}^{\sharp}_{\mathfrak{B}}(B)$ according to Definition 3. Hence, $\partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}$ is defined on at least one nonempty subset of $U$.

Notice that $\mathfrak{C}^{\flat}_{\mathfrak{B}}(X) \subseteq X \subseteq \partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}(X)$ holds provided $X$ is $\mathfrak{B}$-approximatable.

As Theorem 1 shows, the upper and lower $\mathfrak{B}$-approximations form a Galois connection on $(2^U, \subseteq)$ if and only if the base system $\mathfrak{B}$ is a partition of $U$. The question naturally arises whether the Galois connection could be generalized so that the maps $\partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}$ and $\mathfrak{C}^{\flat}_{\mathfrak{B}}$ may form a Galois connection in any sense. Moreover, if the answer is yes, then what conditions have to be fulfilled by a partial $\mathfrak{B}$-approximation space so that $\partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}$ and $\mathfrak{C}^{\flat}_{\mathfrak{B}}$ form a Galois connection of this special type. Recall that $\mathfrak{C}^{\flat}_{\mathfrak{B}}$ is a total and $\partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}$ is a partial map on $2^U$.

To answer this question, first of all, we need a suitable modified notion of Galois connections.

**Definition 8 ([18], Definition 2.2.2).** *The pair of maps $f : P \to Q$ and $g : Q \to P$ forms a* partial Galois connection *between $P$ and $Q$, denoted by $\partial\mathbb{G}(P, \partial f, g, Q)$, if*

1. *$f : P \to Q$ is a monotone partial map,*
2. *$g : Q \to P$ is a monotone total map,*
3. *$f(g(q))$ exists for all $q \in Q$, and*
4. *$\forall p \in P$ and $\forall q \in Q$ such that $f(p)$ is defined, $f(p) \leq_Q q \Leftrightarrow p \leq_P g(q)$.*

*Remark 3.* In [18], A. Miné actually introduced the concept of $\mathcal{F}$-partial Galois connection $\partial\mathbb{G}(P, \partial f, g, Q)$ between the concrete domain $P$ and the abstract domain $Q$, where $\mathcal{F}$ is a set of concrete operators. We apply this notion in the simplest form when $P = Q = 2^U$ and $\mathcal{F} = \emptyset$. It is allowed by Miné's definition.

**Theorem 2 ([6], Theorem 4.22).** *Let $\langle U, \mathfrak{B}, \mathfrak{C}^{\flat}_{\mathfrak{B}}, \mathfrak{C}^{\sharp}_{\mathfrak{B}} \rangle$ be a partial $\mathfrak{B}$-approximation space.*

*The partial upper $\mathfrak{B}$-approximation and the lower $\mathfrak{B}$-approximation form a partial Galois connection $\partial\mathbb{G}(2^U, \partial\mathfrak{C}^{\sharp}_{\mathfrak{B}}, \mathfrak{C}^{\flat}_{\mathfrak{B}}, 2^U)$ on $(2^U, \subseteq)$ if and only if the $\mathfrak{B}$-sets are pairwise disjoint.*

A natural question is how we can form a base system from an arbitrary one of which members are pairwise disjoint. In practice, this problem can be reduced to finite base systems. A possible way to construct such a base system is the following.

First, let us form an intersection structure from an arbitrary finite base system. Formally, a nonempty family $\mathfrak{S} \subseteq 2^U$ is an intersection structure if $\forall \mathfrak{S}' (\neq \emptyset) \subseteq \mathfrak{S} \, (\bigcap \mathfrak{S}' \in \mathfrak{S})$, i.e. it is closed under intersection but does not contain $U$ necessarily [7].

Let us take an arbitrary finite base system $\mathfrak{B}$ and create its intersection structure $IS(\mathfrak{B})$ as the smallest set which satisfies the following two properties:

1. $\mathfrak{B} \subseteq IS(\mathfrak{B})$.
2. If $\mathfrak{B}' \subseteq IS(\mathfrak{B})$, then $\bigcap \mathfrak{B}' \in IS(\mathfrak{B})$.

Having given the intersection structure $IS(\mathfrak{B})$, we can create a family of sets $IS_{\Pi}(\mathfrak{B})$ of which members are pairwise disjoint. $IS_{\Pi}(\mathfrak{B})$ is the smallest family of sets which satisfies the following property:

$$\text{If } u \in U \text{ and } \mathfrak{B}' = \{B : B \in \mathfrak{B} \wedge u \in B\}, \text{ then } \bigcap \mathfrak{B}' \in IS_{\Pi}(\mathfrak{B}).$$

## 3 Rough Set Theory and Formal Concept Analysis

Let $G$ and $M$ denote a set of objects and a finite set of attributes, respectively. Note that the formal concept analysis allows that $G$ and $M$ to be empty sets, but the rough set theory does not.

### 3.1 Information Systems

First, we reformulate the rough set theory [9, 24]. Let $S = \langle G, M, V_{m \in M}, f \rangle$ be an *information system*, where $G$ and $M$ as before, $V_m$ is a nonempty set of values of attribute $m \in M$, and $f : G \times M \to V = \bigcup_{m \in M} V_m$ is an information function with $\forall g \in G \, \forall m \in M \, (f(g, m) \in V_m)$. Informally, $f(g, m)$ represents the value which object $g$ takes at attribute $m$.

The information system is often represented by a table, as shown in Table 1. It is an information table containing a shortened student grade history from an information technology course held for hospital nurses at the Faculty of Health, University of Debrecen. It contains 20 students and their results in three homework assignments, and one final examination.

**Table 1.** Information system of a shortened student grade history (complete)

| Student | Hw1 | Hw2 | Hw3 | Final exam |
|---|---|---|---|---|
| $S_1$ | 1 | 1 | 1 | **1** |
| $S_2$ | 1 | 1 | 2 | **2** |
| $S_3$ | 1 | 1 | 1 | **1** |
| $S_4$ | 1 | 2 | 1 | **1** |
| $S_5$ | 1 | 1 | 1 | **1** |
| $S_6$ | 1 | 1 | 2 | **1** |
| $S_7$ | 4 | 1 | 3 | **1** |
| $S_8$ | 2 | 4 | 1 | **2** |
| $S_9$ | 1 | 3 | 1 | **2** |
| $S_{10}$ | 1 | 1 | 3 | **1** |
| $S_{11}$ | 2 | 1 | 1 | **2** |
| $S_{12}$ | 1 | 1 | 1 | **1** |
| $S_{13}$ | 1 | 2 | 1 | **1** |
| $S_{14}$ | 1 | 1 | 2 | **3** |
| $S_{15}$ | 4 | 3 | 3 | **4** |
| $S_{16}$ | 2 | 1 | 1 | **4** |
| $S_{17}$ | 2 | 2 | 2 | **4** |
| $S_{18}$ | 4 | 4 | 3 | **3** |
| $S_{19}$ | 4 | 3 | 3 | **2** |
| $S_{20}$ | 4 | 4 | 3 | **4** |

**Table 2.** Information system of a shortened student grade history (partial)

| Student | Hw1 | Hw2 | Hw3 | Final exam |
|---|---|---|---|---|
| $S_1$ | 1 | 1 | 1 | **1** |
| $S_2$ | 1 | 1 | 2 | **2** |
| $S_3$ | 1 | 1 | 1 | **1** |
| $S_4$ | | 2 | | **1** |
| $S_5$ | 1 | 1 | 1 | **1** |
| $S_6$ | 1 | 1 | 2 | **1** |
| $S_7$ | | 1 | 3 | **1** |
| $S_8$ | 2 | 4 | 1 | **2** |
| $S_9$ | 1 | 3 | 1 | **2** |
| $S_{10}$ | 1 | 1 | 3 | **1** |
| $S_{11}$ | 2 | 1 | 1 | **2** |
| $S_{12}$ | 1 | 1 | 1 | **1** |
| $S_{13}$ | 1 | 2 | 1 | **1** |
| $S_{14}$ | 1 | 1 | 2 | **3** |
| $S_{15}$ | 4 | 3 | 3 | **4** |
| $S_{16}$ | 2 | 1 | 1 | **4** |
| $S_{17}$ | | | | **4** |
| $S_{18}$ | 4 | 4 | 3 | **3** |
| $S_{19}$ | 4 | 3 | 3 | **2** |
| $S_{20}$ | 4 | 4 | 3 | **4** |

With each $N \subseteq M$ we associate an equivalence relation $E_N \subseteq G \times G$ by

$$(g_1, g_2) \in E_N \text{ if } \forall n \in N\,(f(g_1, n) = f(g_2, n)).$$

If $g \in G$, then $[g]_{E_N}$ is the equivalence class of $E_N$ containing $g$. Let $G/N$ denote the set of equivalence classes generated by $E_N$.

A concept $X \subseteq G$ is $E_N$-definable or $E_N$-exact if $X$ is a union of some equivalence classes, otherwise $X$ is $E_N$-undefinable or $E_N$-inexact.

Lower and upper $E_N$-approximations of $X$ are:

$$\underline{E_N}(X) = \bigcup\{[g]_{E_N} \in G/N \mid [g]_{E_N} \subseteq X\},$$
$$\overline{E_N}(X) = \bigcup\{[g]_{E_N} \in G/N \mid [g]_{E_N} \cap X \neq \emptyset\}.$$

### 3.2 Formal Context

In formal concept analysis a *formal context* is a triple $\langle G, M, R\rangle$ [11], where $G$ and $M$ as above and $R \subseteq G \times M$ is a binary relation. Choosing

$$\forall m \in M\,(V_m = \{0,1\}) \text{ and } f(g,m) = \begin{cases} 1, \text{ if } (g,m) \in R; \\ 0, \text{ otherwise} \end{cases},$$

we may transform information systems into formal contexts. For instance, Table 3 shows a formal context representation of the same example shown in Table 1.

**Table 3.** Formal context of a shortened student grade history

| | Homework1 | | | | | Homework2 | | | | | Homework3 | | | | | Final |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Student | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | exam |
| $s_1$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_2$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| $s_3$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_4$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_5$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_6$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_7$ | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $s_8$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| $s_9$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| $s_{10}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $s_{11}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| $s_{12}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_{13}$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_{14}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| $s_{15}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| $s_{16}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4 |
| $s_{17}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| $s_{18}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| $s_{19}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| $s_{20}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |

**Table 4.** Incomplete formal context of a shortened student grade history

| | Homework1 | | | | | Homework2 | | | | | Homework3 | | | | | Final |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Student | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | exam |
| $s_1$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_2$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| $s_3$ | | | | | | 1 | 0 | 0 | 0 | 0 | | | | | | 1 |
| $s_4$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_5$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_6$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| $s_7$ | | | | | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $s_8$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| $s_9$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| $s_{10}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| $s_{11}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| $s_{12}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_{13}$ | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $s_{14}$ | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 3 |
| $s_{15}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| $s_{16}$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 4 |
| $s_{17}$ | | | | | | | | | | | | | | | | 4 |
| $s_{18}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 3 |
| $s_{19}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| $s_{20}$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |

Given the formal context $\langle G, M, R\rangle$ we define

$$A^{\triangleright} = \{m \in M \mid \forall g \in A\,((g,m) \in R)\}, \text{ for } A \subseteq G,$$
$$B^{\triangleleft} = \{g \in G \mid \forall m \in B((g,m) \in R)\}, \text{ for } B \subseteq M,$$

called the *polars* of $A$, $B$, respectively [26].

Informally, $A^{\triangleright}$ is the set of attributes common to all the objects in $A$, $B^{\triangleleft}$ is the set of all objects which possess all of the attributes in $B$.

Given $A \subseteq G$ and $B \subseteq M$ we have $A \times B \subseteq R \Leftrightarrow A \subseteq B^{\triangleleft} \Leftrightarrow A^{\triangleright} \supseteq B$. The pair $(A, B)$ is called a *formal concept* if

$$A = B^{\triangleleft} \text{ and } A^{\triangleright} = B.$$

Formal concepts are usually ordered by inclusion on the first co-ordinate and/or reverse inclusion on the second:

$$(A_1, B_1) \preceq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \text{ and } B_1 \supseteq B_2 \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_1 \supseteq B_2.$$

Formal concepts with this ordering form a concept hierarchy for the context $\langle G, M, R\rangle$ and denoted by $\mathcal{B}(G, M, R)$. The fundamental theorem of formal concept analysis states that $\mathcal{B}(G, M, R)$ with the ordering $\preceq$ is a complete lattice called the *concept lattice* [11].

## 4 A Tool-Based Set Theoretic Approximation Framework

Let $U$ be any nonempty set. Let $A^+, A^- \subseteq U$ be two nonempty subsets of $U$ in such a way that $A^+ \cap A^- = \emptyset$. $A^+$ and $A^-$ are called the *positive* and *negative reference set*, respectively. The adjectives *positive* and *negative* claim nothing else but that the sets $A^+$ and $A^-$ are well separated.

In general, the constraint $A^+ \cap A^- = \emptyset$ is the only requirement for $A^+$ and $A^-$. Of course, additional relations between them may be supposed.

Furthermore, let $\mathfrak{T}^+$ and $\mathfrak{T}^- \subseteq 2^U$ be two nonempty *finite* families of subsets of $U$. The members of $\mathfrak{T}^+$ are called *positive* or $\mathfrak{T}^+$*-tools*, whereas the members of $\mathfrak{T}^-$ are called *negative* or $\mathfrak{T}^-$*-tools*.

Requirements for positive and negatives tools are the following:

***(T1)*** For each subset $T^+ \in \mathfrak{T}^+$ (resp., $T^- \in \mathfrak{T}^-$) it is *easy* to decide whether an element of $U$ belongs to $T^+$ (resp., $T^-$) or not.
***(T2)*** Sets in $\mathfrak{T}^+$ are not necessarily pairwise disjoint, neither are those in $\mathfrak{T}^-$.
***(T3)*** $\mathfrak{T}^+ \cap \mathfrak{T}^- = \emptyset$.
***(T4)*** Neither $\bigcup \mathfrak{T}^+$ nor $\bigcup \mathfrak{T}^-$ covers $U$ necessarily.
***(T5)*** It is assumed that

$$\forall T_1^+, T_2^+ \in \mathfrak{T}^+\,(T_1^+ \cap T_2^+ \in \mathfrak{T}^+), \text{ and } \forall T_1^-, T_2^- \in \mathfrak{T}^-\,(T_1^- \cap T_2^- \in \mathfrak{T}^-),$$

i.e. the $\mathfrak{T}^+$ and $\mathfrak{T}^-$ are closed under intersection.

Positive (resp., negative) tools provide an opportunity to locate or approximate the positive (resp., negative) reference set. Positive and negative tools together also yield useful information about the reference sets. To do this, we can use the following three partial approximation spaces relaying on $\mathfrak{T}^+$ and $\mathfrak{T}^-$:

$$\langle U, \mathfrak{D}_{\mathfrak{T}^+}, \mathfrak{C}^{\flat}_{\mathfrak{T}^+}, \mathfrak{C}^{\sharp}_{\mathfrak{T}^+} \rangle, \langle U, \mathfrak{D}_{\mathfrak{T}^-}, \mathfrak{C}^{\flat}_{\mathfrak{T}^-}, \mathfrak{C}^{\sharp}_{\mathfrak{T}^-} \rangle, \langle U, \mathfrak{D}_{\mathfrak{T}^+ \cup \mathfrak{T}^-}, \mathfrak{C}^{\flat}_{\mathfrak{T}^+ \cup \mathfrak{T}^-}, \mathfrak{C}^{\sharp}_{\mathfrak{T}^+ \cup \mathfrak{T}^-} \rangle.$$

Within these spaces, any clump of observed objects can be approximated with the help of the lower and upper $\mathfrak{T}^+(\mathfrak{T}^-, \mathfrak{T}^+ \cup \mathfrak{T}^-)$-approximations.

## 5 An Illustrative Example

To illustrate our framework let us see a simple example. We want to approximately estimate the achievement of students and their results in the final examination in higher education [20, 21]. We have at our disposal an information table (Table 5) containing the student grade history (5 = excellent, 4 = good, 3 = fair, 2 = pass, 1 = fail).

**Table 5.** Information table with student grade history

| Student | Hw1 | Hw2 | Hw3 | Final exam |
|---|---|---|---|---|
| $S_1$ | 1 | 1 | 1 | **1** |
| $S_2$ | 1 | 1 | 2 | **2** |
| $S_3$ | 1 | 1 | 1 | **1** |
| $S_4$ | 1 | 2 | 1 | **1** |
| $S_5$ | 1 | 1 | 1 | **1** |
| $S_6$ | 1 | 1 | 2 | **1** |
| $S_7$ | 4 | 1 | 3 | **1** |
| $S_8$ | 2 | 4 | 1 | **2** |
| $S_9$ | 1 | 3 | 1 | **2** |
| $S_{10}$ | 1 | 1 | 3 | **1** |
| $S_{11}$ | 2 | 1 | 1 | **2** |
| $S_{12}$ | 1 | 1 | 1 | **1** |
| $S_{13}$ | 1 | 2 | 1 | **1** |
| $S_{14}$ | 1 | 1 | 2 | **3** |
| $S_{15}$ | 4 | 3 | 3 | **4** |
| $S_{16}$ | 2 | 1 | 1 | **4** |
| $S_{17}$ | 2 | 2 | 2 | **4** |
| $S_{18}$ | 4 | 4 | 3 | **3** |
| $S_{19}$ | 4 | 3 | 3 | **2** |
| $S_{20}$ | 4 | 4 | 3 | **4** |

Positive tools:

$T^+_{Hw1=4} = \{S_7, S_{15}, S_{18}, S_{19}, S_{20}\}$

$T^+_{Hw2=4} = \{S_8, S_{18}, S_{20}\}$

$T^+_{Hw1=4 \wedge Hw2=4} = \{S_{18}, S_{20}\}$

Negative tools:

$T^-_{Hw1=1} = \{S_1, S_2, S_3, S_4, S_5, S_6, S_9, S_{10}, S_{12}, S_{13}, S_{14}\}$

$T^-_{Hw2=1} = \{S_1, S_2, S_3, S_5, S_6, S_7, S_{10}, S_{11}, S_{12}, S_{14}, S_{16}\}$

$T^-_{Hw3=1} = \{S_1, S_3, S_4, S_5, S_8, S_9, S_{11}, S_{12}, S_{13}, S_{16}\}$

$T^-_{Hw1=1 \wedge Hw2=1} = \{S_1, S_2, S_3, S_5, S_6, S_{10}, S_{12}, S_{14}\}$

$T^-_{Hw1=1 \wedge Hw3=1} = \{S_1, S_3, S_4, S_5, S_9, S_{12}, S_{13}\}$

$T^-_{Hw1=2 \wedge Hw3=1} = \{S_1, S_3, S_5, S_{11}, S_{12}, S_{16}\}$

$T^-_{Hw1=1 \wedge Hw2=1 \wedge Hw3=1} = \{S_1, S_3, S_5, S_{12}\}$

Of course, there is no way to accurately measure the achievement of students and their success or failure on the final exam. Moreover, students cannot exactly appreciate 'what they know' or 'what they do not know'. However, with the

apparatus of partial approximation spaces, we can analyze student grade history contained in Table 5 in order to understand how the results in assignments approximately relate to success or failure on the final exam.

For the sake of simplicity, students' success and failure on homework assignments or the final exam are measured by grade 4 and grade 1, respectively. Based on these prerequisites, the positive tools (Fig. 4) and negative tools (Fig. 5) are the following (see also Table 5):

$$\mathfrak{T}^+ = \{T^+_{Hw1=4},\, T^+_{Hw2=4},\, T^+_{Hw1=4\wedge Hw2=4}\},$$
$$\mathfrak{T}^- = \{T^-_{Hw1=1},\, T^-_{Hw2=1}, T^-_{Hw3=1},\, T^-_{Hw1=1\wedge Hw2=1}, T^-_{Hw1=1\wedge Hw3=1},$$
$$T^-_{Hw1=2\wedge Hw3=1},\, T^-_{Hw1=1\wedge Hw2=1\wedge Hw3=1}\}$$

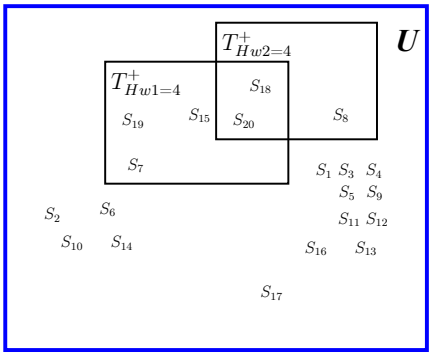



**Fig. 4.** Positive tools (homework = 4). $\langle U, \mathfrak{D}_{\mathfrak{T}^+}, \mathfrak{C}^{\flat}_{\mathfrak{T}^+}, \mathfrak{C}^{\sharp}_{\mathfrak{T}^+}\rangle$ partial approximation space

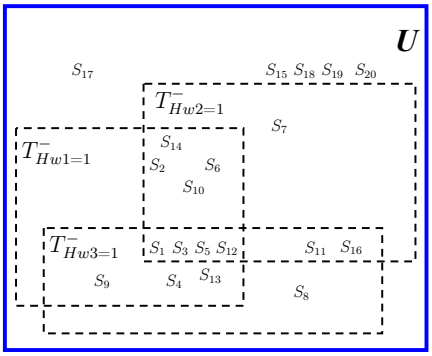



**Fig. 5.** Negative tools (homework = 1). $\langle U, \mathfrak{D}_{\mathfrak{T}^-}, \mathfrak{C}^{\flat}_{\mathfrak{T}^-}, \mathfrak{C}^{\sharp}_{\mathfrak{T}^-}\rangle$ partial approximation space

Students who have successful final exams can be evaluated with both positive and negative tools (Fig. 6, Fig. 7):

- $\mathfrak{C}^{\flat}_{\mathfrak{T}^+}(X_{Final\_exam=4}) = \emptyset$
  Informally: there is no combination of successful homework in which case the final exam *surely* succeeds.
- $\mathfrak{C}^{\sharp}_{\mathfrak{T}^+}(X_{Final\_exam=4}) = T^+_{Hw1=4} \cup T^+_{Hw2=4} \cup T^+_{Hw1=4\wedge Hw2=4}$
  Informally: if one of the Homework 1, 2 or both of the two succeed, the final exam *possibly* succeeds.
- $\mathfrak{C}^{\flat}_{\mathfrak{T}^-}(X_{Final\_exam=4}) = \emptyset$
  Informally: there is no combination of failed homework in which case the final exam *surely* succeeds.
- $\mathfrak{C}^{\sharp}_{\mathfrak{T}^-}(X_{Final\_exam=4}) = T^-_{Hw3=1}$
  Informally: if the Homework 3 fails, the final exam *may* succeed.

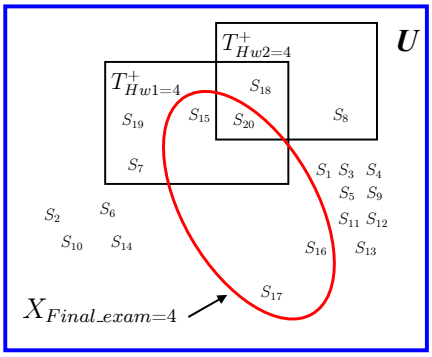



**Fig. 6.** Evaluation of successful final exams with positive tools

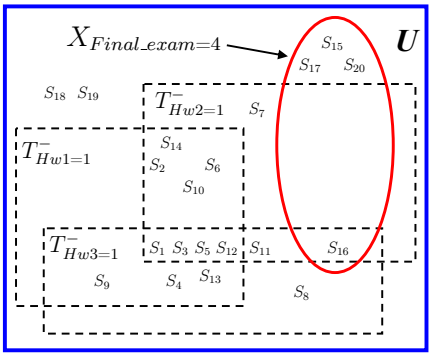



**Fig. 7.** Evaluation of successful final exams with negative tools

Students who have failed their final exams can also be evaluated with both positive and negative tools (Fig. 8, Fig. 9):

- $\mathfrak{C}^{\flat}_{\mathfrak{T}^+}(X_{Final\_exam=1}) = \emptyset$
  Informally: there is no combination of successful homework in which case the final exam *surely* fails.
- $\mathfrak{C}^{\sharp}_{\mathfrak{T}^+}(X_{Final\_exam=1}) = T^+_{Hw1=4}$
  Informally: if the only Homework 1 succeeds, the final exam *possibly* fails (because, e.g., Homework 1 is the simplest part of the course).
- $\mathfrak{C}^{\flat}_{\mathfrak{T}^-}(X_{Final\_exam=1}) = T^-_{Hw1=1 \wedge Hw2=1 \wedge Hw3=1}$
  Informally: If all homework fail, the final exam *surely* fails.
- $\mathfrak{C}^{\sharp}_{\mathfrak{T}^-}(X_{Final\_exam=1}) = \bigcup \mathfrak{T}^-$
  Informally: if at least one homework fails, the final exam *possibly* fails.

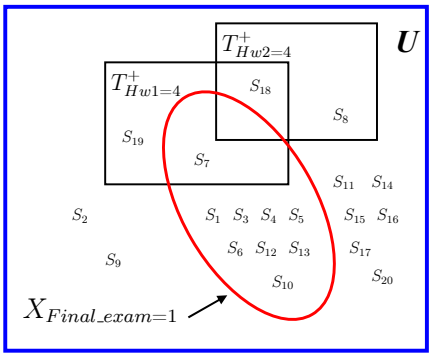



**Fig. 8.** Evaluation of failed final exams with positive tools

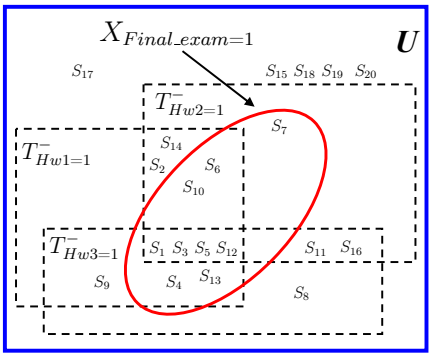
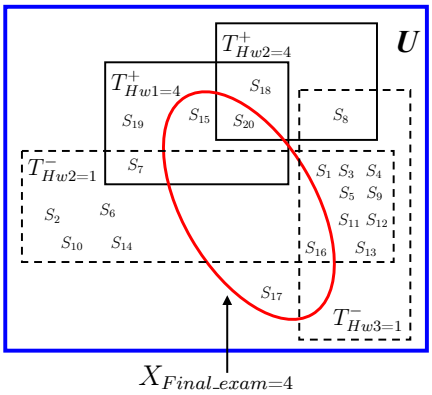



**Fig. 9.** Evaluation of failed final exams with negative tools

Evaluations can also be carried out over positive and negative tools together:

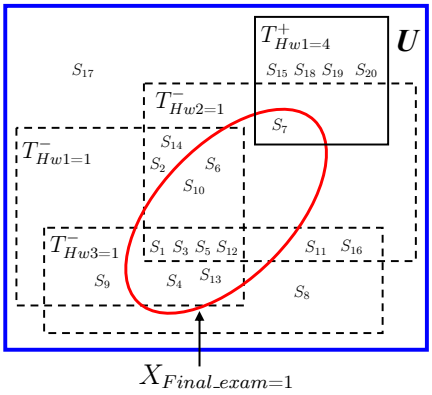
- $\mathfrak{C}^{\flat}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=4}) = \emptyset$ (see Fig. 10) informally means that there is no combination of successful or failed homework in which case the final exam *surely* succeeds.
- $\mathfrak{C}^{\sharp}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=4})$ (see Fig. 10) informally means that if one of the Homework 1, 2 or both of the two succeed, in addition, even if one of the Homework 1, 3 or both of the two fail, then the final exam *possibly* succeed.
- $\mathfrak{C}^{\flat}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=1})$ (see Fig. 11) informally means that if at least one homework fails, the final exam *surely* fails.
- $\mathfrak{C}^{\sharp}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=1})$ (see Fig. 11) informally means that if at least one homework fails, the final exam *possibly* fails even if the Homework 1 succeeds.




$$\mathfrak{C}^{\flat}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=4}) = \emptyset$$

$$\begin{aligned}&\mathfrak{C}^{\sharp}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=4})\\ &= T^+_{Hw1=4} \cup T^+_{Hw2=4}\\ &\quad \cup T^+_{Hw1=4\wedge Hw2=4}\\ &\quad \cup T^-_{Hw2=1} \cup T^-_{Hw3=1}\\ &\quad \cup T^-_{Hw2=1\wedge Hw3=1}\end{aligned}$$

**Fig. 10.** Evaluation of successful final exams with positive and negative tools




$$\begin{aligned}&\mathfrak{C}^{\flat}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=1})\\ &= T^-_{Hw1=1\wedge Hw2=1\wedge Hw3=1}\end{aligned}$$

$$\begin{aligned}&\mathfrak{C}^{\sharp}_{\mathfrak{T}^+\cup\mathfrak{T}^-}(X_{Final\_exam=1})\\ &= T^-_{Hw1=1} \cup T^-_{Hw2=1} \cup T^-_{Hw3=1}\\ &\quad \cup T^-_{Hw1=1\wedge Hw2=1}\\ &\quad \cup T^-_{Hw1=1\wedge Hw3=1}\\ &\quad \cup T^-_{Hw2=1\wedge Hw3=1}\\ &\quad \cup T^-_{Hw1=1\wedge Hw2=1\wedge Hw3=1}\\ &\quad \cup T^+_{Hw1=4}\end{aligned}$$

**Fig. 11.** Evaluation of failed final exams with positive and negative tools

## 6 Conclusion

We have presented in this paper a tool-based set theoretic framework for concept approximation relying on partial approximation spaces. Positive features and their substantially negative features of observed objects can *simultaneously* be approximated with the help of this framework.

We have drawn up a simplified example to demonstrate our approach. We have analyzed a student grade history and we have been able to evaluate the students' achievement, exploring 'what they know' and/or 'what they do not know', and understand how the results in homework assignments approximately relate to success or failure on the final exam. Of course, a more subtle definition of the notions of 'success' and 'failure' could result in a more subtle evaluation. A refined evaluation process can form a basis for *quality insurance in higher education* properly building in the hierarchy of quality management.

## Acknowledgement

The author would like to express his gratitude to the anonymous referees for reading the paper carefully and their insightful comments and suggestions.

## References

1. Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2009, Mragowo, Poland, 12-14 October 2009. Polskie Towarzystwo Informatyczne - IEEE Computer Society Press (2009)
2. Csajbók, Z.: Partial approximative set theory: A generalization of the rough set theory. In: Martin, T., Muda, A.K., Abraham, A., Prade, H., Laurent, A., Laurent, D., Sans, V. (eds.) Proceedings of SoCPaR 2010, December 7-10, 2010., Cergy Pontoise / Paris, France. pp. 51–56. IEEE (2010)
3. Csajbók, Z., Mihálydeák, T.: A general tool-based approximation framework based on partial approximation of sets. In: Kuznetsov, S.O., et al. (eds.) Proceedings of RSFDGrC 2011, June 25-27, 2011, Moscow, Russia. LNAI, vol. 6743, pp. 52–59. Springer-Verlag Berlin Heidelberg (2011)
4. Csajbók, Z., Mihálydeák, T.: On the general set theoretical framework of set approximation. pp. 12–15 (2011), Proceedings of RST 2011, 14-16 September 2011, Milan, Italy (2011)
5. Csajbók, Z., Mihálydeák, T.: Partial approximative set theory: A generalization of the rough set theory. International Journal of Computer Information Systems and Industrial Management Applications 4, 437–444 (2012)
6. Csajbók, Z.: Approximation of sets based on partial covering. Theoretical Computer Science 412(42), 5820–5833 (2011)
7. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge University Press, Cambridge (2002)
8. Denecke, K., Erné, M., Wismath, S. (eds.): Galois Connections and Applications. Kluwer Academic Publishers (2004)
9. Düntsch, I., Gediga, G.: Statistical evaluation of rough set dependency analysis. International Journal of Human-Computer Studies 46(5), 589–604 (1997)

10. Ganter, B., Kuznetsov, S.O.: Formalizing hypotheses with concepts. In: Mineau, G., Ganter, B. (eds.) Proc. 8th Int. Conf. on Conceptual Structures, ICCC'2000. Lecture Notes in Artificial Intelligence, vol. 1867, pp. 342–356. Springer, Berlin (2000)
11. Ganter, B., Wille, R.: Formal concept analysis: Mathematical foundations. Springer, Berlin-Heidelberg (1999)
12. Gartmeier, M., Bauer, J., Gruber, H., Heid, H.: Negative knowledge: Understanding professional learning and expertise. Vocations and Learning: Studies in Vocational and Professional 1(2), 87–103 (2008)
13. Järvinen, J.: Lattice theory for rough sets. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI, LNCS, vol. 4374, pp. 400–498. Springer-Verlag (2007)
14. Keefe, R.: Theories of Vagueness. Cambridge Studies in Philosophy, Cambridge University Press, Cambridge, UK (2000)
15. Kuznetsov, S.O.: Machine learning and formal concept analysis. In: Eklund, P.W. (ed.) Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings. Lecture Notes in Computer Science, vol. 2961, pp. 287–312. Springer-Verlag, Berlin Heidelberg (2004)
16. Kuznetsov, S.O.: Galois connections in data analysis: Contributions from the soviet era and modern russian research. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis, Foundations and Applications. LNAI, vol. 3626, pp. 196–225. Springer (2005)
17. Marek, V.W., Truszczyński, M.: Approximation schemes in logic and artificial intelligence. In: Peters, J.F., Skowron, A., Rybinski, H. (eds.) Transactions on Rough Sets IX. LNCS, vol. 5390, pp. 135–144. Springer-Verlag (2008)
18. Miné, A.: Weakly Relational Numerical Abstract Domains. Ph.D. thesis, École Polytechnique, Palaiseau, France (December 2004)
19. Minsky, M.: Negative expertise. International Journal of Expert Systems 7(1), 13–19 (1994)
20. Narli, S., Ozelik, Z.A.: Data mining in topology education: Rough set data analysis. International Journal of the Physical Sciences 5(9), 1428–1437 (2010)
21. Narli, S., Yorek, N., Sahin, M., Usak, M.: Can we make definite categorization of student attitudes? a rough set approach to investigate students implicit attitudinal typologies toward living things. Journal of Science Education and Technology 19, 456–469 (2010)
22. Pagliani, P., Chakraborty, M.: A Geometry of Approximation: Rough Set Theory Logic, Algebra and Topology of Conceptual Patterns (Trends in Logic). Springer Publishing Company, Incorporated (2008)
23. Parviainen, J., Eriksson, M.: Negative knowledge, expertise and organisations. International Journal of Management Concepts and Philosophy 2(2), 140–153 (2006)
24. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
25. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
26. Priestley, H.A.: Ordered sets and complete lattices: A primer for computer science. In: Backhouse, R.C., Crole, R.L., Gibbons, J. (eds.) Algebraic and Coalgebraic Methods in the Mathematics of Program Construction. LNCS, vol. 2297, pp. 21–78. Springer (2000)
27. Revett, K., Gorunescu, F., Salem, A.B.M.: Feature selection in parkinson's disease: A rough sets approach. In: IMCSIT [1], pp. 425–428

28. Salem, A.B.M., Revett, K., El-Dahshan, E.S.A.: Machine learning in electrocardiogram diagnosis. In: IMCSIT [1], pp. 429–433
29. Skowron, A.: Rough sets in perception-based computing. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) Pattern Recognition and Machine Intelligence, First International Conference, PReMI 2005, Kolkata, India, December 20-22, 2005, Proceedings. LNCS, vol. 3776, pp. 21–29. Springer (2005)
30. Skowron, A., Stepaniuk, J., Swiniarski, R.: Approximation spaces in rough-granular computing. Fundamenta Informaticae 100(1–4), 141–157 (2010)
31. Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)

# Decision Aiding Software Using FCA

Florent Domenach and Ali Tayari

Computer Science Department, University of Nicosia, 46 Makedonitissas Av.,
P.O.Box 24005, 1700 Nicosia, Cyprus, domenach.f@unic.ac.cy

**Abstract.** The consensus problem arises from social choice theory and systematic biology where we are looking for the common information shared by a series of trees. In this paper we present a decision aiding software to help systematic biologist to choose the consensus function the most appropriate for their need. This software is based on a previous study between consensus functions and axiomatic properties, and their underlined concept lattice.

## 1 Introduction

The consensus problem, which [11] deemed a "problem for the future", consists of summarizing a series of structures, usually trees, into one representative structure. Axiomatic studies of consensus functions is often [26] described as an "ideal situation [in which] the researcher formulates a list of desirable axioms that a consensus function should satisfy, and search for the best method that satisfies these axioms" [33]. We present here a software following this approach almost to the letter. Unfortunatly, it is still missing critical GUI features and is not available yet.

The motivation for the software is originating from the separation existing between theorizers and practitioners of consensus theory, what [7] denotes as abstract consensus theory and concrete consensus theory. On one hand, mathematicians are developing sophisticated mathematical tools. The modern development of the consensus problem originates from Arrow's work [3] (followed by [25]) who considered the problem of aggregating votes and showed that any voting system is either inconsistent, arbitrary or unstable. Since then, a lot of functions, together with a set of equivalent axioms, were developed (see [15, 22] for a comprehensive survey).

On the other hand, practitioners like systematic biologists are rarely using more than a handful of consensus functions. If you consider the most popular software available like PAUP*[1] [37] (majority), PHYLIP[2] [20] (majority, strict), or COMPONENT 2.0[3] [30] (strict, majority-rule, loose, Nelson and Adams consensus trees), only a handful are available for use. It was pointed out [38] that this gap between the two communities was detrimental to both.

[1] http://paup.csit.fsu.edu/
[2] http://evolution.genetics.washington.edu/phylip.html
[3] http://taxonomy.zoology.gla.ac.uk/rod/cpw.html

The goal of this paper is to present an approach – based on FCA – to the consensus problem that would fill the gap between both communities. We created a software that asks the user to think of desirable properties that a consensus method should possess, and then we advise on which consensus function satisfying these properties he/she should use. Each step is described in detail in the paper.

This paper is organized in four sections, the first one being this introduction. In Section 2, we give a precise definition of the consensus problem, as well as the definitions of the consensus functions (Section 2.1) and of the axiomatic properties (Section 2.2) that we implemented. We present in Section 3 the structure of our program, and explain for every step why and how we are doing it. Finally a brief conclusion is given in Section 4.

## 2 The Consensus Problem

Consider a finite set $S$, $|S| = n$. In phylogeny, the elements of $S$ are called operational taxonomic units, or *taxa*. A hierarchy $H$ on $S$, also called $n$-tree, is a family of subsets of $S$ (called the *classes* or clusters of $H$) such that $S \in H$, $\emptyset \notin H$, $\{s\} \in H$ for all $s \in S$, and $A \cap B \in \{\emptyset, A, B\}$ for all $A, B \in H$. We will indifferently use the terms trees or hierarchies in the paper. We denote the set of all hierarchies on $S$ by $\mathcal{H}$. Fig. 1 shows the graphical representation of different trees; usually the internal nodes are simply denoted by the leaves underneath.

Consensus trees are summarizations of the information shared by two or more classification trees of the same set of taxa. Given a profile $H^*$ of trees on $S$, i.e. a series of trees, we want to know what they have in common - we want to aggregate $H^*$ in a unique tree $H$. We consider in this paper the case where all the trees of the profile are defined on the same set of taxa, as the generalization to super-trees [34] (where the trees can have different sets of taxa) can create computational problems.

### 2.1 Consensus Functions

Let $H^* = (H_1, H_2, ..., H_k)$ be a profile of hierarchies on $S$, and $K$ will denote the set of indices of the hierarchies of $H^*$, $K = \{1, ..., k\}$. Formally, a *consensus function* on $\mathcal{H}$ is a map $c : \mathcal{H}^k \to \mathcal{H}$ with $k \geq 2$ and $\mathcal{H}^k$ the $k$ cartesian product, which associate to any profile $H^*$ a unique hierarchy consensus, $c(H^*)$. We do not aim to have an exhaustive list of consensus functions, a classification based on refinement is available in [13]. Consensus functions can be divided in three main categories:

*Quota-based consensus functions.* Consider a grouping and the associated index defined as:

$$N_{H^*}(A) = \{i \in K : A \in H_i\} \text{ and } n_{H^*}(A) = |N_{H^*}(A)|$$

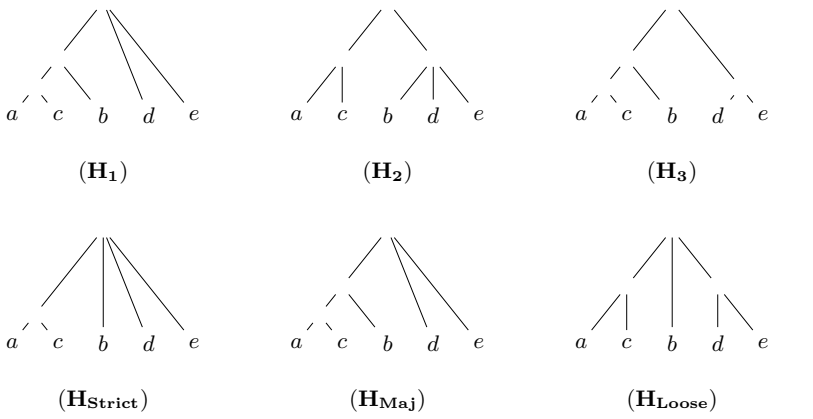



**Fig. 1.** Different trees defined on the set of taxa $S = \{a, b, c, d, e\}$. For the profile $H^* = (H_1, H_2, H_3)$, the strict consensus tree is given by $(H_{Strict})$, the majority by $(H_{Maj})$ and the loose by $(H_{Loose})$

We associate the consensus function $c_{(p)} : \mathcal{H}^k \to \mathcal{H}$ to the index $n_{H^*}$ for any $p \in K$. A subset $A$ is called *p-frequent* if $n_{H^*}(A) \geq p$, and the *p-frequent consensus* of $H^*$, denoted as $c_{(p)}(H^*)$, is the family of all $p$-frequent subsets. Quota-based consensus functions are particular cases of federation consensus functions [23]. Recall that a federation (simple game) is a family $\mathcal{F}$ of subsets of $K$ such that $A \in \mathcal{F}, B \supseteq A$ imply $B \in \mathcal{F}$. A *federation consensus function* $c_{\mathcal{F}}$ is then defined as $c_{\mathcal{F}}(H^*) = \vee_{S \in \mathcal{F}}(\cap_{i \in S} H_i)$. If we take the simple case where, for some $j \in K$, $\mathcal{F} = \{S \subseteq K : j \in S\}$, we have $c_{\mathcal{F}}(H^*) = H_j$, a single hierarchy dictating the result of the consensus, the so called *projection* consensus function.

$$\text{Projection: } \exists j \in K : Prj(H^*) = H_j$$

When we extend this to a subset $J$ of $K$, we have the *oligarchic* consensus function using $\mathcal{F} = \{S \subseteq K : J \subseteq S\}$, and $c_{\mathcal{F}}(H^*) = \bigcap_{j \in J} H_j$.

$$\text{Oligarchy: } \exists J \subseteq K : Ol(H^*) = \bigcap_{j \in J} H_j$$

In the family of quota-based consensus functions, one can notice $c_{(k)}(H^*) = \bigcap_{i \in K} H_i$ the set of classes present in all trees of the profile, i.e. the *strict* consensus function [36]. In Fig. 1, $(H_{Strict})$ is the strict consensus of the profile $(H_1, H_2, H_3)$.

$$\text{Strict: } Str(H^*) = \bigcap_{i \in K} H_i$$

If we take $p = \lceil \frac{k+1}{2} \rceil$, the smallest natural number greater than $\frac{k}{2}$, we have the *majority* consensus function [24] which considers clusters appearing in at least

half of the trees. An example of the majority consensus function is given in Fig. 1, where $(H_{Maj}) = Maj(H_1, H_2, H_3)$.

$$\text{Majority: } Maj(H^*) = \{X \subseteq S : n_{H^*}(X) > \frac{k}{2}\}$$

Unfortunately, if $p$ is less than $\lceil \frac{k+1}{2} \rceil$, it cannot be guaranteed that the resulting family will be a tree. In order to keep the structure of a tree, different strategies can be used.

*Frequency-based consensus functions* A first approach considers the idea of compatibility, i.e. two sets $A$ and $B$ are *compatible* if $A \cap B \in \{\emptyset, A, B\}$, denoted as $A \| B$, and a set $A$ is compatible with a hierarchy $H$ if it is compatible with every cluster of $H$ (or, equivalently, if $A \cup H \in \mathcal{H}$). We then can define a consensus function called *loose* consensus [6] (originally called combinable component [12], also called semi-strict) which considers subsets as long as they are compatible with every tree of the profile. Fig. 1 shows $(H_{Loose})$, the loose consensus tree obtained from $(H_1, H_2, H_3)$.

$$\text{Loose: } L(H^*) = \bigcup \{X \subseteq S : \exists j \in K, X \in H_j \text{ and } \forall i \in K, X \cup H_i \in \mathcal{H}\}$$

The loose consensus function was extended by [18] to two different consensus functions. The first one is combining the classes obtained by the majority consensus function with those of the loose consensus function:

$$\text{Loose and Majority Function Property: } LM(H^*) = Maj(H^*) \cup L(H^*)$$

The second extension is to add classes that are more often compatible than not. Define $N_{H^*}(\overline{X}) = \{i \in K : X \cup H_i \notin \mathcal{H}\}$ as the set of trees not compatible with a subset $X$, then the *majority (+)* consensus function will take subsets that are more often compatible than incompatible. It obviously contains all the classes obtained by the majority function and by the loose function.

$$\text{Majority-rule } (+) : Maj^+(H^*) = \{X \subseteq S : |N_{H^*}(X)| > |N_{H^*}(\overline{X})|\}$$

Consider the weight function $w(X) = n_{H^*}(X) - 1$ on classes. The *Nelson-Page* consensus tree is the tree constructed from the clique $G$ containing the components most frequently replicated in the profile. If two or more cliques have the same, maximal number of replications of components, then the consensus tree is constructed from those components common to all those cliques. In the literature, the Nelson-Page tree [27, 29] has often been confused with the strict consensus tree.

The *frequency difference* consensus function consider the subsets of $S$ that are more frequent than any other subsets non-compatible.

$$\text{Freq. Diff.: } FD(H^*) = \{X : n_{H^*}(X) > max_Y \text{ not compatible with } _X\{n_{H^*}(Y)\}\}$$

Previous consensus functions may miss some structural features of the trees, particularly if the data is noisy. For example, a desirable feature would be that

if two taxa are closer than a third one, we want these two taxa to be separated from the third one in the consensus hierarchy - which is what Adams' function [1] achieves. Historically the first one, an *Adams* consensus tree contains the nestings common to all trees in a profile. $X$ nests in $Y$ in $H$, denoted as $X <_H Y$ if and only if $X \subset Y$ and there is $Z \in H$ such that $X \subseteq Z$ and $Y \not\subseteq Z$. $\pi(H)$ is the maximal cluster partition for $H$ with blocks equal to the maximal clusters of $H$. Adams' consensus function is best described algorithmically (from [13]):

Procedure AdamsTree($H_1, ..., H_k$)
  Construct $\pi(H)$, the product of $\pi(H_1), ..., \pi(H_k)$.
  For each block $B$ of $\pi(H)$ do
    AdamsTree($H_1|_B, ..., H_k|_B$)

*Distance-based consensus functions* Another consensus family is based on distance, either as a height function, or as distance between trees. *Durchschnitt* [28] consensus function takes the intersection of all classes at the same height. The canonical height $\eta_0(X)$ of a class $X \subseteq S$ is defined as $\eta_0(S) = 0$ and $\eta_0(X) = h$ if and only if there is a maximal sequence $S \supset X_1 \supset ... \supset X_{h-1} \supset X_h = X$. Define $\omega = min_{i \in K} max_{X \in H_i} \eta_0(X)$ as the height of the smallest tree of the profile.

$$\text{Durchschnitt: } Dur(H^*) = \bigcup_{j=1}^{\omega} \{ \bigcap_{i \in K} X_i : X_i \in H_i \text{ and } \eta_0(X_i) = j \}$$

The median and asymmetric median consensus functions both use a distance between trees, i.e. a distance on $\mathcal{H}$. The *median* consensus is the tree minimizing the distance of the symmetric difference from it to every tree of the profile. The median consensus was extensively studied, particularly in the case of semi-lattices [35] (as trees can be seen as semi-lattices).

$$\text{Median: } Med(H^*) = min_{H \in \mathcal{H}} \sum_{i=1}^{k} |H \triangle H_i|$$

The *asymmetric median* consensus [32] on the other hand is the tree minimizing the distance between each tree and the consensus tree, i.e. minimizing the number of classes in $H_i$ that are not present in $c(H^*)$.

$$\text{Asymmetric Median: } AMed(H^*) = min_{H \in \mathcal{H}} \sum_{i=1}^{k} |H_i - H|$$

### 2.2 Axiomatic Properties of Consensus Functions

Historically, consensus functions were studied through a series of (desirable) axioms proved to be equivalent to the function. Arrow's pioneer work proved the impossibility of a non-dictatorial consensus function satisfying fundamental axioms (transitivity, Pareto and independence of irrelevant alternatives) on linear

orders. We implemented a series of axioms that a user may find desirable or undesirable.

A consensus function is Pareto relatively to a specific kind of relationships (classes, triplets, nestings) when the consensus tree will contain the relationship present in all the trees, i.e. will contain the intersection of the trees of the profile with respect to the relationship. For example, when we are interested in the common classes, we have the *Pareto optimal* [31] axiom:

$$\text{Pareto Optimality: } (\forall X \subseteq S)(X \in \bigcap_{i=1}^{k} H_i \Rightarrow X \in c(H^*))$$

Trees can also be defined [14] through triplets $ab|c$, $a, b, c \in S$, denoting the grouping of $a$ and $b$ relative to $c$. We say that $ab|c \in H$ if there exists a class $X \in H$ such that $a, b \in X$ but $c \notin X$. The Pareto property on triplets is that a common separation of two taxa from a third taxon among every input tree must be respected and applied in the consensus tree.

$$\text{Ternary Pareto Optimality: } (\forall x, y, z \in S)((\forall i \in K)(xy|z \in H_i) \Rightarrow xy|z \in c(H^*))$$

Adams [2] extended that idea to nestings, where if two clusters are separated from each other in every input tree, therefore they must also be separated in the consensus tree:

$$\text{Nesting Preservation: } (\forall \emptyset \neq X, Y \subseteq S)((\forall i \in K)(X <_{H_i} Y) \Rightarrow (X <_{c(H^*)} Y))$$

Conversely, a consensus function is *co-Pareto* for a particular relationship if one can find every relationship of that kind of the consensus tree in one or more tree of the profile. Every cluster from the consensus tree must appear in at least one of the input tree, or in other words it should be a member of the union of all input trees. We will consider here only co-Pareto optimally for classes.

$$\text{co-Pareto Optimality: } (c(H^*) \subseteq \bigcup_{i=1}^{k} H_i)$$

In order to characterize his consensus function, Adams introduced a reciprocal property of nesting preservation, although stronger than just a co-Pareto property. It states that if two subsets are nested in the consensus tree, they must be nested in all the trees of the profile.

$$\text{Strong Presence: } (\forall \emptyset \neq X, Y \subseteq S)(X <_{c(H^*)} Y \Rightarrow (\forall i \in K)(X <_{H_i} Y))$$

It happened that Strong Presence property was too constraining, so instead of considering all possible nested subsets, Adams considered only the nested classes. Any two clusters of the consensus tree that are separated from each other must also be separated in every input tree.

$$\text{Qualified Strong Presence: } (\forall X, Y \in c(H^*))(X <_{c(H^*)} Y \Rightarrow (\forall i \in K)(X <_{H_i} Y))$$

Qualified strong presence was weakened to consider the clusters of the consensus tree to be nested in $S$ in each tree of the profile:

$$\text{Upper Strong Presence: } (\forall X \in c(H^*))(X <_{c(H^*)} S \Rightarrow (\forall i \in K)(X <_{H_i} S))$$

The *dictatorship* property (an input tree dictates over the consensus tree by having all of its clusters included in the consensus tree) is often consider undesirable; however, this can change if there is a particular tree that can be consider an oracle, i.e. for which we want the consensus tree to refine it.

$$\text{Dictatorship: } (\exists j \in K)(\forall X \subseteq S)(X \in H_j \Rightarrow X \in c(H^*))$$

Another desirable property, also called faithful, is the following: for every group of clusters containing only one cluster from each input tree there must be a cluster in the consensus tree such that it includes the intersection of the group of the group of clusters and it is included in the union of the groups of the group of clusters.

$$\text{Betweenness: } (\forall i \in K \text{ with } X_i \in H_i)(\exists Y \in c(H^*))(\bigcap_{i=1}^{k} X_i \subseteq Y \subseteq \bigcup_{i=1}^{k} X_i)$$

## 3 Decision Aiding Software

We used Formal Concept Analysis (FCA) [21] as our formal background. FCA is particularly suitable as it provides a structure on the power set of attributes, here the consensus functions and axioms, and allow calculations of distances on that structure. Since we assume the reader familiar with FCA, we will only briefly recall main terminologies and results used in our program: a *formal context* $(G, M, I)$ is defined as a set $G$ of objects, a set $M$ of attributes, and a binary relation $I \subseteq G \times M$. $(g, m) \in I$ is read as "object $g$ has attribute $m$". To this formal context, one can associate to a set of objects $A \subseteq G$ its intension $A' = \{m \in M : \forall g \in A, (g, m) \in I\}$ of all properties shared by $A$. Dually, we can define $B' = \{g \in G : \forall m \in B, (g, m) \in I\}$, the extension of a set of properties $B \subseteq M$. A pair $(A, B), A \subseteq G, B \subseteq M$, is a *formal concept* if $A' = B$ and $B' = A$. The set of all formal concepts, ordered by inclusion, forms a lattice [5], called *concept lattice*. For more terms and definitions on lattice theory, one can refer to [10, 16].

This D.A. software has three different functional layout (see Fig. 2): a pre-processing is first done on consensus functions and axioms in order to create the context that then will be used, with the associated lattice, in order to advise users on which consensus function to use. The last layer is concerned with the obtainment of the tree itself from some input profile.

### 3.1 Pre-processing

The first layer of the D.A. software concerns the pre-processing of the data that will be used. In order to insure scientific validity of the decision aiding, we implemented the previous consensus functions of Sec. 2.1 and the axiomatic properties

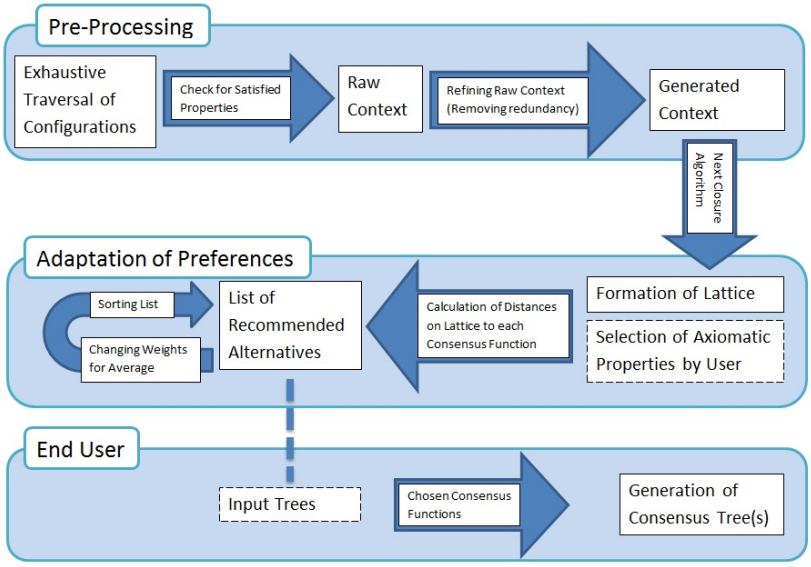



**Fig. 2.** D.A. software functional layout.

of Sec. 2.2 in C++ on a laptop Intel Core i5, 2.3 GHz. Initially, it generates all possible hierarchies based on a given set of $n$ taxa, and traverses through all possible profiles of $k$ hierarchies, together with all possible consensus trees. Then we exhaustively list what we called configurations, each configuration is a pair consisting of a profile and a consensus tree. Every configuration was systematically compared against axiomatic properties and consensus functions in order to create a first (raw) context. Attributes of the context are the consensus functions and the axiomatic properties, while the objects are every possible configuration. We discussed in [17] the implications generated by the context.

During the pre-processing phase, we encountered a series of computational challenges, as the number of $n$-trees grows exponentially [19] and some consensus functions are NP-hard [32]. We were able to exhaustively investigate the configurations only up to $n = 5$, for which we obtained around $9.57 \times 10^{12}$ configurations. Since the running time of the simulation increases exponentially with slight addition to $n$ or $k$, in order to have partial results, controlled randomly selected configurations were chosen in order to have a more accurate - and so a more refine - context.

### 3.2 Underlined Structure

Given the number of objects in our context (over one trillion), we first eliminate duplicates. If several configurations share the same attributes, we simply keep the first one as representative. No information is lost as we are interested in

the structure of the attributes, and the objects (the configurations) sole purpose is to systematically investigate this structure. Our simplified context has 5379 objects for 23 attributes, and Fig. 3 shows the overall concept lattice, having 3718 concepts. In order to derive the lattice, we followed The Next Closure [21] algorithm. This algorithm uses the lectic order on the set of attributes $M$, which is a total order on $\mathcal{P}(M)$. Given two subsets $A$ and $B$ of $M$, $A$ is said to be lectically smaller than $B$ at position $i$, and we denote it by $A \prec_i B$, if and only if $i = min(A \Delta B)$ and $i \in B$. Finally, we say that $A$ is lectically smaller than $B$ if $A = B$ or $A \prec_i B$ for some $i \in K$. We used Next Closure algorithm as it is an efficient and easy to implement.

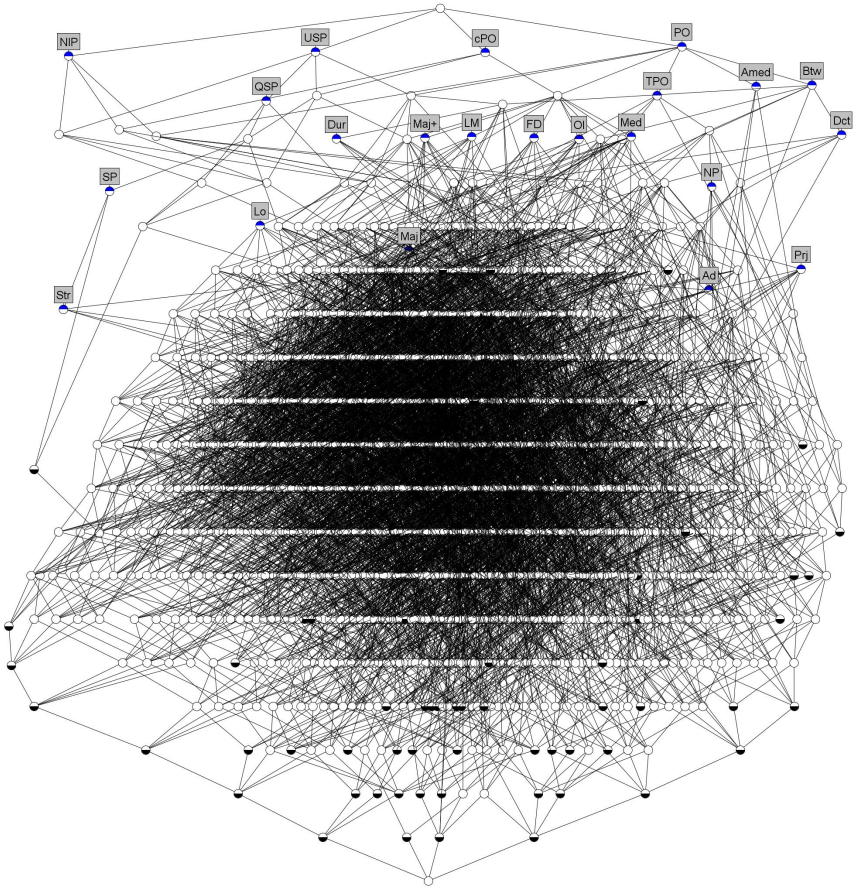



**Fig. 3.** Concept lattice associated with the configurations with minimal labeling of the properties (drawn with ConExp [39]).

After constructing the list of concepts and listing them in ascending order, the program also keeps track of the children and parents of each concept of the lattice. The user can then select a set of axiomatic properties depending on the one he/she finds desirable or undesirable: each axiom can be preferred (positive), disliked (negative), or neutral. Based on that input, the program finds the meet of the selected properties, i.e. the concept $C$ associated to his/her choices. Concept $C$ is the smallest concept containing all the positive user's choices and no negative ones if it exists. If the user's choices are conflicting, i.e. $C$ doesn't exist, positive choices will be given priority over negative ones.

### 3.3 Distances in the lattice

In order to advise which consensus function would be suitable depending on the user's choices, for each consensus function, we first find the smallest concepts $C_i$ containing that consensus function. Then we used different distances between $C$ (the concept representing the user's choice) and each $C_i$ (the concepts associated with consensus function $i$) in order to find the consensus function the closest to the user's choice. The use of different distances lets the user freely choose which distance is more suitable.

We can consider two types of distances in a lattice: distances based on concepts and distances on the covering graph (or Hasse diagram) of the lattice. For the first type, we used the *distance of the symmetric difference* between concepts, $d_1(C, C_i) = |C \Delta C_i|$, i.e the number of properties present in either $C$ or $C_i$ but not in both. For the second type, distances in the covering graph, we considered four distinct distances:

- *Any Path Distance*: weight of the shortest path (topological distance) between the corresponding attribute concepts; the closer the two concepts are in the graph, the greater their likelihood.
- *Any Path Distance Without $\bot_L$ and $\top_L$*: we remove the top and the bottom concepts of the lattice to compute the topological distance because such concepts don't bear any information (even if $1_L$ can have attributes associated, it still doesn't have any meaning). It is particularly important when we consider the co-atoms of the lattice (such as Pareto Optimal or co-Pareto Optimal, see Fig. 3), as the shortest path could go through $1_L$ and short-circuit the "real" distance.
- *Meet Distance*: It is the topological distance between $C$ and $C_i$ passing through their meet, i.e. the distance from $C$ to $C \wedge C_i$ plus the distance from $C_i$ to $C \wedge C_i$.
- *Join Distance*: Dual to the meet distance, it is the topological distance between $C$ and $C_i$ passing through their join.

Since each previous distance has its own advantages and disadvantages, we also implemented a weighted average distance for which the user can freely assign the weights. It is a weighted average of all the above distances based on user's preference. Fig. 4 shows an example of user's choice and the advised consensus function.

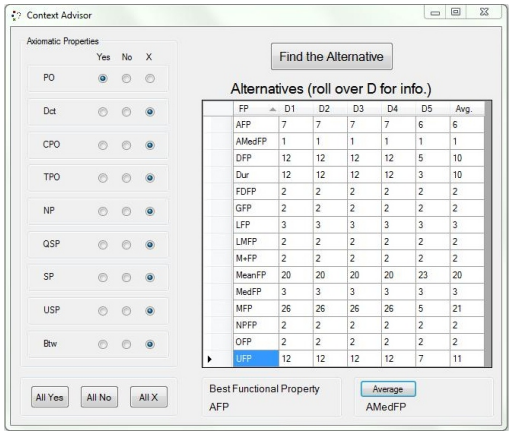



**Fig. 4.** Screen shot of the second layer of the software, with an example of user's choices.

### 3.4 Decision Aiding

In the third layer, the D.A. Software recognizes input trees which are given in Newick format. The Newick tree format is a well-known representation of graph-theoretical trees which denotes trees using parentheses and commas. The simplicity and standard nature of Newick makes it a suitable method for scientists to provide the software with their input. There are several ways through which trees can be represented, however the representation that contains only the information about the leaves are recognized as the valid ones. For example, in Fig. 1, $(H_1)$ has the Newick format $(((a, c), b), d, e)$, while $(H_2)$ is $((a, c), (b, d, e))$.

Upon selection of consensus functions by the user, the D.A. software generates the unique (or set of all possible consensus trees) for the selected functions, so that the user can compare them with each other. Using this feature, the user is able to find out which model would be more suitable for the nature of their work, for which he/she will be provided with respective consensus tree(s). This allows the user to have a narrowed list of candidates for the representative consensus trees as well as having a hands-on experience to find out the most suitable functional property and consensus tree.

## 4 Conclusion and Future Work

In this paper, we presented a decision aiding software which explore via Formal Concept Analysis the space of consensus functions and their axioms. It provides the user with means to generate consensus tree(s) representative(s) depending on their choices. It initially imports the raw context obtained via pre-processing, constructs the associated lattice and, depending on the user's preferences, advise

based on distances in the lattice on which function to use. Upon selection of functions, the program generates the consensus trees of the collection of user's input tree using selected functional properties.

In continuation of this project, we are planning to expand the capabilities of this software. Firstly, besides the (rooted) trees that are currently supported as input and output, the program will be able to support super-trees as well as unrooted trees as its input and output. Another possibility would be the addition of other types of structures of sets such as pyramids [9], weak-trees [4], and, more generally, lattices. In addition, the concept of *independence* and *neutrality* as axiomatic properties are planned to be incorporated. Moreover, other commonly used consensus functions are going to be added to the result, therefore with a further refined and exhaustive approach, the program's precision and usefulness would be improved.

## References

[1] Adams III, E.N.: Consensus Techniques and the Comparison of Taxonomic Trees. Systematic Zoology **21** (1972) 390–397
[2] Adams III, E.N.: N-trees as nestings: Complexity, similarity, and consensus. J. Classif **3** (1986) 299-317
[3] Arrow, K.J.: Social Choice and Individual Values. Wiley, New York (1951)
[4] Bandelt, H.-J., Dress, A.: Weak hierarchies associated with similarity measures: an additive clustering technique Bull. Math. Biology **51** (1989) 133-166
[5] Barbut, M., Monjardet, B.: Ordres et classification: Algèbre et combinatoire (tome II). Hachette, Paris (1970)
[6] Barthélemy, J.-P., McMorris, F.R., Powers, R.C: Dictatorial consensus functions on n-trees. Math. Soc. Sci. **25** (1992) 59-64
[7] Barthélemy, J.-P., Brucker, F.: Average Consensus in Numerical Taxonomy. In Data analysis, Eds. W. Gaul, O. Opitz, and M. Schader, Springer (2000) 95-104
[8] Bertrand, P.: Set Systems and Dissimilarities. Europ. J. Combinatorics **21** (2000) 727-743
[9] Bertrand, P., Diday, E.: A visual representation of compatibility between an order and a dissimilarity index: the pyramids. Comput. Stat. Quart. **2** (1985) 31-44
[10] Birkhoff, G.: Lattice Theory, 3rd ed. Amer. Math. Soc., Providence (1967)
[11] Bock, H.H.: Classification and clustering: Problems for the future. In Diday, E., Lechevallier, Y., Schader, M., Bertrand, P., Burtschy, B.: New approaches in classification and data analysis. Springer-Verlag, Berlin (1994) 3–24
[12] Bremer, K.: Combinable component consensus. Cladistics **6** (1990) 369-372
[13] Bryant, D.: A Classification of Consensus Methods for Phylogenetics. In: Janowitz, M., Lapointe, F.J., McMorris, F., Mirkin, B., Roberts, F. (eds.) Bioconsensus, DIMACS (2003) 163-184
[14] Colonius, H., Schulze, H.-H.: Tree structure for proximity Data. Brit. J. of Math. Stat. Psych. **34** (1981) 167-180
[15] Day, W.H.E., McMorris, F.R.: Axiomatic Consensus Theory in Group Choice and Biomathematics. Siam, Philadelphia (2003)
[16] Davey, B.A., Priestley, H. A.: Introduction to Lattices and Order, 2nd ed. Cambridge University Press (2002)

[17] Domenach, F., Tayari, A.: Implications of Axiomatic Consensus Properties. To appear (2012)
[18] Dong, J., Fernández-Baca, D., McMorris, F.R., Powers. R.C.: An Axiomatic Study of Majority-rule (+) and associated Consensus Functions on Hierarchies. Disc. App. Math. **159** (2011) 2038-2044
[19] Felsenstein, J.: The Number of Evolutionary Trees. Syst. Zool. **27** (1978) 27-33
[20] Felsenstein, J.: PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics **5** (1989) 164-166
[21] Ganter, B., Wille, R.: Formal Concept Analysis : Mathematical Foundations. Springer (1996)
[22] Hudry, O., Monjardet, B.: Consensus Theories. An Oriented Survey. Math. Sci. hum **190** (2010) 139-167
[23] Leclerc, B. Monjardet, B.: Latticial theory of consensus. In: Barnett, V., Moulin, H., Salles M., Schofield N. (eds.) Social choice, Welfare and Ethics. Cambridge University Press, Cambridge (1995) 145-159
[24] Margush, T., McMorris, F.R.: Consensus n-trees. Bull. Math. Biol. **43** (1981) 239-244
[25] May, K.O.: A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decision. Econometrica **20** (1952) 680-684
[26] McMorris, F.R.: : Axioms for consensus functions defined on undirected phylogenetic trees. Math. Biosciences **74** (1985) 77-80
[27] Nelson, G.: Cladistic analysis and synthesis: Principles and definitions, with a historical note on Adanson's Famille des Plantes (1763-1764). Syst. Zool. **28** (1979) 1-21
[28] Neumann, D.A.: Faithful consensus methods for n-trees. Math. Biosci **63** (1983) 271-287
[29] Page, R.D.M.: Tracks and Trees in the Antipodes: A Reply to Humphries and Seberg. Syst. Zool. **39** (1990) 288-299
[30] Page, R.D.M.: User's manual for COMPONENT, Version 2.0. Natural History Museum, London (1993)
[31] Pareto, V.: Cours d'économie politique. F. Rouge, Lausanne (1896)
[32] Phillips, C., Warnow, T.J.: The aymmetric median tree - A new model for building consensus trees. Disc. App. Math. **71** (1996) 311-335
[33] Powers, R.C., White, J.M.: Wilson's theorem for consensus functions on hierarchies. Disc. Appl. Math. **156** (2008) 1321–1329
[34] Semple, M., Steel, C.: A supertree method for rooted trees. Disc. App. Math. **105** (2000) 147-158
[35] Sholander M.: Medians, Lattices, and Trees. Proceedings of the American Mathematical Society **5** (1954) 808-812
[36] Sokal, R.R., Rohlf, F.J., Taxonomic congruence in the Leptopodomorpha reexamined. Syst. Zool. **30** (1981) 309-325
[37] Swofford, D.L.: PAUP: Phylogenetic Analysis Using Parsimony, version 3.0. Illinois Natural History Survery, Champaign (1990)
[38] Wilkinson, M., Thorley, J.L., Pisani, D.E., Lapointe, F.-J., McInerney, James O.: Some Desiderata for Liberal Supertrees. In: Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, Kluwer Academic Publishers (2004) 564-582
[39] Yevtushenko, S.A.: System of data analysis "Concept Explorer". Proceedings of the 7th national conference on Artificial Intelligence KII-2000, Russia (2000) 127-134

# Analyzing Chat Conversations of Pedophiles with Temporal Relational Semantic Systems

Paul Elzinga[1], Karl Erich Wolff[2], Jonas Poelmans[3,4], Guido Dedene[3,5], and Stijn Viaene[3,6]

[1] Amsterdam-Amstelland Police
James Wattstraat 84, 1097DM Amsterdam, The Netherlands
paul.elzinga@amsterdam.politie.nl
[2] Ernst-Schröder-Center, Darmstadt University of Technology
Schloßgartenstr. 7, D-64289 Darmstadt, Germany
karl.erich.wolff@t-online.de
[3] K.U.Leuven, Faculty of Business and Economics, Naamsestraat 69, 3000 Leuven, Belgium
[4] National Research University Higher School of Economics (HSE)
Pokrovskiy boulvard 11, 101000 Moscow, Russia
jonas.poelmans@econ.kuleuven.be
[5] Universiteit van Amsterdam Business School
Roetersstraat 11, 1018 WB Amsterdam, The Netherlands
Guido.Dedene@econ.kuleuven.be
[6] Vlerick Leuven Gent Management School, Vlamingenstraat 83, 3000 Leuven, Belgium
Stijn.Viaene@econ.kuleuven.be

**Abstract.** Grooming is the process by which pedophiles try to find children on the internet for sex-related purposes. In chat conversations they may try to establish a connection and escalate the conversation towards a physical meeting. Till date no good methods exist for quickly analyzing the contents, evolution over time, the present state and threat level of these chat conversations. In this paper we propose a novel method based on Temporal Relational Semantic Systems, the main structure in the temporal and relational version of Formal Concept Analysis. For rapidly gaining insight into the topics of chat conversations we combine a linguistic ontology for chat terms with conceptual scaling and represent the dynamics of chats by life tracks in nested line diagrams. To showcase the possibilities of our approach we used chat conversations of a private American organization which actively searches for pedophiles on the internet.

**Keywords:** Formal Concept Analysis, Temporal Concept Analysis, Conceptual Scaling, Relational Systems, Nested Line Diagrams, Transition Diagrams

**Acknowledgment** Jonas Poelmans is Aspirant of the "Fonds voor Wetenschappelijk Onderzoek – Vlaanderen" (FWO) or Research Foundation – Flanders.

## References


[1] Chein, M., Mugnier, M.-L.: Graph-based Knowledge Representation. Computational Foundations of Conceptual Graphs. Springer-Verlag London Limited (2009)

[2] Chen, H., Chung, W., Xu, J.J., Wang, G., Qin, Y., Chau, M.: (2004) Crime data mining: a general framework and some examples. IEEE Computer, April (2004)

[3] Dau, F.: The Logic System of Concept Graphs with Negation And Its Relationship to Predicate Logic. LNAI 2892, Springer, Heidelberg (2003).

[4] Dombrowski, S.C., Gischlar, K.L., Durst, T.: Safeguarding young people from cyber pornography and cyber sexual predation: a major dilemma of the internet. Child abuse review, Vol. 16, pp. 153–170 (2007).

[5] Elzinga, P., Poelmans, J., Viaene, S., Dedene, G., Morsing, S. : Terrorist threat assessment with Formal Concept Analysis. Proc. IEEE International Conference on Intelligence and Security Informatics. May 23-26, 2010 Vancouver, Canada. ISBN 978-1-42446460-9/10, pp.77–82. (2010)

[6] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999); German version: Springer, Heidelberg (1996)

[7] Gottschalk, P.: A dark side of computing and information sciences: characteristics of online groomers. The Journal of Emerging Trends in Computing and Information Sciences, Vol. 2, No. 9, pp. 447–455, September (2011).

[8] Huchard, M., Rouane-Hacene, M., Cyril Roume, Valtchev, P.: Relational concept discovery in structured datasets. Ann. Math. Artif. Intell. 49(1–4), pp. 39–76.(2007)

[9] IALEIA: Law Enforcement Analytic Standards. Richmond, VA: Global Justice Information Sharing Initiative.(2004)

[10] Krippendorf, K.: The Content analysis Reader. With M. A. Bock (Eds.). Thousand Oaks, CA: Sage, 481 pp.(2008)

[11] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. . A case of using formal concept analysis in combination with emergent self organizing maps for detecting domestic violence, Lecture Notes in Computer Science, 5633, pp. 247 - 260, Advances in Data Mining. Applications and Theoretical Aspects, 9th Industrial Conference (ICDM), Leipzig, Germany, July 20–22, 2009, Springer (2009)

[12] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G. : Curbing domestic violence: Instantiating C-K theory with Formal Concept Analysis and Emergent Self Organizing Maps. Intelligent Systems in Accounting, Finance and Management 17, (3–4), pp. 167–191. Wiley and Sons, Ltd. doi 10.1002/isaf.319 (2010)

[13] Poelmans, J., Elzinga, P., Viaene, S., Dedene, G., Kuznetsov, S. : A concept discovery approach for fighting human trafficking and forced prostitution. Lecture Notes in Computer Science 6828, pp. 201–214, 19th International conference on conceptual structures, July 25–29, Derby, England. Springer (2011)

[14] Poelmans, J., Elzinga, P., Neznanov, A., Kuznetsov, S., Dedene, G., Ignatov, D., Viaene, S. : Concept relation discovery and innovation enabling technology (CORDIET). D. Ignatov et al. (Eds.): Proceedings of the International Workshop on Concept Discovery in Unstructured Data, 25 June, Moscow, Russia, pp. 53 – 62. ISSN 1613-0073.(2011)

[15] Prediger, S.: Kontextuelle Urteilslogik mit Begriffsgraphen. Ein Beitrag zur Restrukturierung der mathematischen Logik. Dissertation, TU Darmstadt 1998. Shaker, Aachen (1998)

[16] Ratcliffe, J.: Intelligence-Led Policing. Collumpton, UK Willan Publishing (2008)

[17] Sowa, J.F.: Conceptual structures: information processing in mind and machine. Adison-Wesley, Reading (1984)
[18] Sowa, J.F.: Knowledge representation: logical, philosophical, and computational foundations. Brooks Cole Publ. Comp., Pacific Grove, CA (2000)
[19] Wille, R.: Restructuring Lattice Theory: an Approach based on Hierarchies of Concepts. In: Rival, I. (ed.): Ordered Sets. pp. 445–470, Reidel, Dordrecht-Boston (1982). Reprinted in: Ferr'e, S., Rudolph, S. (eds.): Formal Concept Analysis. ICFCA 2009. LNAI 5548, pp. 314–339. Springer, Heidelberg (2009)
[20] Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: D. Lukose, H. Delugach, M. Keeler, L. Searle, J.F. Sowa (eds.): Conceptual Structures: Fulfilling Peirce's Dream. LNAI 1257, pp. 290–303. Springer, Heidelberg (1997)
[21] Wolak, J., Finkelhor, D., Mitchell, K.J., Ybarra, M.L.: Online predators and their victims - myths, realities and implications for prevention and treatment. American Psychologist Vol. 63, No. 2, pp. 111–128 (2008).
[22] Wolff, K.E.: Temporal Concept Analysis. In: E. Mephu Nguifo et al. (eds.): ICCS-2001 International Workshop on Concept Lattices-Based Theory, Methods and Tools for Knowledge Discovery in Databases, Stanford University, Palo Alto, CA, 91–107 (2001)
[23] Wolff, K.E.: 'Particles' and 'Waves' as Understood by Temporal Concept Analysis. In: K.E. Wolff, H.D. Pfeiffer, H.S. Delugach (eds.): Conceptual Structures at Work. LNAI 3127, pp. 126–141. Springer, Heidelberg (2004)
[24] Wolff, K.E.: States of Distributed Objects in Conceptual Semantic Systems. In: F. Dau, M.L.Mugnier, G.Stumme (eds.): Common Semantics for Sharing Knowledge. LNAI 3596, pp. 250–266. Springer, Heidelberg (2005)
[25] Wolff, K.E.: States, Transitions, and Life Tracks in Temporal Concept Analysis. In: B. Ganter, G. Stumme, R. Wille (eds.): Formal Concept Analysis State of the Art. LNAI 3626, pp. 127–148. Springer, Heidelberg (2005)
[26] Wolff, K.E.: Relational Semantic Systems, Power Context Families, and Concept Graphs. In: Wolff, K.E. et al (eds.): Contributions to ICFCA 2009, pp. 63–78. Verlag Allgemeine Wissenschaft, Darmstadt (2009)
[27] Wolff, K.E.: Relational Scaling in Relational Semantic Systems. In: Rudolph, S. et al (eds.): Conceptual Structures: Leveraging Semantic Technologies. LNAI 5662, pp. 307–320. Springer-Verlag, Heidelberg (2009)
[28] Wolff, K.E.: Temporal Relational Semantic Systems. In: Croitoru et al (eds.): Conceptual Structures: From Information to Intelligence. LNAI 6208, pp. 165–180. Springer-Verlag, Heidelberg (2010)
[29] Wolff, K.E.: Applications of Temporal Conceptual Semantic Systems. In: Wolff, K.E. et al (eds.): Knowledge Processing and Data Analysis. LNAI 6581, pp. 59–78. Springer-Verlag, Heidelberg (2011)
[30] Wollbold, J., Wolff, K.E., Huber, R., Kinne, R.: Conceptual Representation of Gene Expression Processes. In: Wolff et al (eds.): Knowledge Processing and Data Analysis. LNAI 6581, pp. 79–100. Springer, Heidelberg (2011)

# Closures and Partial Implications in Educational Data Mining

Diego García-Saiz[1], Marta Zorrilla[1], and José L. Balcázar[2]

[1] Mathematics, Statistics and Computation Department, University of Cantabria
Avda. de los Castros s/n, Santander, Spain
garciasad@unican.es marta.zorrilla@unican.es
[2] LSI Department, UPC, Campus Nord, Barcelona
jose.luis.balcazar@upc.edu

**Abstract.** Educational Data Mining (EDM) is a growing field of use of data analysis techniques. Specifically, we consider partial implications. The main problems are, first, that a support threshold is absolutely necessary but setting it "right" is extremely difficult; and, second, that, very often, large amounts of partial implications are found, beyond what an EDM user would be able to manually inspect. Our program *yacaree*, recently developed, is an associator that tackles both problems. In an EDM context, our program has demonstrated to be competitive with respect to the amount of partial implications output. But "finding few rules" is not the same as "finding the right rules". We extend the evaluation with a deeper quantitative analysis and a subjective evaluation on EDM datasets, eliciting the opinion of the instructors of the courses under analysis to assess the pertinence of the rules found by different association miners.

**Keywords:** Closure Lattices, Partial Implications, Association Rules

## 1 Introduction

Education is evolving at all levels since the appearance of e-learning environments: Learning Content Management Systems (LCMS), Intelligent Tutoring Systems, or Adaptive Educational Hypermedia Systems. These systems log all the activity carried out by students and instructors, and this raw data, adequately analyzed, might help instructors to obtain a better understanding of the students and of their learning processes. In remote learning, instructors may never see their students in person. Data analysis techniques could help them to detect problems (lack of motivation, under-performance, drop-out...) and, possibly, to take action. Yet, unless the course itself is on data mining, it is unlikely that the instructors know much about data mining techniques. If we want to help teachers of, say, philology or law, we need to work out data mining tools that do not require much tuning or technical understanding.

Here we focus on the particular case of mining partial implications [1] (a relaxed form of implication analysis in concept lattices [2]), and their close relatives: association rules [3]. Most of the available algorithms depend on one or more parameters whose value is to be set by the user, and whose semantics are unlikely to be easy to understand by teachers of other disciplines.

We have explored the output of five association algorithms on datasets from educational sources, and evaluated not only the amounts of partial implications found but also the subjective pertinency of the rules obtained. For this last task we kept close cooperation with the end user, namely, the teachers of the online courses from which the datasets were obtained. Our conclusions are in the form of strengths and weaknesses of each of the five algorithms compared.

One of the algorithms participating in the evaluation was a contribution of our group, demonstrated at [4] and described in more detail in [5]: the *yacaree* association miner. This associator extracts partial implications from the "iceberg" (frequent part of the) FCA lattice [6]; it attempts at offering a more user-friendly, parameter-less interface, through self-tuning the support threshold and a threshold on a relative form of confidence studied in [7]: the closure-based confidence boost.

In [8], a two-page poster publication, we have provided a preliminary initial description of this study, containing only the quantitative analysis (a part of Table 2 below) but using a version of *yacaree* which did not report yet rules of confidence 100%. This paper extends it largely with further quantitative analyses and a qualitative, user-based, subjective evaluation of the usefulness of the resulting rules. The main question to study is whether a price, in terms of usefulness of the output for the end user, was being paid for the parameter-less interface. Any parameter-free alternative should stand a comparison of its output with that of other, "expert"-oriented algorithms, to clarify whether, for the subjective perception of the teacher, the outcome does make sense and results useful. Actually, our main conclusion is that they do, and that, developed according to our strategy, a self-tuning associator is able to provide sensible quantities of partial implications that result useful and informative to the end user.

### 1.1 Related work

In the educational context, data mining techniques are used in order to understand learner behaviour [9], to recommend activities or topics [10], to offer learning experiences [11] or to provide instructional messages to learners [12] with the aim of improving the effectiveness of the course, promoting group-based collaborative learning [13], or even predicting students' performance [9]. Two interesting papers which detail and summarize the application of data mining to educational systems are [14] and [15].

The FCA community has also contributed in this arena. We must name Romashkin et al. [16] who used closed sets of students and their marks to reveal some interesting patterns and implications in student assessment data, especially to trace dynamic; and Ignatov et al. [17] who showed that FCA taxonomies are a useful tool for representing object-attribute data which helps to reveal some

frequent patterns and to present dependencies in data entirely at a certain level of details. They carried out the analysis of university applications to the Higher School of Economics as case study. Another interesting work in this research line was previously carried out by Belohlávek et al. [18] in order to evaluate questionnaires.

In the particular case of the association rules technique, we find works such as [19] in which association rules are used to find mistakes often made together while students solve exercises in propositional logic, [20] where rules are used to discover the tools which virtual students employ frequently together during their learning sessions, and [21] where association rules and collaborative filtering are used inside an architecture for making recommendations in courseware.

However, association rule algorithms still have some drawbacks, as analyzed in [22]: mainly, first, as most often the instructors are not data mining experts, the decisions about setting to useful values the parameters of the algorithms present difficulties. Then, a second difficulty is the large number of rules often obtained as output, most of which are redundant and non-interesting for decision making and, in many occasions, exhibit low understandability. The authors of [22] offer some solutions although none of them is automatized or gathered in an algorithm. For example, they propose to use Predictive Apriori, rather than the implementation of Apriori in Weka [23], since it only requires one parameter which is the number of rules that the user wants to obtain. In [24], it is argued that cosine and added value (or equivalently lift) are well suited to educational data, and that instructors can interpret their results easily. In our opinion, these measures lack actionability since they are symmetric, which reduces the use of the rules in decision making tasks. Orientation is a crucial and very suggestive property of association rules and partial implications, and we consider that it must be preserved in an effective but asymmetric measure, as close as possible to confidence. Many measures of intensity of implication are described e.g. in [25],[26].

## 2 Case Studies

This section contains our major contributions: we compare the output of five well-known association rule miners on five educational datasets and evaluate the subjective pertinency of the rules obtained in close cooperation with the teachers involved in the two virtual courses analyzed.

### 2.1 Association rule miners

There is a long list of association rule miners; large sets of references and surveys appear e.g. in http://michael.hahsler.net/research/bib/association rules/ and in all main Data Mining reference works. Among them, we have chosen the following algorithms for our comparison: the implementation of Apriori by Borgelt [27], the implementation of Apriori in the Weka package [23], the Predictive Apriori

implementation in Weka [28], the implementation of ChARM [29] available in the Coron System [30], and our own closure-lattice-based associator *yacaree* [4].

The implementation of Apriori by Borgelt [27] is a representative of the standard usage of association rules in data mining, as per [3], particularly in the way support and confidence parameters are handled, as well as in the restriction to association rules with a single item in the consequent. In this fully standard approach, first, one constructs all frequent sets, and then each item in each frequent set is tried as consequent with the rest of the frequent itemset as antecedent, and the confidence of the rule evaluated; the rule is reported if its confidence is high enough. This implementation is amazingly well streamlined for speed. It offers, additionally, an ample repertory of additional evaluation measures (lift, normalized chi-square...), and we must warn that a specific flag must be set (as we did, "`-o`") so that support is computed accordingly with the notion of support in other tools.

Weka is one of the oldest and most extended open-source data mining suites, and all implementations there are widely used. The implementation of Apriori in the Weka package is similar to the one just described, employing confidence and support constraints; it departs slightly from [3], though. First, the rules generated can have more than one item in the consequent. Also, instead of fixing the support at the given threshold at once, the user is requested to indicate a number of rules and a "delta" parameter. Then, support is set initially at 100% and iteratively reduced by "delta" until either the support threshold is reached or the requested number of rules is collected.

The Predictive Apriori implementation in Weka follows [28]. The advantage of this algorithm is that it only requires from the user to set the number of rules to be discovered, which is appropriate for users that are not data mining experts, provided that, in some sense, "the right rules" are found. The algorithm automatically attempts at balancing optimally support and confidence on the basis of Bayesian criteria related to the so-called expected predictive accuracy. A disadvantage of this method is that it often requires longer running times than the previous ones.

These three implementations construct partial implications on the basis of all frequent itemsets. Our other two systems work on the basis of frequent closures, which allow one to know the support of any frequent itemset without storing all of them. The Coron system [30] offers several implementations of different closed-set-based algorithms. These methods return the same set of closure-based partial implications, although they compute them in different ways. We have used ChARM [29], but the specific method is not relevant here because we do not include yet running times in our evaluation: we concentrate on the usefulness of the output.

The fifth implementation is our own association miner *yacaree* [4]. Like ChARM, it is based on closures, and allows for several items in the consequent of the partial implications. In the partial implications output by this system, both antecedent and total set of items in each rule will be closed sets. The currently most recent version 1.2.0 is the first to report rules of confidence 100%.

First, it constructs the Closure Lattice up to a support bound that is adjusted autonomously during the run, on the basis of the technological limitations, so that the user does not need to select it. Second, it constructs a basis of partial implications out of these closures. Third, it filters the partial implications along the way, on the basis of the closure-based confidence boost [7], whereby the confidence of an association rule is compared to that of other similar rules: a rule must offer a clear improvement on similar ones to be considered useful.

### 2.2 Datasets

For the case studies, we used the data from two courses offered in the University of Cantabria. Both courses are eminently practical. The first one, entitled "Introduction to multimedia methods", has the objective of teaching the students how to use a particular multimedia tool (in what follows, we refer to it as the multimedia dataset) and the second one, "Basic administration of a UNIX-LINUX system" (the Linux dataset) teaches the students the basic utilities and tools to install and configure correctly a LINUX operating system.

The multimedia course is designed by means of web pages and includes some video tutorials, flash animations and interactive elements. The students must perform 4 exercises, 2 projects and one final exam online. The course is open to all degrees and the number of students enrolled was 79.

Unlike the multimedia course, the Linux course only allows 24 students to be enrolled, all of them from a telecommunications degree. All materials of the course are available since the first day of the course. Furthermore, the contents of a previous edition of the course is also offered in pdf; these files have the advantage that they can be kept locally and used for study in case any technical problem would prevent access to the updated files, but do not include all the contents of the present edition. Additionally, during the course, the students must deliver 6 practical exercises and pass two online exams. The course includes 38 self-tests, one for each topic of the course. The instructor indicates the topics and self-tests that they must perform every week on the calendar.

We worked with five datasets. The first one, "linux materials", gathers the access logs to materials prepared by the instructor (html pages, pdf files, tests, and so on) as used by each student in each learning session of the Linux course. The datasets "linux resources" and "multimedia resources" are the session-wise log of the resources and tools used by each student in each learning session(assessment, content-pages, forum, and so on). It was immediately apparent that, in these datasets, one specific resource led to some "noise": the "organizer" resource acts as front page of most sessions (near 84% in Linux and 85% in multimedia, as the only other alternative is the access through the forum) and hence it appears in many rules and creates many variants, mostly of low information contents. Thus, we prepared two datasets, named "linux resources reduced" and "multimedia resources reduced" respectively, which are identical to the second and third dataset, except that the "organizer" resource is fully removed. The number of different items and transactions of each dataset is shown in Table 1. For the

sake of better understanding, we show a diagram of the intents of the concept lattice of the linux dataset above 13% support in Fig. 1.

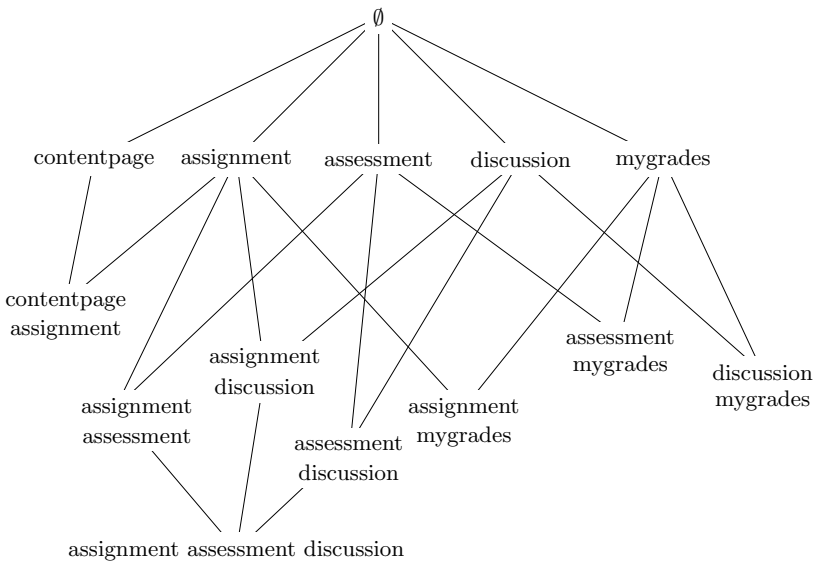



**Fig. 1.** Intents of at least 13% support.

**Table 1.** Datasets description

| Name | Transactions | Items |
|---|---|---|
| Dataset1 (linux materials) | 407 | 22 |
| Dataset1 (linux resources) | 2486 | 27 |
| Dataset2 (linux resources reduced) | 2346 | 26 |
| Dataset4 (multimedia resources) | 5892 | 27 |
| Dataset5 (multimedia resources reduced) | 5643 | 26 |

### 2.3 Datasets results

With the aim of comparing several association programs, one difficulty is always the setting of the parameters, particularly the support, as the value chosen might favor one particular algorithm in larger degree. In our case, there is an extra level of difficulty, as one of the participating algorithms, *yacaree*, self-tunes the

support on itself. In order to find fair comparison grounds, we performed a brief preprocessing.

Running on one of the "Linux resources" dataset, *yacaree* took about four minutes (a bit long for a non-expert to wait) and delved down to 0.02% support; however, for this low threshold, both Weka alternatives were substantially worse (Predictive Apriori took 40 minutes and Apriori led to overflow even when given 2GB of memory). Similar facts happened for the other datasets.

Given this information, we decided to fix at 1% the support threshold for all the computations, and at 66% the confidence threshold (initial value set up by *yacaree*). In all the runs, we left unbounded, or, in the case of Weka tools, we set very high (10000) the number of rules to be found, even if this meant overriding their default value for this quantity. We show the number of rules obtained utilizing the different algorithms on our datasets in Table 2. The entries marked "—" on the table are cases where the corresponding algorithm was unable to complete in 6 hours.

**Table 2.** Number of rules obtained on our datasets with the five algorithms

| Dataset | Number of rules s=1% c=66% | | | | |
|---|---|---|---|---|---|
| | Weka Apriori | Predictive Apriori | Borgelt Apriori | ChARM | *yacaree* |
| Dataset1 (linux materials) | 2272 | 1730 | 524 | 366 | 40 |
| Dataset2 (linux resources) | 7523 | over 10000 | 3751 | 5610 | 255 |
| Dataset3 (linux resources reduced) | 4249 | over 10000 | 1876 | 2586 | 93 |
| Dataset4 (multimedia resources) | 1442 | — | 1023 | 1427 | 182 |
| Dataset5 (multimedia resources reduced) | 488 | — | 404 | 469 | 46 |

**Results from "resources reduced" datasets** If we analyze the results obtained with Apriori from Weka, we can see that the number of rules is unmanageable, e.g. 4249 rules for Linux resources reduced dataset. The first 243 are implications of full confidence, 100%, low support, and high redundancy: see rules 2 and 3 and 235 and 236 and the followings in Table 3. Had we used the tool's default settings of the parameters, we would have found essentially no information. The same happens with multimedia dataset (we do not show the table for space reasons).

The analysis of the results obtained from Predictive Apriori is very costly, as it generates as many rules as we allow it to. With 10000 rules required, they are obtained on dataset2 and dataset3 waiting for more than 20 minutes, and the accuracy is still high, so that many further rules could be obtained. If we restrict ourselves to the first few rules returned, they turn out to offer a very low support and quite some redundancy (see Table 4).

The output offered by Borgelt's implementation presents a large number of rules: 1876 and 404 rules in Linux and multimedia reduced datasets respectively,

**Table 3.** Subset of association rules obtained with Apriori from Weka on the "Linux resources reduced" dataset

| No. | Association rule | (Sup., Conf.) |
|---|---|---|
| 2 | announcement tracking ⇒ assessment | (1.7, 100) |
| 3 | announcement mygrades tracking⇒ assessment | (1.6, 100) |
| 235 | assignments calendar contentpage discussion medialibrary syllabus ⇒ assessment | (1.0, 100) |
| 236 | assessment calendar contentpage discussion medialibrary syllabus ⇒ assignments | (1.0, 100) |
| 2523 | announcement assessment calendar syllabus ⇒ assignments contentpage | (1.2, 78.0) |
| 2524 | announcement assessment calendar syllabus ⇒ assignments discussion | (1.2, 78.0) |
| 2530 | announcement calendar mail ⇒ contentpage | (1.0, 78.0) |
| 2534 | announcement assignments calendar chat ⇒ contentpage | (1.0, 78.0) |

**Table 4.** Subset of association rules obtained with Predictive Apriori from Weka on the "linux resources reduced" dataset

| No. | Association rule | (Support, Accuracy) |
|---|---|---|
| 122 | assignments calendar search ⇒ syllabus | (0.85, 0.95439) |
| 123 | assignments chat weblinks ⇒ assessment syllabus | (0.85, 0.95439) |
| 124 | assignments chat weblinks ⇒ discussion syllabus | (0.85, 0.95439) |
| 125 | assignments discussion search ⇒ assessment syllabus | (0.85, 0.95439) |

of which 141 and 2 are implications. Coming up with specific conclusions becomes harder. The rules tend to be small, exhibit high redundancy and involve low-support tools that are almost never used, so that they offer little interest to the instructor. As shown in Table 5, where the rules 11, 12, 13 differ slightly from the rules 99, 100 and 101 which contain the announcement tool in the antecedent with a very low support and similar confidence.

**Table 5.** Subset of association rules obtained with Borgelt's apriori implementation on the "linux resources reduced" dataset

| No. | Association rule | (Supp. , Conf. ) |
|---|---|---|
| 11 | chat ⇒ discussion | (3.7, 84.9) |
| 12 | chat ⇒ assignments | (3.7, 75.6) |
| 13 | chat ⇒ assessment | (3.7, 81.4) |
| 99 | chat announcement ⇒ discussion | (2.0, 84.8) |
| 100 | chat announcement ⇒ assignments | (2.0, 87.0) |
| 101 | chat announcement ⇒ assessment | (2.0, 93.5) |

ChARM returns a higher number of rules, 2586 and 469 with 193 and 2 implications in Linux and multimedia resources reduced datasets respectively.

As in previous cases, the rules also present high redundancy (see rules 3 to 6 and 7 and 8 in Table 6 and rules 10,11,12 and 31,32,33 in Table 7).

**Table 6.** Subset of association rules obtained with ChARM on the "linux resources reduced" dataset

| No. | Association rule | (Supp. , Conf. ) |
|---|---|---|
| 3 | announcement, contentpage, medialibrary, syllabus ⇒ assessment | (1.02, 96.00) |
| 4 | announcement, assessment, medialibrary, syllabus ⇒ contentpage | (1.02, 88.89) |
| 5 | announcement, assessment, contentpage, medialibrary ⇒ syllabus | (1.02, 70.59) |
| 6 | announcement, medialibrary, syllabus ⇒ assessment, contentpage | (1.02, 82.76) |
| 7 | announcement, medialibrary, syllabus ⇒ contentpage | (1.07, 86.21) |
| 8 | announcement, contentpage, medialibrary ⇒ syllabus | (1.07, 67.57) |

**Table 7.** Subset of association rules obtained with ChARM algorithm on the "multimedia resources reduced" dataset

| No. | Association rule | (Supp. , Conf. ) |
|---|---|---|
| 10 | chat, contentpage, discussion ⇒ assessment | (1.13, 81.01) |
| 11 | assessment, chat contentpage ⇒ discussion | (1.13, 94.12) |
| 12 | chat, contentpage ⇒ assessment, discussion | (1.13, 71.91) |
| 31 | contentpage, discussion, syllabus, ⇒ assessment | (1.12, 84.00) |
| 32 | assessment, discussion, syllabus, ⇒ contentpage | (1.12, 66.32) |
| 33 | assessment, contentpage, syllabus, ⇒ discussion | (1.12, 79.75) |

Despite the fact that the number of rules obtained with *yacaree* on reduced resources datasets is a bit high, 93 for dataset3 and 46 for dataset5, it is possible to discover the resources which students use frequently together in each learning session and, at the same time, the kind of sessions which they perform. It is remarkable the reduction in the number of rules due to the use of confidence boost parameter. A subset of the most relevant rules obtained with *yacaree* on Linux resources reduced dataset is shown in Table 8. However, there appear as well quite a few trivial and non-interesting rules for the instructor. For instance, rule 1 is trivial because it is obvious that to send a task is necessary to use the file manager tool. The rules 6, 18 and 19 do not offer new information to the instructor given that he uses the forum in order to establish the date of the exams. So that these kind of sessions are known to the instructor. The rules 7, 12, 36 and 50 gather sessions in which students want to know specific dates: deadlines for tasks or assessments, exam dates. Rule 16 indicates quite a few sessions in which the students are interested in knowing their progress, and rules 8 and 10 gather the study sessions in which the students combine reading of content pages with tackling self-tests.

Table 9 depicts a subset of the most relevant rules obtained with *yacaree* on multimedia resources reduced dataset. As in the previous result, there are

**Table 8.** Subset of association rules obtained with *yacaree* on the "linux resources reduced" dataset

| No. | Association rule | (Supp., Conf., Lift, Cboost) |
|---|---|---|
| 1 | filemanager ⇒ assignments | (4.6, 93.9, 1.908, 1.908) |
| 6 | discussion whoisonline ⇒ assessment | (3.0, 75.5, 1.648, 1.379) |
| 18 | discussion mail ⇒ assessment | (3.2, 72.1, 1.574, 1.268) |
| 19 | announcement mail ⇒ assessment discussion | (1.6, 80.9, 3.381, 1.267) |
| 7 | announcement ⇒ assessment | (7.6, 88.1, 1.923, 1.369) |
| 12 | calendar ⇒ assessment | (9.1, 75.9, 1.656, 1.337) |
| 36 | calendar ⇒ assignments | (8.1, 67.0, 1.362, 1.219) |
| 50 | announcement calendar ⇒ assessment assignments | (2.6, 77.2, 2.941, 1.200) |
| 16 | tracking ⇒ mygrades | (6.8, 80.3, 2.409, 1.272) |
| 8 | contentpage mygrades ⇒ assessment | (3.8, 84.8, 1.850, 1.369) |
| 10 | contentpage discussion ⇒ assessment | (7.3, 75.1, 1.639, 1.339) |

some trivial and non-interesting rules for the instructor. For example, rule 1 already explained, and rule 2 and 40 which gather sessions in which students wanted to know specific dates for assignments. Instead, other rules as rule 7, 14 and 36 allowed the teacher to discover the students visited the content pages and the forum in working sessions with the aim at solving problems or doubts in the resolution of the tasks. Furthermore, she was happy when observed that learning objectives tool was used while studying the contents (rule 3). This means that students played the videotutorials which she had recorded with great effort. Additionally, rule 4 informed her about the joint use of contents and weblinks tools. This last one contains the links to downloadable material. This reinforced her idea that the material should be presented in both formats, online and downloadable.

**Table 9.** Subset of association rules obtained with *yacaree* on the "multimedia resources reduced" dataset

| No. | Association rule | (Supp., Conf., Lift, Cboost) |
|---|---|---|
| 1 | filemanager ⇒ assignments | (5.1, 71.5, 1.871, 1.871) |
| 2 | calendar ⇒ assignments | (6.1, 74.9, 1.961, 1.610) |
| 40 | announcement ⇒ assignments | (3.9, 67.2, 1.759, 1.153) |
| 3 | weblinks ⇒ contentpage | (3.7, 78.2, 2.105, 1.588) |
| 4 | learningobjectives ⇒ contentpage | (4.5, 81.4, 2.192, 1.530) |
| 7 | contentpage mygrades ⇒ assignments | (2.7, 66.7, 1.746, 1.421) |
| 14 | assignments whoisonline ⇒ discussion | (1.7, 72.5, 1.612, 1.301) |
| 36 | discussion weblinks ⇒ assignments | (1.9, 73.4, 1.923, 1.180) |

**Results from "resources" datasets, not reduced** From the point of view of a virtual course instructor who is not an expert in Data Mining, the decision

of removing the "organizer" item from the "resources" dataset is debatable. This would be rather an action typical of a Data Mining expert. We consider that it was appropriate to do it, as the designers of the e-learning platform could easily predict that this "organizer" item was to be extremely frequent, and thus the option of discarding it could be incorporated by design into a set of related data mining tools ahead of time. However, we briefly discuss now what happens if one works with the complete "resources" dataset.

With *yacaree* we obtain 255 and 182 rules in dataset2 and dataset4 respectively. In both cases, one of them indicates that "organizer" is used in near 84% and 85% of the sessions respectively (see Tables 10 and 11). For this format of rule, with empty antecedent, support and confidence clearly must coincide. Essentially, the output of *yacaree* is not that different from the previous cases: many rules from the previous analysis reappear now in pairs, once with "organizer" and once without; when such a pair appears, the rule having "organizer" may look sometimes redundant, but its confidence boost value shows that it has high enough confidence so as to make it nonredundant (see Tables 10 and 11).

**Table 10.** Subset of association rules obtained with *yacaree* on the "Linux resources" dataset

| No. | Association rule | (Supp., Conf., Lift, Cboost) |
|---|---|---|
| 2 | ⇒ organizer | (83.9, 83.9, 1.000, 1.982) |
| 158 | mygrades tracking ⇒ assessment organizer | (4.6, 71.7, 1.888, 1.109) |
| 287 | mygrades tracking ⇒ assessment | (5.0, 78.6, 1.818, 1.096 ) |

**Table 11.** Subset of association rules obtained with *yacaree* on the "multimedia resources" dataset

| No. | Association rule | (Supp., Conf., Lift, Cboost) |
|---|---|---|
| 1 | ⇒ organizer | (84.9, 84.9, 1.000, 2.421) |
| 9 | chat ⇒ discussion organizer | (2.0, 77.6, 2.324, 1.283) |
| 113 | chat ⇒ discussion | (2.2, 84.2, 1.954, 1.085) |

The extra effort to be spent on the *yacaree* output is not that high compared with the alternative algorithms. ChARM and Borgelt's Apriori runs into the same difficulties indicated for the reduced datasets, increased by the fact that the number of rules is, with ChARM, 5610 in dataset2 and 1427 in dataset4, and with Borgelt, 3751 in dataset2 and 1023 in dataset4, which include a considerable number of rules whose only consequent is "organizer". Intuitively, all of them are pointing out to the fact that this item is so prevalent. Similarly, Weka Apriori obtains over 7000 rules in dataset2 and 1442 in dataset4, of which the first 568 are implications of 100% confidence, 474 of which are again rules that only have "organizer" as consequent. Predictive Apriori, beyond taking 45 minutes

to complete, also generates a large amount of rules (which we limited to 10000 again); and again the first ones have as single consequent "organizer", and the next ones are long rules of very low support.

**Results from the "linux materials" dataset** We show in the Table 12 some of the most relevant rules among the 40 rules, of which 16 are implications of confidence 100%, selected by *yacaree* on this dataset. Such a limited output size allows for easy inspection by the instructor.

**Table 12.** Subset of association rules obtained with *yacaree* on the "materials" dataset

| No. | Association rule | (Supp., Conf., Lift, Cboost) |
|---|---|---|
| 1 | topic6 ⇒ topic-pdf | (13.3, 1.0, 2.544, 2.544) |
| 2 | topic7 ⇒ topic-pdf | (9.8, 1.0, 2.544, 2.500) |
| 3 | topic4 topic-pdf ⇒ topic5 | (6.4, 76.5, 5.764, 2.266) |
| 18 | topic1 topic3 ⇒ topic2 | (3.9, 72.7, 4.055, 1.377) |
| 6 | topic9 ⇒ topic10 topic-pdf | (0.057, 1.0, 7.537, 1.917) |
| 7 | topic10 topic7 ⇒ topic8 topic-pdf | (0.037, 1.0, 14.536, 1.875) |
| 23 | topic-pdf topic10 topic6 ⇒ topic8 | (2.9, 66.7, 9.690, 1.286) |
| 40 | exam2 topic-pdf ⇒ topic10 | (1.7, 77.8, 5.862, 1.167) |
| 9 | test2 ⇒ test1 test3 | (4.9, 71.4, 13.844, 1.667) |
| 10 | test9 ⇒ test6 test7 test8 topic-pdf topic10 | (2.5, 66.7, 27.133, 1.667) |
| 14 | test7 topic-pdf topic10 ⇒ test6 test8 test9 | (2.5, 76.9, 31.308, 1.538) |
| 23 | test9 ⇒ test8 topic-pdf topic10 | (3.4, 93.3, 23.742, 1.273) |
| 28 | test3 test4 ⇒ test5 topic-pdf | (2.7, 73.3, 14.213, 1.222) |

The rules show that the course is divided clearly in two parts, up to topic and test number 5 and the followings (see rules 2 and 18 and 6, 7 and 23 as well as the set of rules from 9 to 28). The instructor observed that not all topics get really studied: some are worked out only through self-tests (set rule from 9 to 28 with a higher support than the corresponding to topic rules). He was very interested by these rules: first, as many of them indicate that students do not really study their assigned materials, but rather they undertake the tests and only look at the study materials when they do not know the answer, hence reversing the intended order of use of the materials; second, because they show that the outdated, incomplete materials from the earlier edition of the course (topic-pdf appears in most rules), which were thought of as a remedial offer for cases of technical connectivity difficulties only, were actually used much more than intended, even in sessions devoted to learning through self-tests. The first seven rules shown in the table also seems to suggest that students checked at what extent the contents of each topic differs from the old compiled version and as it was easier to manage and carry out searches, they frequently used it with tests. Another piece of interesting information, as judged by the teacher, is the fact that the topics in the second half of the course were consulted in more sessions than the first; this did match his perception that he had had to offer

more "moral support" to students on the brink of failure towards the end of the course. Rule 38 shows a good support for exam2, which is not the case for exam1; in fact, the exams are one-shot events. This unexpected support for exam2 was due to technical problems: half the students lost their connections and had to reconnect later in order to finish their exams, accounting for a misleadingly high number of sessions. (The instructor was surprised that our association rules could detect this.).

With Coron's ChARM many of the rules generated are somewhat redundant variants of the rules found by *yacaree*. Many other rules are also found: essentially, longish rules of confidence 100% (see Table 13). The task of browsing through the hundreds of rules, however, is slow and not user-friendly, and we do not believe a regular instructor would display enough patience to find out the most instructive rules among those returned by the algorithm.

**Table 13.** Subset of association rules obtained with Coron's ChARM implementation on the "materials" dataset

| No. | Association rule | (Supp. , Conf. ) |
|---|---|---|
| 6 | topic7 topic9 topic10 topic-pdf ⇒ topic8 | (1.23, 100.00) |
| 7 | topic7 topic8 topic9 topic-pdf ⇒ topic10 | (1.23, 100.00) |
| 8 | topic7 topic8 topic9 topic10 ⇒ topic-pdf | (1.23, 100.00) |
| 9 | topic7 topic9 topic-pdf ⇒ topic8 topic10 | (1.23, 100.00) |
| 65 | test5 test7 test8 test9 topic10 topic-pdf ⇒ test6 | (1.47, 100.00) |
| 66 | test5 test6 test8 test9 topic10 topic-pdf ⇒ test7 | (1.47, 100.00) |
| 67 | test5 test6 test7 test9 topic10 topic-pdf ⇒ test8 | (1.47, 100.00) |

This objection also happens in Borgelt's implementation and worsens with the Weka Apriori, which produces 2272 rules, of which 1522 are again longish implications of confidence 100%. Still, one can see that some of the rules having several items as consequent subsume into a single line several rules that the classical scheme separates into one rule per consequent item. Predictive Apriori generates 1730 rules, of which the first handful are 100% confidence implications with topic-pdf (the old material) as consequent, and the rest consists mostly of rules of rather low support.

## 3 Conclusions

One of the drawbacks of some data mining algorithms is a dependence on suitable parameter settings which can be difficult for "non-expert data miners" to determine. Another aspect is the degree of difficulty of interpretation of the results. Although the results obtained by association rule miners can be considered easy to interpret by end-users, the large number of rules generated by the more commonly used algorithms, most of which contain facts that, intuitively, will be seen as redundant by users, makes their interpretation and comprehension difficult.

Our comparison of different associators shows that they are vastly different in mere quantitative terms (already advanced in [8] and confirmed in this work); most associators lead to voluminous output; on the other hand, *yacaree* provides several dozen rules that may contain good knowledge yet will not overwhelm the user.

The main question, then, is: are they "the right ones?" Our educational datasets seem to require a low support threshold, but do include items of rather high support; and this combination seriously hinders the ability of traditional association miners to offer interesting output. On the other hand, the most recent version of *yacaree*, which includes implications of confidence 100%, seems particularly well-suited to these cases, and finds rules of both high and low supports; and indeed we find that in most cases these rules "say different things". All our conclusions have been thoroughly discussed with the instructors of the virtual courses to which the datasets refer.

Summarizing, we can say that *yacaree* offers several advantages for non-expert data miners. First, it offers a parameter-less interface, which makes its usage easier. Second, it generates a reduced number of rules, as it works with closed frequent itemsets, mines only a rule basis, and prunes the rules through the confidence boost parameter. Third, it shows the support, confidence, lift and confidence boost in the output at the same time, which allows end-users to better assess the rules, once these measures are conveniently explained.

The current (and previous) versions of *yacaree* present a limitation: by default, it sets up the number of output rules to 50; our study reveals that this condition should be removed or, at least, relaxed. Previous versions did not search for full implications, and only the latest current version (1.2.0) does; our studies confirm that this must be maintained, as a number of interesting implications for our external user were missed in previous versions.

As final conclusion, our interaction with the instructors involved in the virtual courses analyzed indicates that the results of *yacaree* are superior, in the case of analyzing datasets coming from logs of educational learning systems, in comparison with the rest of the algorithms used in our case study. This program can be freely downloaded from SourceForge, and a link has been provided in the web page on FCA software kindly maintained by prof. Uta Priss.

## References

1. Luxenburger, M.: Implications partielles dans un contexte. Mathématiques et Sciences Humaines **29** (1991) 35–55
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag (1999)
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press (1996) 307–328
4. Balcázar, J.L.: Parameter-free association rule mining with yacaree. In Khenchaf, A., Poncelet, P., eds.: EGC. Volume RNTI-E-20 of Revue des Nouvelles Technologies de l'Information., Hermann-Éditions (2011) 251–254

5. Balcázar, J.L., García-Sáiz, D., de la Dehesa, J.: Iterator-based algorithms in self-tuning discovery of partial implications. ICFCA, Supplementary proceedings (2012)
6. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with Titanic. Data Knowl. Eng. **42**(2) (2002) 189–222
7. Balcázar, J.L.: Formal and computational properties of the confidence boost in association rules. Available at: [http://personales.unican.es/balcazarjl]. Extended abstract appeared as [31] (2010)
8. Zorrilla, M.E., García-Sáiz, D., Balcázar, J.L.: Towards parameter-free data mining: Mining educational data with yacaree. [32] 363–364
9. Hung, J.L., Zhang, K.: Revealing online learning behaviors and activity patterns and making predictions with data mining techniques in online teaching. Journal of Online Learning and Teaching **4**(4) (2008) 426–436
10. Zaïane, O.R.: Building a recommender agent for e-learning systems. In: Proc. of the International Conference on Computers in Education (ICCE), Washington, DC, USA, IEEE Computer Society (2002) 55–59
11. Au, T.W., Sadiq, S., Li, X.: Learning from experience: Can e-learning technology be used as a vehicle? In: Proceed ings of the fourth International Conference on e-Learing, Toronto: Academic Publishing Limited (2009) 32–39
12. Ueno, M., Okamoto, T.: Bayesian agent in e-learning. IEEE International Conference on Advanced Learning Technologies (2007) 282–284
13. Perera, D., Kay, J., Koprinska, I., Yacef, K., Zaïane, O.R.: Clustering and sequential pattern mining of online collaborative learning data. IEEE Transactions on Knowledge and Data Engineering **21**(6) (2009) 759–772
14. Romero, C., Ventura, S.: Educational data mining: A review of the state-of-the-art. IEEE Tansactions on Systems, Man and Cybernetics, part C: Applications and Reviews **40**(6) (2010) 601–618
15. Castro, F., Vellido, A., Nebot, A., Mugica, F.: Applying data mining techniques to e-learning problems. In Kacprzyk, J., Jain, L., Tedman, R., Tedman, D., eds.: Evolution of Teaching and Learning Paradigms in Intelligent Environment. Volume 62 of Studies in Computational Intelligence. Springer Berlin Heidelberg (2007) 183–221 10.1007/978-3-540-71974-8_8.
16. Romashkin, N., Ignatov, D.I., Kolotova, E.: How university entrants are choosing their department? mining of university admission process with fca taxonomies. [32] 229–234
17. Ignatov, D.I., Mamedova, S., Romashkin, N., Shamshurin, I.: What can closed sets of students and their marks say? [32] 223–228
18. Belohlávek, R., Sklenar, V., Zacpal, J., Sigmund, E.: Evaluation of questionnaires supported by formal concept analysis. In Eklund, P.W., Diatta, J., Liquiere, M., eds.: CLA. Volume 331 of CEUR Workshop Proceedings., CEUR-WS.org (2007)
19. Merceron, A., Yacef, K.: Mining student data captured from a web-based tutoring tool: Initial exploration and results. Journal of Interactive Learning Research **15**(4) (2004) 319–346
20. Zorrilla, M.E., García-Saiz, D.: Mining service to assist instructors involved in virtual education. In Zorrilla, M.E., Mazón, J.N., Óscar Ferrández, Garrigós, I., Daniel, F., Trujillo, J., eds.: Business Intelligence Applications and the Web: Models, Systems and Technologies. Information Science Reference (IGI Global Publishers) (September 2011)
21. García, E., Romero, C., Ventura, S., de Castro, C.: An architecture for making recommendations to courseware authors using association rule mining and collaborative filtering. User Model. User-Adapt. Interact. **19**(1-2) (2009) 99–132

22. García, E., Romero, C., Ventura, S., Calders, T.: Drawbacks and solutions of applying association rule mining in learning management systems. In: Procs of the International Workshop on Applying Data Mining in e-Learning. (2007) 13–22
23. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques (2ed). Morgan Kaufmann (2005)
24. Merceron, A., Yacef, K.: Interestingness measures for associations rules in educational data. In de Baker, R.S.J., Barnes, T., Beck, J.E., eds.: EDM, www.educationaldatamining.org (2008) 57–66
25. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Comput. Surv. **38**(3) (2006)
26. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. European Journal of Operational Research **184**(2) (2008) 610–626
27. Borgelt, C.: Efficient implementations of apriori and eclat. In Goethals, B., Zaki, M.J., eds.: FIMI. Volume 90 of CEUR Workshop Proceedings., CEUR-WS.org (2003)
28. Scheffer, T.: Finding association rules that trade support optimally against confidence. In: In: 5th European Conference on Principles of Data Mining and Knowledge Discovery. (2001) 424–435
29. Zaki, M.J., Hsiao, C.J.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Transactions on Knowledge and Data Engineering **17**(4) (2005) 462–478
30. Kaytoue, M., Marcuola, F., Napoli, A., Szathmary, L., Villerd, J.: The Coron System. In Boumedjout, L., Valtchev, P., Kwuida, L., Sertkaya, B., eds.: 8th International Conference on Formal Concept Analsis (ICFCA) - Supplementary Proceedings. (2010) 55–58 (demo paper).
31. Balcázar, J.L.: Objective novelty of association rules: Measuring the confidence boost. In Yahia, S.B., Petit, J.M., eds.: EGC. Volume RNTI-E-19 of Revue des Nouvelles Technologies de l'Information., Cépaduès-Éditions (2010) 297–302
32. Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., Stamper, J.C., eds.: Procs of the 4th International Conference on Educational Data Mining, Eindhoven, The Netherlands, July 6-8, 2011. In Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., Stamper, J.C., eds.: EDM, www.educationaldatamining.org (2011)

# Attribute Exploration in a Fuzzy Setting

Cynthia Vera Glodeanu

Technische Universität Dresden,
01062 Dresden, Germany
Cynthia_Vera.Glodeanu@mailbox.tu-dresden.de

**Abstract.** Since its development attribute exploration was successfully applied in different fields, proving itself as a strong tool for knowledge acquisition. However, the disadvantage of this method is that it can be applied only for binary data. The growing number of applications of fuzzy logic in numerous domains including formal concept analysis makes it a natural wish to generalise the powerful technique of attribute exploration for fuzzy data. It is this paper's purpose to fulfill this wish and present a generalisation of attribute exploration to the fuzzy setting.

**Keywords:** Attribute exploration, knowledge discovery, fuzzy data

## 1 Introduction

Attribute exploration, as introduced in [1], is a tool for knowledge discovery by interactive determination of the implications holding between a given set of attributes. This method is especially useful when the examples, objects having the considered attributes, are infinite, hardly to enumerate or (partially) unknown. The user is asked whether some implications (the smallest set of implications from which all the other implications can be derived) hold. If the answer is affirmative, the next implication is considered. If, however, the implication is false, the user has to provide a counterexample. This method assumes that the user can distinguish between true and false implications and that he can provide counterexamples for false implications. The result of the attribute exploration is a set of implications which are true in general for the attributes under consideration and a representative set of examples for the whole theory.

Attribute exploration was successfully applied in different areas of research, for a brief overview see Subsection 2.1.

Formal fuzzy concept analysis goes back to [2, 3]. Its need arose by the fact that objects can have attributes with some truth degree instead of either having or not having them, reflecting that life is not just black and white. In such a fuzzy setting one can also be interested in the implications between attributes. These are formulas like $A \Rightarrow B$, where $A$ and $B$ are fuzzy sets of attributes. Such implications can be interpreted in fuzzy contexts, meaning that *if objects have the attributes from $A$ to at least the degree $a$, then they also have the attributes from $B$ to at least the degree $b$.* Attribute implications in a fuzzy setting were mainly developed and investigated by R. Belohlavek and V. Vychodil in

a series of papers, see for example [4, 5]. Due to the large number of fuzzy attribute implications in a formal fuzzy context, one is interested in the smallest set of attribute implications, the so-called *stem base*, from which all the other implications can be derived. The problem of determining the stem bases for the crisp case was studied in [6], see also [1]. However, in the fuzzy setting these stem bases need neither to be unique nor to exist. These facts split the problem of fuzzy attribute exploration into two cases, as we will see in Sections 3 and 4. We will show under which conditions an attribute exploration in a fuzzy setting can be performed successfully. The research in attribute exploration in the fuzzy setting is still at its beginning. We expect for it at least the same popularity in applications as its crisp variant has gained.

The article is structured as follows: In Section 2 we give short introductions to attribute exploration in the crisp setting, fuzzy sets and fuzzy logic, formal fuzzy concept analysis and implications in such a setting. Section 3 first presents how the stem bases can be computed in a fuzzy setting using the globalisation and afterwards it focuses on attribute exploration in such a setting. In Section 4 we treat the same subject as in the section before but this time we use a general hedge in the residuated lattice for the exploration. The last section contains concluding remarks and further topics of research.

## 2 Preliminaries

### 2.1 Crisp Attribute Exploration

We assume basic familiarities with Formal Concept Analysis and refer the reader to [1].

Attribute exploration ([1]) permits the interactive determination of the implications holding between the attributes of a given context. However, there are situations when the object set of a context is too large (possibly infinite) or difficult to enumerate. With the examples (possibly none) of our knowledge we build the object set of the context step-by-step. The stem base of this context is built stepwise and we are asked whether the implications of the base are true. If an implication holds, then it is added to the stem base. If however, an implication does not hold, we have to provide a counterexample. While performing an attribute exploration we have to be able to distinguish between true and false implications and to provide correct counterexamples for false implications. This is a crucial point since the algorithm is naive and will believe whatever we tell it. Once a decision was taken about the validity of an implication the choice cannot be reversed. Therefore, the counterexamples may not contradict the so-far confirmed implications. The procedure ends when all implications of the current stem base hold in general. This way we obtain an object set which is representative for the entire theory, theory which may also be infinite.

The following proposition justifies why we do not have to reconsider the already confirmed implications:

**Proposition 1.** *([1]) Let $\mathbb{K}$ be a context and $P_1, P_2, \ldots, P_n$ be the first $n$ pseudo-intents of $\mathbb{K}$ with respect to the lectic order. If $\mathbb{K}$ is extended by an object $g$ the*

*object intent $g^{\uparrow}$ of which respects the implications $P_i \to P_i^{\downarrow\uparrow}$, $i \in \{1, \ldots, n\}$, then $P_1, P_2, \ldots, P_n$ are also the lectically first $n$ pseudo-intents of the extended context.*

As mentioned in the introductory section, attribute exploration was successfully applied in both theoretical and practical research domains. On the one hand it facilitated the discovery of implications between properties of mathematical structures, see for example [7–9]. On the other hand it was also used in real-life scenarios, for instance in civil engineering ([10]), chemistry ([11]), information systems ([12]), etc.

The algorithm is implemented in different formal concept analytical tools, as for example in ConExp[1] and Conexp-clj[2].

There are also further variants of attribute exploration, for instance *attribute exploration with background knowledge* for the case that the user knows in advance some implications between the attributes that hold ([13, 14]). Another possibility is to perform *concept exploration* as presented in [15]. By replacing the implications with Horn clauses from predicate logic one obtains the so-called *rule exploration* developed in [16].

### 2.2 Fuzzy Sets and Fuzzy Logic

In this subsection we present some basics about fuzzy sets and fuzzy logic. The interested reader may find more details for instance in [17, 3].

A **complete residuated lattice with truth-stressing hedge** (shortly, a hedge) is an algebra $\mathbf{L} := (L, \wedge, \vee, \otimes, \to, ^*, 0, 1)$ such that: $(L, \wedge, \vee, 0, 1)$ is a complete lattice; $(L, \otimes, 1)$ is a commutative monoid; 0 is the least and 1 the greatest element; the adjointness property, i.e., $a \otimes b \leq c \Leftrightarrow a \leq b \to c$, holds for all $a, b, c \in L$. The hedge $^*$ is a unary operation on $L$ satisfying the following:

i) $a^* \leq a$,
ii) $(a \to b)^* \leq a^* \to b^*$,
iii) $a^{**} = a^*$,
iv) $\bigwedge_{i \in I} a_i^* = (\bigwedge_{i \in I} a_i)^*$,

for every $a, b, a_i \in L$ ($i \in I$). Elements of $L$ are called **truth degrees**, $\otimes$ and $\to$ are (truth functions of) "fuzzy conjunction" and "fuzzy implication". The hedge $^*$ is a (truth function of) logical connective "very true", see [17, 18]. Properties (i)-(iv) have natural interpretations, i.e., (i) can be read as "if $a$ is very true, then $a$ is true", (ii) can be read as "if $a \to b$ is very true and if $a$ is very true, then $b$ is very true", etc. From the mathematical point of view, the hedge operator is a special kernel operator which controls the size of the fuzzy concept lattice.

A common choice of $\mathbf{L}$ is a structure with $L = [0, 1]$, $\wedge$ and $\vee$ being minimum and maximum, $\otimes$ being a left-continuous t-norm with the corresponding $\to$. The

[1] http://conexp.sourceforge.net/
[2] http://daniel.kxpq.de/math/conexp-clj/

three most important pairs of adjoint operations on the unit interval are:

$$\text{Lukasiewicz: } a \otimes b := max(0, a+b-1) \text{ with } a \to b := min(1,\ 1-a+b),$$

$$\text{Gödel: } \quad a \otimes b := min(a,b) \text{ with } a \to b := \begin{cases} 1, & a \leq b \\ b, & a \not\geq b \end{cases},$$

$$\text{Product: } \quad a \otimes b := ab \text{ with } a \to b := \begin{cases} 1, & a \leq b \\ b/a, & a \not\geq b \end{cases}.$$

Typical examples for the hedge are the *identity*, i.e., $a^* := a$ for all $a \in L$, and the *globalization*, i.e., $a^* := 0$ for all $a \in L \setminus \{1\}$ and $a^* := 1$ if and only if $a = 1$.

Let $\mathbf{L}$ be the structure of truth degrees. A **fuzzy set** (**L-set**) $A$ in a universe $U$ is a mapping $A : U \to L$, $A(u)$ being interpreted as "the degree to which $u$ belongs to $A$". If $U = \{u_1, \ldots, u_n\}$, then $A$ can be denoted by $A = \{{}^{a_1}/_{u_1}, \ldots, {}^{a_n}/_{u_n}\}$ meaning that $A(u_i)$ equals $a_i$ for each $i \in \{1, \ldots, n\}$. Let $\mathbf{L}^U$ denote the collection of all fuzzy sets in $U$. The operations with fuzzy sets are defined component-wise. For example, the intersection of fuzzy sets $A, B \in \mathbf{L}^U$ is a fuzzy set $A \cap B$ in $U$ such that $(A \cap B)(u) = A(u) \wedge B(u)$ for each $u \in U$, etc. Binary fuzzy relations (**L**-relations) between $G$ and $M$ can be thought of as fuzzy sets in the universe $G \times M$. For $A, B \in \mathbf{L}^U$, the **subsethood degree** is defined as

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

which generalises the classical subsethood relation $\subseteq$. Therefore, $S(A,B)$ represents a degree to which $A$ is a subset of $B$. In particular, we write $A \subseteq B$ iff $S(A,B) = 1$.

### 2.3 Formal Fuzzy Concepts and Concept Lattices

In the following we give brief introductions to Formal Fuzzy Concept Analysis [2, 3].

A triple $(G, M, I)$ is called a **formal fuzzy context** if $I : G \times M \to L$ is a fuzzy relation between the sets $G$ and $M$ and $L$ is the support set of some residuated lattice. Elements from $G$ and $M$ are called **objects** and **attributes**, respectively. The fuzzy relation $I$ assigns to each $g \in G$ and each $m \in M$ the truth degree $I(g,m) \in L$ to which the object $g$ has the attribute $m$. For fuzzy sets $A \in \mathbf{L}^G$ and $B \in \mathbf{L}^M$ the **derivation operators** are defined by

$$A^{\uparrow}(m) := \bigwedge_{g \in G} (A(g)^* \to I(g,m)),\ B^{\downarrow}(g) := \bigwedge_{m \in M} (B(m) \to I(g,m)), \qquad (1)$$

for $g \in G$ and $m \in M$. Then, $A^{\uparrow}(m)$ is the truth degree of the statement "$m$ is shared by all objects from $A$" and $B^{\downarrow}(g)$ is the truth degree of "$g$ has all attributes from $B$". The operators $^{\uparrow}, ^{\downarrow}$ form a so-called Galois connection with hedges ([19]). A **formal fuzzy concept** is a tuple $(A, B) \in \mathbf{L}^G \times \mathbf{L}^M$ such that

$A^{\uparrow} = B$ and $B^{\downarrow} = A$. Then, $A$ is called the **(fuzzy) extent** and $B$ the **(fuzzy) intent** of $(A, B)$. We denote the set of all fuzzy concepts of a given context $(G, M, I)$ by $\mathfrak{B}(G^*, M, I)$. Concepts serve for classification. Consequently, the super- and subconcept relation plays an important role. A concept is called superconcept of another if it is more general, i.e., if it contains more objects. More formally, $(A_1, B_1)$ is a **subconcept** of $(A_2, B_2)$, written $(A_1, B_1) \leq (A_2, B_2)$, iff $A_1 \subseteq A_2$ (iff $B_1 \supseteq B_2$). Then, we call $(A_2, B_2)$ the **superconcept** of $(A_1, B_1)$. The set of all fuzzy concepts ordered by this concept order forms a complete fuzzy lattice (with hedge), the so-called **fuzzy concept lattice** which is denoted by $\underline{\mathfrak{B}}(G^*, M, I) := (\mathfrak{B}(G^*, M, I), \leq)$, see [20].

The *fuzzy lectic order* ([21]) is defined as follows: Let $L = \{l_0 < l_1 < \cdots < l_n\}$ be the support set of some residuated lattice. For $a := (i, j)$ and $b := (h, k)$, where $a, b \in M \times L$, we write

$$a \leq b :\Longleftrightarrow (i < h) \text{ or } (i = h \text{ and } l_j \geq l_k).$$

For $B \in \mathbf{L}^M$ and $(i, j) \in M \times L$ we define

$$B \oplus (i, j) := ((B \cap \{1, 2, \ldots, i-1\}) \cup \{{}^{a_j}/_i\})^{\downarrow\uparrow}.$$

Furthermore, for $B, C \in \mathbf{L}^M$ define

$$B <_{(i,j)} C :\Longleftrightarrow B \cap \{1, \ldots, i-1\} = C \cap \{1, \ldots, i-1\} \text{ and } B(i) < C(i) = a_j.$$

We say that $B$ is **lectically smaller** than $C$, written $B < C$, if $B <_{(i,j)} C$ for some $(i, j)$. As in the crisp case we have that $B^+ := B \oplus (i, j)$ is the least intent which is greater than a given $B$ with respect to $<$ and $(i, j)$ is the greatest with $B <_{(i,j)} B \oplus (i, j)$.

*Example 1.* Consider the formal fuzzy context $(G, M, I)$ given in Figure 1. Using the Lukasiewicz logic with the identity as hedge we obtain 15 formal fuzzy concepts. For example $(\{Mo, T, {}^{0.5}/_W\}, \{c, r\})$ is a fuzzy concept. We could name it the concept of *cold and rainy days* because of its intent. Then, Monday, Tuesday and partially Wednesday belong to this concept, i.e., they are cold and rainy days. Another example is $(\{{}^{0.5}/_W, Th, F\}, \{w\})$ which corresponds to *warm days*. Yet another example are the *warm and partially rainy days* given by $(\{{}^{0.5}/_W, Th, {}^{0.5}/_F\}, \{w, {}^{0.5}/_r\})$. The fuzzy concept lattice is displayed on the left side in Figure 2. For better legibility we did not use all the labels. Using the globalisation instead of the identity, we obtain 10 formal fuzzy concepts which are displayed on the right in Figure 2. The concepts obtained through the globalisation need not be a subset of those obtained with the identity. In this example this case does not appear. Using the Gödel structure one obtains 13 concepts with the identity and 10 with the globalisation.

### 2.4 Fuzzy Implications and Non-redundant Bases

As already mentioned, fuzzy implications were studied in a series of papers by R. Belohlavek and V. Vychodil, for instance in [4, 5].

| | warm (w) | cold (c) | rainy (r) |
|---|---|---|---|
| Monday (Mo) | 0 | 1 | 1 |
| Tuesday (T) | 0 | 1 | 1 |
| Wednesday (W) | 0.5 | 0.5 | 1 |
| Thursday (Th) | 1 | 0 | 0.5 |
| Friday (F) | 1 | 0 | 0 |

**Fig. 1.** Example of a fuzzy formal context

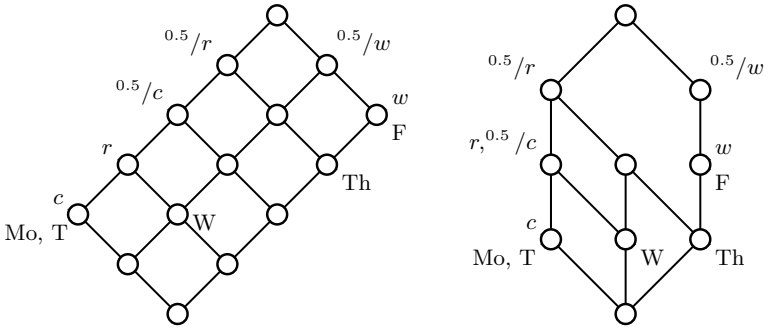



**Fig. 2.** Formal fuzzy concept lattices

A **fuzzy attribute implication** (over the attribute set $M$) is an expression $A \Rightarrow B$, where $A, B \in \mathbf{L}^M$. The verbal meaning of $A \Rightarrow B$ is: "if it is (very) true that an object has all attributes from $A$, then it also has all attributes from $B$". The notions "being very true", "to have an attribute", and logical connective "if-then" are determined by the chosen $\mathbf{L}$. For a fuzzy set $N \in \mathbf{L}^M$ of attributes, the degree $||A \Rightarrow B||_N \in L$ to which $A \Rightarrow B$ is valid in $N$ is defined as

$$||A \Rightarrow B||_N := S(A, N)^* \to S(B, N).$$

If $N$ is the fuzzy set of all attributes of an object $g$, then $||A \Rightarrow B||_N$ is the truth degree to which $A \Rightarrow B$ holds for $g$. For a set $\mathcal{N} \subseteq \mathbf{L}^M$, the degree $||A \Rightarrow B||_{\mathcal{N}} \in L$ to which the implication $A \Rightarrow B$ holds in $\mathcal{N}$ is defined by

$$||A \Rightarrow B||_{\mathcal{N}} := \bigwedge_{N \in \mathcal{N}} ||A \Rightarrow B||_N.$$

For a fuzzy context $(G, M, I)$, let $I_g \in \mathbf{L}^M$ $(g \in G)$ be a fuzzy set of attributes such that $I_g(m) = I(g, m)$ for each $m \in M$. Clearly, $I_g$ corresponds to the row labelled $g$ in $(G, M, I)$. The degree $||A \Rightarrow B||_{(G,M,I)} \in L$ to which $A \Rightarrow B$ holds in (each row of) $\mathbb{K} = (G, M, I)$ is defined by

$$||A \Rightarrow B||_{\mathbb{K}} = ||A \Rightarrow B||_{(G,M,I)} := ||A \Rightarrow B||_{\mathcal{N}},$$

where $\mathcal{N} := \{I_g \mid g \in G\}$. Denote by

$$\mathrm{Int}(G^*, M, I) := \{B \in \mathbf{L}^M \mid (A, B) \in \mathfrak{B}(G^*, M, I) \text{ for some } A\}$$

the set of all intents of $\mathfrak{B}(G^*, M, I)$. Since $N \in \mathbf{L}^M$ is the intent of some concept if and only if $N = N^{\downarrow\uparrow}$, we have $\mathrm{Int}(G^*, M, I) = \{N \in \mathbf{L}^M \mid N = N^{\downarrow\uparrow}\}$. The degree $||A \Rightarrow B||_{\mathfrak{B}(G^*,M,I)} \in L$ to which $A \Rightarrow B$ holds in (the intents of) $\mathfrak{B}(G^*, M, I)$ is defined by

$$||A \Rightarrow B||_{\mathfrak{B}(G^*,M,I)} := ||A \Rightarrow B||_{\mathrm{Int}(G^*,M,I)}.$$

**Lemma 1.** *([22]) Let $(G, M, I)$ be a fuzzy context. Then,*

$$||A \Rightarrow B||_{(G,M,I)} = ||A \Rightarrow B||_{\mathfrak{B}(G^*,M,I)} = S(B, A^{\downarrow\uparrow})$$

*for each fuzzy attribute implication $A \Rightarrow B$.*

*Example 2.* Consider once again the fuzzy context given in Figure 1. Using the Lukasiewicz logic and the globalisation as the hedge we have $||c \Rightarrow r||_{(G,M,I)} = 1$, i.e., this is a true implication. However, in the fuzzy case, there are implications which are valid to a certain degree different from 1, for instance we have the implication $||c \Rightarrow \{^{0.5}/w, r\}||_{(G,M,I)} = 0.5$. We obtain the same truth value for these implications also by using the identity. Consider the Gödel logic with the globalisation. For example, we have the implication $||w, r \Rightarrow c||_{(G,M,I)} = 1$ but using the identity this implication holds with the truth value 0. This is due to the fact that we have $\{w, r\}^{\downarrow\uparrow} = \{w, r, c\}$ with the globalisation and $\{w, r\}^{\downarrow\uparrow} = \{w, r\}$ with the identity.

Due to the large number of implications in a fuzzy and even in a crisp formal context, one is interested in the stem base of the implications. The **stem base** is a set of implications which is non-redundant and complete. The problem for the fuzzy case was studied in [5, 22, 23]. Neither the existence nor the uniqueness of the stem base for a given fuzzy context is guaranteed in general. How these problems can be overcome is the topic of the rest of this subsection. For a more detailed description we refer the reader to the papers cited above.

Let $T$ be a set of fuzzy attribute implications. A fuzzy attribute set $N \in \mathbf{L}^M$ is called a **model** of T if $||A \Rightarrow B||_N = 1$ for each $A \Rightarrow B \in T$. The set of all models of $T$ is denoted by $Mod\,(T)$, i.e.,

$$Mod\,(T) := \{N \in \mathbf{L}^M \mid N \text{ is a model of } T\}.$$

The degree $||A \Rightarrow B||_T \in L$ to which $A \Rightarrow B$ **semantically follows** from $T$ is defined by $||A \Rightarrow B||_T := ||A \Rightarrow B||_{Mod(T)}$. $T$ is called **complete** (in $(G, M, I)$) if $||A \Rightarrow B||_T = ||A \Rightarrow B||_{(G,M,I)}$ for each $A \Rightarrow B$. If $T$ is complete and no proper subset of T is complete, then T is called a **non-redundant basis**.

**Theorem 1.** *([5]) $T$ is complete iff $Mod\,(T) = Int(G^*, M, I)$.*

As in the crisp case the stem base of a given fuzzy context can be obtained through the pseudo-intents.

**Definition 1.** *$\mathcal{P} \subseteq \mathbf{L}^M$ is called a **system of pseudo-intents** if for each $P \in \mathbf{L}^M$ we have:*

$$P \in \mathcal{P} \iff (P \neq P^{\downarrow\uparrow} \text{ and } ||Q \Rightarrow Q^{\downarrow\uparrow}||_P = 1 \text{ for each } Q \in \mathcal{P} \text{ with } Q \neq P).$$

For each $(G, M, I)$ there exists a unique system of pseudo-intents, if $^*$ is the globalisation and $M$ is finite (this does not hold for the other hedges in general).

**Theorem 2.** *([22]) $T := \{P \Rightarrow P^{\downarrow\uparrow} \mid P \in \mathcal{P}\}$ is complete and non-redundant. If $^*$ is the globalization, then $T$ is unique and minimal.*

## 3 Fuzzy Attribute Exploration with Globalisation

Attribute exploration is a very powerful tool. However, its theoretical basis lies in Proposition 1 which represents its key to success. Thus, the crucial step is to generalise this proposition to the fuzzy setting. After developing the theoretical ingredients for a successful attribute exploration in a fuzzy setting, we turn our attention to its practical parts. First, we develop an appropriate algorithm for this technique and afterwards illustrate the method by an example.

In case we choose for $^*$ the globalisation, then the formalisation of pseudo-intents from Definition 1 becomes: $\mathcal{P} \subseteq \mathbf{L}^M$ is a system of pseudo-intents if

$$P \in \mathcal{P} \iff (P \neq P^{\downarrow\uparrow} \text{ and } Q^{\downarrow\uparrow} \subseteq P \text{ for each } Q \in \mathcal{P} \text{ with } Q \subsetneq P). \qquad (2)$$

**Theorem 3.** *([22]) Let $\mathbf{L}$ be a residuated lattice with globalization. Then, for each $(G, M, I)$ with finite $M$ there is a unique system of pseudo-intents $\mathcal{P}$ given by* (2).

For $Z \in \mathbf{L}^M$ we put

$$\begin{aligned} Z^{T^*} &:= Z \cup \bigcup \{B \otimes S(A, Z)^* \mid A \Rightarrow B \in T \text{ and } A \neq Z\}, \\ Z^{T_0^*} &:= Z, \\ Z^{T_n^*} &:= (Z^{T_{n-1}^*})^{T^*}, \text{ for } n \geq 1, \end{aligned}$$

where $B \otimes S(A, Z)^*$ is computed component-wise, and we define an operator $cl_{T^*}$ on $\mathbf{L}$-sets in $M$ by

$$cl_{T^*}(Z) := \bigcup_{n=0}^{\infty} Z^{T_n^*}.$$

**Theorem 4.** *([5]) If $^*$ is the globalisation, then $cl_{T^*}$ is an $\mathbf{L}^*$-closure operator and*

$$\{cl_{T^*}(Z) \mid Z \in \mathbf{L}^M\} = \mathcal{P} \cup Int(X^*, Y, I).$$

According to this theorem, if $^*$ is the globalisation, then we can obtain all intents and all pseudo-intents of a given fuzzy context by computing the fixed points of $cl_{T^*}$. In [5] an algorithm for the computation of all intents and all pseudo-intents in lectic order was proposed. Therefore, the following result holds:

**Proposition 2.** *Let* $\mathbf{L}$ *be a residuated lattice with hedge and let* $^*$ *be the globalisation. Further, let* $\mathcal{P}$ *be the unique system of pseudo-intents of the fuzzy context* $(G, M, I)$ *such that* $P_1, P_2, \ldots, P_n \in \mathcal{P}$ *are the first* $n$ *pseudo-intents in* $\mathcal{P}$ *with respect to the lectic order. If* $(G, M, I)$ *is extended by an object* $g$ *the object intent* $g^\uparrow$ *of which respects the implications* $P_i \to P_i^{\downarrow\uparrow}$, $i \in \{1, \ldots, n\}$, *then* $P_1, P_2, \ldots, P_n$ *remain the lectically first* $n$ *pseudo-intents of the extended context.*

*Proof.* Easy, by induction on the number of pseudo-intents in $\mathcal{P}$.

With this result we are able to generalise the attribute exploration algorithm to the fuzzy setting, as displayed below.

(1) $\mathcal{L} := \varnothing$; $A := \varnothing$
(2) **if** $(A = A^{\downarrow\uparrow})$
(3) **then add** $A$ to $\mathrm{Int}(\mathbb{K})$
(4) **else** Ask expert whether $||A \Rightarrow A^{\downarrow\uparrow}||_{\mathbb{K}} = 1$
(5) If yes, **add** $A \Rightarrow A^{\downarrow\uparrow}$ to $\mathcal{L}$
(6) **else** ask for counterexample $g$ and **add** it to $\mathbb{K}$
(7) **end if**
(8) **do while** $(A \neq M)$
(9) **for** $i = n, \ldots, 1$ **and for** $l = \max L, \ldots, \min L$ **with** $A(i) < l$ **do**
(10) $B :=cl_{T^*}(A)$
(11) **if** $(A \searrow i = B \searrow i)$ **and** $(A(i) < B(i))$ **then**
(12) $A := B$
(13) **if** $(A = A^{\downarrow\uparrow})$
(14) **then add** $A$ to $\mathrm{Int}(\mathbb{K})$
(15) **else** Ask expert whether $||A \Rightarrow A^{\downarrow\uparrow}||_{\mathbb{K}} = 1$
(16) If yes, **add** $A \Rightarrow A^{\downarrow\uparrow}$ to $\mathcal{L}$
(17) **else** ask for counterexample $g$ and **add** it to $\mathbb{K}$
(18) **end if**
(19) **end if**
(20) **end for**
(21) **end do**

**Fig. 3.** Algorithm for attribute exploration with globalisation

The first intent or pseudo intent is the empty set. If it is an intent, add it to the set of intents of the context. Otherwise, ask the expert whether the implication is true in general. If so, add this implication to the stem base else ask for a counterexample and add it to the context (line $2 - 6$). Until $A$ is different from the whole attribute set, repeat the following steps: Search for the largest attribute $i$ in $M$ with its largest value $l$ such that $A(i) < l$. For this attribute compute its closure with respect to the $cl_{T^*}$-closure operator and check whether the result is the lectically next intent or pseudo-intent (line $9 - 12$). Thereby,

$A \searrow i := A \cap \{1, \ldots, i-1\}$. If the result is an intent, add it to the set of intents (line $13-14$), otherwise ask the user whether the implication provided by the pseudo-intent holds. If the implication holds, add it to the stem base otherwise ask the user for a counterexample (line $15-17$).

The algorithm generates interactively the stem base of the formal fuzzy context. As in the crisp case we enumerate the intents and pseudo-intents in the lectic order. Hence, we go through the list of all such elements. Due to Proposition 2 we are allowed to extend the context by objects whose object intents respect the already confirmed implications. This way, the pseudo-intents already used in the stem base do not change. Hence, the algorithm is sound and correct.

*Example 3.* We want to explore the size and distance of the planets. We include some of them into the object set and obtain the context given in Figure 4. In this example we will be using the Lukasiewicz logic with the globalisation as hedge.

| | small (s) | large (l) | far (f) | near (n) |
|---|---|---|---|---|
| Earth | 1 | 0 | 0 | 1 |
| Mars | 1 | 0 | 0.5 | 1 |
| Pluto | 1 | 0 | 1 | 0 |

**Fig. 4.** Initial context

We start the attribute exploration. The first pseudo-intent is $\varnothing$ and we are asked

*All objects have the attribute s to degree* $1$*?*

This is of course not true and we provide a counterexample:

| | small (s) | large (l) | far (f) | near (n) |
|---|---|---|---|---|
| Jupiter | 0 | 1 | 1 | 0.5 |

The next pseudo-intent is $n$ and we are asked

*Objects having attribute* $n$ *to degree* $1$ *also have attribute* $s$ *to degree* $1$*?*

This is a true implication and we confirm it. The next pseudo-intent is $\{f, {}^{0.5}/n\}$ which yields the following question:

*Objects having attribute* $f$ *and* $n$ *to degree* $1$ *and* $0.5$*, respectively, also have attribute* $l$ *to degree* $1$*?*

This is a true implication and we confirm it. The algorithm proceeds with

*Objects having attribute* $l$ *to degree* $0.5$ *also have the attributes* $l, f, n$ *to degree* $1, 1, 0.5$*, respectively?*

This implication is not true for our planet system and we give a counterexample:

| | small (s) | large (l) | far (f) | near (n) |
|---|---|---|---|---|
| Uranus | 0.5 | 0.5 | 1 | 0 |

The following four implications are true, so we will confirm them:

$$
\begin{aligned}
&{}^{0.5}/l \Rightarrow f,\\
&l, f \Rightarrow^{0.5} /n,\\
&{}^{0.5}/s, {}^{0.5}/n \Rightarrow s, n,\\
&s, {}^{0.5}/l, f \Rightarrow l, n.
\end{aligned}
$$

And the attribute exploration has stopped. Now we have an extended formal fuzzy context, namely the one containing Jupiter and Uranus besides the objects given in Figure 4. Note that we did not have to include all the planets into the object set, just a representative part of them. The other planets with their attributes are displayed in Figure 5. These objects contain just redundant information and the knowledge provided by them is already incorporated into the stem base of the extended context.

| | small (s) | large (l) | far (f) | near (n) |
|---|---|---|---|---|
| Mercury | 1 | 0 | 0 | 1 |
| Venus | 1 | 0 | 0 | 1 |
| Saturn | 0 | 1 | 1 | 0.5 |
| Neptune | 0.5 | 0.5 | 1 | 0 |

**Fig. 5.** Superfluous planets

## 4 Fuzzy Attribute Exploration with General Hedges

As the title of this section suggests, we will now turn our attention to attribute exploration with general hedges. After introducing the necessary background information, we will focus on the exploration. As it turns out, there are several obstacles that make a straight-forward generalisation of attribute exploration in such a setting impossible. At the end of the section we will discuss which approaches may lead to a successful exploration. However, it is also an open question whether an exploration in such a setting is desirable.

The computation of the systems of pseudo-intents for general hedges was studied in [23]. For a fuzzy context $(G, M, I)$ we compute the following:

$$V := \{P \in \mathbf{L}^M \mid P \neq P^{\downarrow\uparrow}\}, \tag{3}$$

$$E := \{(P, Q) \in V \times V \mid P \neq Q \text{ and } ||Q \Rightarrow Q^{\downarrow\uparrow}||_P \neq 1\}. \tag{4}$$

In case of a non-empty $V$, $\mathbf{G} := (V, E \cup E^{-1})$ is a graph. For $Q \in V$, $\mathcal{P} \subseteq V$ define the following subsets of $V$:

$$Pred\,(Q) := \{P \in V \mid (P, Q) \in E\},$$
$$Pred\,(\mathcal{P}) := \bigcup_{Q \in \mathcal{P}} Pred\,(Q).$$

Described verbally, $Pred\,(Q)$ is the set of all elements from $V$ which are predecessors of $Q$ (in $E$). $Pred\,(\mathcal{P})$ is the set of all predecessors of any $Q \in \mathcal{P}$.

We will compute the systems of pseudo-intents through maximal independent sets. Therefore, the following result is useful:

**Lemma 2.** *([23]) Let $\varnothing \neq \mathcal{P} \subseteq \mathbf{L}^M$. If $V \setminus \mathcal{P} = Pred\,(\mathcal{P})$, then $\mathcal{P}$ is a maximal independent set in $\mathbf{G}$.*

The next theorem characterises the systems of pseudo-intents of a fuzzy context using general hedges:

**Theorem 5.** *([23]) Let $\mathcal{P} \subseteq \mathbf{L}^M$. $\mathcal{P}$ is a system of pseudo-intents if and only if $V \setminus \mathcal{P} = Pred(\mathcal{P})$.*

It is well-known that the maximal independent sets of a graph can be efficiently enumerated in lexicographic order with only polynomial delay between the output of two successive independent sets ([24]). In [25] it was shown that the pseudo-intents cannot be enumerated in lexicographic order with polynomial delay unless P = NP. These two results do not contradict each other because they address different issues. The first one in encountered when we enumerate the maximal independent sets of the graph $\mathbf{G}$ which is the input of the corresponding algorithm. These sets correspond to the systems of pseudo-intents. Whereas the result from [25] is for the globalisation and takes as input a formal context enumerating its pseudo-intents.

In the following we will exemplify the computation of the systems of pseudo-intents. Afterwards, we illustrate how an attribute exploration with general hedge could be performed.

*Example 4.* We start with a very simple example. Let $(\{g\}, \{a, b\}, I)$ be the formal fuzzy context with $I(g, a) = 0.5$ and $I(g, b) = 0$. Further, we use the three-element Lukasiewicz chain with $^*$ being the identity. First, we compute $V$ as given by (3) and obtain

$$V = \{\{{}^{0.5}/a, {}^{0.5}/b\}, \{{}^{0.5}/b\}, \{\}, \{{}^{0.5}/a, b\}, \{b\}, \{a\}\}.$$

Afterwards, we compute the binary relation $E$ as given by (4) which is displayed in Figure 6. Considering the undirected diagram of Figure 6 we obtain the graph $\mathbf{G}$. There, we have four maximal independent sets, namely

$$\mathcal{P}_1 = \{\{\}, \{{}^{0.5}/a, b\}, \{a\}\},$$
$$\mathcal{P}_2 = \{\{{}^{0.5}/b\}, \{a\}\},$$
$$\mathcal{P}_3 = \{\{b\}, \{a\}\},$$
$$\mathcal{P}_4 = \{\{{}^{0.5}/a, {}^{0.5}/b\}, \{a\}\}.$$

$\mathcal{P}_1$ and $\mathcal{P}_3$ do not satisfy the condition of Theorem 5 and are therefore not

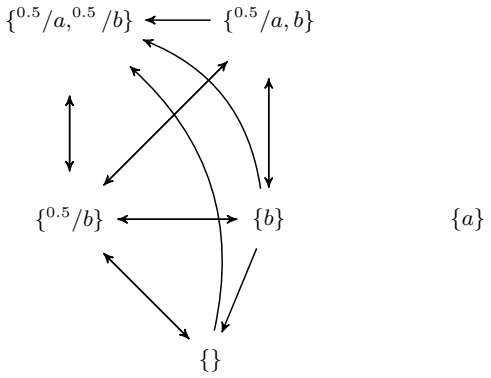



**Fig. 6.** Binary relation $E$ for $(G, M, I)$

systems of pseudo-intents. $\mathcal{P}_2$ and $\mathcal{P}_4$ do satisfy this condition and hence they are systems of pseudo-intents yielding the stem bases displayed in Figure 7.

| $\mathcal{T}_2$ | | $\mathcal{T}_4$ | |
|---|---|---|---|
| (1) | $^{0.5}/b \Rightarrow a$ | (3) | $^{0.5}/a, ^{0.5}/b \Rightarrow a$ |
| (2) | $a \Rightarrow^{0.5}/b$ | (4) | $a \Rightarrow^{0.5}/b$ |

**Fig. 7.** Stem bases

Now we could start an attribute exploration, for instance in $\mathcal{T}_2$. The algorithm would ask us:

*Objects having attribute b to degree* 0.5 *also have attribute a to degree* 1*?*

Let us answer this question affirmatively. The next question is:

*Objects having attribute a to degree* 1 *also have attribute b to degree* 0.5*?*

We deny this implication and provide a counterexample, namely the object $h$ with $I(h, a) = 1$ and $I(h, b) = 0$. This counterexample obviously respects the already confirmed implication so the context is extended by the new object $h$. For this extended context we can compute the sets $V$ and $E$. The binary relation

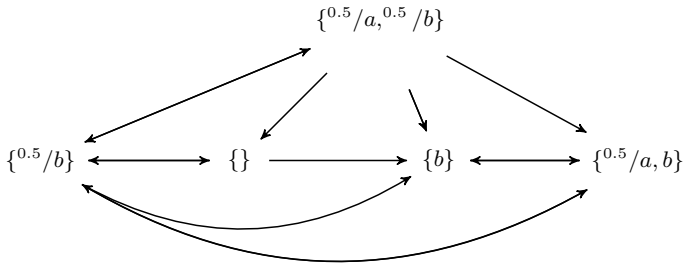



**Fig. 8.** Binary relation $E$ for the extended context

$E$ for the extended context is given in Figure 8. From this graph we obtain four maximal independent sets, three of which form systems of pseudo-intents. The stem bases which they induce are displayed in Figure 9. At the beginning we

| $\mathcal{T}_2^{\mathrm{I}}$ | $\mathcal{T}_2^{\mathrm{II}}$ | $\mathcal{T}_2^{\mathrm{III}}$ |
|---|---|---|
| (5) $^{0.5}/b \Rightarrow a$ | (6) $\{\} \Rightarrow^{0.5}/a$<br>(7) $^{0.5}/a, b \Rightarrow b$ | (8) $\{b\} \Rightarrow a$ |

**Fig. 9.** Stem bases of the extended context

have confirmed implication (1) from Figure 7. However, this implication is now not present any more in the stem bases $\mathcal{T}_2^{\mathrm{II}}$ and $\mathcal{T}_2^{\mathrm{III}}$. This is also reflected in the stem base $\mathcal{T}_4$. Even though the counterexample respects implication (3), the pseudo-intent belonging to this implication also disappears.

Concluding, by extending the context with objects which respect the already confirmed implications, the latter may disappear from the stem base of the extended context. Hence, we do not have an analogon of Proposition 2 for general hedges.

The attribute exploration with general hedges raises a lot of questions and open problems. First of all it is unclear whether such an exploration is desirable. We have more than one stem base for a context. These bases are equally powerful with respect to their expressiveness. The major problem however is how to perform an attribute exploration successfully. It is an open problem how to enumerate the pseudo-intents obtained by general hedges such that the already confirmed implication still remain in the stem base of the extended context. One

could for instance make some constraints on the counterexamples. However, such an approach is not in the spirit of attribute exploration.

## 5 Conclusion

We presented a generalisation of attribute exploration to the fuzzy setting. The problem is two-sided. If one uses the globalisation in the residuated lattice, the stem base is unique. For such a setting the results regarding the exploration from the crisp case can be transferred without problems and one can perform successfully an attribute exploration with attributes having fuzzy values. Using hedges different from the globalisation one obtains more than one system of pseudo-intents. This alone would not cause such a big problem. The major difficulty comes with the fact that the already confirmed pseudo-intents are not necessarily pseudo-intents of the extended context. This is therefore an open problem, how to perform an attribute exploration using a general hedge.

In the future we will focus on the problem regarding the general hedge and on extensions of this method, as for instance on fuzzy attribute exploration with background knowledge. There, the user can enter in advance some implications which he/she knows to hold between the attributes. Using such background knowledge one usually has to provide less examples and answer to fewer questions.

We are expecting that the method will have many practical applications, as its crisp variant has. Therefore, we will also focus on applications using attribute exploration in a fuzzy setting.

## References

1. Ganter, B., Wille, R.: Formale Begriffsanalyse: Mathematische Grundlagen. Springer (1996)
2. Pollandt, S.: Fuzzy Begriffe. Springer Verlag, Berlin Heidelberg New York (1997)
3. Belohlávek, R.: Fuzzy Relational Systems: Foundations and Principles. Volume 20 of IFSR Int. Series on Systems Science and Engineering. Kluwer Academic/Plenum Press (2002)
4. Belohlávek, R., Vychodil, V.: Attribute implications in a fuzzy setting. In: ICFCA. (2006) 45–60
5. Belohlávek, R., Chlupová, M., Vychodil, V.: Implications from data with fuzzy attributes. In: AISTA 2004 in Cooperation with the IEEE Computer Society Proceedings. (2004)
6. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de donnes binaires. Math. Sci. Humaines **24**(95) (1986) 5–18
7. Sacarea, C.: Towards a theory of contextual topology. PhD thesis, TH Darmstadt, Aachen (2001)
8. Kwuida, L., Pech, C., Reppe, H.: Generalizations of boolean algebras. an attribute exploration. Math. Slovaca **56**(2) (2006) 145–165
9. Revenko, A., Kuznetsov, S.: Attribute exploration of properties of functions on ordered sets. In: Proc. CLA 2010. (2010) 313–324

10. Eschenfelder, D., Kollewe, W., Skorsky, M., Wille, R.: Ein Erkundungssystem zum Baurecht: Methoden der Entwicklung eines TOSCANA-Systems. Volume 2036. Techn. Univ., FB 4, Darmstadt (Januar 1999) Ersch. ebenf. in: Begriffliche Wissensverarbeitung: Methoden und Anwendungen. Hrsg.: G. Stumme, R. Wille. - Berlin, Heidelberg (u.a.): Springer, 2000. S. 254-272.
11. Bartel, H.G., Nofz, M.: Exploration of nmr data of glasses by means of formal concept analysis. Chemom. Intell. Lab. Syst. **36** (1997) 53–63
12. Stumme, G.: Acquiring expert knowledge for the design of conceptual information systems. In Fensel, D., Studer, R., eds.: EKAW. Volume 1621 of Lecture Notes in Computer Science., Springer (1999) 275–290
13. Ganter, B.: Attribute exploration with background knowledge. Theor. Comput. Sci. **217**(2) (1999) 215–233
14. Stumme, G.: Attribute exploration with background implications and exceptions. In Bock, H.H., Polasek, W., eds.: Data Analysis and Information Systems. Statistical and Conceptual approaches. Proc. GfKl'95. Studies in Classification, Data Analysis, and Knowledge Organization 7, Heidelberg, Springer (1996) 457–469
15. Wille, R.: Bedeutungen von Begriffsverbänden. In Ganter, B., Wille, R., Wolff, K.E., eds.: Beiträge zur Begriffsanalyse. B.I.–Wissenschaftsverlag, Mannheim (1987) 161–211
16. Zickwolff, M.: Rule exploration: first order logic in formal concept analysis. Technische Hochschule Darmstadt. (1991)
17. Hájek, P.: The Metamathematics of Fuzzy Logic. Kluwer (1998)
18. Hájek, P.: On very true. Fuzzy Sets and Systems **124**(3) (2001) 329–333
19. Belohlávek, R., Funioková, T., Vychodil, V.: Galois connections with hedges. In Liu, Y., Chen, G., Ying, M., eds.: Eleventh International Fuzzy Systems Association World Congress,. Fuzzy Logic, Soft Computing & Computational Intelligence, Tsinghua University Press and Springer (2005) 1250–1255
20. Belohlávek, R., Vychodil, V.: Fuzzy concept lattices constrained by hedges. JACIII **11**(6) (2007) 536–545
21. Belohlávek, R.: Algorithms for fuzzy concept lattices. In: Proc. Fourth Int. Conf. on Recent Advances in Soft Computing. (2002) 200–205
22. Belohlávek, R., Vychodil, V.: Fuzzy attribute logic: attribute implications, their validity, entailment, and non-redundant basis. In Liu, Y., Chen, G., Ying, M., eds.: Eleventh International Fuzzy Systems Association World Congress,. Volume 1 of Fuzzy Logic, Soft Computing & Computational Intelligence., Tsinghua University Press and Springer (2005) 622–627
23. Belohlávek, R., Vychodil, V.: Fuzzy attribute implications: Computing non-redundant bases using maximal independent sets. In: Australian Conference on Artificial Intelligence. (2005) 1126–1129
24. Johnson, D.S., Yannakakis, M., Papadimitriou, C.H.: On generating all maximal independent sets. Information Processing Letters **27**(3) (1988) 119–123
25. Distel, F., Sertkaya, B.: On the complexity of enumerating pseudo-intents. Discrete Applied Mathematics **159**(6) (2011) 450–466

# On Open Problem - Semantics of the Clone Items

Juraj Macko

Dept. Computer Science
Palacky University, Olomouc
17. listopadu 12, CZ-77146 Olomouc
Czech Republic
email: {juraj.macko}@upol.cz

**Abstract.** There was presented a list of open problems in the Formal Concept Analysis area at the conference ICFCA 2006. The problem number seven deals with the semantics of the clone items. Namely, for whom can clone items make sense and for whom can make sense the item, which can cause, that clones disappear in the collection of itemsets. In this paper we propose the semantics behind clone items with the couple of examples. Definition of the clone items is very strict and theirs use could be very limited in the real datasets. We introduce method, how to deal with items, which properties are very near to the clones. We also have a look on the items, which causes the disappearing of the clones, or decrease (increase) the degree of property "to be clone". In the experiment part we analyze some known datasets from the clone items point of view. The results bring a couple of new questions for the future research.

**Keywords:** formal concept analysis, clone items

## 1 Introduction

This paper is structured as follows: The first part, which is actually cited from the source, where the problem were defined [2] describes and defines the whole problem - the semantics of the clone items. In the second part is proposed the semantics of the clone items by putting the problem into the other point of view. There is also a discussion here, about another possible definitions of the clones as presented in [1]. In this part three comprehensive examples can be found. The third part tries to set a quite new approach to the clone items. The attributes, which are not clones, but they have properties very close to clones are considered. A nearly clones are defined. In this part some results from the introductory experiments about the clones and nearly clones are presented. Finally, the conclusion is divided in two parts - conclusion of defined problem and conclusion of other proposed issues.

## 2 The Problem Setting

The proposed problem of the semantics of the clone items were proposed and defined in [2] as follows: Let $J$ be a set of items $x_1, ..., x_{|J|}$, let $\mathcal{F}$ be a collection of subsets of $J$ and let $\varphi_{a,b}$ be the mapping $\varphi_{a,b} : 2^J \to 2^J$ defined by following formula:

$$X \to \varphi_{a,b}(X) = \begin{cases} (X \backslash \{a\}) \cup \{b\} \text{ if } b \notin X \text{ and } a \in X \\ (X \backslash \{b\}) \cup \{a\} \text{ if } a \notin X \text{ and } b \in X \\ X \text{ elsewhere} \end{cases}$$

It means swapping items $a$ and $b$, which are called clone items in $\mathcal{F}$ iff for any $F \in \mathcal{F}$, we have $\varphi_{a,b}(F) \in \mathcal{F}$. A Clone-free collection is, if it does not contain any clone items.

Let $(X, Y, I)$ be a formal context such that attributes $a \in Y$ and $b \in Y$ are not clones. Consider the formal sub-context $(X, Z, I)$, where $Z \subset Y$, such that $a$ and $b$ are clone in $(X, Z, I)$. Let $c \in Y \backslash Z$ such that $a$ and $b$ are no longer clone in $(X, Z \cup \{c\}, I)$. Attributes $a$ and $b$ has symmetrical behaviour in $(X, Z, I)$, but this behaviour is lost when we add the attribute $c$ to the formal context. The following question are asked:

1. Does such symmetrical behaviour of $a$ and $b$ make sense for someone?
2. Does it make the sense, that such symmetrical behaviour disappears, when the attribute $c$ is added?
3. What is semantics behind the attributes $a$, $b$, and $c$?

## 3 Semantics behind Clones

### 3.1 Semantics behind Clones - Auxiliary Formal Definitions

The collection of itemsets will be defined as a formal context $(X, Y, I)$, where $X$ is a set of objects and $Y$ is a set of attributes. Objects and attributes are related by $I \subseteq X \times Y$, which means, that the object $x \in X$ has the attribute $y \in Y$. For $A \subseteq X$, $B \subseteq Y$ and formal context $(X, Y, I)$ we define operators

$$A^{\uparrow_I} = \{y \in Y \mid \text{ for each } x \in X : \langle x, y \rangle \in I\}$$
$$B^{\downarrow_I} = \{x \in X \mid \text{ for each } y \in Y : \langle x, y \rangle \in I\}$$

The two given attributes $a, b \in Y$ will be investigated, whether are clones or not. For this purpose the **pivot table** will be defined as the relation $R \subseteq P \times \mathcal{N}$, where $P = \{a, b\} \subseteq Y$ and $\mathcal{N}$ is a set of all $N_j$, where $j \in [1; |\mathcal{N}|]$. $N_j \in \mathcal{N}$ represents the set of attributes $N_j = \{x\}^{\uparrow_I} \cap (Y \backslash P)$ for each $x \in X$ such that $\{a, b\} \cap \{x\}^{\uparrow_I} \neq \emptyset$ and $\{a, b\} \not\subseteq \{x\}^{\uparrow_I}$. The investigated attributes $a, b \in P \subseteq Y$ will be named the **pivot attributes** and all other considered attributes, hence $n \in \bigcup_{j=1}^{|\mathcal{N}|} N_j$, we denote as the **non-pivot attributes**. $N_j$ is a set generated by pivot attributes (or shortly the **generated set**). The pivot table has two

rows. The "cross" $\times$ in pivot table will represent the fact, that in the formal context there exists at least one row, where the investigated attribute $a$ (or $b$ respectively) appears together with the attributes in the particular $N_j$. Formally,

$\langle a, N_j \rangle \in R$ iff in context $(X, Y, I)$ exists $x \in X$ such that $x^{\uparrow_I} = \{a\} \cup N_j$,
$\langle b, N_j \rangle \in R$ iff in context $(X, Y, I)$ exists $x \in X$ such that $x^{\uparrow_I} = \{b\} \cup N_j$.

Based on pivot attributes, non-pivot attributes and formal context $(X, Y, I)$ consider **pivot table** which is as new formal context $(P, \mathcal{N}, R)$ with operators for $C \subseteq P$ and $\mathcal{D} \subseteq \mathcal{N}$ defined as follows

$$C^{\uparrow_R} = \{N_i \in \mathcal{N} \mid \text{ for each } p \in P : \langle p, N_j \rangle \in R\},$$
$$\mathcal{D}^{\downarrow_R} = \{p \in P \mid \text{ for each } N_i \in \mathcal{N} : \langle p, N_j \rangle \in R\}$$

In the pivot table $(P, \mathcal{N}, R)$ we are trying to find whether $\{a\}^{\uparrow_R} = \{b\}^{\uparrow_R}$. In

| | $a$ | $b$ | $n_1$ | $n_2$ | $n_3$ |
|---|---|---|---|---|---|
| $x_1$ | × | | × | × | |
| $x_2$ | | × | × | × | |
| $x_3$ | × | | | × | × |
| $x_4$ | | × | | × | × |
| $x_5$ | × | | × | | × |
| $x_6$ | | × | × | | × |
| $x_7$ | × | | × | × | × |
| $x_8$ | | × | × | × | × |

(i) Formal context $(X, Z, I)$

| | $N_1 = \{n_1, n_2\}$ | $N_2 = \{n_2, n_3\}$ | $N_3 = \{n_1, n_3\}$ | $N_4 = \{n_1, n_2, n_3\}$ |
|---|---|---|---|---|
| a | × | × | × | × |
| b | × | × | × | × |

(ii) Pivot table $(P, \mathcal{N}, R)$

| | a | b | $n_1$ | $n_2$ | $n_3$ | $c$ |
|---|---|---|---|---|---|---|
| $x_1$ | × | | × | × | | × |
| $x_2$ | | × | × | × | | × |
| $x_3$ | × | | | × | × | |
| $x_4$ | | × | | × | × | |
| $x_5$ | × | | × | | × | × |
| $x_6$ | | × | × | | × | |
| $x_7$ | × | | × | × | × | |
| $x_8$ | | × | × | × | × | × |

(iii) Formal context $(X, Z \cup \{c\}, I_c)$

| | $N_1 = \{n_1, n_2, c\}$ | $N_2 = \{n_2, n_3\}$ | $N_3 = \{n_1, n_3\}$ | $N_4 = \{n_1, n_3, c\}$ | $N_5 = \{n_1, n_2, n_3\}$ | $N_6 = \{n_1, n_2, n_3, c\}$ |
|---|---|---|---|---|---|---|
| a | × | × | | × | × | |
| b | × | × | × | | | × |

(iv) Pivot table $(P, \mathcal{N}_c, R_c)$

**Fig. 1.** Formal context and pivot tables with and without clones

other words, we want to know, whether the attribute $a$ appears in given formal context with the same combination of other attributes, as $b$ appears (in the same formal context). If yes, the pivot attributes $a, b$ in context $(X, Y, I)$ generates the same generated sets. In other words, $a, b$ are not unique with respect to the non-pivot attributes. Such attributes we call clones. When $\{a\}^{\uparrow_R} \neq \{b\}^{\uparrow_R}$, attributes $a, b$ are unique with respect to the non-pivot attributes, because generates at least one different generated set. The attribute $c \in Y$, which makes $a, b$ unique with respect to generated sets is called the **originality factor** of $a, b$. In Figure 1 we show examples of the contexts and pivot tables with clones or with the originality factor respectively. By introducing the pivot table, the whole problem have been put to the other point of view. The proposed semantics will be explained based on the previous definitions.

### 3.2 Discussion and Remarks

Before the comprehensive examples will be proposed, it is necessary to discuss previous auxiliary definition of the clones using the pivot table. There are couple of problems mainly dealing with ambiguity of the pivot table definition with respect to the various definitions of the clones used by the several authors in the other works. In the pivot table definition the set $N_j \in \mathcal{N}$ is defined as $N_j = \{x\}^{\uparrow_I} \cap (Y \backslash P)$ for each $x \in X$ such that

1. $\{a, b\} \cap \{x\}^{\uparrow_I} \neq \emptyset$ and
2. $\{a, b\} \not\subseteq \{x\}^{\uparrow_I}$.

The first condition tells, that we ignore the itemsets (rows), where neither $a$ nor $b$ is present. Such items are not interesting when we investigate whethe $a$ and $b$ are clones, so we will ignore them when the pivot table is defined. The second condition excludes itemsets, where we have the both pivot attributes $a$ and $b$ and the question is: Why we exclude such itemsets from pivot table, when we can see it in original definition of the clone items? Recall the original definition of the clones:

$$X \to \varphi_{a,b}(X) = \begin{cases} (X \backslash \{a\}) \cup \{b\} \text{ if } b \notin X \text{ and } a \in X \\ (X \backslash \{b\}) \cup \{a\} \text{ if } a \notin X \text{ and } b \in X \\ X \text{ elsewhere} \end{cases}$$

Items $a$ and $b$, which are called clone items in $\mathcal{F}$ iff for any $F \in \mathcal{F}$, we have $\varphi_{a,b}(F) \in \mathcal{F}$. So we need to have the original itemset and swapped itemset as well in the whole collection of itemsets. In definition of $\varphi$ are interesting the rows 1 and 2. The row 3 is only technical condition. It means, that fulfillment of swapping condition of itemsets, which does not contain any of $a$ or $b$ or conversely, when it contains both, is trivial. So we could add them in the pivot table by skipping the condition $\{a, b\} \not\subseteq \{x\}^{\uparrow_I}$, but we consider such information redundant and hence useless. However, the semantics of the clones remains unchanged. But on the other side, it can influence the value of the degree of clones $d^I_{(a,b)}$ (which will be defined later). In such case we need to investigate, which definition would be more precise for the user. The basic idea of our semantics of clones (and nearly clones defined later as well) is, of how original are items $a$ and $b$ in the whole collection of itemsets. The itemsets which does not include either $a$ or $b$ will not tell us anything about originality of such items, the itemsets which include both as well.

The other point for the discussion comes from the problem number six (presented in [2]), which deals with the size of a clone-free Guigues-Duquenne basis. Namely, whether the clone items are responsible for the combinatorial explosion of some Guigues-Duquennes basis. The Guigues-Duquennes basis is non-redundant. All other attribute implications, which holds in given context, can be derived from this base. In the paper [1] there are presented some partial results, which includes definitions and propositions dealing with the clones. The

clones are defined with respect to pseudo-closed sets in the collection of the closed itemsets. The one of the basic results is, that in order to detect clone items, one has to consider meet-irreducible itemsets only (for details see [1]). The definition of the clone items given in [2] is defined in more general manner. It is based not only on the pseudo-closed itemset collection, but it is defined for arbitrary collection of itemsets. This fact can cause, that two items may not appear as clone according the definition in [2], but the are still clones in definition according to [1]. In the rest of the paper there will be considered the definition used in [2] only. However, the proposed semantics would be slightly modified, when we would need to use it in the meaning of [1].

The other important part is to compare proposed solution with other attempts or solutions, but the author has no information either about such attempts or about some real solutions. Hence, according to the author's best knowledge, the author's proposed solution seems to be novel.

### 3.3 Semantics behind Clones - Examples

In this part we would like to show on couple of examples, how the clone items and the originality factor can be used. The originality factor can be desired under some conditions, but undesired under the other conditions. Inall examples the same formal context and the pivot tables will be used, but always with the different meaning of the objects and attributes. The Table 1 represents the original formal context $(X, Z, I)$ with the clones $a$ and $b$ and it also represents the formal context $(X, Z \cup \{c\})$, where the originality factor $c$ is added. The corresponding pivot tables $(P, \mathcal{N}, R)$ and $(P, \mathcal{N}_c, R_c)$ can be seen in the Table 2. A labeling of the objects and the pivot attributes is done according to the particular sets $X$ and $Y$ defined in each example below.

**The sales analysis** Let $X = \{Customer1, \ldots, Customer8\}$ be a set of customers and the set of attributes is defined as $Y = \{Man, Woman, n_1, n_2, n_3, c\}$. The attributes $Man$ and $Woman$ represents the sex of customer and the other attributes represents the products bought by each customer. The following formal contexts represents a marketing research of the sales company (the customers and theirs attributes). In the formal context $(X, Z, I)$ attributes $Man$ and $Woman$ are clones. In the pivot table $(P, \mathcal{N}, R)$ attributes $Man$ and $Woman$ are pivot attributes and $n_j$ is product bought by customer. On the other hand, in the formal context $(X, Z \cup \{c\})$ and the corresponding pivot table $(P, \mathcal{N}_c, R_c)$ the attributes $Man$ and $Woman$ are no longer clones and the attribute $c$ (Product $c$) is the originality factor in this case. Namely, for the itemset $\{Man, n_1, n_3, c\}$ there is no corresponding itemset $\{Woman, n_1, n_3, c\}$.

How can this information be used for the marketing department? Imagine, that the sales company wants to create packages based on the marketing research. These packages should consist of the particular products $n_j$. In the first

| | *Man* / *Europe* / *Gene*1<br>*Woman* / *America* / *Gene*2 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | $n_1$ | $n_2$ | $n_3$ | | | $c$ |
| *Customer* 1 / *Animal* 1 / *Organism* 1 | × | | × | × | | | | × |
| *Customer* 2 / *Animal* 2 / *Organism* 2 | | × | × | × | | | | × |
| *Customer* 3 / *Animal* 3 / *Organism* 3 | × | | | | × | × | | |
| *Customer* 4 / *Animal* 4 / *Organism* 4 | | × | | | × | × | | |
| *Customer* 5 / *Animal* 5 / *Organism* 5 | × | | × | | | × | | × |
| *Customer* 6 / *Animal* 6 / *Organism* 6 | | × | × | | | × | | |
| *Customer* 7 / *Animal* 7 / *Organism* 7 | × | | × | × | × | | | |
| *Customer* 8 / *Animal* 8 / *Organism* 8 | | × | × | × | × | | | × |

**Table 1.** Formal contexts $(X, Z, I)$ and $(X, Z \cup \{c\}, I_c)$

| | $\mathcal{N}$ | | | | $\mathcal{N}_c$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $N_1 = \{n_1, n_2\}$ | $N_2 = \{n_2, n_3\}$ | $N_3 = \{n_1, n_3\}$ | $N_4 = \{n_1, n_2, n_3\}$ | $N_1 = \{n_1, n_2, c\}$ | $N_2 = \{n_2, n_3\}$ | $N_3 = \{n_1, n_3\}$ | $N_4 = \{n_1, n_3, c\}$ | $N_5 = \{n_1, n_2, n_3\}$ | $N_6 = \{n_1, n_2, n_3, c\}$ |
| *Man* / *Europe* / *Gene*1 | × | × | × | × | × | × | | × | × | |
| *Woman* / *America* / *Gene*2 | × | × | × | × | × | × | × | | | × |

**Table 2.** Pivot tables $(P, \mathcal{N}, R)$ and $(P, \mathcal{N}_c, R_c)$

case of the formal context $(X, Z, I)$ the company can create the same packages for man and for woman, because male and female customers buy the same combinations of products $n_j$. The same packages for two different groups can reduce the total cost of production, because we need to produce only four types of the packages, namely the packages $N_1 = \{n_1, n_2\}$, $N_2 = \{n_2, n_3\}$, $N_3 = \{n_1, n_3\}$ and $N_4 = \{n_1, n_2, n_3\}$. With the attribute $c$ added to the formal context, we need six different packages, because only the packages $N_1 = \{n_1, n_2\}$ and $N_2 = \{n_2, n_3\}$ can be produced for men and women at the same time. Other packages are different for the male and female customers. From this point of view, the originality factor is undesired and the clones are desired.

But we can use this information in the other way. Suppose, that the cost difference of producing four or six package types is not significant, but significant can be a targeted marketing on the male and female customers. The formal context $(X, Z, I)$, where we have the clone attributes $Man$ and $Woman$, does not provide differentiated information about the male and female customers. On the other hand, the formal context $(X, Z \cup \{c\})$ does. The attribute $c$ provides desired information, that the Product $c$ influences the different combination of the products bought by the male and female customer. It means, that we can make targeted marketing (namely, the different type of packages for the different type of customers) based on the originality factor Product $c$ and its combinations with the other products. Some combinations of the products with the originality factor can be used as a topic for advertising to highlight the difference between man and woman preferences. However, the clone analysis can provide the marketing department with the useful information in both cases.

**Analysis of the animals** Let $X = \{Animal1, \ldots, Animal8\}$ be a set of animals and a set of attributes is defined as $Y = \{Europe, America, n_1, n_2, n_3, c\}$. The formal context $(X, Z, I)$ in the Table 1, shows the attributes *Europe* and *America* as clones. This fact can be interpreted as follows: In Europe and in America they live the same types of animals, when we consider the attributes of the animals $n_1$, $n_2$ and $n_3$ only. The same information can be seen in the pivot table Table 2. When we add the attribute $c$, we can see the different types of animals (with the different generated sets) in Europe and in America as well (see Table 2). The information, that exists the originality factor $c$ for attributes *Europe* and *America* can be interpreted as follows: It shows, that *Europe* and *America* are somehow specific. In Europe are some different combinations of animal's attributes than in America and vice versa and at the same time we see, that this difference somehow deals with the attribute $c$. Biologist can investigate in more details, what is specific in Europe and in America, which specific attribute of Europe leads to the different attributes of the animals in Europe (and vice versa). Other use of such information is following: From a background knowledge we know, that there is no reason for differentiating the animals in Europe and America just on attribute $c$. In our dataset we do not have in America the animal with attributes $n_1$, $n_3$ and $c$, but with respect to the attribute $c$ we

expect to have the same types of animals in Europe and in America. Thus, we need to look for such animal in America as well. Our hypothesis is, that in America lives such animal, because it lives in Europe and based on our background knowledge there is no reason for $c$ to be the originality factor. From the formal point of view, we do not have the complete dataset (formal context). Some rows are missing, and we need to find such objects in the reality (in this case we are looking for the animal).

**Analysis of genes and the morphological attributes of organisms** The last example use set $X = \{Organism1, \ldots, Organism8\}$ and set of attributes $Y = \{Gene1, Gene2, n_1, n_2, n_3, c\}$ The attributes $Gene1$ and $Gene2$ are clones in formal context $(X, Z, I)$ in Table 1 and the other attributes represents the morphological property of the organism. The interpretation can be following: Organisms with $Gene1$ and $Gene2$ has the same combination of morphological properties $N_j$, when we consider the morphological properties of organisms $n_1$, $n_2$ and $n_3$. The same information can be seen in the pivot table Table 2. When we add the morphological attribute $c$, we get the formal context $(X, Z \cup \{c\})$, which means, that based on attribute $c$ there are some different types of the morphological attributes of organisms with the $Gene1$ and $Gene2$ (see the Table 1 and Table 2). It shows, that the $Gene1$ and $Gene2$ probably does not influence the sets of the morphological attributes containing only $n_1, n_2, n_3$, but this $Gene1$ and $Gene2$ influence the sets of the morphological attributes containing $c$. Thus, $c$ as the originality factor makes the difference between these two genes. This information could be useful for a hypothesis creation in genetics.

## 4 Nearly Clones

### 4.1 Degree of Clones and Degree of Originality

The definition of the clone items is very strict. Recall, that condition $\varphi_{a,b}(F) \in \mathcal{F}$ needs to be true for any $F \in \mathcal{F}$. We can see, that adding only one "cross" into the huge formal context can cause, that two clones disappear. We expect, that in real dataset such condition can be true very rarely. When we want to use the clone items meaningfully, we need to have a weaker definition. For practical purposes it suffices, that condition $\varphi_{a,b}(F) \in \mathcal{F}$ can be true in some reasonable amount of $F \in \mathcal{F}$. We define **degree of clone** as

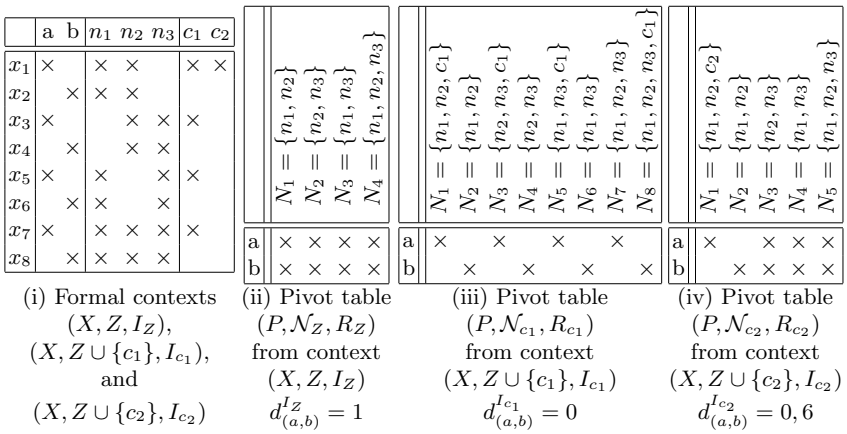
$$d^I_{(a,b)} = \frac{|\{a\}^{\uparrow_R} \cap \{b\}^{\uparrow_R}|}{|\{a\}^{\uparrow_R} \cup \{b\}^{\uparrow_R}|},$$

which can be read as follows: The attributes $a$ and $b$ with respect to the formal context $I$ are clones in the degree $d$. For a priori given threshold $\theta$ we define $a$ and $b$ as **nearly clones** iff $d_{(a,b)} \geq \theta$. Note, that for $d_{(a,b)} = 1$ the attributes $a$ and $b$ are clones and for $d_{(a,b)} = 0$ we say, that they are **original attributes**. Consider now the formal context $(X, Z, I_Z)$ and the corresponding pivot table $(P, \mathcal{N}_Z, R_Z)$ (see Figure 2). We can see, that $a$ and $b$ are clones with the degree

$d^{I_Z}_{(a,b)} = 1$. Adding either attribute $c_1$ or attribute $c_2$ to the formal context leads to decreasing of clone degree for $a$ and $b$. Namely, $d^{I_{c_1}}_{(a,b)} = 0$ and $d^{I_{c_2}}_{(a,b)} = 0,6$. In both cases degree has decreased, but the resulted clone degree is different. In the first case attributes are original, in the second case attributes are nearly clones for arbitrary $\theta \leq 0,6$. Such situation can be formalized, and define the **degree of originality** for given $c_i$ and context $(X, Z, I)$ as

$$g^{I_c}_{(a,b)} = d^{I}_{(a,b)} - d^{I_c}_{(a,b)}$$

The degree of originality shows, how the attribute, added to the context, does influence the degree of clone for given attributes $a, b \in Y$ and the formal context $(X, Z, I)$.




(i) Formal contexts $(X, Z, I_Z)$, $(X, Z \cup \{c_1\}, I_{c_1})$, and $(X, Z \cup \{c_2\}, I_{c_2})$

(ii) Pivot table $(P, \mathcal{N}_Z, R_Z)$ from context $(X, Z, I_Z)$ $d^{I_Z}_{(a,b)} = 1$

(iii) Pivot table $(P, \mathcal{N}_{c_1}, R_{c_1})$ from context $(X, Z \cup \{c_1\}, I_{c_1})$ $d^{I_{c_1}}_{(a,b)} = 0$

(iv) Pivot table $(P, \mathcal{N}_{c_2}, R_{c_2})$ from context $(X, Z \cup \{c_2\}, I_{c_2})$ $d^{I_{c_2}}_{(a,b)} = 0,6$

**Fig. 2.** Formal contexts and pivot tables with different degrees of clone.

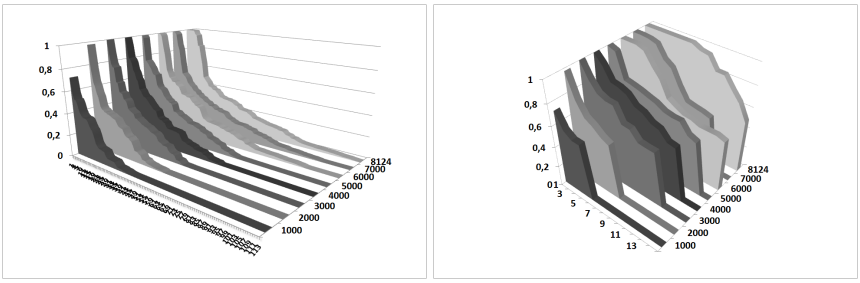
### 4.2 Experiment Nr. 1 - Amounts and Degrees of Nearly Clones in Datasets

For the purpose of this paper we arranged two introductory experiments with the nearly clones, in which we use datasets *Mushroom* [3], *Adults* [4] and *Anonymous* [5] from well known UC Irvine Machine Learning Repository (for the details see Table 3). In the experiments we used a naive algorithm (the brute-force search, but with polynomial complexity) for finding the degrees of clones as defined above. Looking for more efficient algorithm is out of scope of this paper. The algorithm was implemented in C, and all experiments have been run on the computer with an Intel Core i5 CPU, 2.54 Ghz, 6 GB RAM, 64bit W7 Professional.

In the first experiment we were focused on finding all nearly clone pairs, with $d_{(a,b)} > 0$, especially we investigated, if there are some clones (where $d_{(a,b)} = 1$) in the real datasets. The results of the first experiment are shown in Table 3.

| | Mushroom [3] | Adults [4] | Anonymous [5] |
|---|---|---|---|
| Number of objects | 8 124 | 48 842 | 32 713 |
| Number of attributes | 119 | 104 | 295 |
| Number of nearly clones $d_{(a,b)} > 0$ | 113 | 1 568 | 382 |
| Maximal $d_{(a,b)} > 0$ | 1,00000 | 0,02252 | 0,00187 |
| Minimal $d_{(a,b)} > 0$ | 0,00123 | 0,00014 | 0,00143 |
| Average $d_{(a,b)} > 0$ | 0,24423 | 0,00449 | 0,00160 |
| Median $d_{(a,b)} > 0$ | 0,14537 | 0,00195 | 0,00159 |
| Slope | 0,99402 | 0,77895 | 0,55004 |

**Table 3.** Overview of the datasets and results of the first experiment (source of datasets: *http://archive.ics.uci.edu/ml/index.html*)




(i) nearly clone pairs for $d_{(a,b)} > \theta = 0$
x-axis - number of nearly clone pairs
y-axis - degree of clone $d_{(a,b)}$
z-axis - number of objects processed

(ii) nearly clone pairs for $d_{(a,b)} \geq \theta = 0.5$
x-axis - number of nearly clone pairs
y-axis - degree of clone $d_{(a,b)}$
z-axis - number of objects processed

**Fig. 3.** Mushroom - distribution of $d_{(a,b)}$ in dataset scaled by 1000 objects.

In case of the dataset *Mushroom*, we present also the distribution of the clone degrees and some other details as well. Figure 3 shows the volume of all pairs $a$ and $b$ and clone degree $d_{(a,b)} > 0$, for each scale pattern (from 1000 to 8124 by 1000). In (i) are displayed all pairs with $d_{(a,b)} > 0$ and part (ii) is more focused on the amount of pairs where $d_{(a,b)} \geq 0,5$ for each investigated scaled pattern. Note, that the results from numbers of the processed objects in the dataset $Mushroom$ (namely from 1000 to 7000 depicted in z-axis in the Figure 3) depends on an order of the processing objects. This fact were not investigated

more deeply. However, when we have processed all 8124 objects, the order will not influence the result. Figure 4 shows some interesting details. In (i) there are presented the pairs $a$, $b$ with $d_{(a,b)} = 1$, in (ii) the same for $1 > d_{(a,b)} \geq 0,5$ . We have found 4 clone pairs, and one clone triple. In the clone triple $(103, 104, 105)$ we can see the transitivity (i.e. when $(a, b)$ are clones and $b, c$ are clones, also $a$ and $c$ are clones). Such transitivity is not surprising and is direct consequence of the clone definition.

What does such results show and does it appear reasonable? The Figure 4 part (i) shows the clones $a$ and $b$. The original dataset $Mushroom$ consists of 22 attributes with non-binary values. For the purposes of clone investigation, this dataset were nominally scaled to the formal context, which is binary indeed. It is interesting to see, that all clone items represents the value of the same original attribute. E.g. clones 019 and 021 represents the original attribute $Cap\ Color$, thus its values $Purple$ or $White$ respectively. Another example is clone triple 103, 104 and 105 which represents the original attribute $Spore\ Print\ Color$ with the corresponding values $Orange$, $Purple$ and $White$. It can be interpreted as follows: Purple and white color generates the same sets of the non-pivot attributes. In other words, to each mushroom with the purple cap (the pivot attribute), there exists corresponding mushroom with the white cap (the pivot attribute), but all other properties remains he same (non-pivot attributes). Similarly to each mushroom with the purple spore print color, there exists corresponding mushroom with the white spore print color and the corresponding mushroom with the orange one. When we look on the nearly clones in the Figure 4 part (ii), the attributes 69 and 70 represents the same original attribute *stalk color above ring* with the values *cinnamon* and *gray* (the details are not shown in the table). These attributes are not the clones, but the nearly clones with the clone degree $d_{(a,b)} = 0,96$. It can be interpreted similarly as by the clones. Only the difference will be in a quantifier. By clones the quantifier was "for each", by the nearly clones we will have fuzzy quantifier, in this case "for the most". Hence the interpretation is: For the most mushroom with cinnamon stalk above the ring exists corresponding mushroom with the corresponding gray stalk above the ring (and vice versa). The clone degree is very high in this case ($d_{(a,b)} = 0,96$) it means there are only couple of mushrooms with cinnamon stalk above the ring color, which do not have corresponding mushroom with the gray stalk above the ring color. However, the for the deeper understanding of such examples, it is required to ask an expert in mycology.

### 4.3 Experiment 2 - Structure of Nearly Clones in Datasets

In the second experiment we investigated the structure of nearly clones. Namely, we have defined a fuzzy relation $T : Y \times Y \to L$, where $L = [0; 1]$ is defined as $T(a, b) = d_{(a,b)} \in L$. In other words, the fuzzy relation express the degree of the clone for each pair $a, b \in Y$. For the better visualization we display such relation in so called "bubble chart". The bubble chart displays three dimensional data in two dimensional chart. The position of the bubble is given by two dimensions

(x and y axis) and the size of the bubble shows the third dimension. The results from the first experiments are displayed in bubble chart, where the pairs of attributes $a$ and $b$ represents two dimensions and the degree of clone $d_{(a,b)}$ is represented by the size of the bubble. Note, that the fuzzy relation $T$ is indeed symmetric (i.e. $d_{(a,b)} = d_{(b,a)}$), but we show only part of the relation, where $a < b$. Figures 5, 6 and 7 show the structure of the nearly clones for the datasets *Mushroom*, *Anonymous* and *Adults*. We can observe very different structure of the nearly clones in each dataset. In *Mushroom* we can see, that the structure of the nearly clones is approximately linear. All nearly clones are clustered near to the line defined as $(y, y)$. In the case of *Adults* dataset we can see more spread, but still approximately linear structure, except of one cluster near point $(0, |Y|)$. The nearly clones of the data set *Anonymous* forms the different, but kind of regular structure as well. This results leads to the question, what kind of properties has fuzzy relation of the nearly clones and if properties of such fuzzy relation correlates with the properties of the formal context, or with properties of the concept lattice. Until now we know, that such relation is transitive for clone items $d_{(a,b)} = 1$ and symmetric for the arbitrary nearly clone items, but this two properties are trivial. I would be also interesting to find the semantics of such fuzzy relation defined on the nearly clones. All this will be part of the future investigation.

| original attributes | a | b | $d_{(a,b)} = 1$ |
|---|---|---|---|
| 03. cap-color | 019 purple=u | 021 white=w | 1,00 |
| 05. odor | 025 almond=a | 026 anise=l | 1,00 |
| 05. odor | 030 musty=m | 031 none=n | 1,00 |
| 17. veil-color | 086 brown=n | 087 orange=o | 1,00 |
| 20. spore-print-color | 103 orange=o | 104 purple=u | 1,00 |
| 20. spore-print-color | 103 orange=o | 105 white=w | 1,00 |
| 20. spore-print-color | 104 purple=u | 105 white=w | 1,00 |

(i) Clones

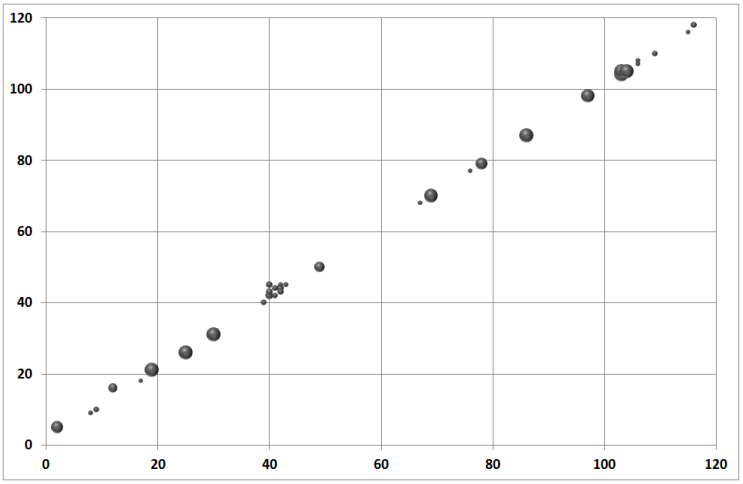
| a | b | $d_{(a,b)} \geq 0.5$ |
|---|---|---|
| 69 | 70 | 0,96 |
| 97 | 98 | 0,95 |
| 78 | 79 | 0,84 |
| 2 | 5 | 0,84 |
| 49 | 50 | 0,74 |
| 12 | 16 | 0,66 |
| 40 | 42 | 0,51 |

(ii) Nearly clones

**Fig. 4.** Dataset Mushroom - experiment on nearly clones

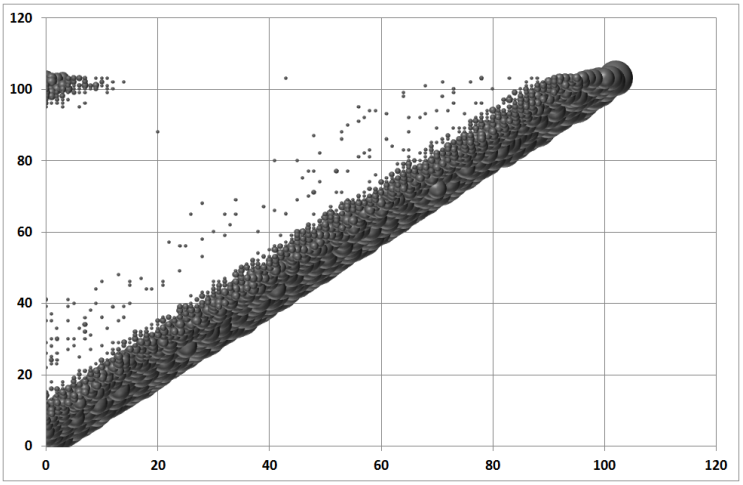
## 5 Conclusion and Future Perspectives

The paper was motivated by open problem proposed at ICFCA 2006 [2]. We hope, that this small open problem is solved now and the reason is presented in the first part of the conclusion. This part is structured as a direct answers on proposed questions. The second part of the conclusion describes ideas, which overlaps the original open problem and come with some new questions.



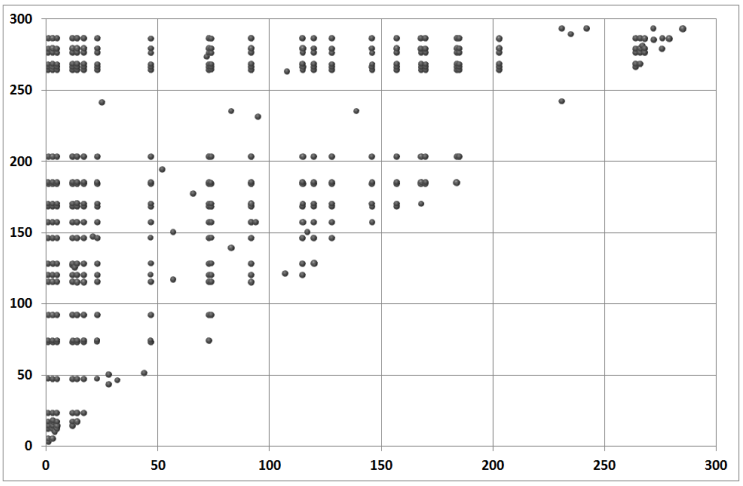

Nearly clone pairs for $d_{(a,b)} > \theta = 0$ , where x,y-axis - $a$ and $b$ pairs
size of bubble = $d_{(a,b)}$

**Fig. 5.** Dataset Mushroom




Nearly clone pairs for $d_{(a,b)} > \theta = 0$ , where x,y-axis - $a$ and $b$ pairs
size of bubble = $d_{(a,b)}$

**Fig. 6.** Dataset Adults




Nearly clone pairs for $d_{(a,b)} > \theta = 0$ , where x,y-axis - $a$ and $b$ pairs
size of bubble = $d_{(a,b)}$

**Fig. 7.** Dataset Anonymous

### 5.1 Conclusion for Open Problem Questions

**Question 1**: Does the symmetrical behaviour of $a$ and $b$ make sense for someone?
**Answer 1**: Yes, such symmetrical behaviour can identify the same combination of the non-pivot attributes with respect to pivot attributes and can make sense:

- for the marketing department to reduce cost of packages - the clone items enable the same packages for the different types of customers (e.g. man and woman)
- for biologists to complete the the dataset - the clone items are expected, because the originality factor c, has no sense based on the background knowledge. Hence, we some miss rows in the dataset (e.g. we need to find the new animals)
- for genetics - it bring an information that the two genes has no influence on a combination of the morphological properties of organisms
- generally for everyone, who needs an information about the same combination of non-pivot attributes with respect to the pivot attributes

**Question 2**: Does it make sense, that such symmetrical behaviour disappear, when $c$ is added?
**Answer 2**: Yes, such attribute is called the originality factor for the items $a$ and $b$ and can be useful:

- for the marketing department to make a targeted marketing for the different types of customers (e.g. man and woman) using unique combination of the non-pivot attributes
- for biologist to find the difference between two pivot attributes (e.g. Europe and America) with respect to other non-pivot properties. The originality factor $c$ reveals, that the pivot attributes are original and this originality needs to be investigated deeper.
- for genetics - it brings an information, that two genes has an influence on a combination of the morphological properties of organisms
- generally for everyone, who needs an information about the reason, why the non-pivot attributes has the different combinations with respect to the pivot attributes.

**Question 3**: What is semantics behind $a$, $b$, and $c$?
**Answer 3**: The attributes $a$ and $b$ are the pivot attributes, all other attributes are the non-pivot attributes and $c$ is moreover the originality factor for the attributes $a$ and $b$. The pivot attributes generates a combination of the non-pivot attributes in the given context. The attribute $c$ make the attributes $a$ and $b$ unique, which can be "good" or "bad". It depends on a goal of the analysis.

### 5.2 Conclusion and Future Perspectives

The second part of conclusion shows, that the clones are very strictly defined. Therefore the nearly clones were introduced. The nearly clones operates with the degree, in which two attributes are clones. Such formalization asks itself for study of the nearly clones under fuzzy setting (e.g. we have already mentioned, that structure of nearly clones can be seen as fuzzy relation indeed). The introductory experiments shows, that the nearly clones in dataset have an interesting structure, which needs to be investigated more deeply. This paper was introductory for the nearly clones. As a future work we plan to describe more efficient algorithm to compute the nearly clones for the given threshold $\theta$, and algorithm for identifying the originality factors for another given threshold $\omega$. Finally we hope, that this paper, even it does not come with a great mathematical or experimental results, brings some interesting ideas to FCA community.

**Acknowledgements.** The author is very grateful to the reviewers for their helpful comments and suggestions. Partly supported by IGA (Internal Grant Agency) of the Palacky University, Olomouc is acknowledged.

## References

1. Gély A,, Medina A., Nourine L. and Renaud Y.: *Uncovering and Reducing Hidden Combinatorics in Guigues-Duquenne Bases.* Springer, Lecture Notes in Computer Science, 2005, Volume 3403/2005, 235-248, Heidelberg 2005
2. more authors: *Some open problems in Formal Concept Analysis.* ICFCA 2006, Dresden, http://www.upriss.org.uk/fca/fcaopenproblems.html

3. Schlimmer,J.S. : *Concept Acquisition Through Representational*, Adjustment (Technical Report 87-19). (1987). Doctoral disseration, Department of Information and Computer Science, University of California, Irvine.
4. Kohavi R.: *Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid* Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996
5. Breese J., Heckerman D., Kadie C.: *Empirical Analysis of Predictive Algorithms for Collaborative Filtering.* Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July, 1998.

# Computing the Skyline of a Relational Table Based on a Query Lattice

Nicolas Spyratos, Tsuyoshi Sugibuchi, Ekaterina Simonenko, and Carlo Meghini

Laboratoire de Recherche en Informatique, Université Paris-Sud 11, France
{Nicolas.Spyratos, Tsuyoshi.Sugibuchi, Ekaterina.Simonenko}@lri.fr
Istituto di Scienza e Tecnologie della Informazione del CNR, Pisa, Italy
Carlo.Meghini@isti.cnr.it

**Abstract.** We propose a novel approach to computing the skyline set of a relational table $R$, with respect to preferences expressed over one or more numerical attributes. Our approach is based on what we call the *query lattice* of $R$, and our basic algorithm constructs the skyline set as the union of the answers to a subset of queries from that lattice - hence *without* directly accessing the table R. Therefore, in contrast to all existing techniques, our approach is independent of how the table R is implemented or how its tuples are indexed. We demonstrate the generality of our approach by computing the skyline set of the join of two tables based on the product of their individual query lattices - therefore *without* performing the join. The paper presents basic concepts and algorithms leaving experimentation and performance evaluation to a forthcoming paper.

**Keywords:** skyline, relational table, query lattice

## 1 Introduction

In many multicriteria decision-making applications, dominance analysis is an important aspect. As an example, consider a person looking for a vacation package using two criteria, or "attributes": hotel rating and price. Intuitively, a package $P = \langle r, p \rangle$ is better than a package $P' = \langle r', p' \rangle$ if $P$ is better than $P'$ in one attribute and not worse than $P'$ in the other attribute. If this is the case then we say that $P$ *dominates* $P'$.

For example, consider the following three packages:

- $P_1 = \langle 2, 100 \rangle, P_2 = \langle 3, 130 \rangle, P_3 = \langle 2, 120 \rangle$

Since a higher rating and a lower price are more preferable, $P_1$ dominates $P_3$. On the other hand, $P_1$ and $P_2$ don't dominate each other because $P_1$ has a lower rating than $P_2$ and $P_2$ has a higher price. Similarly, $P_2$ and $P_3$ don't dominate each other because $P_2$ has a higher rating than $P_3$ and $P_3$ has a lower price.

A package, or "tuple" that is not dominated by any other tuple is said to be a skyline tuple or to be in the skyline. The tuples in the skyline are the best

possible trade-offs among the attribute values appearing in the tuples. Thus in our example, packages $P_1$ and $P_2$ are in the skyline, while $P_3$ is not.

In order to conduct a skyline analysis, two items must be specified:

1. A set of attributes over which preferences are expressed (such as Hotel Rating and Price, in our example).
2. An ordering of the attribute values (either total or partial) according to which preference is expressed (such as the ordering of the integers for Hotel Rating to express that rating $r$ is preferred to rating $r'$ if $r > r'$; and similarly for Price to express that price $p$ is preferred to price $p'$ if $p < p'$).

In recent years, skyline analysis has gained considerable interest in the area of information systems in general, and in the area of databases in particular. However, skyline analysis (i.e. computing non dominated points) existed well before the concept appeared in database research; it is known as the maximum vector problem or the Pareto optimum [11][18]. The popularity of skyline analysis in the area of information systems is mainly due to its applicability for decision making applications. Indeed, as information systems store larger and larger volumes of data today, data management and in particular query processing present difficult challenges. From the user viewpoint, large volumes of data imply answers of large size. By returning the best tuples (in terms of user preferences), the skyline query relieves the user from having to deal with answers of large size in order to find the best tuples.

The skyline operator was first introduced in [3], where the authors also present two basic algorithms: the Block Nested Loops (BNL) and the Divide and Conquer (D&C). In order to improve the performance of BNL algorithm, the SFS (*sort-filter-skyline*) algorithm was proposed in [5]. SFS runs on data sorted according to a monotonic function (namely, *entropy* descending). Such sorting guarantees the non-dominance of each object by those that follow in the order. Therefore, once an object is put into the buffer window, it can be reported as part of the skyline. Not only this makes the SFS algorithm progressive, but also allows to reduce the number of comparisons needed, since SFS compares only against the non-dominated tuples, whereas BNL often compares against dominated tuples [7]. Despite this improvement, all objects have to be scanned by the algorithm at least once. Succeeding approaches tend to avoid scanning the complete data set. Namely, SaLSa [1] uses the *minimal coordinate* of each object as a sorting function, and during the filter-scan step checks if all remaining objects are dominated by a so-called *stop* object, which can be determined in $O(1)$ from the data accessed so far. The shortcoming is that the performance of SaLSa algorithm is affected by the data distribution and increasing dimensionality, since in higher dimensions instances of the problem the pruning power of the stop object is limited [29].

More generally, all sort-based techniques share the same drawback, namely the number of computations during the filter-scan step, as every input object should be compared with the skyline points in the buffer (which can potentially become large).

An alternative to the sort-based techniques is the use of indexes, which allows to avoid scanning all the input objects. The basic idea is to rely on an index in order to determine dominance between tuples, and to exclude tuples from further processing as early as possible. Two index-based algorithms, Bitmap and Index were first introduced in [21]. Bitmap uses the bitmap encoding of the data so that the dominating points are determined by a bit-wise "and" operation. The Index approach partitions the objects into a set of lists. Each list is sorted by minimum coordinate and indexed by a B-tree. The objects are accessed in batches defined by the values of the minimal coordinate, while the algorithm computes local skylines in each "batch" of the lists and then merges them into a global skyline. However, besides the computation cost of Bitmap, and the necessity to construct a B-tree for every combination of dimensions having potential interest for the user, the order in which skyline points are returned by these algorithms is fixed and depends on the data distribution, so it cannot be adapted to the users preferences.

Two other index-based skyline algorithms, NN (nearest-neighbor) [13] and BBS (*branch-and-bound skyline*) [19], are based on the observation that the object closest to the origin has to be part of the skyline. Nearest neighbor search is used to retrieve such point by using the R-tree. The pitfall of the NN algorithm is that in order to iteratively find the next nearest neighbors it divides the data set into overlapping partitions, and therefore duplicates have to be removed by traversing the R-tree multiple times. To avoid that, BBS rather accesses partially dominated nodes of the R-tree.

The main drawback of all index-based approaches is that not all data can be indexed (namely when data is dynamically produced). Also, R-trees and other multidimensional indexes have their own limitations, namely the curse of dimensionality.

Concerning related work specifically targeting the multi-relational skyline (or skyline join), two progressive algorithms are proposed in [9]. The idea is to combine the join with nested-loop and sort-merge algorithms. However, each relation has to accessed multiple times in order to compute the skyline for each join value, and then the global skyline. In addition, each input object has to be scanned at least once.

More recent work also considers how to compute the skyline over the join of two or more relational tables without actually computing the join [23]. Apart from its applicability to computing skylines in a centralized database, the interest of such work lies in the fact that it is also applicable to distributed environments. Earlier work related to this topic can be found in [1][2][8][10][20][25][26].

A lattice-based approach to single-relation skyline computation is introduced in [15], with the aim of proposing a data-distribution independent algorithm. A lattice structure is used to answer skyline queries over dimensions with low-cardinality domains, or those that can be mapped to low-cardinality domains (such as *Price*, that can be mapped to price ranges). The principle is to organize all the values combinations into a lattice based on the dominance relationship, and then to retrieve those that (a) are present in the input data set, and (b)

are not reachable by the dominance relationship from another element of the lattice, also belonging to the data set. However, no early pruning is done, so the entire data set has to be read twice in order to determine the skyline tuples, and the skyline join problem is not investigated. Also, the mapping of the values of a domain to a set of ranges has to be carefully tuned in order to deliver a meaningful skyline result, which is not discussed in the paper.
Several variants of skyline were introduced in [19], such as constrained, subspace and dynamic skyline queries (see also [6][16][22][27][28]). Skyline queries have also been studied in various other domains, outside traditional databases. These include probabilistic skyline computations over uncertain data [17](e.g. data in sensor networks); skyline computations over incomplete data [12](e.g. data with missing values); over data whose attributes have partially-ordered domains [4](e.g. preferences expressed by users online); over stream data[14]; or even bandwidth-constrained skyline computations over mobile devices [24].

In this paper, we present a novel approach to computing skylines which represents a major deviation from existing approaches. Indeed, instead of accessing individual tuples in a database table, our approach relies on the definition of skyline as the union of the answers to a set of queries. In doing so, our basic algorithm avoids accessing the table directly: access to the table is through queries, hence independent of how the table is implemented or how its tuples are indexed.

Given a relational table $R$, our approach is based on what we call the query lattice of $R$; and our basic algorithm constructs the skyline set as the union of the answers to a subset of queries from that lattice - hence *without* directly accessing the table $R$. We demonstrate the generality of our approach by computing the skyline of the join of two tables based on the product of their individual query lattices - therefore *without* performing the join. The paper presents basic concepts and algorithms leaving experimentation and performance evaluation to a forthcoming paper.

The paper is organized as follows. In section 2 we give some preliminary definitions and introduce our notation. In section 3 we present our basic algorithm for computing skylines through queries. In section 4 we apply our approach to computing skylines over joins, thus demonstrating the generality of the approach. Finally, in section 5, we offer some concluding remarks and discuss further research.

## 2 Basic definitions

Let $R$ be a relational table, with $A_1, \ldots, A_n$ as attributes. Let $\mathcal{B} = \{B_1, \ldots, B_k\}$, $k \leq n$, be the set of *preference attributes,* that is a set of attributes of the table whose domains are numeric and over which preferences are declared.

A *preference* over $B_i$ is an expression of one of two forms: $B_i \rightarrow min$ or $B_i \rightarrow max$. If the preference $B_i \rightarrow min$ is expressed by a user of the table, then this is interpreted as follows: given two values $x$ and $y$ in the domain of $B_i$, $x$ *is preferred to* $y$ or $x$ *preceeds* $y$ iff $x < y$; and similarly, if the preference $B_i \rightarrow max$

is expressed by a user, then this is interpreted as follows: given two values $x$ and $y$ in the domain of $B_i$, *$x$ is preferred to $y$* or *$x$ preceeds $y$* iff $x < y$;

In order to simplify the presentation, and without loss of generality, we shall consider only one form of preference, namely $B_i \rightarrow min$. However, all methods discussed in this paper can be applied with any combination of the preferences $B_i \rightarrow min$ and $B_i \rightarrow max$.Therefore, from now on, given two values $x$ and $y$ in the domain of $B_i$, we shall say that *$x$ is preferred to $y$* or *$x$ preceeds $y$* iff $x < y$.

**Definition 1** *(Pareto domination) Let $\mathcal{B} = \{B_1, \ldots, B_k\}$ be a set of preference attributes of a relational table $R$ and let $s$ and $t$ be tuples of $R$. We say that* s is equivalent to t, *denoted as $s \equiv t$ iff $s.B_i = t.B_i$ for all $B_i \in \mathcal{B}$. Moreover, we say that* s Pareto dominates t, *denoted as $s <_{Pa} t$, iff $s \not\equiv t$ and for all $B_i \in \mathcal{B}$, $s.B_i \leq t.B_i$.*

In order to simplify the presentation we will simply say "dominates" instead of "Pareto dominates", and we shall drop the subscript in the notation, writing $s < t$ instead of $s <_{Pa} t$.

We shall call *Pareto preference query,* or simply *preference query* over $R$, any expression of the form $(B_1 = b_1) \wedge \ldots \wedge (B_k = b_k)$, where each $b_i$ is a value in the domain of attribute $B_i$. For simplicity of notation we shall denote a preference query simply by $\langle b_1, \ldots, b_k \rangle$.

Note that $\langle b_1, \ldots, b_k \rangle$ denotes also a tuple in the projection of $R$ over the preference attributes; however, context will always disambiguate. Also note that a preference query $\langle b_1, \ldots, b_k \rangle$ returns the set of tuples in $R$ whose projection over the preference attributes is the tuple $\langle b_1, \ldots, b_k \rangle$; therefore the answer to each query of the form $\langle b_1, \ldots, b_k \rangle$ is a Pareto equivalence class.

It is easy to verify that Pareto domination is irreflexive (*i.e.*, $s < s$ is false for each tuple $s$) and transitive (*i.e.*, $s < t$ and $t < u$ imply $s < u$ for all tuples $s$, $t$ and $u$), hence a strict order over $R$. A partial order $\leq$ over $R$ can be defined from Pareto domination as follows:

$$s \leq t \text{ iff } (s < t \text{ or } s = t)$$

for all tuples $s$ and $t$ in $R$. We shall say that *$s$ Pareto preceeds $t$*, or simply that *$s$ preceeds $t$*, whenever $s \leq t$.

Clearly, Pareto precedence defines a partial order also over preference queries. Moreover, given preference queries $s$ and $t$ we define the following operations:

- $s \otimes t = \langle min\{s.B_1, t.B_1\}, \ldots, min\{s.B_k, t.B_k\} \rangle$
- $s \oplus t = \langle max\{s.B_1, t.B_1\}, \ldots, max\{s.B_k, t.B_k\} \rangle$

It is easy to check that these operations make the set of preference queries over $R$ into a complete lattice, with $\otimes$ defining the least upper bound and $\oplus$ defining the greatest lower bound of any two preference queries. We notice that this lattice may be infinite, as some of the domains of the preference attributes may be infinite. However, if we require that each $b_i$ in a preference query be in the active domain of $B_i$ (*i.e.*if we require that each $b_i$ appear in $R$), then the lattice

becomes finite and therefore it has a top and a bottom query, denoted as $\top$ and $\bot$, respectively. These extreme elements are given by:

$$\top = \langle m_1, \ldots, m_k \rangle$$
$$\bot = \langle M_1, \ldots, M_k \rangle$$

where $m_i$ and $M_i$ are the minimum value and the maximum value appearing in the active domain of $B_i$, respectively. We shall call this (finite) lattice the *query lattice,* of $R$ and we shall denote it as $(Q, \leq)$, where $Q$ is the (finite) set of preference queries over $R$.

In this paper, we shall use the query lattice for two purposes: (a) as a tool for computing skylines of relational tables, and (b) as a means for comparing our approach to existing approaches. First, however, let's define skylines formally.

**Definition 2** *The* skyline *of a table $R$ over preference attributes $\mathcal{B}$, denoted by $SKY(R, \mathcal{B})$, is the set of tuples from $R$ defined as follows:*

$$SKY(R, \mathcal{B}) = \{t \in R \mid \nexists s \in R : \ s \leq t\}$$

In other words, the skyline of $R$ is the set of non-dominated tuples of $R$.

As customary, given a query $q$ and a relation $R$, we will let $ans(q, R)$ stand for the *answer of $q$ over $R$,* that is the set of tuples obtained by asking query $q$ against relation $R$. Moreover, we define a *skyline query* of $R$ over preference attributes $\mathcal{B}$, to be a query $q(R, \mathcal{B})$ over $R$ whose answer is a skyline of $R$ over $\mathcal{B}$, that is:

$$ans(q(R, \mathcal{B}), R) = SKY(R, \mathcal{B})$$

Skyline queries will play a central role in our approach to skyline computation.

## 3 Computing the skyline of a relational table

In this Section, we present an algorithm for computing the skyline of a given table $R$. Our algorithm obtains the skyline by constructing a skyline query. To find the skyline query, our algorithm traverses part of the query lattice and collects a set of non-dominated queries whose disjunction is the skyline query.

In contrast to existing methods that actually construct the skyline, and as such are sensitive to the ways the table $R$ is implemented or how its tuples are accessed, our algorithm obtains the skyline while making no assumptions on how the relation $R$ is accessed by the system while processing the skyline query.

Our algorithm uses the notion of *successors* of a query, defined as follows. First, for each preference attribute $B_i$ and value $b_i$ in the domain of $B_i$, let's denote as $succ(b_i)$ the successor of $b_i$ in the linear order of the domain of $B_i$. For instance, let $B_i$ be the Hotel Rating attribute of our earlier example, having as domain the interval [1,5]. As this interval is linearly ordered by the $<$ relation, we have $succ(1) = 2$, $succ(2) = 3$, and so on. This makes *succ* a partial function over the domain of $B_i$, undefined for the maximum $M_i$. Now, we extend the *succ* function to preference queries as follows.

Let $q$ be a preference query $q = \langle b_1, \ldots, b_k \rangle$ such that $q \neq \bot$. This means that $b_j \neq M_j$ for at least one $j \in [1, k]$, where $M_j$ is the maximum value in the active domain of $B_j$. With no loss of generality, we shall assume that the $m$ values of $q$ that are not maximal, where $1 \leq m \leq k$, occur in the first $m$ positions of $q$, (*i.e.*, $q = \langle b_1, \ldots, b_m, M_{m+1}, \ldots, M_k \rangle$). Then, the *successors of q, succ*$(q)$, is defined to be the set of queries $succ(q) = \{q_1, \ldots, q_m\}$, such that, for all $1 \leq i \leq m$,

$$q_i = \langle b_1, \ldots, b_{i-1}, succ(b_i), b_{i+1}, \ldots, b_m, M_{m+1}, \ldots, M_k \rangle$$

Clearly, the *succ* function is undefined on the bottom of the lattice $\bot$. The following Lemma gives two important properties of the successors of $q$ for the establishment of the correctness of the following algorithm for computing a skyline query of the table $R$.

**Lemma 1.** *Let $q$ be a preference query. Then, for each query $q' \in succ(q)$ :*

1. $q \leq q'$
2. *there is no query $q''$ such that $q < q'' < q'$.*

We are now ready to give the algorithm for computing a skyline query of a table $R$.

**Algorithm** Skyline Query over a Single Table (*SQST*)

*Input* A non-empty table $R$, a non-empty set $\mathcal{B} = \{B_1, \ldots, B_k\}$ of preference attributes in $R$, and the projections of $R$ over $B_1, \ldots, B_k$.

*Data* We use the following variables for accumulating data during the execution of the algorithm:

- The variable $F$ is a set variable called *frontier;* it is initialized to empty, and it is used to accumulate the queries whose disjunction will be the result of the algorithm (*i.e.*, whose disjunction will be the skyline query).
- The variable $C$ is a set variable containing the set of all current candidate queries; it is initialized to the top $\top$ of the query lattice.
- The variable $S$ is a set variable used to accumulate successors of current candidate queries.
- The variable $C'$ is a (auxiliary) set variable for accumulating candidate queries for the next while-loop iteration in the algorithm; it is initialized to empty at the beginning of each iteration.

*Output* A set of preference queries over $R$, whose disjunction is a skyline query.

*Method*

$C \leftarrow \{\top\}; F \leftarrow \emptyset$
**while** $C \neq \emptyset$ **do**
  **for all** $c \in C$ such that $ans(c, R) \neq \emptyset$ **do**
    $C \leftarrow C \backslash \{c\}; F \leftarrow F \cup \{c\}$

```
    end for
    C' ← ∅
    for all c ∈ C do
      for all s ∈ succ(c) such that no query in F Pareto dominates s do
        C' ← C' ∪ {s}
      end for
    end for
    C ← C'
  end while
  return F
```

Informally, the algorithm works as follows:

- If the top query $\top$ of the lattice is non-empty (*i.e.*if its answer over $R$ is non-empty) then the algorithm terminates; $\top$ is the output of the algorithm, and the answer of $\top$ over $R$ is the skyline.
- Otherwise, each successor query $c$ of $\top$ might be a candidate for contributing to the skyline, and this is checked as follows:
  - if the query $c$ is non-empty then it is added to $F$ (*i.e.*to the variable that accumulates the queries contributing to the skyline);
  - otherwise, each successor of $c$ that is not dominated by a query in $F$ becomes a candidate, by being added to the variable $C'$ (*i.e.*to the variable that accumulates all candidate queries for the next iteration). The so selected successors are finally transferred to the variable $C$.

This process is repeated until there is no more candidate left (*i.e.*until $C$ is empty).

Upon termination of the algorithm, $F$ contains all queries $q$ such that: (a) the answer to $q$ is non-empty, and (b) $q$ is not dominated by another query. The disjunction of all queries in $F$ is then the skyline query, and the answer of the skyline query over $R$ is the skyline of $R$.

$R$

| HID | Price | Rating |
|---|---|---|
| $h_1$ | 200 | 2 |
| $h_2$ | 150 | 5 |
| $h_3$ | 100 | 3 |
| $h_4$ | 300 | 2 |
| $h_5$ | 350 | 1 |
| $h_6$ | 100 | 8 |

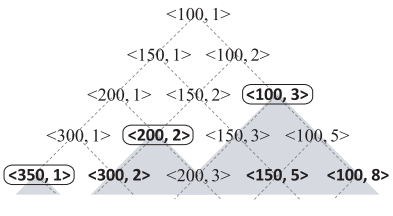



**Fig. 1.** Example relation and query lattice.

Let us illustrate the algorithm by using the example table $R$ in Fig 1. We assume the preference attributes to be *Price* and *Rating* and the preference

to be $Price \rightarrow min$ and $Rating \rightarrow min$. The projections of $R$ over $Price$ and $Rating$, sorted in ascending order, are as follows:

- $Price : 100, 150, 200, 300, 350$
- $Rating : 1, 2, 3, 5, 8$

The diagram in Fig 1 shows a part of the query lattice derived from $R$. In this diagram, queries having non-empty answers are emphasized by bold letters. Queries contributing to the skyline are enclosed by rounded rectangles. Gray triangles in the diagram represent areas dominated by queries in the skyline. As we can see in the diagram, the top of the query lattice is $\top = \langle 100, 1\rangle$. Therefore, we start the first iteration of the algorithm with $C = \{\langle 100, 1\rangle\}$ and $F = \emptyset$. At the end of each subsequent iteration of the while-loop, the contents of $C$ and $F$ change as follows:

- end of 1st iteration: $C = \{\langle 150, 1\rangle, \langle 100, 2\rangle\}, F = \emptyset$
- end of 2nd iteration: $C = \{\langle 200, 1\rangle, \langle 150, 2\rangle, \langle 100, 3\rangle\}, F = \emptyset$

In the third iteration, the query $\langle 100, 3\rangle \in C$ has a non-empty result therefore $\langle 100, 3\rangle$ leaves $C$ and enters $F$. We then consider the successors of the queries left in $C$. $\langle 150, 3\rangle \in succ(\langle 150, 2\rangle)$ is dominated by $\langle 100, 3\rangle \in F$ therefore $\langle 150, 3\rangle$ is omitted from $C'$, which accumulates candidate queries for the next iteration.

- end of 3rd iteration: $C = \{\langle 300, 1\rangle, \langle 200, 2\rangle\}, F = \{\langle 100, 3\rangle\}$
- end of 4th iteration: $C = \{\langle 350, 1\rangle\}, F = \{\langle 100, 3\rangle, \langle 200, 2\rangle\}$
- end of 5th iteration: $C = \emptyset, F = \{\langle 100, 3\rangle, \langle 200, 2\rangle, \langle 350, 1\rangle\}$
  (the algorithm stops here)

The correctness of the algorithm is easily established by observing that the algorithm explores the lattice completely (this is guaranteed by the second property of the *succ* function in the above Lemma), retaining only maximal queries (this is guaranteed by the test on dominance performed by the algorithm and also by the fact that the successors of a query are all dominated by it, as stated by the first property of the *succ* function in the above Lemma).

Formally, we will denote as $SQST(R, \mathcal{B})$ the result of the $SQST$ algorithm having $R$ and $\mathcal{B}$ as the input non-empty relation and preference attributes, respectively. On the basis of the above observations, we state the following:

**Proposition 1** *For every relation $R$ and preference attributes $\mathcal{B}$ over $R$,*

$$ans(\bigvee SQST(R, \mathcal{B}), R) = SKY(R, \mathcal{B})$$

We note that, as all queries in $F$ are conjunctive and any two queries in $F$ differ in at least one conjunct, the answers making up the skyline actually form a partition of the skyline. This is an interesting observation when combined with the notion of rank of a query in the query lattice.

**Definition 3 (Rank of a query)** *The rank of a query $q$ in the query lattice is defined as follows: if $q$ is the root query then $rank(q) = 0$ else $rank(q)$ is the maximum length of path from the root query to $q$*

Clearly, the higher the rank of a query the less the tuples in its answer are preferred.

Now, as the skyline of $R$ is partitioned by the answers to the queries in $F$, one can ask new kinds of queries. For example, one can ask the query: "give me the best tuples from the skyline". The answer to this query will be the answer to the query of lowest rank in $F$. Similarly, one can ask the query: "give me the ranks of all tuples in the skyline". This query will return a set of ranks, thereby giving a useful information as to how far are the tuples of the skyline from the most preferred tuples. A detailed discussion of the relationship between "most preferred" and "non-dominated" tuples is given in a forthcoming paper.

## 4 Skylines of joins

We now consider the computation of the skyline over the join of two tables $R_1$ and $R_2$. To this end, we introduce the necessary concepts.

Let $R_1$ and $R_2$ be relations with $A_1^1, \ldots, A_1^{n_1}$ and $A_2^1, \ldots, A_2^{n_2}$ as attributes, respectively, and let $\mathcal{B}_i = \{B_i^1, \ldots, B_i^{k_i}\}$, $k_i \leq n_i$, be a set of attributes of $R_i$, called the *preference attributes* of $R_i$, for $i = 1, 2$.

We shall denote as $(Q_1, \leq_1)$ and $(Q_2, \leq_2)$ the query lattices over $R_1$ and $R_2$, respectively. Moreover, $\otimes_i$ and $\oplus_i$ will stand for the least upper bound and the greatest lower bound of any two preference queries over $R_i$, respectively.

Let $\mathcal{J}_i = \{J_i^1, \ldots, J_i^l\}$, be a set of attributes of $R_i$ disjoint form the preference attributes $\mathcal{B}_i$, for $i = 1, 2$. A *join* over $\mathcal{J}_1$ and $\mathcal{J}_2$ is a relational equijoin $R_1 \bowtie_e R_2$, whose join expression $e$ is given by $J_1^1 = J_2^1, \ldots, J_1^l = J_2^l$. In order to simplify the model and with no loss of generality, we will consider the join attributes to be the same for the two relations, that is $\mathcal{J}_1 = \mathcal{J}_2$, and moreover to consist of a single attribute $J$, that is $e$ is given by $J = J$.

Intuitively, a preference query over a join $R_1 \bowtie_e R_2$ is a $(k_1 + k_2)$-tuple whose first $k_1$ elements make up a query in $Q_1$ and whose last $k_2$ elements make up a query in $Q_2$. In order to simplify notation, we will commit a slight abuse and write $\langle q_1, q_2 \rangle$ to represent a preference query over $R_1 \bowtie_e R_2$, where $q_1 \in Q_1$ and $q_2 \in Q_2$. As a consequence, the set of preference queries over $R_1 \bowtie_e R_2$, that we shall denote as $Q_{\bowtie}$, is given by the Cartesian product of the set of preference queries over $R_1$ and $R_2$, that is:

$$Q_{\bowtie} = Q_1 \times Q_2$$

Now, let $\leq_{\bowtie}$ be the Pareto preference relation over $Q_{\bowtie}$. As we have already seen in the previous Section, $(Q_{\bowtie}, \leq_{\bowtie})$ is a complete lattice. Moreover, it is not difficult to see that:

**Proposition 2** *$(Q_{\bowtie}, \leq_{\bowtie})$ is the product of $(Q_1, \leq_1)$ and $(Q_2, \leq_2)$. That is, letting $q$ and $q'$ be preference queries in $Q_{\bowtie}$ such that $q = \langle q_1, q_2 \rangle$ and $q' = \langle q'_1, q'_2 \rangle$, where $q_1, q'_1 \in Q_1$ and $q_2, q'_2 \in Q_2$, we have:*

1. *$q \leq_{\bowtie} q'$ iff $q_1 \leq_1 q'_1$ and $q_2 \leq_2 q'_2$*
2. *$q \otimes_{\bowtie} q' = \langle q_1 \otimes_1 q'_1, q_2 \otimes_2 q'_2 \rangle$*
3. *$q \oplus_{\bowtie} q' = \langle q_1 \oplus_1 q'_1, q_2 \oplus_2 q'_2 \rangle$*

From now on we shall simplify notation by omitting subscripts, unless this creates ambiguity.

We shall call *join* values the set of tuples $V = X_1 \cap X_2$, where:

$$X_i = \pi_J(R_i) \quad \text{for } i = 1, 2$$

By definition of join, a tuple $t$ in $R_i$ contributes to the join if and only if its projection over the join attribute $J$ is in $V$, that is $t.J \in V$, for $i = 1, 2$. Likewise, a query $q$ in $Q_i$ *may* contribute to the skyline of the join only if it occurs in the join. For each join value $v \in V$, we define the $v-$partition of $R_i$, denoted $S_i(v)$, as follows :

$$S_i(v) = \pi_{\mathcal{B}_i}(\sigma_{J=v}(R_i)) \quad \text{for } i = 1, 2$$

In practice, each $v-$partition includes the queries that contribute to the join $R_1 \bowtie R_2$. In particular:

$$\pi_{\mathcal{B}_1 \cup \mathcal{B}_2}(R_1 \bowtie R_2) = \bigcup_{v \in V} S_1(v) \times S_2(v)$$

$v-$partitions play an important role in determining the skyline of the join $R_1 \bowtie R_2$ *without* computing the join.

**Proposition 3** *A query $\langle q_1, q_2 \rangle \in \pi_{\mathcal{B}_1 \cup \mathcal{B}_2}(SKY(R_1 \bowtie R_2, \mathcal{B}_1 \cup \mathcal{B}_2))$ iff for some join value $v \in V$, $q_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v), \mathcal{B}_i))$ for $i = 1, 2$ and for no other $v' \in V$ there exists skylines $q'_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v'), \mathcal{B}_i))$ such that $q'_i \leq q_i$ for $i = 1, 2$.*
*Proof: ($\rightarrow$) If for some join value $v \in V$, $q_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v), \mathcal{B}_i))$ for $i = 1, 2$ then $\langle q_1, q_2 \rangle \in \pi_{\mathcal{B}_1 \cup \mathcal{B}_2}(R_1 \bowtie R_2)$, and moreover for 1 in Proposition 2 there exists no other query $\langle q'_1, q'_2 \rangle \in \pi_{\mathcal{B}_1 \cup \mathcal{B}_2}(R_1 \bowtie R_2)$ where $q'_i \in S_i(v)$ for $i = 1, 2$ such that $\langle q'_1, q'_2 \rangle \leq \langle q_1, q_2 \rangle$. If for no other $v' \in V$ there exists skylines $q'_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v'), \mathcal{B}_i))$ such that $q'_i \leq q_i$ for $i = 1, 2$ then again from 1 in Proposition 2 there exists no $\langle q'_1, q'_2 \rangle$ such that $\langle q'_1, q'_2 \rangle \leq \langle q_1, q_2 \rangle$. Hence $\langle q_1, q_2 \rangle \in \pi_{\mathcal{B}_1 \cup \mathcal{B}_2}(SKY(R_1 \bowtie R_2, \mathcal{B}_1 \cup \mathcal{B}_2))$.*
*($\leftarrow$) Suppose not. Then, either (a) for no join value $v \in V$, $q_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v), \mathcal{B}_i))$ for $i = 1, 2$ or (b) there exists a join value $v' \in V$ and skylines $q'_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v'), \mathcal{B}_i))$ such that $q'_i \leq q_i$ for $i = 1, 2$. In case (a), there are two sub-cases: (a1) for no join value $v \in V$, $q_i \in S_i(v)$ for $i = 1, 2$. In this case, $\langle q_1, q_2 \rangle \notin \pi_{\mathcal{B}_1 \cup \mathcal{B}_2}(R_1 \bowtie R_2)$, against the hypothesis. (a2) for some join value $v \in V$, $q_i \in S_i(v)$, but either $q_1 \notin \pi_{\mathcal{B}_1}(SKY(S_1(v), \mathcal{B}_1))$ or $q_2 \notin \pi_{\mathcal{B}_2}(SKY(S_2(v), \mathcal{B}_2))$. Then let $q'_i$ be such that $q'_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v), \mathcal{B}_i))$. Such $q'_i$ are guaranteed to exist because $S_i(v)$ is finite and partially ordered by Pareto preference. By hypothesis, either $q'_1 \neq q_1$ or*

$q'_2 \neq q_2$. *Then* $\langle q'_1, q'_2\rangle \in \pi_{\mathcal{B}_1\cup\mathcal{B}_2}(R_1 \bowtie R_2)$ *and by 1 in Proposition 2* $\langle q'_1, q'_2\rangle \leq \langle q_1, q_2\rangle$, *therefore* $\langle q_1, q_2\rangle \notin \pi_{\mathcal{B}_1\cup\mathcal{B}_2}(SKY(R_1 \bowtie R_2, \mathcal{B}_1 \cup \mathcal{B}_2))$, *against the hypothesis. (b) If there exists a join value* $v' \in V$ *and skylines* $q'_i \in \pi_{\mathcal{B}_i}(SKY(S_i(v'), \mathcal{B}_i))$ *such that* $q'_i \leq q_i$ *for* $i = 1, 2$ *then* $\langle q'_1, q'_2\rangle \in \pi_{\mathcal{B}_1\cup\mathcal{B}_2}(R_1 \bowtie R_2)$ *and by 1 in Proposition 2* $\langle q'_1, q'_2\rangle \leq \langle q_1, q_2\rangle$, *therefore* $\langle q_1, q_2\rangle \notin \pi_{\mathcal{B}_1\cup\mathcal{B}_2}(SKY(R_1 \bowtie R_2, \mathcal{B}_1 \cup \mathcal{B}_2))$, *against the hypothesis.*

We now provide an algorithm for computing the skyline queries over the join of two tables $R1$ and $R_2$ (without computing the join.). The algorithm exploits Proposition 3, and uses the $SQST$ algorithm for computing skylines over $v$-partitions of the given tables. As a result, the algorithm returns two sets of queries, one over $R_1$, the other over $R_2$, for selecting from each table the tuples that will generate the skyline of the join $R_1 \bowtie R_2$, and only those.

**Algorithm** The Skyline Queries over a Join ($SQJ$)

*Input* Non-empty relations $R_1$ and $R_2$, non-empty sets $\mathcal{B}_1$ and $\mathcal{B}_2$ of preference attributes in $R_1$ and $R_2$, respectively, and the join values $V$.

*Data* $G$ is a set variable initialized to empty and used to accumulate all candidate skyline queries, resulting from the Cartesian products of the skyline queries over the same join value. $R$ is a set variable where the skyline of $G$ is computed. $P_i$ and $F_i$ (for $i = 1, 2$) are auxiliary set variables, used to store $v$-partitions and final results, respectively.

*Output* Two sets of queries, one over $R_1$ and the other over $R_2$.

*Method*

$G \leftarrow \emptyset$
**for all** $v \in V$ **do**
  $P_1 \leftarrow SQST(S_1(v), \mathcal{B}_1)$
  $P_2 \leftarrow SQST(S_2(v), \mathcal{B}_2)$
  $G \leftarrow G \cup (P_1 \times P_2 \times \{v\})$
**end for**
$R \leftarrow SQST(G, \mathcal{B}_1 \cup \mathcal{B}_2)$
$F_1 \leftarrow \pi_{\mathcal{B}_1\cup\{J\}}(R)$
$F_2 \leftarrow \pi_{\mathcal{B}_2\cup\{J\}}(R)$
**return** $F1$, $F_2$

We note that the algorithm operates in three passes:

1. In the first pass, it gathers (in $G$) all candidate results by looping over all join values. This is in fact required by the first condition of Proposition 3, which states that $\langle q_1, q_2\rangle$ is in a skyline query of the join if both $q_1$ and $q_2$ are skyline queries over the $v$-partitions for the same join value $v$.
2. In the second pass, it removes from $G$ the compound queries that are dominated by some other compound query, as required by the second condition of Proposition 3.

3. Finally, it slices the compound queries vertically, in order to obtain queries over $R_1$ (these are stored in $F_1$) and queries over $R_2$ (in $F_2$). Notice that the join attribute $J$ must be transferred all along in order to generate the correct queries.

Clearly, there is no other way of proceeding since it is necessary to obtain *all* candidate queries in order to apply the second condition of Proposition 3.

Formally, we will denote as $SQJ(R_1, R_2, \mathcal{B}_1, \mathcal{B}_2)_i$ the $i-$th result of the $SQJ$ algorithm having $R_1$ and $R_2$ as the input non-empty relations and $\mathcal{B}_1$ and $\mathcal{B}_2$ as preference attributes, respectively. For readability, we will abbreviate $SQJ(R_1, R_2, \mathcal{B}_1, \mathcal{B}_2)_i$ as $SQJ_i$.

On the basis of the above observations and of Proposition 3, we therefore state the following:

**Proposition 4** *For every pair of relations $R_1$ and $R_2$ and preference attributes $\mathcal{B}_1$ and $\mathcal{B}_2$ over them,*

$$SQST(ans(\bigvee SQJ_1, R_1) \bowtie ans(\bigvee SQJ_2, R_2), \mathcal{B}_1 \cup \mathcal{B}_2) = SKY(R_1 \bowtie R_2, \mathcal{B}_1 \cup \mathcal{B}_2)$$

Let us demonstrate the SQJ algorithm by using table $R_1$ (hotels) and $R_2$ (restaurants) in Fig. 2. Suppose we want to find best combinations of hotels and restaurants in the same "Location", by minimizing "Price", "Rating", "Distance" and "Location" (we took this example from [23]). In this case, the preference attributes are $\mathcal{B}_1 = \{Price, Rating\}$, $\mathcal{B}_2 = \{Distance, Ranking\}$ and the join attribute is $\mathcal{J} = \{Location\}$. From the intersection of values appearing in the *Location* attribute in $R_1$ and $R_2$, we can obtain join values $V = \{A, B, C\}$.

In the algorithm, firstly we gather candidate skyline queries for each join value. In the example, for a join value $A \in V$, we obtain v-partition $S_1(A)$ and $S_2(A)$. Then we compute skylines $P_1, P_2$ (emphasized in tables $S_1$, $S_2$ by bold letters) from $S_1(A)$, $S_2(A)$ by the applying the SQST algorithm. Finally we make a Cartesian product $P_1 \times P_2 \times \{A\}$ and append it to $G$ that accumulates candidate skyline queries.

After iterating this candidate gathering process for every join value, we apply SQST to $G$ to obtain the skyline (emphasized in tables $G$ by bold letters) over the join $R_1 \bowtie R_2$. It is important to note the difference of size between $R_1 \bowtie R_2$ and $G$. In this example, $R_1 \bowtie R_2$ produces 12 tuples but $G$ contains 8 tuples. Therefore, we can compute the skyline from $G$ with less cost than by computing directly from $R_1 \bowtie R_2$.

In the example, $\langle h_5, r_5 \rangle$ is not a skyline in the join because for the join value $B$, we have $h_2$ dominating $h_5$ and $r_1$ dominating $r_5$. On the other hand, neither $h_6$ nor $r_4$ are skyline in their table, but they form a skyline in the join because they are skylines in their join group and there is no other group in which both are dominated ($h_6$ is dominated by a query $h_3$ in the $A$ group, whereas $r_4$ is dominated by $r_2$ in the $C$ group, and there is no single group in which both are dominated.)

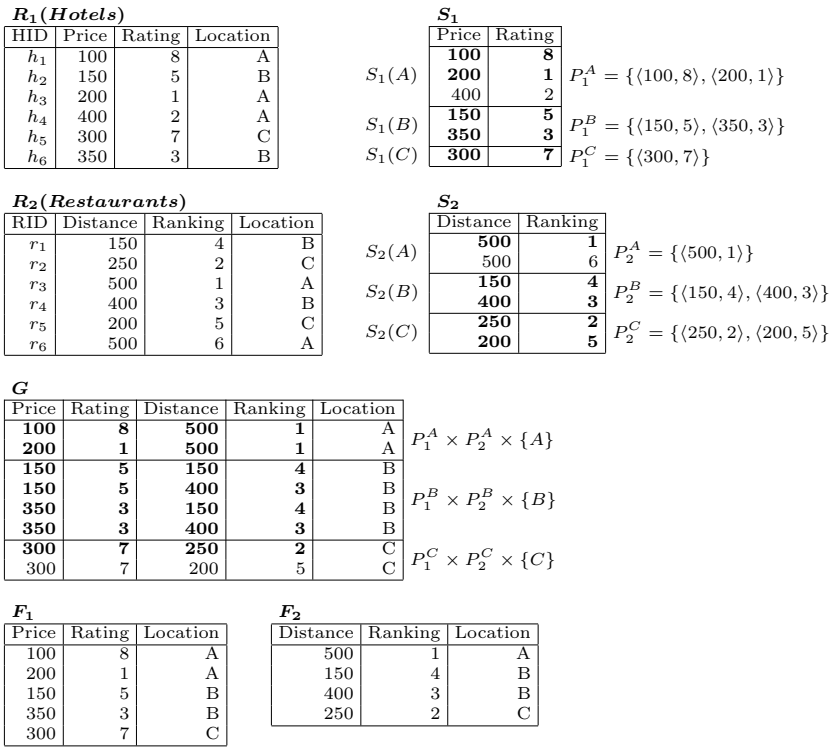



**Fig. 2.** An example of skyline over the join of two tables

## 5 Concluding Remarks

We have seen a novel approach to computing the skyline of a relational table with respect to preferences expressed over one or more attributes with ordered domains. Our approach is based on what we called the query lattice of the table, and our basic algorithm constructs the skyline as the union of the answers to a subset of queries from that lattice — hence *without* directly accessing the table $R$. Therefore, in contrast to all existing techniques, our approach is independent of how the table $R$ is implemented or how its tuples are indexed. We have demonstrated the generality of our approach by computing the skyline over the join of two tables based on the product of their individual query lattices — therefore *without* performing the join.

We note that our method is applicable to a computational geometry setting as well. Indeed, a discrete (finite) $n$-dimensional Euclidean space $S$ can be thought of as a relational table $T(S)$ in which: (a) the attributes are the $n$ dimensions

of $S$ and (b) each tuple of $T(S)$ represents a point in $S$. Moreover, the answer to a query $q = \langle b_1, \ldots, b_k \rangle$ from the query lattice of $T(S)$ is the set of all points of $S$ such that $(B_1 = b_1) \wedge \ldots \wedge (B_k = b_k)$, where $B_1, \ldots, B_k$ are the corresponding dimensions; in other words, the answer to $q$ is the set of points having the same coordinate values over the dimensions $B_1, \ldots, B_k$. Additionally, our method can be applied to the Cartesian product of two or more spaces through the product lattice of the individual spaces. We are currently pursuing work in two different directions, namely refining skyline analysis and applying our approach to a distributed setting:

1. *Refining skyline analysis*
   As we mentioned in section 3, the skyline query returned by our basic algorithm is the disjunction of a set of queries from the query lattice, say $q_1 \wedge \ldots \wedge q_m$; and the answers to these queries actually partition the skyline into disjoint subsets. Moreover, these queries have different ranks, in general. Therefore it now becomes possible to ask finer queries regarding the skyline such as "give me the best tuples of the skyline" (meaning the answer to the query $q_i$ of highest rank), or "return the skyline by presenting the answers to $q_i$'s in increasing order of rank", and so on. In this respect, we would also like to investigate in more detail the relationship between "most preferred tuple" and "non-dominated tuple".
2. *Applying our method to a distributed setting*
   As information systems store bigger and bigger volumes of data today, data management and in particular query processing presents difficult challenges. From the user viewpoint, big volumes of data imply answers of large size. By returning the best tuples (in terms of user preferences), the skyline query relieves the user from having to deal with answers of large size; and having the possibility to further refine the skyline (as mentioned above) further contributes in that direction. However, in recent years, data management and data storage have become increasingly distributed, and distribution presents additional challenges for query processing. Adapting the skyline operator to a distributed setting is one of the research lines that we are currently pursuing. We believe that our approach to skyline computation through query lattices is particularly well suited in a distributed environment, where computation can be distributed and recomposed in the form of the product lattice.

## References

1. I. Bartolini, P. Ciaccia, and M. Patella. SaLSa: computing the skyline without scanning the whole sky. In Proceedings of CIKM, 2006.
2. I. Bartolini, P. Ciaccia, and M. Patella. Efficient sort-based skyline evaluation. ACM Trans. Database Syst., 33(4), 2008.
3. S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In Proceedings of ICDE, pages 421-430, 2001.
4. C.-Y. Chan, P.-K. Eng, K.-L. Tan. Stratified computation of skylines with partially-ordered domains. In Proc. of SIGMOD 2005, pp. 03–214, 2008.

5. J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting. In Proceedings of ICDE, pages 717-816, 2003.
6. B. Cui, H. Lu, Q. Xu, L. Chen, Y. Dai, Y. Zhou. Parallel Distributed Processing of Constrained Skyline Queries by Filtering. In Proc. of ICDE 2008, pp. 546-555, 2008.
7. P. Godfrey, R. Shipley, and Jarek Gryz. Algorithms and analyses for maximal vector computation. The VLDB Journal, vol 16(1), pp. 5–28, 2007.
8. W. Jin, M. Ester, Z. Hu, and J. Han. The multi-relational skyline operator. In Proceedings of ICDE, 2007.
9. W. Jin, M. Morse, J. Patel, M. Ester, and Z. Hu. Evaluating skylines in the presence of equi-joins. In Proc. of ICDE, 2010.
10. N. Koudas, C. Li, A. K. H. Tung, and R. Vernica. Relaxing join and selection queries. In Proceedings of VLDB, 2006.
11. H.T. Kung, F. Luccio, F.P. Preparata. On finding the maxima of a set of vectors. Journal of the ACM, vol. 22(4), pp. 469–476, 1975.
12. M.E. Khalefa, M.F. Mokbel, J.J. Levandoski. Skyline Query Processing for Incomplete Data. In Proc. of ICDE 2008, pp. 556-565, 2008.
13. D. Kossmann , F. Ramsak , S. Rost. Shooting Stars in the Sky: An Online Algorithm for Skyline Queries. In Proc. of VLDB 2002, pp. 275–286, 2002.
14. X. Lin , Y. Yuan , W. Wangnicta , S. Wales. Stabbing the sky: Efficient skyline computation over sliding windows. In Proc. of ICDE 2005, pp. 502–513, 2005.
15. M. Morse , J. Patel , H. V. Jagadish. Efficient Skyline Computation over Low-Cardinality Domains. In Proc. of VLDB 2007, pp. 267–278, 2007.
16. J. Pei , W. Jin , M. Ester , Y. Tao. Catching the best views of skyline: A semantic approach based on decisive subspaces. In Proc. of VLDB 2005, pp. 253–264, 2005.
17. J. Pei , B. Jiang , X. Lin , Y. Yuan. Probabilistic skylines on uncertain data. In Proc. of VLDB 2007, pp. 15–26, 2007.
18. F.P. Preparata, M.I. Shamos. Computational Geometry Computational Geometry, Springer-Verlag, 1985.
19. D. Papadias, Yufei Tao, Greg Fu, Bernhard Seeger. An Optimal and Progressive Algorithm for Skyline Queries. In Proc. of SIGMOD 2003, pp. 467–478, 2003.
20. V. Raghavan, E. Rundensteiner. Progressive result generation for multi-criteria decision support queries. In Proceedings of ICDE, 2010.
21. K.-L. Tan, P.-K. Eng, B.C. Ooi. Efficient Progressive Skyline Computation. In Proc. of VLDB 2001, pp. 301-310, 2001.
22. Y. Tao, X. Xiao, and J. Pei. Subsky: Efficient computation of skylines in subspaces. In Proceedings of ICDE, 2006.
23. A. Vlachou, C. Doulkeridis, N. Polyzotis. Skyline query processing over joins. In Proceedings of SIGMOD 2011, pp. 73–84, 2011.
24. A. Vlachou, K. Nørvåg. Bandwidth-constrained distributed skyline computation. In Proc. of MobiDE 2009, pp. 17–24, 2009.
25. D. Sun, S. Wu, J. Li, and A. K. H. Tung. Skyline-join in distributed databases. In ICDE Workshops, 2008.
26. Q. Wan, R. C.-W. Wong, I. F. Ilyas, M. T. Özsu, and Y. Peng. Creating competitive products. PVLDB, vol. 2(1), 898-909, 2009.
27. P. Wu , C. Zhang , Y. Feng , B.Y. Zhao , D. Agrawal , A.E. Abbadi. Parallelizing skyline queries for scalable distribution. In Proc. of EDBT 2006, pp. 112-130, 2006.
28. Y. Yuan , X. Lin , Q. Liu , W. Wang , J.X. Yu , Q. Zhang. Efficient Computation of the Skyline Cube. In Proc. of VLDB 2005, pp. 241–252, 2005.

29. S. Zhang , N. Mamoulis , S.W. Cheung. Scalable skyline computation using object-based space partitioning. In Proc. of SIGMOD 2009, pp. 483-494, 2009.

# Using FCA for Modelling Conceptual Difficulties in Learning Processes

Uta Priss, Peter Riegler, Nils Jensen

Zentrum für erfolgreiches Lehren und Lernen
Ostfalia University of Applied Sciences
Wolfenbüttel, Germany
www.upriss.org.uk, {p.riegler,n.jensen}@ostfalia.de

**Abstract.** In the natural sciences, mathematics and technical subjects, universities often observe generally low pass rates and high drop out rates in the first years. Many students seem to have conceptual difficulties with technical and mathematical materials. Furthermore, physics education research appears to indicate that even students who are able to pass exams may still not have a good understanding of basic physics concepts. Some researchers use the notion of "misconception" to describe conceptual differences between intuitive notions and accepted scientific notions. A significant body of educational research exists dedicated to overcoming such didactic challenges, but so far not much Formal Concept Analysis (FCA) research has been dedicated to these topics. The aim of this paper is to develop a better understanding of the structure of conceptual difficulties in learning processes using FCA. It is not intended in this paper to develop new educational methods or to collect new data, but instead to analyse existing data and models from an FCA viewpoint.

## 1 Introduction

Education is an interesting application area for Formal Concept Analysis[1] (FCA) because the analysis, representation and development of conceptual structures - in the mind of the learner, and maybe also of the teacher - is an inherent feature of learning and teaching. Because modern e-learning materials and environments tend to accumulate and provide large amounts of data, any technology, such as FCA, developed for structuring and retrieval of information or semantic, conceptual and ontological analysis is implicitly applicable to learning materials as well.

Rudolf Wille the founder of FCA also pioneered the use of FCA for teaching mathematics (Wille, 1995). He published a number of subsequent papers on mathematics restructuring and education - most of them are more general, of philosophical nature and not specifically about FCA. Otherwise, there do not appear to be significant numbers of FCA publications in the educational domain. Examples of FCA applications in this area focus on ontological representations, such as the structuring, retrieval and visualisation of learning materials (Lee, 2005) or the development of an ontology-based

[1] Because this conference is dedicated to FCA, this paper does not provide an introduction to FCA. Information about FCA can be found, for example, on-line (http://www.fcahome.org.uk) and in the main FCA textbook by Ganter & Wille (1999).

courseware management system (Tane et al., 2004) which facilitates browsing, querying and clustering of materials and ontology evolution. Other FCA applications relate to the use of FCA with computer algebra systems (Priss, 2010) and to the meta-analysis of learning materials. For example Pecheanu et al. (2011) use FCA to evaluate and compare open source learning platforms.

Apart from the general data analysis and knowledge representation applications, it is of interest to use FCA to directly analyse the cognitive structures involved in learning processes because presumably learning consists of concept formation, ordering and structuring. Applying FCA in this area is not fundamentally different from other applications where different concept lattices might represent the views of different experts except that in teaching there is an expectation that some conceptual structures are correct and some are not and that the conceptual structures of the students are intended to change.

One obvious difficulty is that it is not easy to obtain representations of such cognitive structures. Psychologists have developed methods for eliciting and externally representing mental models. A number of applications of FCA in the psychological domain have been described, for example, by Spangenberg and Wolff (1991). Al-Diban and Ifenthaler (2011) discuss the comparison of two methods for eliciting and analysing mental models of students, one of which uses FCA. These methods build on a qualitative analysis of data, including transcribed and coded textual protocols, and data collected from specific tests where subjects order concepts in if-then relations. A disadvantage of these methods is that it is not clear whether they could be applied to data observed in real teaching situations (instead of collected from tests) because real data might not have sufficient structure and detail. Furthermore at least in the Al-Diban and Ifenthaler study, the focus was on declarative knowledge, that is whether students know certain facts, not so much on degrees of understanding. An advantage of these methods is that, for example, conceptual gaps and differences among different students and between students and teachers can be detected and analysed.

In addition to the analysis of conceptual structures in learning processes, one might also want to model the conceptual space of a domain for teaching purposes. This has been achieved by Falmagne et al. (2006) who describe a "knowledge state" as the set of particular problems a student can answer in a mathematical topic area. Feasible knowledge states are represented with respect to a precedence graph. This graph is a partially ordered set and not a lattice, but it could be embedded into a lattice and thus modelled with FCA. The idea is that knowledge is ordered: if someone masters a certain mathematical problem then that person can also solve problems that are simpler but may still have to learn to solve problems that are more difficult. Because the precedence graph is not a linear order, different students can take different learning paths. The position of the knowledge state of a student in the graph shows exactly which problems the student can attempt to learn to solve next. Furthermore the student's progress can be exactly measured. Establishing a precedence graph for a mathematical topic area which might consist of hundreds of states is labour-intensive but feasible in a commercial environment such as Falmagne et al.'s ALEKS software tool. There are different means for building such a precedence graph: by questioning experts about the difficulties and pre-

requisites of problems, by analysing student data collected from an e-learning tool or by analysing problem solving processes in the domain.

Currently, ALEKS focuses on mathematics and science topics. It is not clear how far such approaches would be suitable for other non-science domains where it would be difficult to establish a precise ordering of problems. Furthermore, it it may be difficult to evaluate how accurate and useful a particular precedence graph is because user testing of complex e-learning tools is notoriously difficult. If student learning improves while they are using ALEKS, it would be difficult to determine whether that is because of the precedence graph or because of any other of ALEKS's many features.

In summary, general knowledge representation and retrieval aspects of e-learning tools are not any different to such aspects of other textual databases and are covered sufficiently in other domains than educational research. But the analysis of conceptual structures involved in learning processes and the conceptual structuring of domain knowledge for learning purposes is specific to educational research. Currently, FCA appears to be underrepresented in these tasks but it should be very applicable to both of these tasks. It is of particular interest to study differences between the conceptual structures of a learner and of an expert, such as knowledge gaps, discrepancies and misconceptions. A goal for this paper is to involve FCA in the analysis, description and detection of conceptual difficulties, including misconceptions. Section 2 of this paper provides an overview of challenges encountered in teaching conceptually difficult topics. The following three sections show examples of conceptual difficulties for selected mathematical topics: equality in Section 3, translating text into algebraic expressions in Section 4 and the notion of "function" in Section 5. The lattices in these examples are developed from the viewpoint of a teacher who is exploring the difficulties in these areas by modelling formal contexts based on the description of misconceptions in the literature and based on student data.

## 2 Successful and unsuccessful teaching

Physics Education Research studies the problems students have in acquiring physics concepts. Hake (1998) explains that students have initial common-sense beliefs about 'motion' which are in contradiction to current physics theory and which are not improved on by traditional educational methods. Hestenes et al. (1992) published a Force Concept Inventory (FCI) designed to test the students' conceptual understanding of Newtonian mechanics which can be used before and after a physics course to evaluate the educational success of the course. The test is written in a language that is accessible to people who have never taken physics courses but the test is quite different from standard exams which students may be able to pass by memorising and applying formulas. The FCI test purely examines conceptual knowledge. It shows that there is no correlation between standard exam results and test results of individual students. Many students do not change their incorrect beliefs about physics concepts when they are taking physics classes. The revelation that traditional teaching methods are largely ineffective (Hake, 1998) led physics professors to search for alternative teaching methods and led to the establishment of Physics Education Research as a field of study. It seems that misconceptions are particularly visible in physics education because, on the one

hand, people have naive physics theories about natural laws based on observation and experience and, on the other hand, scientific physics theory describes concepts and laws with mathematical precision which are experimentally verifiable but which sometimes contradict naive observation and experience. Outside the natural sciences, concepts are not usually definable and verifiable with such rigour and precision. But the insights of Physics Education Research should also be relevant for the other natural sciences and mathematics (the latter as discussed by Riegler (2010)).

A conclusion of Physics Education Research is that teaching methods involving interactive engagement (Hake, 1998) tend to be more successful in improving the conceptual understanding of students than traditional teaching methods. Interactive engagement is achieved by questioning and challenging students to think instead of just memorising facts. Several factors appear to be contributing to the success of interactive engagement teaching, including cognitive, social constructive and psychological factors.

From a cognitive viewpoint, it has been known for many years (Auble and Franks, 1978) that effort toward comprehension improves recall, i.e., if someone makes an effort at finding a solution before the solution is presented, recall is higher than if a solution is presented right away. Furthermore, active recall is more beneficial for long-term retention than passive exposure (Ellis, 1995). Conway et al. (1992) report that coursework marks are a better predictor for long-term retention than exam marks, possibly because creating a piece of coursework requires the students to be involved with the subject matter at a deeper level than when they reiterate facts during an exam. Conway et al.'s (1992) paper also confirms other observations of Physics Education Research with respect to other domains: procedural knowledge (where students learn something by doing it) is retained much better than declarative knowledge. Students who take only one course in a subject domain tend to forget it completely after a few years. In particular although they might remember some isolated facts, their understanding of the subject is first to disappear - presumably because they never really understood it in the first place. Students who take several courses on a topic and achieve a certain level of proficiency and understanding will retain a large portion of their knowledge for a long time. Thus if interactive engagement teaching leads to a better understanding of a subject, it will help students to remember what they have learnt more permanently than just until the end of the term.

An example of interactive engagement teaching is Mazur's (1996) peer instruction which is even feasible in large classes. Using peer instruction, a lecturer pauses a lecture with challenging questions which the students discuss among each other. Apparently students do not change their conceptual knowledge just because a teacher provides them with facts or a good explanation or even with a demonstration. But if they discuss questions among each other, the students who do have correct conceptual understanding tend to be able to convince their peers. In addition to the cognitive aspects of interactive engagement learning, there seems to be a social component involved: peer pressure seems to be a stronger motivation for people to question and change their beliefs than explanation or observation.

Last but not least, psychological aspects are involved in learning processes. Devlin (2000) argues that mathematicians are psychologically different to non-mathematicians because mathematicians think about mathematical objects in an emotional, associa-

tive manner in the same way as other people think about physical or even animate objects. For example, mathematicians might attribute emotional features to numbers and other abstract objects. Other psychological aspects are involved when people experience clashes between observation and scientific explanations, as for example in optical illusions which are clashes of visual perception and logical, geometrical explanations. Some people perceive clashes as emotionally upsetting. A famous example is the Monty Hall problem[2] about the winning chances in a game show. When Marilyn Vos Savant discussed it and similar problems in her column in the TV magazine Parade, readers responded with angry, emotional letters: "I will never read your column again[3]" or "As a professional mathematician, I'm very concerned with the general public's lack of mathematical skills. Please help by confessing your error and in the future being more careful[4]." - written by someone with a Ph.D who was wrong! One can speculate that animals have evolved probabilistic intuitions in order to make survival decisions which evoke strong emotional responses when challenged. Another example of intuitions contradicting mathematical probability is the belief which many people have that the longer they have played in the lottery without winning, the more likely it is that they are going to win the next time they play. Again this belief tends to have an emotional component as anybody can observe who has ever discussed it with lottery players.

In summary, teaching a topic which contradicts the existing conceptual structures which the students have is challenging. Methods such as peer instruction can help to overcome cognitive, social constructivist and emotional obstacles. Clearly, not all topics evoke such difficulties and some can be taught with more standard teaching methods. Thus it would be useful for a teacher to know in advance which areas of the subject domain are going to produce conceptual difficulties and which not. McDermott (2001) argues that there is only a limited number of re-occurring conceptual difficulties which tend to be experienced by all students similarly. The idea for this paper is that FCA might provide useful methods for detecting and analysing conceptual difficulties. Although McDermott (2001) emphasises that just detecting misconceptions is not sufficient for improving teaching, providing a better understanding of the conceptual structures of misconceptions is going to be beneficial for teachers. In the following, three examples of conceptual difficulties in mathematics education are analysed using FCA.

## 3 Conceptual difficulties of the equality sign

Prediger (2010) discusses problems pupils are having with developing an appropriate conceptual model of equality. In primary school, pupils tend to experience the equal sign as a request to calculate something. For example, "$2 + 3 =$" might be printed in a textbook. Prediger calls this the operational use because pupils are requested to perform an operation. Apparently, this can lead to difficulties later when the equal sign is used in its more general algebraic meaning (or its "relational" meaning). For example, Prediger quotes the case of a pupil who says that the equation $24 \times 7 = 20 \times 7 + 4 \times 7$ is

[2] http://en.wikipedia.org/wiki/Monty_Hall_problem

[3] Parade Magazine, July 27, 1997

[4] http://www.marilynvossavant.com/articles/gameshow.html

wrong because "$24 \times 7$ does not equal 20" and the case of a pupil who writes "$1 \times 10 = 10 + 110 = 120$". Prediger then discusses the difficulties which prospective teachers might encounter in understanding the problems these pupils are having. In her analysis she distinguishes operational, relational and specification uses (such as defining $x := 4$) of the equal sign. She divides the relational use further into symmetric identities ($4 + 5 = 5 + 4$), general equivalences ($(a - b)(a + b) = a^2 - b^2$), searching for unknowns ($x^2 = 6 - x$) and contextual uses ($a^2 + b^2 = c^2$) where the variables are meaningful in a context, such as characterising a right-angled triangle.

To demonstrate the use of FCA in this area we have modelled the problem as a formal context. The formal objects are examples of uses of the equal sign, inequality ($>$) and equivalence ($\Leftrightarrow$). Furthermore we added basic operations from programming languages: not-equal ($!=$), test for equality ($==$) and Boolean operators (&&). Four of the formal attributes are from Prediger's classification: "operation", "contextual", "definition" (i.e. specification) and "law" (i.e equivalence). Here, "operation" refers to rule-based drills where the students solve a problem in a precisely taught manner and the equality sign is always read from left to right. A "definition" for other symbols than "$=$" defines a set of possible values for a variable (e.g., $i > 1$). Furthermore, two attributes have been added which distinguish whether the statements are true for all values of the variables or just for some. Prediger's "unknowns" has been replaced with "test" as a request to evaluate an expression with respect to variables with given values.

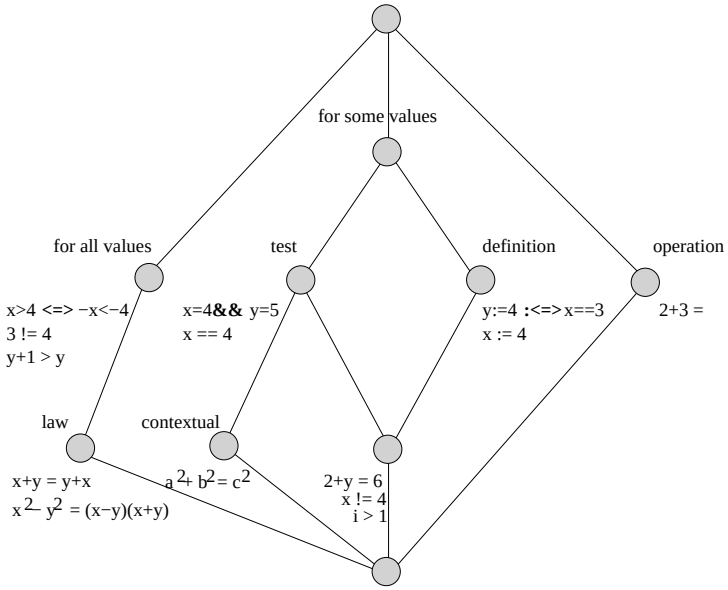



**Fig. 1.** Equation, assignment and comparison operators

The resulting concept lattice (Fig. 1) shows a classification which is slightly different from Prediger's list[5]. The operational use of the equal sign is not connected to any of the other uses. Although this results directly from the definition of the formal attribute "operation", it represents implicit structure which the authors were not aware of before the lattice was constructed. The separation of "operation" from the other concepts provides a graphical explanation as to why students may find it particularly difficult to progress from an operational use to the more general algebraic use.

The extensions of the concepts under "for all values" contain tautologies. But there is a distinction made between those which students have to specifically learn (under "law") in order to understand how the operators work and those which just happen to be true. There are three different reasons why a statement might only be true for some values of the variables: the variables are defined in the statement; it is to be tested whether the statement is true (or for which variables it is true); and in the contextual use, the statement is only true in some contexts and thus describes such contexts. Some interesting cases are under both "test" and "definition": an equation $2 + y = 6$ is both an implicit definition of $y$ and a request to evaluate which values of $y$ yield the equation to be true. For the use of $!=$ and $<$, it depends on the context whether the statements are meant to be evaluated for their truth value or whether they are meant to define a range for their variables.

## 4 Conceptual difficulties of translating text into algebraic expressions

A well-known conceptual difficulty that mathematics students experience pertains to the translation of text into algebraic expressions. Clement (1982) conducts an experiment where he asks students to write an equation using the variables $S$ and $P$ to represent "there are six times as many students as professors". His findings are that only 40-60% of the students produce a correct answer ($S = 6P$). The most common incorrect answer is $6S = P$. He provides two reasons for the incorrect answer: first, some students simply translate the sentence into mathematical symbols in the same word order. Second, some students use a static comparison pattern or, in other words, an incorrect schema where $S$ and $P$ do not represent numbers, but instead units of students and professors. This is in the same manner as how $m$ and $km$ are used in $1000m = 1km$. In this case, $m$ and $km$ are not variables but represent a fixed "1 to 1000" relationship. It is not possible to substitute arbitrary values for $m$ or $km$, but $m$ can be substituted with $\frac{1}{1000}km$ and $km$ can be substituted with $1000m$, yielding, for example, $2000m = 2000 \times \frac{1}{1000}km = 2km$. One difference between units and variables for numbers is that it is not usually acceptable to insert a multiplication sign between a number and its unit.

Table 1 summarises the differences between the two conceptual systems: in the first one the letters represent units, in the second, algebraic one the letters represent variables for numbers. The first conceptual system has a meronymic (part-whole) quality. A certain, fixed aggregate of the smaller units constitutes the larger unit. The extension

[5] In Fig. 1, in the statements with more than one operator the relevant one is printed in bold face.

of $6s = 1p$ is really a fixed "6 to 1" relation which is expressed in $s$ and $p$. In contrast, the algebraic conceptual system represents normal algebraic use of variables. The extension of $s = 6 * p$ consists of the pairs of values that can be substituted for $s$ and $p$. The table also shows examples of intensionally equivalent and implied expressions. In the meronymic conceptual system, $s$ is indeed smaller than $p$ because it represents the unit "student" which somehow has less value than the unit "professor". It is possible to interpret the units $s$ and $p$ as algebraic variables but not as "numbers of". For example, $s$ could represent the money paid by a student and $p$ the money earned by a professor.

**Table 1.** Two conceptual systems for the use of letters in equations

| the letter means: | unit | variable for number |
|---|---|---|
| conceptual system: | meronymic | algebraic operation |
| representation: | $6s = p$ | $s = 6 * p$ |
| extension: | relation: 6 to 1 | substitution: $\{(6,1),(12,2),(18,3),...\}$ |
| intensionally equivalent: | $s = \frac{1}{6}p$ | $s/p = 6/1$ |
| intensionally implied: | $s < p$ | $s > p$ |

It should be emphasised that both conceptual systems in Table 1 are consistent. In everyday experience, meronymic, unit-based conceptual systems may be much more common than algebraic ones. Thus it should be expected that students who have not yet made much progress towards learning algebra or people who have not recently used algebra would prefer the meronymic, unit-based representation. Ben-Ari (1998) argues that from a constructivist educational viewpoint, students always already have existing mental models which may contradict scientific models. Teachers need to understand the students' mental models and to build on them instead of discarding them as simply being incorrect. In this case the algebraic use of variables must be taught to people who already employ a different, meronymic conceptual system. They need to learn to use the different systems in different circumstances.

As a further analysis, we have coded the data from two student interviews (Clement, 1982) in a content analytic manner and converted them into a formal context. The formal objects are mathematical notations as used or implied by the students. The formal attributes are verbal descriptions made by the students converted into a slightly more formal language. A cross in the formal context means that the student used a verbal expression with respect to a mathematical notation. Mathematical notations and verbal expressions that were used algebraically incorrectly by a student have been highlighted in bold face.

The following observations can be made from the resulting concept lattice in Fig. 2. Even though it was argued that the two conceptual systems in Table 1 are both consistent conceptual systems, in Fig. 2 it appears that the correct statement $s = 6p$ is conceptually better refined than the incorrect one $6s = p$ which is more isolated in the lattice. This is because the lattice combines the data from two student interviews: one student with a correct answer of the problem who provided detailed explanations and reasons for why his answer was correct and another student who produced an incorrect

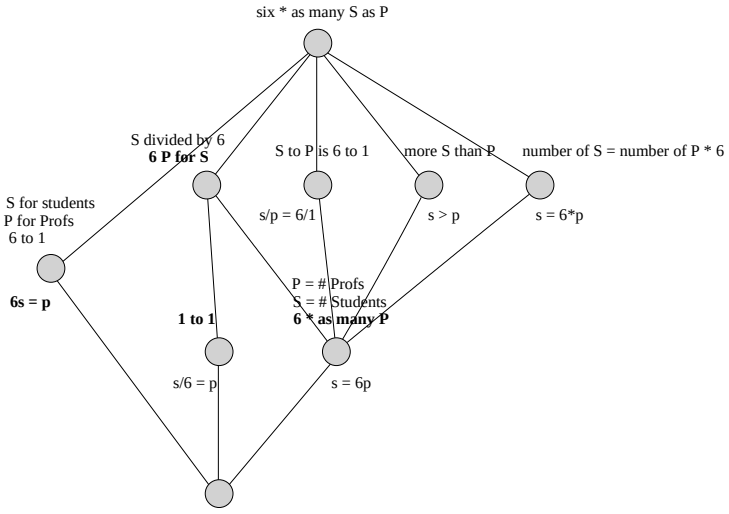



**Fig. 2.** Data from student interviews

solution which does not appear to be very coherent according to the lattice. The student with the correct answer understood that the variables represent "numbers of". The other student said that "S stands for student". It is interesting to observe that the student with the incorrect solution focussed more on the relationship ("6 to 1" and "1 to 1") which is indeed the extension of the incorrect representation according to Table 1. When he talked about "1 to 1" he really meant to express a "fixed correspondence". The student with the correct answer on the other hand demonstrated detailed understanding of algebraic transformations which is why his arguments contained intensionally equivalent and implied statements.

Clement (1982) observes that different, unsuccessful strategies have been tried to help students in finding correct solutions. It is our opinion that all of the strategies mentioned by Clement are methods from within the algebraic conceptual system (for example telling the student to substitute numbers for variables or to determine whether there are more students or professors). Presumably all instructors involved in the experiments were of the opinion that a student's attempt was plain wrong, not that it was part of an internally coherent, but different conceptual system. One can speculate what would happen if the students were somehow taught that there are different conceptual systems for use of letters in equations and how to determine which conceptual system is appropriate for which problem. We suspect that in general in most basic mathematics teaching the modelling aspect (how to determine which type of solution belongs to which type of problem) is not significantly highlighted. Thus most students will not be aware that there are different conceptual structures involved in using mathematics and will not have been taught to analyse their strategies from that aspect on a meta-level. They might be aware that they are not "very good at mathematics" without knowing any reasons for the difficulties encountered.

## 5 Conceptual difficulties of the notion of "function"

The third example discussed in this paper refers to the conceptual difficulties encountered by students in learning the notion of "function". The problem is well-known and has been discussed numerous times (e.g., Leinhardt et al. (1990) and Breidenbach et al. (1992)). Quite often students can recall a correct formal definition of a function, but misconceptions become obvious when they are asked to determine whether something can be represented as a function or not. Leinhardt et al. provide the following list of misconceptions:

- Too narrow understanding of "function". Only functions with certain characteristics (regularity, symmetry, linearity, one-to-one, causal relationship, etc) or which are represented in a certain manner (formula, graph, table) are recognised as functions.
- Correspondence: students often believe that functions must be one-to-one and they might be confused about the difference between many-to-one and one-to-many.
- Linearity: students have a tendency towards linearity. They tend to prefer straight lines in graphs.
- Continuous versus discrete: historically, functions were not allowed to be discontinuous. Students have problems understanding the notion of continuity. They discretise continuous data.
- Representations: problems translating graphs into formulas and vice versa.
- Interpretation of graphs: students have problems with confusing intervals and points, slope and height. They might interpret graphs in a literal, iconic manner.
- Variables: students have problems with the notion of "variable". Some do not accept constant functions as functions.
- Notation: students have problems understanding axes and scales in a graph.

Breidenbach et al. (1992) emphasise the "process conception of function". They argue that an understanding of "function" proceeds from a pre-function over an action to a process stage. In our opinion the notion of "process" is misleading in this case because it implies a temporal progression which is not involved in functions. This is in contrast to functions implemented in a computer where an input is converted into an output in real time so that the output is generated temporally after the input has been processed and the input may be purged from memory after it has been used. Although both Clement (1982) and Breidenbach et al. observe that students often develop a better understanding of mathematical operations if they execute them as computer programs, many features of mathematical objects cannot be adequately represented on a computer (for example infinity) and thus there are limits to the use of computer programs for representing mathematical ideas.

In our opinion, it is not the "process conception of function" that is relevant but instead simply the "concept of function". Breidenbach et al.'s tests for whether students understand the notion of function include: asking students to provide a definition (i.e., an intensional description); asking students to decide whether something is a function or not (i.e., evaluating whether something is in the extension of "function"); and asking students to perform operations with functions (composition and reversion) which demonstrates an understanding of the implied intensional features. Thus all of

the tests are aimed at demonstrating whether or not students have an acceptable concept of function, including extension, intension, subsumption, implication and equivalence. Initially, students appear to have incomplete or disconnected concepts of "function". For example, Breidenbach et al. report that the examples of functions provided by students are more sophisticated than their definitions whereas Leinhardt et al. (1990) state that students can recall an accurate definition of "function" but cannot apply it. In either case there is a mismatch between the extension and intension of "function".

Breidenbach et al. also test whether students develop an abstract understanding of particular functions. For example, they define a complicated function F(a)(b)(c) the meaning of which is "the cth character in the string which is the name of the integer given by the ath power of the integer b". We would argue that although translating a function from one representation (formula) to another (textual representation) is an important aspect of using functions, this particular example is really more a test for intelligence than for an understanding of the notion of "function". Similarly, they use strings as examples of functions. Depending on how much experience students have with programming languages, they may or may not be familiar with the representation of strings as arrays of characters. Thus, interpreting a string as a function depends on the students' programming knowledge, not on an understanding of function. On the other hand, if students do have a programming background, then functions from programming languages can be used to emphasise to students that not all functions are of the form "$f(x) =$".

From an FCA viewpoint, data collected from student interviews and exams can be represented as concept lattices to visualise such conceptual differences. The following attributes are examples of how to characterise an understanding of "function".

- representation: set, equation, graph, verbal description, table, computer program, ...
- constant, linear, quadratic, ...
- causal, non-causal
- discrete, continuous
- 1-to-1, 1-to-many (i.e., the reverse is a function), many-to-1, many-to-many
- finite domain, infinite domain

In Fig. 3 some of these attributes are selected as formal attributes and applied to examples of functions as formal objects. The lattice shows a conceptual structure of "function". In this case, continuous functions with infinite domains tend to have more attributes. For computer programs, it is a matter of choice whether one considers "min(x)" as an abstract function with an infinite domain of possible values $x$ (where $x$ is a set or other container object) or as an actually implemented function with a finite domain. The use of such an expert-designed concept lattice is in comparison with lattices obtained from student data (which we have not included in this paper).

## 6 Conclusion

This paper argues that FCA provides useful methods for analysing conceptual difficulties in learning processes. Teachers can use the construction of formal contexts and concept lattices in order to explore the implicit structures in mathematical notions. The

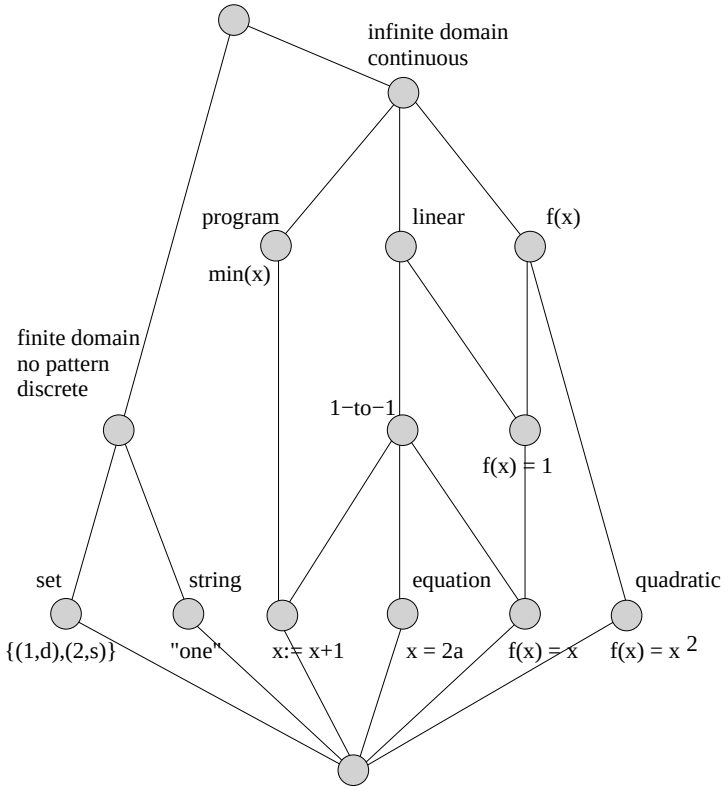



**Fig. 3.** Attributes of functions

data for the lattices can be obtained from the literature on misconceptions, from student interviews or from assessment data. In this manner FCA becomes a tool for exploration and for making underlying assumptions explicit.

Traditional teaching methods are often not successful in teaching conceptually challenging topics. While teaching methods have been developed that are more promising, teachers still need to know when to apply such methods. Thus they need to determine what the specific conceptual challenges are with respect to a certain domain. FCA can be employed as a tool by teachers to familiarise themselves with the materials and to structure difficult topics. Based on the improved understanding of the topics, teachers can then design interactive engagement teaching exercises that focus on the conceptually challenging aspects which the students need to learn.

## References

1. Al-Diban, Sabine; Ifenthaler, Dirk (2011). *Comparison of Two Analysis Approaches for Measuring Externalized Mental Models.* Educational Technology & Society, 14, 2, p. 16-30.

2. Auble, Pamela M.; Franks, Jeffery J. (1978). *The effects of effort toward comprehension on recall.* Memory & Cognition, 6, 1, p. 20-25.
3. Ben-Ari, Mordechai (1998). *Constructivism in computer science education.* SIGCSE Bull, 30, 1, p. 257-261
4. Breidenbach, Daniel; Dubinsky, Ed; Hawks, Julie; Nichols, Devilyna (1992). *Development of the Process Conception of Function.* Educational Studies in Mathematics, 23, p. 247-285.
5. Clement, John (1982). *Algebra Word Problem Solutions: Thought Processes Underlying a Common Misconceptions.* Journal for Research in Mathematics Education, 13, 1, p. 16-30.
6. Conway, M. A., Cohen, G.; Stanhope, N. (1992). *Very long-term memory for knowledge acquired at school and university.* Applied Cognitive Psychology, 6, p. 467-482.
7. Devlin, Keith. 2000. *The Maths Gene. Why everyone has it, but most people don't use it.* Weidenfeld & Nicolson, UK.
8. Ellis, N. C. (1995). *The Psychology of Foreign Language Vocabulary Acquisition: Implications for CALL.* International Journal of Computer Assisted Language Learning (CALL), 8, p. 103-128.
9. Falmagne, Jean-Claude; Cosyn, Eric; Doignon, Jean-Paul; Thiery, Nicolas (2006). *The Assessment of Knowledge, in Theory and in Practice.* LNCS 3874, 949, Springer, p. 61-79.
10. Ganter, Bernhard, & Wille, Rudolf (1999). *Formal Concept Analysis. Mathematical Foundations.* Berlin-Heidelberg-New York: Springer.
11. Hake, Richard R. (1998) *Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses.* American Journal of Physics, 66, 1, p. 64-74.
12. Hestenes, D.; Wells, M.; Swackhamer, G. (1992) *Force Concept Inventory.* Phys. Teach., 30, p. 141-158.
13. Lee, Chien-Sing (2005). *A Formal Context-aware Visualization tool for Adaptive Hypermedia Learning.* WSEAS International Conference on Engineering Education, Athens, Greece.
14. Leinhardt, Gaea; Zaslavsky, Orit; Stein, Mary Kay (1990). *Functions, Graphs, and Graphing: Tasks, Learning, and Teaching.* Review of Educational Research, 60, 1, p. 1-64.
15. Mazur, Eric (1996). *Peer Instruction: A User's Manual*. New Jersey, Prentice Hall.
16. McDermott, Lillian Christie (2001). *Oersted Medal Lecture 2001: "Physics Education Research-The Key to Student Learning"*. Am. J. Phys., 69, 11, p. 1127-1137.
17. Pecheanu, E.; Stefanescu, D.; Dumitriu, L.; Segal, C. (2011). *Methods to evaluate open source learning platforms* Global Engineering Education Conference (EDUCON), IEEE.
18. Prediger, Susanne (2010). *How to develop mathematics-for-teaching and for understanding: the case of meanings of the equal sign.* J. Math. Teacher Educ., 13, p. 73-93.
19. Priss, Uta (2010). *Combining FCA Software and Sage.* In: Kryszkiewicz; Obiedkov (eds.), Proceedings of the 7th International Conference on Concept Lattices and Their Applications, p. 302-312.
20. Riegler, Peter (2010). *Towards Mathematics Education Research - Does Physics Education Research serve as a model?* Proceedings of the 15th MWG Seminar and 8th GFC Workshop, Wismar.
21. Spangenberg, N.; Wolff, K.E. (1991). Interpretation von Mustern in Kontexten und Begriffsverbänden. Actes 26e Séminaire Lotharingien de Combinatoire, p. 93-113.
22. Tane, Julian; Schmitz, Christoph; Stumme, Gerd (2004). *Semantic Resource Management for the Web: An E-Learning Application* Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM.
23. Wille, Rudolf (1995). *"Allgemeine Mathematik" als Bildungskonzept für die Schule.* In: Mathematik allgemeinbildend unterrichten. Biehler et al. (eds). Aulis, p. 41-55.

# Author Index