CLA 2008

Proceedings of the Sixth International Conference on
Concept Lattices and Their Applications

Palacký University, Olomouc, Czech Republic

ISBN 978–80–244–2111–7

Palacký University, Olomouc, Czech Republic

The Sixth International Conference on Concept Lattices and Their Applications

CLA 2008

Olomouc, Czech Republic
October 21–23, 2008

Edited by

Radim Belohlavek
Sergei O. Kuznetsov

CLA 2008

Technical Editor:
Jan Outrata, jan.outrata@upol.cz

Page count:	xii+243
Impression:	70
Edition:	1^{st}
First published:	2008

Published and printed by:
Palacký University, Olomouc, Czech Republic

Organization

CLA 2008 was organized by the Palacký University, Olomouc, and the State University of New York at Binghamton.

Steering Committee

Radim Belohlavek	State University of New York at Binghamton, USA
Sadok Ben Yahia	Faculté des Sciences de Tunis, Tunisia
Jean Diatta	Universit de la Réunion, France
Peter Eklund	University of Wollongong, Australia
Michel Liquiere	LIRMM, Montpellier, France
Engelbert Mephu Nguifo	CRIL CNRS FRE 2499 - IUT de Lens, France
Václav Snášel	VŠB-TU Ostrava, Czech Republic

Program Chairs

Radim Belohlavek	State University of New York at Binghamton, USA
Sergei O. Kuznetsov	State University Higher School of Economics, Moscow, Russia

Program Committee

Radim Belohlavek	State University of New York at Binghamton, USA
Sadok Ben Yahia	Faculté des Sciences de Tunis, Tunisia
Claudio Carpineto	Fondazione Ugo Bordoni, Roma, Italy
Jean Diatta	Université de la Réunion, France
Vincent Duquenne	Université Pierre et Marie Curie, Paris, France
Peter Eklund	University of Wollongong, Australia
Samir Elloumi	Faculté des Sciences de Tunis, Tunisia
Sebastien Ferre	Irisa/Université de Rennes 1, France
Mohamed M. Gammoudi	ISG, Kairouan, Tunisia
Bernhard Ganter	TU-Dresden, Dresden, Germany
Marianne Huchard	LIRMM, Montpellier, France
Ali Jaoua	University of Qatar, Qatar
Bjoern Koester	Webstrategy, GmbH, Germany
Stanislav Krajci	UPJŠ Košice, Slovakia
Sergei Kuznetsov	State University Higher School of Economics, Moscow, Russia
Léonard Kwuida	Bern University of Applied Science, Switzerland
Engelbert Mephu Nguifo	CRIL CNRS FRE 2499 - IUT de Lens, France
Boris Mirkin	Birkbeck College, University of London, UK
Rokia Missaoui	UQO, Gatineau, Canada
Amedeo Napoli	LORIA, Nancy, France
Lhouari Nourine	LIMOS, Université de Clermont Ferrand, France
Sergei Obiedkov	State University Higher School of Economics, Moscow, Russia
Uta Priss	Napier University, Edinburgh, United Kingdom
Sebastian Rudolph	Institute AIFB, University of Karlsruhe, Germany
Yahya Slimani	Faculty of Sciences, Tunis, Tunisia
Petko Valtchev	DIRO, Université de Montréal, Canada
Vilem Vychodil	State University of New York at Binghamton, USA
Rudolf Wille	TU Darmstadt, Germany
Serhiy Yevtushenko	Kiev, Ukraine

Additional Reviewers

Olivier Bedel	Irisa/Université de Rennes 1, France
Chaima Ben Youssef	Faculté des Sciences de Tunis, Tunisia
Ines Bouzouita	Faculté des Sciences de Tunis, Tunisia
Peggy Cellier	Irisa/Université de Rennes 1, France
Alain Gély	LIMOS, Clermont Ferrand, France
Tarek Hamrouni	Faculté des Sciences de Tunis, Tunisia
Mehdi Kaytoue	LORIA, Nancy, France
Mondher Maddouri	Faculté des Sciences de Tunis, Tunisia
Jan Outrata	Palacký University, Olomouc, Czech Republic

Organization Committee

Michal Krupka (Chair)	Palacký University, Olomouc, Czech Republic
Eduard Bartl	State University of New York at Binghamton, USA
Jan Konecny	State University of New York at Binghamton, USA
Tomas Kuhr	Palacký University, Olomouc, Czech Republic
Petr Osicka	Palacký University, Olomouc, Czech Republic
Jan Outrata	Palacký University, Olomouc, Czech Republic

Sponsoring Institutions

PIKE ELECTRONIC, Czech Republic – main sponsor

PIKE Electronic is one of the leading Central European IT companies providing complex service in the field of software application development. The Company delivers or participates on large-scale projects around the globe, cooperating with partners such as Siemens AG, Siemens E&A, SMS DEMAG, TIBCO, Hewlett–Packard, Accenture or Oracle. The main strategic target is specialization in "core business" areas, consolidation, effective distribution and use of long-term experiences and know-how in segments of Process Automation, SOA & Enterprise Application Integration , Information Systems for Health Insurance, Business Intelligence, Telecommunication, Banking, and Logistic.
PIKE ELECTRONIC and the Department of Computer Science of Palacký University have been cooperating for more than ten years. In the recent years, PIKE ELECTRONIC participated in a large research project on Relational Data Analysis which was supported by the Academy of Sciences of the Czech Republic during 2004–2008. The principal investigator was Professor Radim Belohlavek from the Department of Computer Science, the principal co-investigator was Dr. Stanislav Opichal from PIKE. The research results of this project were published in premier international journals and in proceedings of recognized international conferences.

SwissCentrum software, Czech Republic

Faculty of Science, Palacký University, Czech Republic

Table of Contents

Preface

The present volume contains regular papers from CLA 2008, the Sixth International Conference on Concept Lattices and Their Applications. CLA 2008 was held in Olomouc, Czech Republic, from October 21 to October 23, 2008, and was jointly organized by the Palacký University, Olomouc, and the State University of New York at Binghamton. The areas of interest for CLA include various topics related to formal concept analysis, such as foundational aspects, concept lattices and related structures, data mining, attribute implications and data dependencies, algorithms, visualization, data preprocessing, redundancy and dimensionality reduction, information retrieval, classification, clustering, ontologies, and applications in various domains.
The conference received 29 initial submissions from which 19 were accepted as regular papers (acceptance rate for regular papers is 0.66). Contributions to CLA 2008 were refereed by at least two reviewers (2.88 reviews per paper on average) on the basis of their originality, quality, significance, and presentation. When one of the program chairs was involved in a paper, the reviewing process of this paper was managed independently by the other chair.
The program of CLA 2008 consisted of presentations of regular papers and posters, and four invited talks, namely by Bernhard Ganter (TU Dresden, Germany), Petr Hájek (Academy of Sciences of the Czech Republic), George Karypis (University of Minnesota, USA), and Dominik Slezak (Infobright Inc., Canada).
We would like to express our thanks to the authors who submitted their papers to CLA 2008, to the invited speakers, to the members of Program Committee who managed the review of papers, to additional reviewers, to the members of the Organization Committee, as well as to the conference attendees, who all helped make CLA 2008 a successful event.

October 2008

Radim Belohlavek
Sergei O. Kuznetsov
Program Chairs of CLA 2008

Recent Interfaces for Formal Concept Analysis

Bernhard Ganter

TU-Dresden, Dresden, Germany

Abstract. We report on recent developments connecting FCA with other research areas, focussing on such with connections to Dresden. These include approaches to Rough Sets, to Dependencies, to Machine Learning, to Description Logics, and to Algebraic Biology. It seems that the systematic theoretical foundation of FCA paves the way to many different fields and that FCA has a potential to bridge some gaps between different areas.

The GUHA Method and its Meaning for Data Mining

Petr Hájek

Institute of Computer Science,
Academy of Sciences of the Czech Republic

Abstract. The talk presents the history and present state of the GUHA method, its theoretical foundations and its relation and meaning for data mining. (Joint work with M. Holena a J. Rauch.)

Biclustering Methods Meets Formal Concept Analysis

George Karypis

University of Minnesota, Twin Cities, Minneapolis, USA

Abstract. The purpose of this talk is to provide an overview of the problem of bi-clustering, review the various state-of-the-art methods that have been developed in recent years for solving it, and discuss how Formal Concept Analysis methods can benefit from or can be used in bi-clustering.

Rough Sets and Formal Concept Analysis: Foundations and the Case Studies of Feature Subset Selection and Knowledge Structure Formation

Dominik Ślęzak

Infobright Inc., Canada/Poland

Abstract. The theories of Rough Sets (RS) and Formal Concept Analysis (FCA) are well-established from the point of view of both mathematical foundations and real-life applications. The interest in searching for similarities and dissimilarities between RS and FCA has been constantly growing, both with respect to pure theory, as well as with an objective of developing hybrid techniques, better adjusted to practical problems. In this talk, we outline introductory notions of RS and we draw basic lines of its comparison with FCA. As the first case study, we consider the KDD-related problem of feature subset selection and show how to model some new approaches to approximate selection (a more flexible and more practically applicable extension of the classical RS-based feature subset selection principles) in the FCA terminology. As the second case study, we consider the latest Infobright's open source data warehouse platform (www.infobright.org) and we discuss possibilities of improving its performance by using new RS-FCA-based knowledge structures automatically calculated from data.

A Formal Concept Analysis of Harmonic Forms and Interval Structures

Tobias Schlemmer and Stefan E. Schmidt

Technische Universität Dresden,
Fachrichtung Mathematik, 01062 Dresden, Germany
Tobias.Schlemmer@mailbox.tu-dresden.de, midt1@msn.com
http://www.math.tu-dresden.de/~schlemme/

Abstract. While small concept lattices are often represented by line diagrams to better understand their full structure, large diagrams may be too complex to do this. However, such a diagram may still be used to receive new ideas about the inherent structure of a concept lattice. This will be demonstrated for a certain family of formal contexts arising from mathematical musicology. In particular, we investigate how chord patterns can be characterised by their interval structure. For such contexts of pattern structures, it turns out that each corresponding concept lattice incorporates two competing building principles, one emanating from the top the other from the bottom of the lattice.

Key words: formal concept lattice, harmonic form, musicology, interval

1 Introduction

Harmonic forms provide basic notions for the descriptions of chords. Well-known examples are the harmonic form of a major triad which stands for all major chords and, similarly, the minor triad which is the harmonic form of minor chords.

Besides harmonies and chords, harmonic forms play an important role for tuning software like *Mutabor* [1]. In general, applications of mathematical musicology to music software unfold different questions about the mathematical structure of harmonic forms in tone systems.

In the past, formal concept analysis has been applied to various fields of music already (e. g. [3], [4], [5], and [6]). Here as well as in other fields of mathematical musicology (cf. Mazzola et al. [7]), harmonic forms have been analysed to a certain extent, however, there are still plenty of open problems to address.

The description of the structure of harmonic forms leads to concept lattices that are often considered as too large to be drawn meaningfully. However, the diagrams will serve us as a source of information useful for finding more adequate mathematical models.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 9–22,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

2 Qualitative analysis of harmonic forms

To describe musical objects, we need mathematical notions of them. A fundamental one is that of a tone system, which can be modelled as a collection of tones and their interval structure (see also [9], [10]):

A triple $\mathrm{T} = (T, \delta, \mathrm{I})$ is called an *(algebraic) tone system*, if T is a set, $\mathrm{I} = (I, +, 0)$ is an Abelian group and $\delta : T \times T \to I$ is a map such that for all $t_1, t_2, t_3 \in T$ the following hold:

$$\delta(t_1, t_2) + \delta(t_2, t_3) = \delta(t_1, t_3) \text{ and} \tag{1}$$

$$\delta(t_1, t_2) = 0 \text{ iff } t_1 = t_2. \tag{2}$$

The elements of the set T are called *tones* and each subset of T is called a *chord*. The elements of I are considered as *intervals*. For $s, t \in T$, the *interval from s to t* is given by $i \in I$ if $i = \delta(s, t)$ holds; in this case we can agree upon $s + i := t$. For a tone t, the set of *all intervals from t* is given by $I(t) := \delta[\{t\} \times T]$; it follows $t + I(t) = T$ and $I(s) = \delta(s, t) + I(t)$ for all $s, t \in T$.

We call T *homogeneous* if $I(t)$ is a subgroup of I for some tone t; in this case, we observe that $I(s) = I(t) = \delta[T \times T]$ holds for all tones s and t in T. We refer to T as a *(freely) n-generated* tone system if T is homogeneous, $\delta[T \times T] = I$, and I is a (freely) n-generated group.

A *transposition* by the interval $i \in I$ is defined as the map $\tau_i : T \to T$ such that $t \mapsto t + i$ (if $t + i$ exists for every tone t). Obviously, in case of a homogeneous tone system, a transposition τ_i exists for every $i \in \delta[T \times T]$. In particular, an n-generated tone system allows a transposition by any interval of I, and we observe, that the set of all transpositions forms a transformation group canonically isomorphic to the interval group I.

A *morphism* from a tone system $\mathrm{T} = (T, \delta, I)$ to a tone system $\mathrm{T}' = (T', \delta', I')$ is defined as a pair $\phi := (\phi_T, \phi_I)$, consisting of a map $\phi_T : T \to T'$ and a group homomorphism $\phi_I : \mathrm{I} \to \mathrm{I}'$, such that for all $s, t \in T$ we have

$$\phi_I\big(\delta(s, t)\big) = \delta'\big(\phi_T(s), \phi_T(t)\big). \tag{3}$$

If, in addition, ϕ_T and ϕ_I are bijections then ϕ is called an isomorphism (from T to T′). Every transposition τ induces via (τ, id_I) an automorphism on T.

For every positive integer n, a freely n-generated tone system is always isomorphic to the tone system $(\mathbb{Z}^n, \delta, \mathbb{Z}^n)$ where $\delta(x, y) := y - x$ for all $x, y \in \mathbb{Z}^n$.

In the following we consider the 1-generated tone system $\mathrm{T} = (\mathbb{Z}, \delta, \mathbb{Z})$ and we fix a positive integer $\mathfrak{O} \in \mathbb{Z}_+$, which we consider as an interval called *octave*. Let $\mathbb{Z}_{\mathfrak{O}}$ denote the residue ring of integers modulo $\mathfrak{O}$ and let $\mathrm{T}_{\mathfrak{O}} := (\mathbb{Z}_{\mathfrak{O}}, \delta_{\mathfrak{O}}, \mathbb{Z}_{\mathfrak{O}})$ be the 1-generated algebraic tone system (where $\delta_{\mathfrak{O}}(x, y)$ denotes the difference $y - x$ in $\mathbb{Z}_{\mathfrak{O}}$). Following the language of musicology, $T_{\mathfrak{O}}$ is called a *chroma system*, and its elements are refereed to as *chromas*. More specifically, we will refer to $\mathrm{T}_{\mathfrak{O}}$ as $\mathfrak{O}$*-tone equal tempered* chroma system, in short $\mathfrak{O}$-tet. The most commonly used of these are the 12-tet (T_{12}) and the 7-tet (T_7).

Table 1. parameters describing concept lattices of harmonic forms.

Group	# of harmonic forms	# of irreducibles (rows/columns)	# of concepts
$\mathbb{Z}_1$	2	1	2
$\mathbb{Z}_2$	3	2	3
$\mathbb{Z}_3$	4	3	4
$\mathbb{Z}_4$	6	4	6
$\mathbb{Z}_5$	8	6	9
$\mathbb{Z}_6$	14	11	18
$\mathbb{Z}_7$	20	13	42
$\mathbb{Z}_8$	36	25	142
$\mathbb{Z}_9$	60	39	1 460
$\mathbb{Z}_{10}$	108	73	9 325
$\mathbb{Z}_{11}$	188	112	1 798 542
$\mathbb{Z}_{12}$	352	212	208 946 771

The canonical group homomorphism $\phi_{\mathfrak{O}} : \mathbb{Z} \to \mathbb{Z}_{\mathfrak{O}}$ (which maps every integer x to its residue modulo $\mathfrak{O}$, denoted by $x_{\mathfrak{O}}$) induces via $(\phi_{\mathfrak{O}}, \phi_{\mathfrak{O}})$ a morphism from T onto $T_{\mathfrak{O}}$. Every chord X in T is mapped to the chord $X_{\mathfrak{O}} := \{x_{\mathfrak{O}} \mid x \in X\}$ in $T_{\mathfrak{O}}$, which will be called the *harmony of* X.

Chords and harmonies can efficiently be classified by the occurrence of intervals and chromas. In particular, two chords or harmonies have the same pattern if they are related by a transposition. The corresponding pattern classes we refer to as *chordal forms* or *harmonic forms*, respectively.

The degree of consonance or dissonance of a harmony is mostly influenced by its pattern.

Harmonic forms and their hierarchical order have been studied by Rudolf Wille and other authors ([5]). Though the corresponding concept lattice for the 7-tet T_7 has a nice diagram (see figure 1), the number of concepts of $\mathrm{T}_{\mathfrak{O}}$ is rapidly growing for increasing octave $\mathfrak{O}$. Table 1 shows some statistics about these concept lattices. For every chroma system there are printed the number of harmonic forms, the count of rows and columns in the formal context (describing the order of the harmonic forms), and the number of formal concepts in the corresponding concept lattice.

In other important chroma systems the hierarchical order of harmonic forms is too complex to be examined in the fashion above. Therefore, it is interesting to use other properties to clarify the structure of harmonic forms.

One important property is the interval structure of harmonic forms, since the intervals contained in a harmonic form have a major impact on their degree of consonance or dissonance. One example is shown in figure 2. Here, for the 7-tet a formal context is composed of the set of harmonic forms as objects, the set of intervals as attributes, and the interval occurrence as incidence relation. Also, in comparison with figure 1, this concept lattice is significantly simpler (but ordered oppositely).

The other corresponding lattices are relatively small too, as shown in table 2, which enables us to have a view on the 12-tet lattice (fig. 3).

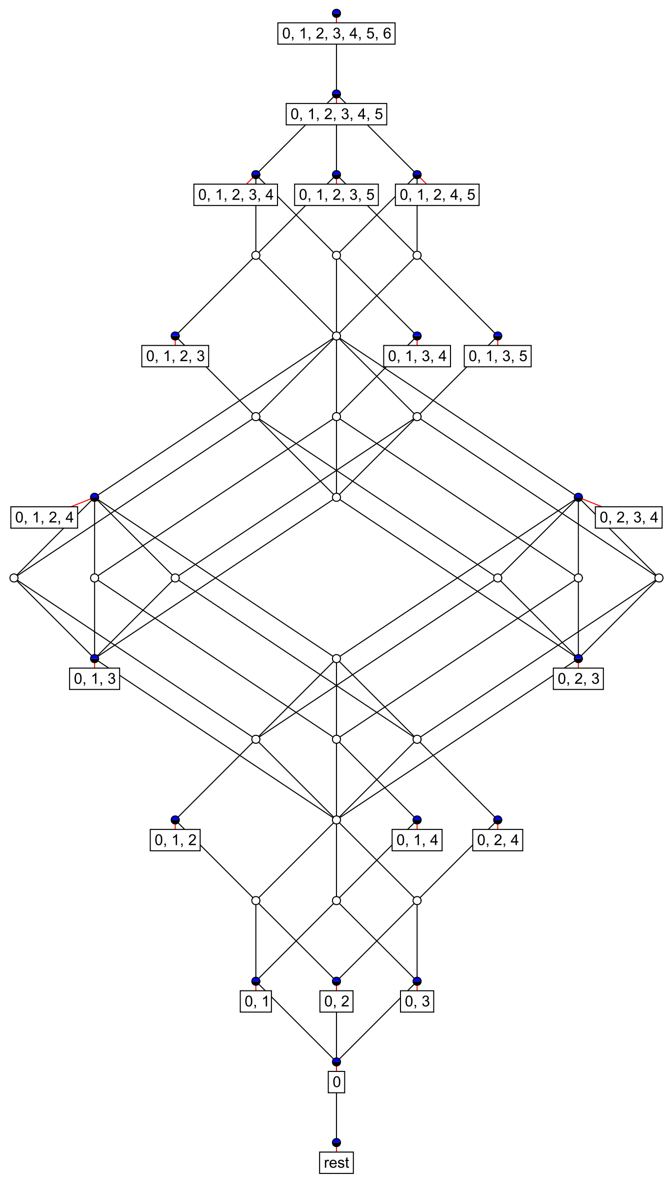


Fig. 1. Dedekind-McNeille completion of the order of harmonic forms in $\mathbb{Z}_7$.

	unison	second	third	fourth
rest				
0	×			
0, 1	×	×		
0, 2	×		×	
0, 1, 2	×	×	×	
0, 3	×			×
0, 1, 3	×	×	×	×
0, 2, 3	×	×	×	×
0, 1, 2, 3	×	×	×	×
0, 1, 4	×	×		×
0, 2, 4	×		×	×
0, 1, 2, 4	×	×	×	×
0, 1, 3, 4	×	×	×	×
0, 2, 3, 4	×	×	×	×
0, 1, 2, 3, 4	×	×	×	×
0, 1, 3, 5	×	×	×	×
0, 1, 2, 3, 5	×	×	×	×
0, 1, 2, 4, 5	×	×	×	×
0, 1, 2, 3, 4, 5	×	×	×	×
0, 1, 2, 3, 4, 5, 6	×	×	×	×

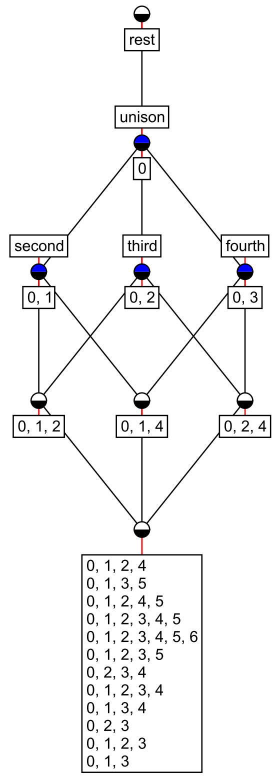


Fig. 2. Formal context and concept lattice which qualitatively describes the contained intervals of harmonic forms in the 7-tet

Table 2. Statistics of harmony interval concept lattices

Group	# of harmonic forms	# forms clar.	# forms red.	# intervals	# int. red.	#concepts
$\mathbb{Z}_1$	2	2	1	1	1	2
$\mathbb{Z}_2$	3	3	2	2	2	3
$\mathbb{Z}_3$	4	3	2	2	2	3
$\mathbb{Z}_4$	6	5	3	3	3	5
$\mathbb{Z}_5$	8	5	3	3	3	5
$\mathbb{Z}_6$	14	7	5	5	4	7
$\mathbb{Z}_7$	20	9	4	4	4	9
$\mathbb{Z}_8$	36	12	6	5	5	13
$\mathbb{Z}_9$	60	13	7	5	5	16
$\mathbb{Z}_{10}$	108	20	6	6	6	33
$\mathbb{Z}_{11}$	188	23	6	6	6	33
$\mathbb{Z}_{12}$	352	32	7	6	7	65

Figures 2 and 3 are largely Boolean lattices (except the node named “rest”). It is also visible in figure 3 that not every node has a label. In the language of music this means that the intervals cannot be combined freely. They have to fulfil certain restrictions.

On the other hand, in this approach we consider neither the order nor the multiplicity of intervals. So with this method some sets of harmonic forms are identified. For example the major triad and the minor triad share the same label since they consist of a minor third, a major third and a fourth as chroma intervals.

In comparison with the concept lattices of the 7-tet and the 12-tet, the ones of the 6-tet and the 8-tet are less symmetric (figure 4). This means, the lattices have a more complex underlying structure.

3 Analysis reflecting interval multiplicities

The Dedekind-McNeille completion of the ordering of harmonic forms does not reflect the notion of an interval. However, it describes a much finer granularity than given by the previously discussed type of lattice (derived from interval occurrence). The gap between these two lattice types can be filled by considering *many-valued* contexts, which reflect multiplicities of intervals within harmonic forms.

Figure 5 shows such a context in the 7-tet. Its ordinary scaled version is shown in figure 6 and the corresponding concept lattice appears in figure 7. This lattice has more concepts than the one presented in figure 2, but it contains less information than the one in figure 1. Though the multiplicity of intervals is reflected by the lattice, their internal arrangement remains neglected. For example, the harmonic forms of $\{0, 1, 3\}$ and $\{0, 2, 3\}$ are different but share the same node. This impacts the musical interpretation, as some harmonic forms (like the major triad and the minor triad in the 12-tet) are indistinguishable.

Furthermore, the lattice of figure 7 has a very regular structure. In the upper part, the free distributive lattice with three generating elements is visible. The concepts below it form a typical configuration for all lattices of this family, which can be seen more clearly in the lattices investigated in the sequel.

After observing the beautiful structure provided by the 7-tet, we investigated higher orders in a similar fashion. Our suggestive approach was to draw the diagrams by ordering the intervals from left to right according to their sizes. For the 8-tet this can be seen in the left diagram of figure 8.

Aiming for further insight, the following question arises: Which lattices allow nice diagrams and how can these diagrams be realised? To get a clue how to answer this question we focus again on the left diagram in figure 8. The right hand side of this diagram shows some interesting structural specialities: There are shorter chains from top to bottom of the lattice than on the left hand side. This phenomenon has the following cause: The concepts on the right hand side share the fifth (distance 4), which is an element of order 2 in the interval group. So these intervals occur only in pairs. This differs from the other intervals, which

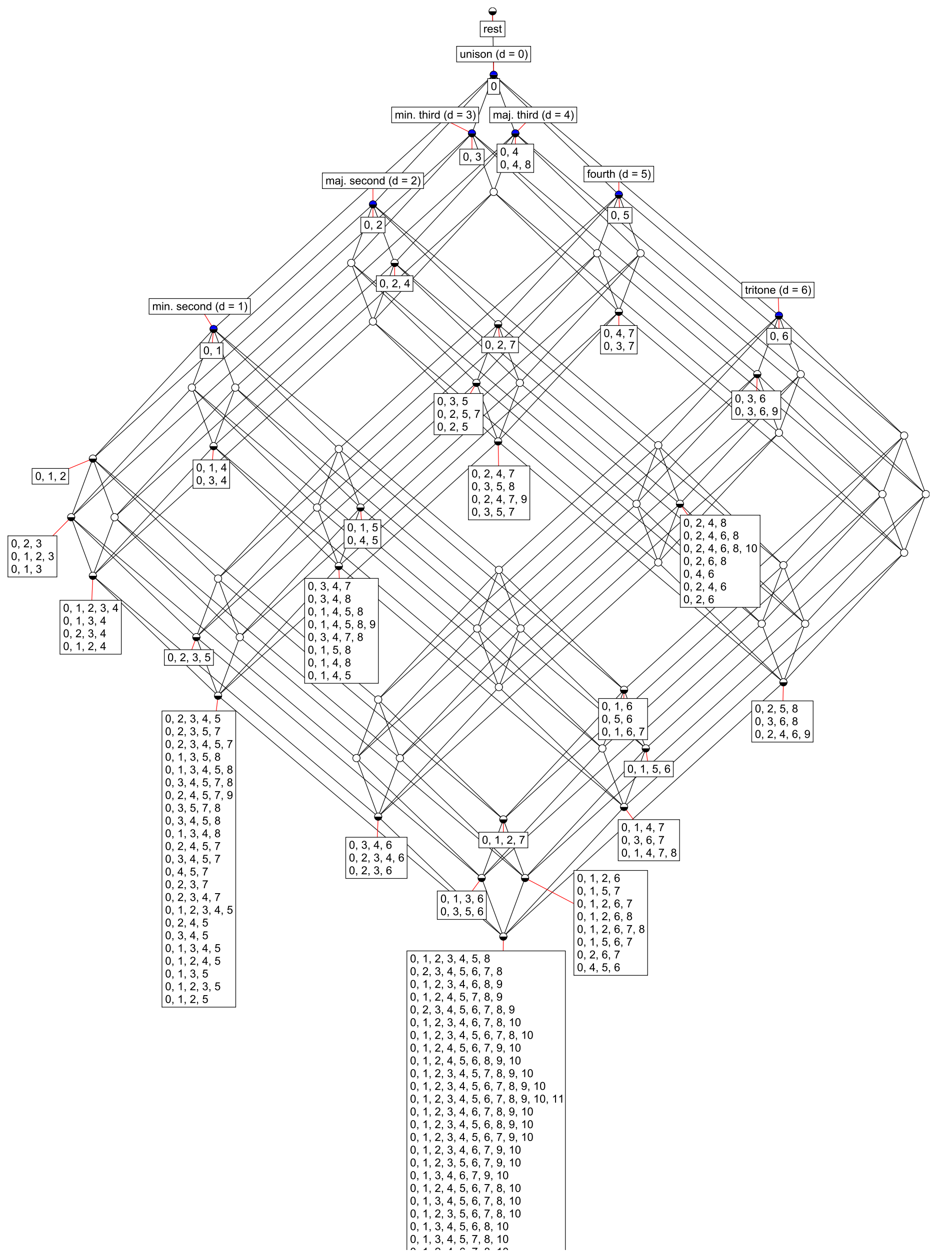


Fig. 3. Concept lattice qualitatively describing the contained intervals of harmonic forms in the 12-tet. The list of harmonic forms in the lower node is truncated.

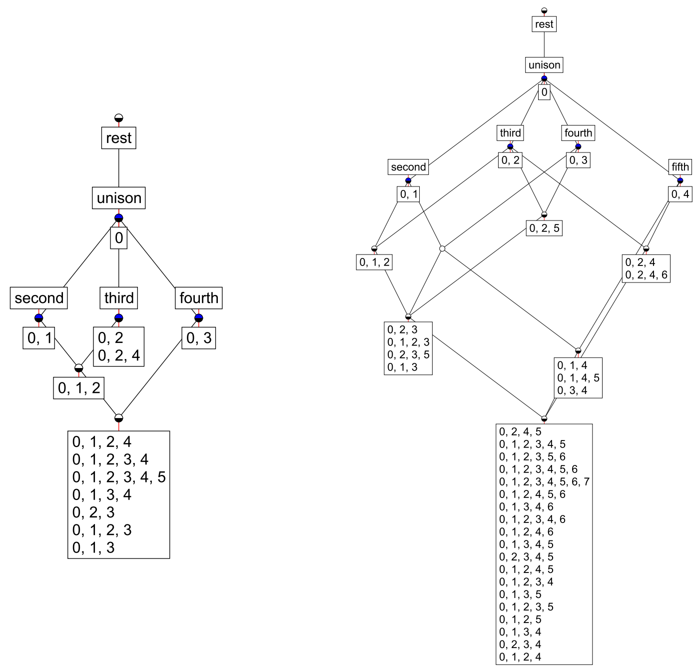


Fig. 4. Concept lattice qualitatively describing the contained intervals of harmonic forms in the 6-tet and the 8-tet

pattern	unison	second	third	fourth
rest				
0	1			
0, 1	2	1		
0, 2	2		1	
0, 1, 2	3	2	1	
0, 3	2			1
0, 1, 3	3	1	1	1
0, 2, 3	3	1	1	1
0, 1, 2, 3	4	3	2	1
0, 1, 4	3	1		2
0, 2, 4	3		2	1
0, 1, 2, 4	4	2	2	2
0, 1, 3, 4	4	2	2	3
0, 2, 3, 4	4	2	2	2
0, 1, 2, 3, 4	5	4	3	3
0, 1, 3, 5	4	1	3	2
0, 1, 2, 3, 5	5	3	4	3
0, 1, 2, 4, 5	5	3	3	4
0, 1, 2, 3, 4, 5	6	5	5	5
0, 1, 2, 3, 4, 5, 6	7	7	7	7

Fig. 5. Many-valued context describing the contained intervals of the 7-tet

	unison (d = 0)	second (d = 1)	2 × second (d = 1)	3 × second (d = 1)	4 × second (d = 1)	5 × second (d = 1)	6 × second (d = 1)	7 × second (d = 1)	third (d = 2)	2 × third (d = 2)	3 × third (d = 2)	4 × third (d = 2)	5 × third (d = 2)	6 × third (d = 2)	7 × third (d = 2)	fourth (d = 3)	2 × fourth (d = 3)	3 × fourth (d = 3)	4 × fourth (d = 3)	5 × fourth (d = 3)		
rest																						
0	×																					
0, 1	×	×																				
0, 2	×								×													
0, 1, 2	×	×	×						×													
0, 3	×															×						
0, 1, 3	×	×							×							×						
0, 2, 3	×	×							×							×						
0, 1, 2, 3	×	×	×	×					×	×						×						
0, 1, 4	×	×														×	×					
0, 2, 4	×								×	×						×						
0, 1, 2, 4	×	×	×						×	×						×	×					
0, 1, 3, 4	×	×	×						×							×	×	×				
0, 2, 3, 4	×	×	×						×	×						×	×					
0, 1, 2, 3, 4	×	×	×	×	×				×	×	×					×	×	×				
0, 1, 3, 5	×	×							×	×	×					×	×					
0, 1, 2, 3, 5	×	×	×	×					×	×	×	×				×	×	×				
0, 1, 2, 4, 5	×	×	×	×					×	×	×					×	×	×	×			
0, 1, 2, 3, 4, 5	×	×	×	×	×	×			×	×	×	×	×			×	×	×	×	×		
0, 1, 2, 3, 4, 5, 6	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

Fig. 6. Scaled formal context quantitatively describing the contained intervals of harmony patterns in the 7-tet

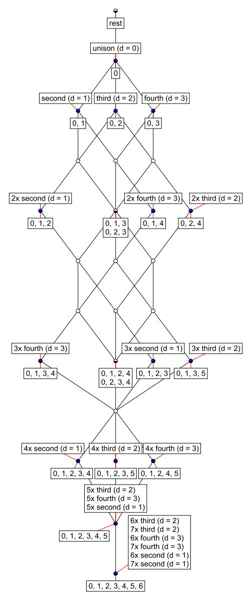


Fig. 7. Concept lattices of the context counting the intervals in the 7-tet

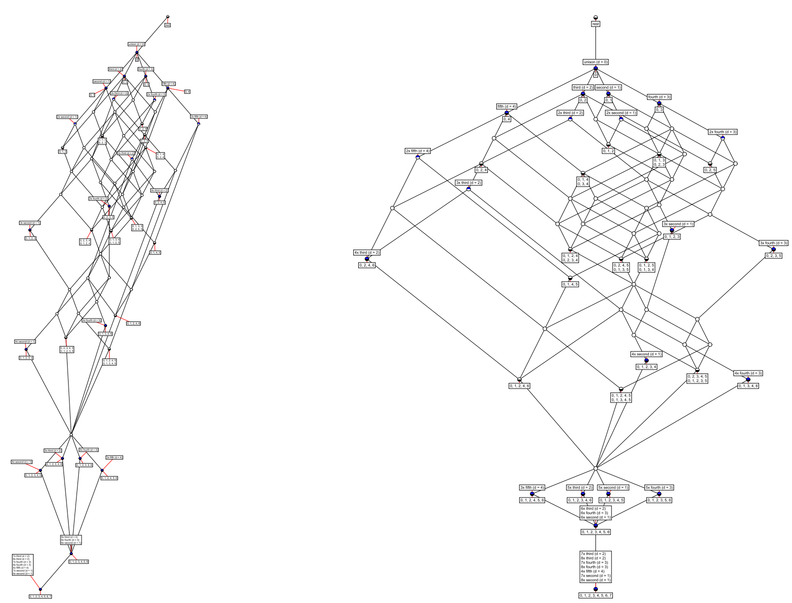


Fig. 8. Concept lattice describing the contained intervals in the 8-tet in two different chain decompositions

can be combined more freely. For example the harmonic form represented by the harmony $\{0, 1, 2, 3, 4\}$ has only three (unordered) pairs of chromas containing a third, namely $\{0, 2\}, \{1, 3\}, \{2, 4\}$.

Thus, divisibility of the group order has an impact on the (potential) layout and also the aesthetics of the generated diagram. On the other hand, harmonic forms of T_4 can be embedded into T_8 in various ways. The most important ones are defined by the mappings of chromas $f_1 : t \mapsto t$ and $f_2 : t \mapsto 2t$. Each of these maps preserves to a reasonable extent the interval structure.

This suggests to rearrange the order of the intervals in the diagram to better unfold the lattice diagram. Coprime intervals (where coprime is meant in the number theoretic sense) should be positioned far apart from each other, while those with small greatest common divisor not equal to 1, should be in close proximity. The unfolded lattice is shown on the right hand side of figure 8.

It turns out that divisibility is the main property which makes the upper part of such a pattern lattice deviate from a product of chains. The lower part of the pattern lattice is of a significantly different shape. To discuss this, we point out the following general fact: In the chroma system $\mathrm{T}_{\mathfrak{O}}$ intervals can be described as LEE distances (see [11]) between chromas.

The lowest point of such a diagram (as given in figure 8) represents the harmonic form of the complete chroma set $\mathbb{Z}_{\mathfrak{O}}$. Here, the LEE distances between unordered pairs of chromas form the set $\{0, \ldots, \lfloor \mathfrak{O}/2 \rfloor\}$. In the diagram the upper neighbour of the concept of the complete harmonic form represents an almost complete chroma set of size one less than $\mathbb{Z}_{\mathfrak{O}}$. That means, all intervals occur with the same frequency $\mathfrak{O} - 2$. Next, selecting $\mathfrak{O} - 2$ chromas, results in a harmonic form with two chromas omitted. Thus, each interval which ends in one of these two points will occur only $\mathfrak{O} - 4$ times in the pattern. But there is one exception: The interval between the two deleted points has been counted twice. Consequently, this interval occurs (still) $\mathfrak{O} - 3$ times in the harmonic form. Obviously, with an increasing number of deleted chromas, the number of intervals additionally vanishing, decreases further.

Because of the above, the lattice is divided into several levels of object concepts according to the number of chromas in the harmonic forms. This is not hard to understand since every harmonic form with k chromas contains $\frac{k(k-1)}{2}$ intervals (where multiplicities are respected). The latter means that the concept of such a harmonic form has $\frac{k(k-1)}{2} + 1$ attribute concepts of interval sets above it. Adding a chroma increases the number of interval concepts by k.

In case of example T_{12}, the above mentioned levels of harmonic forms become increasingly apparent towards the bottom part of the lattice T_{12} (see figure 9). The right hand side of the diagram shows the “lightning rod chains”, which result from the even group order leading to the pairwise occurrence of tritone intervals (distance 6) as described above.

The structure of the left hand side in the diagram of T_{12} is induced by the divisibility of 1, 2, 3 and 4. Though nesting can simplify the diagram in certain cases, the resulting lattices are still too complex for us to analyse. The reason is that every interval chain $< \frac{n}{2}$ generates all the levels described above. An

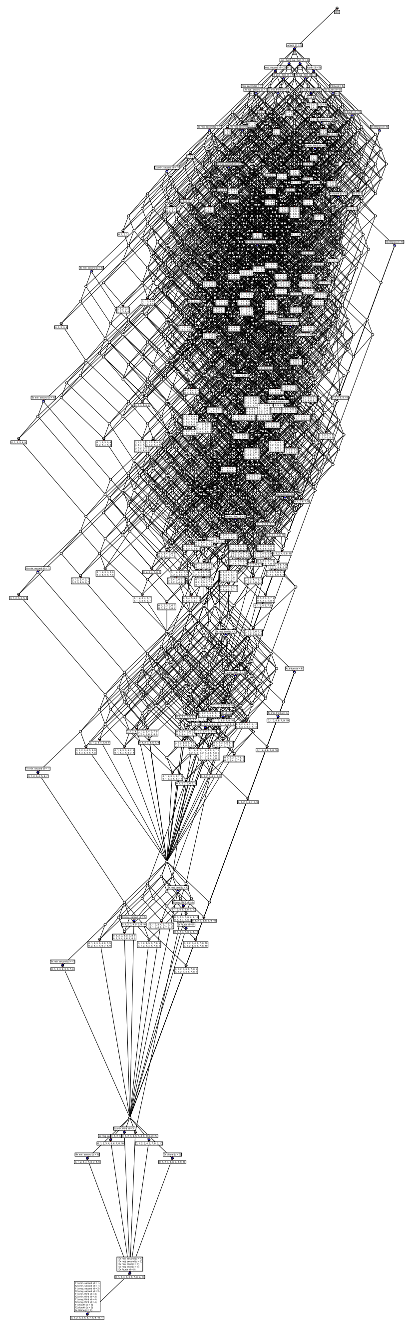


Fig. 9. Concept lattice of the harmony pattern vs. interval count context of the 12-tet

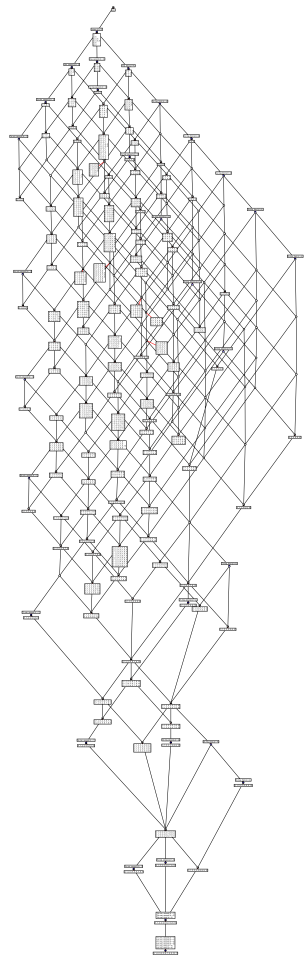


Fig. 10. Concept lattice created omitting all but the intervals 1, 2 and 4 in the 12-tet

example of such a large projection which does *not allow a sensible nesting* is demonstrated in figure 10.

4 Conclusions and further research topics

The current work shows how one can overcome the obstacles of getting a meaningful interpretation of concept lattices of increasing complexity. In particular, we analyse concept lattices describing harmonic forms and their intervals in different ways, focusing on the 7-tet and the 12-tet.

For future analysis we propose that the information hidden in complex diagrams (e. g. as given in figure 9) may be used to further investigate the inherent structure of the concept lattice.

Ongoing work is concerned with a description of the interval structure of a tone system and its influence on the structure of the concept lattices of harmonic forms in case of a totally ordered (or, more generally, lattice ordered) interval group.

Another extension of this work will be the investigation of tone systems with more complicated interval structures, for example the diatonic scale and Leonhard Euler's Tonnetz.

This area of research also aims to have an impact on the further development of the tuning software *Mutabor*.

References

1. Mutabor team: Mutabor. Software `http://www.math.tu-dresden.de/~mutabor/`.
2. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)
3. Wille, R., Wille-Henning, R.: Towards a semantology of music. In Priss, U., Polovina, S., Hill, R., eds.: ICCS. Volume 4604 of Lecture Notes in Computer Science., Springer (2007) 269–282
4. Wille, R., Wille-Henning, R.: Beurteilung von Musikstücken durch Adjektive: Eine begriffsanalytische Exploration. In: Von Intentionalität zur Bedeutung konventionalisierter Zeichen: Festschrift für Gisela Harras zum 65. Geburtstag. Volume Bd. 35. G. Narr, Tübingen (2006) 453
5. Wille, R.: Musiktheorie und Mathematik. In: Musik und Mathematik: Salzburger Musikgespräch 1984 unter Vorsitz von Herbert von Karajan. Springer-Verlag, Berlin (1985) 4–31
6. Wille, R.: Mathematische Sprache in der Musiktheorie. In: Jahrbuch Überblicke Mathematik 1980. Bibliographisches Institut, Mannheim (1980) 167–184
7. Mazzola, G., Göller, S., Müller, S.: The topos of music: geometric logic of concepts, theory, and performance. Birkhauser Verlag, Boston, MA (2002)
8. Wille, R.: Triadic concept graphs. In Mugnier, M.L., Chein, M., eds.: Conceptual Structures: Theory, Tools and Applications. Volume 1453 of Lecture Notes in Computer Science., Springer (1998) 194–208
9. Neumaier, W., Wille, R.: Extensionale Standardsprache in der Musiktheorie – eine Schnittstelle zwischen Musik und Informatik. In Hesse, H., ed.: Mikrotöne. Volume III. Edition Helbing edn. Helbing, Insbruck (1990) 139–167
10. Neumaier, W.: Was ist ein Tonsystem? : eine historisch-systematische Theorie der abendländischen Tonsysteme, gegründet auf die antiken Theoretiker Aristoxenos, Eukleides und Ptolemaios, dargestellt mit Mitteln der modernen Algebra. Volume 9. Lang, Frankfurt am Main (1986)
11. Lee, C.: Some properties of nonbinary error-correcting codes. Information Theory, IEEE Transactions on **4**(2) (Jun 1958) 77–82

Connecting Many-valued Contexts to General Geometric Structures

Tim B. Kaiser

Technische Universität Dresden
Institut für Algebra

Abstract. We study the connection between certain many-valued contexts and general geometric structures. The known one-to-one correspondence between attribute-complete many-valued contexts and complete affine ordered sets is used to extend the investigation to π-lattices and class geometries. The former are identified as a subclass of complete affine ordered sets, which exhibit a close relation to concept lattices which are closely tied to the corresponding context. The latter can be related to complete affine ordered sets using residuated mappings and the notion of a weak parallelism.

1 Introduction

In [5] the notion of an *affine ordered set* enables us to understand a many-valued context in order-theoretic and geometric terms. In [4] affine ordered sets were specialized to *complete affine ordered sets* to allow an algebraic interpretation. Here, we relate complete affine ordered sets to two other known types of general geometric structures, that is,

- π*-lattices* and
- *equivalence class geometries* (or short *class geometries*).

In [6], π-lattices were introduced as an abstraction of affine geometries over rings and modules, yielding the possibility to study geometry in a very general setup. They will turn out to be a well describable specialization of complete affine ordered sets, which opens up an intimate connection to the concept lattices arising from plain conceptual scaling of the corresponding context.
Class geometries are a generalization of *congruence class geometries*. They carry a certain type of parallelism – called *weak parallelism*– arising naturally in the context of coordinatizing geometric closure structures via the congruence classes of an algebra (in the sense of universal algebra), cf. [7]. The weak parallelism of class geometries can be related to the parallelism of complete affine ordered sets by applying a rather abstract result – a residuated pair of mappings between atomic complete lattices where one carries a weak parallelism induces a weak parallelism on the other.
As a first step, we will provide the necessary basic definitions. The second step will lead to an elaboration on the connection between complete affine ordered

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 23–34,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

sets and π-lattices. In a third step, we will show how a weak parallelism on an atomic lattice can induce a weak parallelism on another atomic lattice via an adjunction. The results will be applied in the concluding step to describe a connection between complete affine ordered sets and *class geometries*. Finally, we will give a summary of what we achieved.
Throughout the paper we assume that the reader is knowledgeable of the basic concepts of order theory and formal concept analysis, as provided, for instance, in [2] and [1].

2 Attribute-complete Many-valued Contexts and Complete Affine Ordered Sets

We recall the relevant definitions from [5] and [4].
For a mapping $f : A \longrightarrow B$ between sets A and B, the *kernel* of f is defined as

$$\ker(f) := \{(a_1, a_2) \mid f(a_1) = f(a_2)\}.$$

A many-valued context $\mathbb{K} := (G, M, W, I)$ is called *attribute-complete* if it

- is *complete*, that is, every $m \in M$ can be regarded as a map $m : G \rightarrow W$,
- has a *key* attribute, that is, there exists an attribute $m \in M$ with $\ker(m) = \Delta_G := \{(g, g) \mid g \in G\}$,
- is *simple*, that is, different attributes $m_1, m_2 \in M$ are not functionally equivalent, that is, $\ker(m_1) = ker(m_2)$ implies $m_1 = m_2$, and
- for all $N \subseteq M$ there exists an attribute $m \in M$ such that m and N are functionally equivalent, that is, $\ker(m) = \bigcap_{n \in N} \ker(n)$.

A *system of equivalence relations* (SER) is a pair (G, E) where G is a set and E is a set of equivalence relations on G which contains the identity relation. A SER is called *closed* if its set of equivalence relations forms a closure system.

Let $\uplus$ be the symbol for the disjoint union. Then the *lifting* of an ordered set $(P, \leq)$ is given by $(P \uplus \{\bot\}, \leq \uplus (\{\bot\} \times P \uplus \{\bot\}))$ and denoted as $(P, \leq)_\bot$. Since the following notion is central in this paper we provide it as

Definition 1 ((atomistic, complete) affine ordered set). *We call a triple*

$$\mathbb{A} := (Q, \leq, \parallel)$$

affine ordered set*, if $(Q, \leq)$ is a partially ordered set, $\parallel$ is an equivalence relation (called parallelism) on Q, and the axioms* (A1) - (A4) *hold. Let $A(Q) := \operatorname{Min}(Q, \leq)$ denote the set of all minimal elements in $(Q, \leq)$ and $A(x) := \{a \in A(Q) \mid a \leq x\}$.*

(A1) $\forall x \in Q : A(x) \neq \emptyset$
(A2) $\forall x \in Q \, \forall a \in A(Q) \, \exists! t \in Q : a \leq t \parallel x$
(A3) $\forall x, y, x', y' \in Q : x' \parallel x \leq y \parallel y' \;\; \& \;\; A(x') \cap A(y') \neq \emptyset \Rightarrow x' \leq y'$

(A4) $\forall x, y \in Q \, \exists x', y' \in Q : x \nleq y \ \& \ A(x) \subseteq A(y)$
$\Rightarrow x' \parallel x \ \& \ y' \parallel y \ \& \ A(x') \cap A(y') \neq \emptyset \ \& \ A(x') \nsubseteq A(y')$.

The elements of $A(Q)$ *are called* points *and, in general, elements of* Q *are called* subspaces*. We say that a subspace* x *is* contained *in a subspace* y *if* $x \leq y$*.*
If the lifting of $(Q, \leq)$ *forms a complete lattice* $L(\mathbb{A})$*, the affine ordered set* $\mathbb{A}$ *is called* complete affine ordered set*. We call a complete affine ordered set* $\mathbb{A}$ atomistic *if the corresponding complete lattice* $L(\mathbb{A})$ *is atomistic.*

For a point a and a subspace x we denote by $\pi(a|x)$ the subspace which contains a and is parallel to x. Axiom (A2) guarantees that there is exactly one such subspace. For every $x \in Q$ we observe that $\theta(x) := \{(a, b) \in A(Q)^2 \mid \pi(a|x) = \pi(b|x)\}$ is an equivalence relation on the set of points.

We introduce the following condition for affine ordered sets:

(A34) $\forall x, y \in Q : x \leq y \Longleftrightarrow A(x) \subseteq A(y) \ \& \ \theta(x) \subseteq \theta(y)$

If we assume that only (A1) and (A2) hold in Definition 1 the axioms (A3) and (A4) are equivalent to (A34).

In [4] it was shown that the notions of

- attribute-complete many-valued contexts,
- closed SERs, and
- complete affine ordered sets

with their respective morphisms form categories which are equivalent.

We recall how the objects of these equivalent categories can be translated into each other. To an attribute-complete many-valued context $\mathbb{K} := (G, M, W, I)$ we can assign a closed system of equivalence relations via

$$\mathbf{E}(\mathbb{K}) := (G, \{\ker(m) \mid m \in M\}).$$

To a closed system of equivalence relations $\mathbb{E} := (G, E)$ we can assign a complete affine ordered set − the ordered set of its labeled equivalence classes − via

$$\mathbf{A}(\mathbb{E}) := (\{([x]\theta, \theta) \mid \theta \in E\}, \leq, \parallel)$$

where $\leq$ is defined by

$$([x]\theta_1, \theta_1) \leq ([y]\theta_2, \theta_2) :\Longleftrightarrow [x]\theta_1 \subseteq [y]\theta_2 \ \& \ \theta_1 \subseteq \theta_2$$

and $\parallel$ is defined by

$$([x]\theta_1, \theta_1) \parallel ([y]\theta_2, \theta_2) :\Longleftrightarrow \theta_1 = \theta_2.$$

3 π-Lattices and Complete Affine Ordered Sets

The notion of a π-lattice stems from [6] where it is situated as an abstraction of a geometry over rings. For a complete lattice L, we define

$$L_+ := L \setminus \bigwedge L.$$

A π-lattice is defined as follows:

Definition 2 (π-lattice). *Let V be a complete atomistic lattice with set of atoms $A(V)$. Then an equivalence relation $\| \subseteq V_+ \times V_+$ is called* parallelism *if it satisfies the following axioms*

(E) $\forall p \in A(V)\, \forall x \in V_+ \exists! y \in V_+ : p \leq y \parallel x$
(M) $\forall p \in A(V)\, \forall x, y \in V_+ : x \leq y \implies \pi(p|x) \leq \pi(p|y)$.

We call an atomistic complete lattice with parallelism π-lattice.

It turns out that complete affine ordered sets are a natural generalization of π-lattices.

Proposition 1. *Let V be a π-lattice. Then $(V_+, \leq_V, \|)$ forms a complete affine ordered set.*

Proof. Since V is a π-lattice, $\|$ is an equivalence relation. We have to show that (A1) - (A4) hold for $(V \setminus \{0\}, \leq_V, \|)$. Since V is atomistic (A1) holds. (E) directly implies (A2). For showing (A34), let $x \leq y$ for $x, y \in V_+$. Obviously $A(x) \subseteq A(y)$ follows directly from $x \leq y$. Furthermore, $\theta(x) \subseteq \theta(y)$ by (M). For the other direction, already $A(x) \subseteq A(y)$ implies $x \leq y$ since we have

$$x = \bigvee A(x) \leq \bigvee A(y) = y$$

because V is atomistic. By construction $(V_+)_\perp = V$ is a complete lattice. □

The notion of parallelism for affine ordered sets fulfills the criteria of a parallelism from the definition of π-lattices without problems.

Proposition 2. *Let $\mathbb{A} := (Q, \leq, \|)$ be an affine ordered set. Then* (E) *and* (M) *hold in $L(\mathbb{A})$.*

Proof. (E) follows directly from (A2). To show (M) let $x \leq y$ for $x, y \in L(\mathbb{A})_+$ and $p \in A(L(\mathbb{A}))$. We have to show that $\pi(p \,|\, x) \leq \pi(p \,|\, y)$. By (A34) we have

$$\theta(\pi(p \,|\, x)) = \theta(x) \subseteq \theta(y) = \theta(\pi(p \,|\, y)).$$

Additionally, it follows directly that

$$A(\pi(p \,|\, x)) = [p]\theta(x) \subseteq [p]\theta(y) = A(\pi(p \,|\, y))$$

and therefore by applying the equivalence in (A34) from right to left we get $\pi(p \,|\, x) \leq \pi(p \,|\, y)$ which shows (M). □

Propositions 1 and 2 yield the following characterization of π-lattices in terms of complete affine ordered sets:

Theorem 1. *The atomistic complete affine ordered sets are in one-to-one correspondence with π-lattices. More precisely, moving between the two structures requires only attaching or respectively removing the bottom element while the parallelism can be reused.*

We will illuminate what it means for a complete affine ordered set to be atomistic. We call a system of equivalence relations $\mathbb{E} := (D, E)$ *regular* if its set of equivalence relations E is regular, that is, if there do not exist two different equivalence relations sharing an equivalence class.

Proposition 3. *Let $\mathbb{A}$ be a complete affine ordered set and let $\mathbb{E}$ be a closed system of equivalence relations with*

$$\mathbb{A} \cong \mathbf{A}(\mathbb{E}) \;\;\&\;\; \mathbf{E}(\mathbb{A}) \cong \mathbb{E}.$$

Then $\mathbb{A}$ is atomistic if and only if $\mathbb{E}$ is regular.

Proof. "$\Rightarrow$": Let $\mathbb{E} := (D, E)$ be a regular closed system of equivalence relations and let $\mathbf{A}(\mathbb{E})$ be the associated complete affine ordered set. Then we have to show for a subspace x from $\mathbf{A}(\mathbb{E})$ that $x = \bigvee A(x)$. But since the subspaces of $\mathbf{A}(\mathbb{E})$ are the labeled equivalence classes of $\mathbb{E}$ we know that $x = (X, \theta)$ for an equivalence class X of an equivalence relation $\theta \in E$. Then we have

$$\bigvee A(x) = \bigvee (\{p\}, \Delta) \,|\, p \in X\} = (X, \theta(X)).$$

But $\theta(X) = \theta$ since $\mathbb{E}$ is regular. Therefore, $\bigvee A(x) = (X, \theta(X)) = (X, \theta)$ and hence $\mathbf{A}(\mathbb{E})$ is atomistic.
"$\Leftarrow$": Let $\mathbb{A} := (Q, \leq, \|)$ be an atomistic complete affine ordered set and let $\mathbf{E}(\mathbb{A})$ be the associated closed system of equivalence relations. We have to show that $\mathbf{E}(\mathbb{A})$ is regular, that is, for a point $p \in A(Q)$ where $[p]\theta(x) = [p]\theta(y)$ it follows that $\theta(x) = \theta(y)$. Since $\mathbb{A}$ is atomistic we have

$$x = \bigvee [p]\theta(x) = \bigvee [p]\theta(y) = y.$$

Hence $\mathbf{E}(\mathbb{A})$ is regular. □

The subclass of atomistic complete affine ordered sets can be related to concept lattices arising in a certain fashion from the many-valued context corresponding to the affine ordered set. To be able to formulate this connection, we need the following

Definition 3 (derived context via nominal scaling). *Let $\mathbb{K} := (G, M, W, I)$ be a complete many-valued context. Then the formal context $\mathbb{K}^{nom} := (G, N, J)$ is called* derived context via nominal scaling *of $\mathbb{K}$ if*

$$N := \{(m, w) \in M \times W \,|\, \exists g \in G : m(g) = w\} \text{ and}$$

$$J := \{(g, (m, w)) \in G \times N \,|\, (g, m, w) \in I\}.$$

Now we explain the connection between atomicity of complete affine ordered sets and conceptual scaling.

Proposition 4. *Let $\mathbb{K} := (G, M, W, I)$ be a simple many-valued context with key attribute, let $\mathbb{A} := \mathbf{A}(\mathbb{K})$ be the associated affine ordered set and let $\mathbb{K}^{nom}$ be the derived context of $\mathbb{K}$ via plain nominal scaling. Let $\varphi : \mathbb{A} \to \mathfrak{B}(\mathbb{K}^{nom})$ be a mapping where $(C, \theta) \mapsto (C, C^J)$. Then φ is an order-preserving mapping which is*

- *surjective if and only if $\mathbb{A}$ is complete and*
- *injective if and only if $\mathbb{A}$ is atomistic.*

Proof. To see that $(C, C^J) \in \mathfrak{B}(\mathbb{K}^{nom})$ we have to show that $C = C^{JJ}$, that is that C is an extent of a formal concept of the concept lattice of $\mathbb{K}^{nom}$. It is obvious that $C \subseteq C^{JJ}$ since $\cdot^{JJ}$ is a closure operator. By construction of $\mathbb{A}$ we know that there exists a $h \in G$ and a $m \in M$ such that

$$C = [h]\ker(m) = \{g \in G \mid m(g) = m(h)\} = \{g \in G \mid (g, (m, m(h))) \in J\}.$$

Hence, $(m, m(h)) \in C^J$. But then for all $g \in C^{JJ}$ we have $gJ(m, m(h))$ which shows that if $g \in C^{JJ}$ then $g \in C$. Therefore $C = C^{JJ}$. It is obvious that φ is order-preserving.
"⇒": Let $\mathbb{A}$ be complete. We show that φ is surjective. Since the extents of $\mathfrak{B}(\mathbb{K}^{nom})$ are exactly the meets of equivalence classes induced by $\mathbb{K}$, and the set of equivalence classes induced by $\mathbb{K}$ is already meet-closed it is immediate that φ is surjective.
Let $\mathbb{A}$ be atomistic. We show that φ is injective. Let $\varphi(C_1, \theta_1) = \varphi(C_2, \theta_2)$. Then $(C_1, {C_1}^J) = (C_2, {C_2}^J)$ which implies $C_1 = C_2$. But since we know by Proposition 3 that $\mathbf{E}(\mathbb{A})$ is regular we have $\theta_1 = \theta_2$.
"⇐": Let φ be surjective. Then every extent of $\mathfrak{B}(\mathbb{K}^{nom})$ is an image of φ. But since the extents of $\mathfrak{B}(\mathbb{K}^{nom})$ are exactly the meets of equivalence classes induced by $\mathbb{K}$, we know that the set of equivalence classes is meet-closed and therefore $\mathbb{A}$ is complete.
Let φ be injective. Then whenever $(C_1, {C_1}^J) = \varphi(C_1, \theta_1) = \varphi(C_2, \theta_2) = (C_2, {C_2}^J)$ which is equivalent to $C_1 = C_2$ we have $\theta_1 = \theta_2$. That means, $\mathbf{E}(\mathbb{A})$ is regular. Again, by Proposition 3 we know that $\mathbb{A}$ is atomistic. □

The proof of the previous proposition yields the following

Corollary 1. *Let $\mathbb{K} := (G, M, W, I)$ be an attribute-complete many-valued context, let $\mathbb{A} := \mathbf{A}(\mathbb{K})$ be the associated complete affine ordered set and let $\mathbb{K}^{nom}$ be the derived context of $\mathbb{K}$ via plain nominal scaling. Then*

$$\mathfrak{B}(\mathbb{K}^{nom}) \cong L(\mathbb{A})$$

if and only if $\mathbb{A}$ is atomistic. □

The combination of Propositions 2 and 3 and Corollary 1 can be cast as:

Theorem 2. *Let $\mathbb{K} := (G, M, W, I)$ be an attribute-complete many-valued context. Then the following conditions are equivalent:*

- $\mathbf{E}(\mathbb{K})$ *is regular*
- $\mathbf{A}(\mathbb{K})$ *is atomistic*
- $\mathbf{A}(\mathbb{K})$ *induces a π-lattice*
- $L(\mathbf{A}(\mathbb{K})) \cong \mathfrak{B}(\mathbb{K}^{nom})$

□

Example 1. We get a nice example of an attribute-complete many-valued context if we consider a $\mathbb{K}$-vector space $\mathbb{V}$. Let

$$\mathbb{K}(\mathbb{V}) := (V, \mathrm{End}(\mathbb{V}), V, I)$$

where V is the set of vectors of $\mathbb{V}$, $\mathrm{End}(\mathbb{V})$ is the set of endomorphisms of $\mathbb{V}$, and I is defined as

$$(v, \varphi, w) \in I :\iff \varphi(v) = w.$$

Since for vector spaces, every congruence relation is already representable as the kernel of an endomorphism (and the kernels of endomorphisms are always congruence relations), we know that $\mathbf{E}(\mathbb{K}(\mathbb{V}))$ is closed. The lattice of the corresponding complete affine ordered set is isomorphic to the lattice of affine subspaces of the vector space $\mathbb{V}$. Since $\mathbf{E}(\mathbb{K}(\mathbb{V}))$ is regular we know by Theorem 2 that $\mathbf{A}(\mathbb{K}(\mathbb{V}))$ is atomistic, its lattice is isomorphic to the concept lattice derived by nominal scaling, and it induces a π-lattice.

4 Weak Parallelisms and Affine Ordered Sets

In this section, we will derive insights about trace parallelisms induced by residuated mappings between atomic lattices. An application of these abstract results in the next section will lead to a better understanding of the connection between affine ordered sets and class geometries.
In the following let L and M denote complete lattices. For a lattice L, let $A(L)$ denote the set of the atoms of L and for $s \in L$ let $A(s) := \{p \in A(L) \mid p \leq s\}$ denote the atoms less than or equal to s. A lattice L is called *atomic* if for every $s \in L_+$ we have $A(s) \neq \emptyset$.

Definition 4 (residuated maps). *A map $\varphi : L \longrightarrow M$ is called* residuated *if it is $\bigvee$– preserving. For a residuated map, there exists a map $\varphi^+ : M \longrightarrow L$, called* residual*, which is $\bigwedge$– preserving with*

$$\varphi m \leq l \Leftrightarrow m \leq \varphi^+ l$$

The maps uniquely determine each other. If one of the maps is surjective, the other is injective, and vice versa. The maps are called adjoint *to each other. We call (φ, φ^+) a* residuated pair *or an* adjunction *(this is sometimes also called a* covariant Galois connection*).*

Note, that for a residuated pair (φ, φ^+) where φ is injective, we have $\varphi^+\varphi = \Delta$, since $\varphi\varphi^+\varphi = \varphi$. In general $\varphi^+\varphi$ is a closure operator and $\varphi\varphi^+$ is a kernel operator.

Definition 5 (weak parallelism). *Let L be an atomic complete lattice. We call a relation $\|$ on L_+* weak parallelism *if the following holds for arbitrary $r, s, t, u \in L_+$ and arbitrary $p \in A(L)$.*

(P1) $r \| r$
(P2) $r \| s \geq t \| u \Rightarrow r \|\geq u$
(P3) $r \| s \geq p \Rightarrow r \vee p \geq s$
(P4) $\exists! s : r \| s \geq p$

We say for $r, s \in L_+$ with $r \|\geq s$ that s is *part-parallel* to r. If $\|$ is an equivalence relation the weak parallelism is called *pre-parallelism*.

We will investigate the connection between affine ordered sets and the introduced weak parallelism.

Proposition 5. *Let $\mathbb{A}$ be a complete affine ordered set. Then $L(\mathbb{A})$ is an atomic complete lattice whith pre-parallelism.*

Proof. Obviously, $L(\mathbb{A})$ is atomic (by (A1)) and complete. It remains to verify the axioms (P1)–(P4) for $L(\mathbb{A})$. Axiom (P1) follows from the fact that the parallelism of $\mathbb{A}$ is an equivalence relation. Axiom (A2) grants us that (P4) holds.
To show (P2), let $r \| s \geq t \| u$. Let $p \leq u$ be a point. By (A3) we know that from $u \| t \leq s \| \pi(p|r)$ we get $u \leq \pi(p|r))$. Therefore we have $r \| \pi(p|r) \geq u$.
To show (P3), let $r \| s \geq p$. Let $q \leq s$ be an arbitrary point of s. By Proposition 2 we know that (M) holds. Therefore $r \leq r \vee p$ yields $s = \pi(q|r) \leq \pi(q|r \vee p)$. But $p \leq s \leq \pi(q|r \vee p)$ implies $\pi(q|r \vee p) = r \vee p$. Hence, $s \leq r \vee p$. □

Example 2. The converse of the previous proposition does not hold: In general, atomic complete lattices with pre-parallelism do not induce a complete affine ordered set. If we remove the bottom element of the lattice in Figure 1 we can not consider the resulting structure as an affine ordered set since $\theta(a) = \theta(x)$ would imply $a = x$.

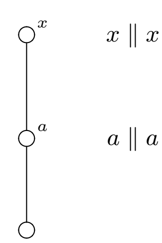


Fig. 1. Complete atomic lattice with trivial pre-parallelism

As in the case of π-lattices – where it was enough to require an affine ordered set to be atomistic to let the concepts coincide – for atomistic lattices a pre-parallelism is already a parallelism.

We will show that a residuated pair between two complete atomic lattices where the latter carries a weak parallelism yields a weak parallelism on the former. This parallelism will also be called *trace parallelism*.

Theorem 3. *Let M and L be complete atomic lattices and let $\|^L$ be a weak parallelism on L. Furthermore, let $\varphi : M \hookrightarrow L$ be a $\bigvee$– preserving, injective* *mapping with $\varphi A(M) \subseteq A(L)$ and let (φ, φ^+) form a residuated pair. Then we define a relation on M_+ as follows*

$$r \parallel^M s :\Leftrightarrow \exists y \in L : \varphi r \parallel^L y \ \ \& \ \ \varphi^+ y = s.$$

The relation $\parallel^M$ is a weak parallelism.

Proof. In the following, let $r, s, t, u \in M$ and $p \in A(M)$.
For (P1), we have to show that $\parallel^M$ is reflexive. Since $\varphi\varphi^+\varphi r = \varphi r$ and φ is injective, we have $\varphi^+\varphi r = r$. Since $\varphi r \parallel^L \varphi r$ we have $r \parallel^M r$ via setting $y := \varphi^+\varphi r$ in the definition of $\parallel^M$.
For (P2), let us assume that $r \parallel^M s \geq t \parallel^M u$. We have to show the existence of an element $v \in M$ with $v \geq u$ and $r \parallel^M v$. From $r \parallel^M s$ we know there exists $y \in L$ such that $\varphi r \parallel^L y$ and $\varphi^+ y = s$. From $t \parallel^M u$ we know there exists $z \in L$ such that $\varphi t \parallel^L z$ and $\varphi^+ z = u$. But since $\varphi^+ y \geq t$ implies $y \geq \varphi t$ we have $\varphi r \parallel^L y \geq \varphi t \parallel^L z$. Applying (P2) yields the existence of an element $q \in L$ with $q \geq z$ and $\varphi r \parallel^L q$. We have $v := \varphi^+ q \geq \varphi^+ z = u$ and $r \parallel^M v$.
For (P3), let us assume $r \parallel^M s \geq p$. From $r \parallel^M s$ we know there exists $y \in L$ such that $\varphi r \parallel^L y$ and $\varphi^+ y = s$. Since $s = \varphi^+ y \geq p$ implies $y \geq \varphi p$ and φ maps atoms to atoms we can apply (P3) in M. This yields $y \leq \varphi r \vee \varphi p = \varphi(r \vee p)$ which implies $s = \varphi^+ y \leq \varphi^+\varphi(r \vee p) = r \vee p$ as required.
For (P4), we have an atom $p \in A(M)$ and an arbitrary element $r \in M_+$. We have to show that there exists exactly one $s \in M_+$ with $r \parallel^M s \geq p$. We can apply (P4) for φp and φr which yields the existence of exactly one $y \in L_+$ with $\varphi r \parallel^L y \geq \varphi p$. We set $s := \varphi^+ y$. Since $y \geq \varphi p$ implies $s = \varphi^+ y \geq p$ and by construction of s we have $r \parallel^M s$ it remains to show that s is unique. Assume we have an element $s' \in M_+$ with $r \parallel^M s' \geq p$. This means that there exists an element $y' \in L_+$ with $\varphi r \parallel^L y'$ and $\varphi^+ y' = s'$. But since $\varphi r \parallel^L y' \geq \varphi p$ (P4) yields $y' = y$ we have $s = s'$. □

In the following theorem we characterize relations which arise from weak parallelisms in the manner described in Theorem 3 by "part-parallelity". This result can be used to see how the two weak parallelisms in Theorem 3 are connected.

Theorem 4. *Let M and L be complete atomic lattices and let $\parallel^L$ be a weak parallelism on L, furthermore, let (φ, φ^+) be a residuated pair for M and L and let $\parallel^M$ be defined as in the previous theorem. Then we have*

$$r \parallel^M \geq s \Leftrightarrow \varphi r \parallel^L \geq \varphi s.$$

Proof. Since $r \parallel^M \geq s$ there exists an $u \in M_+$ with $r \parallel^M u \geq s$. By definition of $\parallel^M$ we have the existence of an element $y \in L$ with $\varphi r \parallel^L y$ and $\varphi^+ y = s$. Since $\varphi\varphi^+$ is a kernel operator we have $\varphi s = \varphi\varphi^+ y \leq y$ which yields that φr is part-parallel to φs. The proof is finished since the argument is symmetric. □

5 Class Geometries and Affine Ordered Sets

Throughout this section, let $\mathbb{E} := (D, E)$ be a closed system of equivalence relations. We know that we can assign a complete affine ordered set, denoted by $\mathbf{A}(\mathbb{E})$, to $\mathbb{E}$. Alternatively, we can also assign the ordered set of equivalence classes $(\{[x]\theta \mid \theta \in E\}, \subseteq)$ to $\mathbb{E}$. It is convenient to attach a bottom element to get a lattice

$$\mathbf{G}(\mathbb{E}) := (S \cup \{\emptyset\}, \subseteq)$$

which we call *class geometry* of $\mathbb{E}$. If the equivalence relations can be regarded as the congruence relations of an algebra (in the sense of universal algebra) we call their class geometry *congruence class geometry*. Congruence class geometries were introduced and characterized geometrically via their closure operators in [7].

Now, we want to relate the class geometry $G := \mathbf{G}(\mathbb{E})$ and the lattice of the affine ordered set $L := L(\mathbf{A}(\mathbb{E}))$ of a closed system of equivalence relations to each other. Let $\varphi^+ : L \to G$ be defined by $\varphi^+(C, \theta) := C$. Since

$$\bigwedge_{i \in I} (C_i, \theta_i) = (\bigcap_{i \in I} C_i, \bigcap_{i \in I} \theta_i),$$

we have

$$\varphi^+ \bigwedge_{i \in I} s_i = \bigcap_{i \in I} C_i = \bigwedge_{i \in I} \varphi^+ s_i$$

for $s_i = (C_i, \theta_i)$. Note that φ^+ is surjective.

From Proposition 9 in [2], p. 14, we know that for any residual map its residuated is given by

$$\varphi s := \bigwedge \{l \mid s \leq \varphi^+ l\}.$$

If we define for a closed system of equivalence relations (D, E) the smallest relation containg $M \subseteq D$ as

$$\theta(M) := \bigcap \{\theta \in E \mid M \times M \subseteq \theta\}$$

the above definition of the residual yields in our context that $\varphi : S \hookrightarrow L$ is defined by

$$\varphi C := (C, \theta(C)).$$

Since φ^+ is surjective, it follows that φ is injective. This implies that φS is a kernel system in L. We summarize the results of the argumentation in

Theorem 5. *Let $\mathbb{E} := (D, E)$ be a closed system of equivalence relations. Let $G := \mathbf{G}(\mathbb{E})$ be its class geometry and let $L := L(\mathbf{A}(\mathbb{E}))$ be the lattice of its affine ordered set. Then (φ, φ^+) (as defined above) forms an adjunction between G and L, where φ is injective and φ^+ is surjective. This implies that G is embedded in L as a kernel system via φ.*

As an illustration of the previous theorem we provide

Example 3. Figure 2 shows the well-known non-modular lattice N_5. Figure 3 shows the lattice of congruence relations of N_5. Figure 4 shows the congruence class geometry of N_5 embedded as a kernel system into the lattice of the affine ordered set of (the congruence relations of) N_5. The kernel system is marked by black dots in Figure 4.

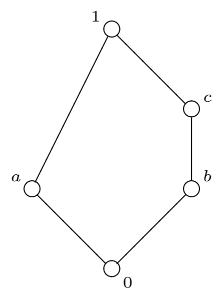


Fig. 2. N_5

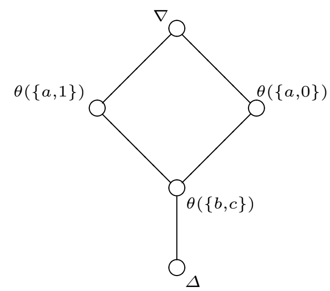


Fig. 3. The congruence lattice of N_5

It is easily observable that both, the class geometry G and the lattice L of the complete affine ordered set, form atomic lattices. By Proposition 5 we know that the parallelism of the affine ordered set constitutes a weak parallelism (even a pre-parallelism) in the sense of Definition 5. We use the residuated pair (φ, φ^+) to apply Theorem 3. Since φ maps atoms to atoms, Theorem 3 yields that

$$r \parallel^S s :\Leftrightarrow \exists l \in L : \varphi r \parallel l \;\;\&\;\; \varphi^+ l = s$$

defines a weak parallelism on S_+.

What does it mean for two equivalence classes C, D to be weakly parallel in S in terms of their equivalence relations? Expanding the definition we get

$$\begin{aligned} & C \parallel^S D \\ \Leftrightarrow\; & \exists (P, \psi) \in L : \theta(C) = \psi \;\;\&\;\; P = D \\ \Leftrightarrow\; & D \text{ is a class of } \theta(C). \end{aligned}$$

Surprisingly, this is exactly the same weak parallelism as is used in [7] on the closed sets of a closure operator to be able to characterize this closure operator as assigning to a set M the smallest congruence class of a suitable algebra containing M.

6 Conclusion

Studying the connection between complete affine ordered sets and π-lattices yielded the fruitful characterization of π-lattices as atomistic affine ordered sets and opened up the possibility to interpret these structures as concept lattices. Through an adjunction between a complete affine ordered set and its corresponding class geometry we could view the class geometry as a kernel system in the affine ordered set and were able to recognize the induced parallelism as known from congruence class spaces, where it is used to coordinatize geometric spaces. We conclude that the findings in this paper support the thesis that affine ordered sets are a conceptually useful paradigm to connect different notions arising when studying geometric structures abstractly.

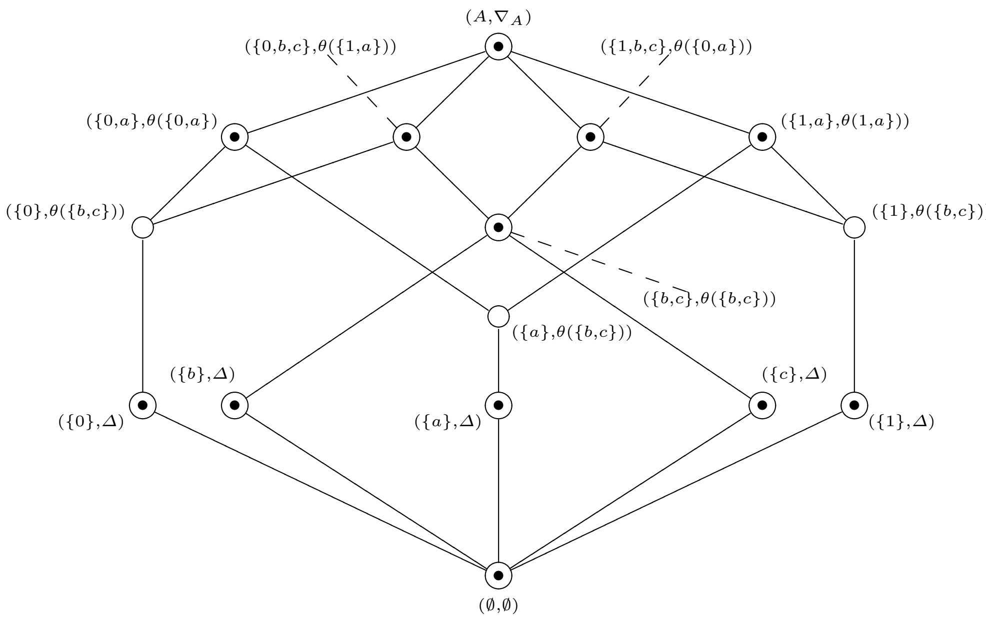


Fig. 4. Congruence class geometry of N_5 embedded as kernel system in the lattice of the labeled congruence classes of N_5

References

1. Davey, B. A., Priestly, H. A.: *Introduction to Lattices and Order*. Cambridge University Press, Cambridge 1990.
2. Ganter, B., Wille, R.: *Formal Concept Analysis, Mathematical Foundations*. Springer, Berlin – Heidelberg – New York 1999.
3. Kaiser, T. B.: *Representation of Data Contexts and their Concept Lattices in General Geometric Spaces*. In F. Dau (Ed.): *Conceptual Structures: Common Semantics for Sharing Knowledge*. LNAI **3596**. Springer, Heidelberg 2005, 195–208.
4. Kaiser, T. B.: *Closure Systems of Equivalence Relations and Their Labeled Class Geometries*. In Sadok B. Yahia, Engelbert M. Nguifo, Radim Belohlavek, ((ed.): *Concept Lattices and Applications*. LNAI **4932**, Springer, Berlin – Heidelberg – New York 2008, 96-106
5. Kaiser, T. B., Schmidt, S. E.: *Geometry of Data Tables*. In P. Eklund (Ed.): *Concept Lattices*. LNAI **2961**. Springer, Heidelberg 2004, 222–235.
6. Schmidt, S. E.: *Grundlegungen zu einer allgemeinen affinen Geometrie*. Birkhäuser, Basel – Boston – Berlin 1995.
7. Wille, R.: *Kongruenzklassengeometrien*. Springer, Berlin – Heidelberg – New York 1970.

Concept Lattice Mining for Unsupervised Named Entity Annotation

Thomas Girault

Orange labs, 2 avenue Pierre Marzin, 22307 Lannion Cedex
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex, France
thomas.girault@orange-ftgroup.com

Abstract. We present an unsupervised method for named entity annotation, based on concept lattice mining. We perform a formal concept analysis from relations between named entities and their syntactic dependencies observed in a training corpus. The resulting lattice contains concepts which are considered as labels for named entities and context annotation. Our approach is validated through a cascade evaluation which shows that supervised named entity classification is improved by using the annotation produced by our unsupervised disambiguation system.

1 Introduction

Lexical ambiguity is a fundamental problem which is central in many tasks involving natural language processing (*e.g.* information retrieval, information extraction, . . .). Our study focuses on a kind of lexical units (LU), named entities (NE), a generic denomination for proper names including persons, locations, organisations. As most LU considered outside a context, NE are ambiguous since their form can potentially refer to different meanings or objects. Our approach to disambiguation is based on *formal concept analysis* (FCA), a generic method for data analysis and knowledge representation which infers *formal concepts* from relational data. In this work, FCA is used to build a knowledge-base that is exploited for NE annotation.

The problem of ambiguity can be considered according to several Word Sense Disambiguation (WSD) approaches [1]. Knowledge-based approaches attempt to select the meaning of words using lexicons, dictionaries or thesauri (*e.g.* WordNet). Corpus-based approaches examine the occurrence of LU and their contexts using machine learning techniques. Supervised learning disambiguates LU according to pre-defined labels whereas unsupervised techniques discriminate the meanings of unlabelled LU thanks to similarity of their contexts.

Since corpus annotation is a tedious and costly task, this work is focused on unsupervised approaches. Among them, *formal concept analysis* (FCA) [2] has been selected : this symbolic unsupervised machine learning technique operates on relational data to infer *formal concepts* which can be structured into a *concept*

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 35–46,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

lattice. FCA is applied on relations between NE and their syntactic dependencies extracted from English news wire articles. The sets of NE sharing the same syntactic dependencies constitute the formal concepts which are considered as units of meaning for the annotation of NE. The concept lattice obtained can be seen as a hierarchical knowledge-base modelling meaning overlapping on several levels of granularity. To our knowledge, these properties attached to concept lattices have not yet been exploited in an unsupervised WSD task. In this context, we propose a conceptual annotation method for NE disambiguation.

In this paper, we address the problem of exploiting a concept lattice for unsupervised NE annotation. We first introduce (Section 2) the problem of NE ambiguity by exposing few examples from our corpus in which relations between NE and their syntactic dependencies are extracted. These relations constitute a formal context from which FCA is performed (section 3). The resulting lattice contains formal concepts which are considered as labels for NE and dependency annotation (Section 4). Our approach is validated through a cascade evaluation (section 5) which shows that supervised NE classification is improved by using the annotation produced by our unsupervised disambiguation system.

2 Corpus-Based Methods for Word Sense Disambiguation

This section introduces corpus-based word sense disambiguation (WSD) with a small sample of a corpus where NE occurrences are semantically labelled. Supervised learning disambiguates LU according to labelled pre-defined meanings whereas unsupervised techniques discriminate the meanings of unlabelled LU thanks to similarity of their lexical contexts. Our unsupervised approach is built upon the study of syntactic relations between NE and other LU occurring in an utterance.

2.1 Tagset Granularity for Supervised NE Classification

Named Entity Recognition (NER) is a subtask of Information Extraction. Different NER systems were evaluated, among others, as a part of the Message Understanding Conferences [3] in 1995 and in the CoNLL 2003 shared task [4]. The most efficient NER systems are built upon supervised corpus-based learning for the detection and classification of NE. They rely on semantically annotated corpora which we can illustrate with the following examples (figure 2.1) :

1. India$_{loc}$ has acquired 120,000 tonnes of diesel in three cargoes, ...
2. Cricket - : India$_{loc}$ wins the toss and bat against Sri Lanka$_{loc}$.
3. Tennis - : Muster$_{per}$ upset, Philippoussis$_{per}$ wins, Stoltenberg$_{per}$ loses.
4. Schumacher$_{per}$ wins Belgian Grand Prix.
5. Clinton$_{per}$ wins democratic re-nomination.
6. Siam Commercial$_{org}$ wins agency bond auctions.

Fig. 1. Samples extracted from the English CoNLL-2003 annotated corpus.

The English CoNLL 2003 data is a collection of news wire articles from the Reuters Corpus in which the NE are manually labelled with respect to the coarse-grained semantic tagset {*person, location, organisation, miscellaneous*}.

The examples (1) and (2) illustrate a case of ambiguity : the NE "India" is labelled as location but a more fined granularity would distinguish the sport nation and the wholesale importer. In addition we could note that LU interacting with NE are ambiguous as well : the LU "wins" occurs with different meanings for the domains of politics, sport or business. Thus, we think that the original tagset should be enriched with a refined semantic description. However, a manual refinement would be a tedious and a costly task. In addition, we cannot define a general semantic tagset since it is domain dependent : for instance a biomedical semantic tagset should discriminate viruses and proteins and it would not be suitable to describe geographic entities such as rivers or mountains.

2.2 Unsupervised Corpus-Based Disambiguation

Instead of assigning predefined labels to LU, an alternative strategy is to discriminate their meanings by analysing their co-occurrences in the utterances of a corpus. This unsupervised approach is founded from the assumption that LU (NE in our case) which occur in similar contexts tends to have close meanings. Distributional methods [5] relying on Harris' hypothesis consider that the share of contexts having common syntactic patterns (*e.g.* subject-verb, modifier-noun) constitutes an indicator of semantic relatedness.

2.3 Named Entity Dependency Extraction

Before applying distributional hypothesis for NE disambiguation, the LU attached syntactically to NE need to be identified. We suppose that the NE frontiers have been already detected. Our method deals with two kinds of dependencies. External dependencies are mainly nouns, verbs and prepositions occurring before or after a NE. They are extracted with patterns defined manually relying on morphosyntactic tagging and phrase chunking available with the CoNLL-2003 corpus[1]. The patterns extracts expressions such as :

- noun + preposition + NE (*e.g.* [election of, Clinton], [results of, European Super League]);
- noun + NE (*e.g.* [champion, Pete Sampras]);
- NE + noun (*e.g.* [Russian, government]);
- NE + verb (*e.g.* [Clinton, signed], [India, wins]).

Internal dependencies correspond to non prepositional tokens occurring in the NE, such as first names or surnames. For example, the list of internal dependencies of *International Boxing Federation*, is {*international, boxing, federation*}.

This work on extraction provides a set of pairs (NE, syntactic dependency) where each element is potentially ambiguous.

[1] Morphosyntactic tagging and chunking have been generated automatically and are therefore noisy.

3 Formal Concept Analysis for Knowledge Base Acquisition

In this section, the approach for knowledge-base acquisition using FCA is exposed. We illustrate FCA with examples taken from our linguistic data. We then discuss the advantages of FCA for dealing with meaning overlapping and granularity of meanings.

3.1 Formal Context of Syntactic Relations

Classical distributional methods could deal with ambiguity of the whole set of LU. However, these methods consider them from a unique point of view whereas for our problem, the data seems more naturally represented according to two interconnected views as the figure (2) shows :

– a view on named entities which is associated to a set of objects $O = \{o_1, o_2, \cdots, o_m\}$.
– a view on their dependencies (*syntactic co-texts + internal components*) represented by a set of attributes $A = \{a_1, a_2, \cdots, a_n\}$

These views are connected by a relation $R \subseteq O \times A$, where $R(o, a)$ means that the object o has the attribute a (*i.e.* the NE o has the dependency a).

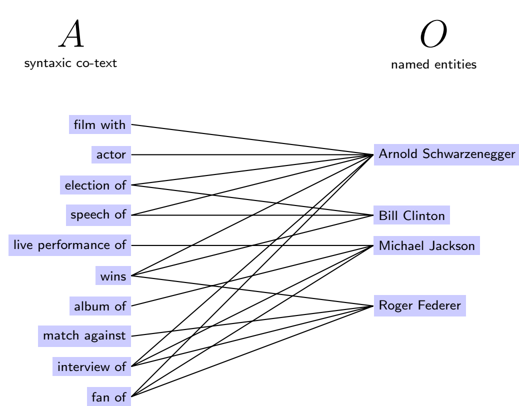


Fig. 2. Relations between NE and their dependencies.

In the FCA terminology, the triple $\mathbb{K} = (O, A, R)$ is called a formal context. It corresponds to a bigraph (from the figure (2)) of objects (NE) in relation with attributes (syntactic co-texts + internal components).

3.2 Formal Concept Analysis

For the understanding of the paper, we introduce standard definitions and notations of FCA [2]. For $E \subseteq O$ and $I \subseteq A$, we define two sets $E' \subseteq A$ and $I' \subseteq O$ extending them : $E' = \{a \in A | \forall o \in E : (o, a) \in R\}$ as the set of all attributes from I that are in relation with all objects from E and $I' = \{o \in O | \forall a \in I : (o, a) \in R\}$, the set of all objects from O that are in relation with all attributes from I. For instance, if $I = \{$speech of, election of$\}$ then $I' = \{$Bill Clinton, Arnold Schwarzenegger$\}$. For $E = \{$Michael Jackson$\}$, we have, $E' = \{$album, live performance of, interview of, fan of$\}$.

We can define a *formal concept* of the formal context $\mathbb{K}$ to be a pair (E, I) satisfying $E \subseteq O, I \subseteq A$, $E' = I$ and $I' = E$. E is called the *extent* and I is called the *intent* of concept. For instance, the pair ($\{$Bill Clinton, Arnold Schwarzenegger$\}$,$\{$wins, election of, speech of$\}$) is a formal concept. The concepts are partially ordered according to the relation $\leq$:

$(E_1, I_1) \leq (E_2, I_2) \Leftrightarrow E_1 \subseteq E_2 \Leftrightarrow I_2 \subseteq I_1$

For instance, we have $C_2 \leq C_0$ for the concepts $C_2 =$ ($\{$Arnold Schwarzenegger,Roger Federer$\}$, $\{$wins,interview of, fan of$\}$) and $C_0 =$ ($\{$Michael Jackson, Roger Federer, Arnold Schwarzenegger$\}$, $\{$interview of, fan of$\}$). The relation $\leq$ form a complete lattice $\mathcal{L}$, called the *concept lattice* of $\mathbb{K}$.

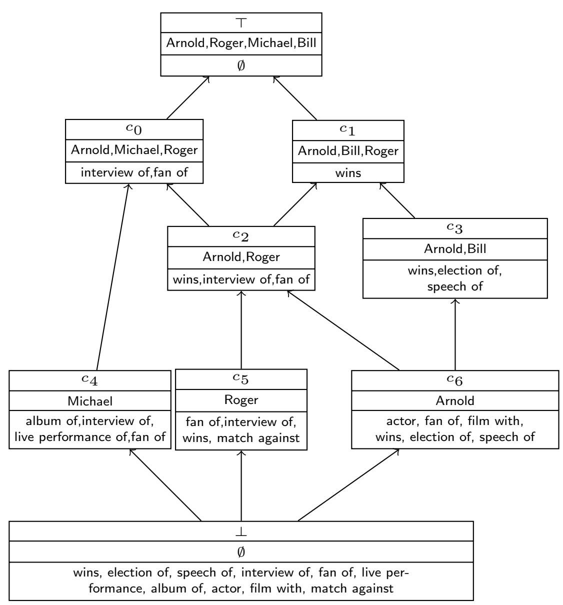


Fig. 3. Concept lattice for the formal context of figure (2). A concept box is contains a name, an extent and an intent.

3.3 The Concept Lattice : a Discriminative Knowledge Base

The general approach for building the concept lattice from linguistic data is similar to the work of Cimiano et al. [6]. The algorithm *AddIntent* [7] has been used for the construction of the lattice. It adopts an incremental procedure allowing dynamic lattice structuring according to new objects or attributes discovered from new utterances. Thus, a lattice could be seen as a knowledge-base already structured which could be adapted to a new corpus. This is an interesting property considering the weak evolutivity of classical lexical resources such as thesauri. According to this perspective, Priss [8] has been able to encapsulate the FrameNet thesauri within *relational concept analysis* framework.

As the figure (3) depicts, the concept lattice structure is organised according to several granularity layers. The upper part of the lattice is represented by general concepts grouping objects which share ambiguous attributes. The opposite part of the lattice has very specific concepts having ambiguous objects. The intermediate zone of the lattice provides concepts which seem more appropriate for LU disambiguation. Although the lattice model is generally considered as symbolic and discrete representation, the intent/extent overlapping reveals potential continuity of meanings. To our knowledge, these properties attached to concept lattices have not been exploited yet for an unsupervised WSD task.

4 Unsupervised Named Entity Annotation

In this section, we describe our FCA based methodology for annotation of relations between a NE and its context in an utterance. FCA is not only used to aggregate data, but also to perform a classification of NE. The unsupervised annotation is based on a selection of formal concepts according to n NE and its dependencies. We illustrate the method with an example and we finally introduce a dimensionality reduction method for the visualisation of formal concepts.

4.1 Concept Lattice Mining for Conceptual Annotation

Formal concepts are now considered as units of meaning potentially useful for LU annotation. As we noticed previously, the overlapping of intents and extents between formal concepts is linked to the intuition that some concepts are more similar than others since they share more objects or more attributes. Thus, the formal concepts could be associated to a metric space where the distance between two concepts measures a degree of semantic similarity.

In a new utterance, we suppose that a new NE $o \in O$ and its dependencies $Atts \subseteq A$ have been detected thanks to the extraction patterns (section 2.3). For a disambiguation task, we consider that the meaning of o relies on the meaning of its dependencies in $Atts$ occurring in the context : in other words, o can be annotated with a formal concept $x \in \mathcal{L}$ according to the concepts for the dependencies in $Atts$.

Our model for conceptual annotation of named entities is based on querying the concept lattice. In the lattice $\mathcal{L}$ the object o is associated to $Co = (\{o\}'', \{o\}')$

and similarly the concepts for the attributes of $Atts$ are the elements Ca_i from $C_{Atts} = \{(\{a_i\}', \{a_i\}'')|a_i \in Atts\}$. We are looking for a representative concept in the lattice which interpolates the concepts Co and Ca_i. We will call this concept x the *prototype* and we search it among the concepts containing o in their extent or at least one dependency a_i in their intent. More formally, $x \in \mathcal{L}(o, Atts)$ where $\mathcal{L}(o, Atts) = \{(E, I) \in \mathcal{L}|o \in E \vee Atts \cap I \neq \emptyset\}$. The prototype x is defined as the concept whose average dissimilarity to the concepts Co and Ca_i is minimal.

$$X = \underset{x \in \mathcal{L}(o,Atts)}{\operatorname{argmin}} \sum_{c \in C_{Atts} \cup \{Co\}} similarity(c, x) \tag{1}$$

In order to deal with similarities, we define two matrices $\mathcal{A}(o, Atts)$ and $\mathcal{O}(o, Atts)$ in which each row corresponds to a formal concept from $\mathcal{L}(o, Atts)$. The columns of $\mathcal{A}(o, Atts)$ are assigned to the intent of the concepts and similarly, the columns of $\mathcal{O}(o, Atts)$ are assigned to the extent of the concepts. Thus, the formal concepts are represented by a vector for extents and a vector for intents. Note that we can also consider $\mathcal{M}(o, Atts)$ which is the concatenation of the matrices $\mathcal{A}(o, Atts)$ and $\mathcal{O}(o, Atts)$.

Similarity measures can then be applied between the concept vectors of $\mathcal{A}(o, Atts)$, $\mathcal{O}(o, Atts)$ or $\mathcal{M}(o, Atts)$: measures such as Euclidean, cosine, correlation, Hamming or Jaccard can be chosen, depending of if we consider the vectors (and the formal context) as boolean or as weighted by the frequency counts of relations (cooccurrences) observed in the corpus. In the last case, the weights assigned to objects and attributes would be respectively $\sum_{a_i \in intent(C)} card(R(o, a_i))$ and $\sum_{o_i \in extent(C)} card(R(o_i, a))$.

4.2 Example from CoNLL Data

To illustrate the method, we propose to annotate the expression "English division" from which the pair $(o, Atts) =$ (English,{division}) is extracted. In a classical dictionary, the LU *division* is typically ambiguous because it can denotes, for instance, a group of military troops or a group of teams in an organised sport. The following list enumerates the concepts associated to $(o, Atts)$

1. (['SCOTTISH PREMIER DIVISION', 'SCOTTISH PREMIER', 'ENGLISH', 'FRENCH', 'SCOTTISH'], ['division', 'premier'])
2. (['DUTCH', 'ENGLISH', 'SCOTTISH'], ['division', 'results', 'league'])
3. (['ENGLISH', 'DUTCH'], ['division', 'draw', 'division leaders', 'league', 'results', 'result', 'news agency'])
4. (['ENGLISH', 'SCOTTISH'], ['league soccer', 'league', 'division', 'premier', 'results', 'league standings', 'league cup', 'summaries'])
5. (['ENGLISH', 'SCOTTISH PREMIER DIVISION', 'DUTCH', 'FRASER', 'MOROCCAN', 'SCOTTISH', 'SWISS', . . . ,'HUNGARIAN'], ['division'])
6. (['WELSH', 'ENGLISH'], ['division', 'referee', 'results'])
7. (['GERMAN', 'DUTCH', 'ENGLISH'], ['division', 'results', 'result'])
8. (['GERMAN', 'ENGLISH'], ['result', 'division', 'law', 'summaries', 'results'])
9. (['ENGLISH'], ['standings', 'premier', 'results', 'county', 'result', 'league cup', 'news agency', 'city', 'langage', 'soccer matches', 'play scores', . . .])
10. (['WELSH', 'SCOTTISH', 'FRENCH', 'GERMAN', 'AUSTRIA', 'DUTCH', 'MOROCCAN', 'ENGLISH', 'SWISS', 'POLISH'], ['division', 'results'])
11. (['ENGLISH', 'FRENCH', 'SCOTTISH'], ['division', 'premier', 'results', 'summaries'])
12. (['AUSTRIA', 'DUTCH', 'ENGLISH'], ['division', 'draw', 'results'])
13. (['SWISS', 'ENGLISH'], ['division', 'results', 'league leaders'])
14. (['NATIONAL LEAGUE EASTERN DIVISION', 'DUTCH', 'ENGLISH', 'AMERICAN LEAGUE EAST DIVISION', 'NATIONAL LEAGUE CENTRAL DIVISION', 'SCOTTISH'], ['league', 'division'])
15. (['GERMAN', 'ENGLISH', 'FRENCH', 'SCOTTISH'], ['division', 'results', 'summaries'])

In the lattice $\mathcal{L}$(English,{division}), the object "English" is represented in the lattice by the concept C_9 and the attribute "division" is represented by the concept C_5. Most concepts appears to denote the *sport division* meaning and it remains to select an appropriate concept for the annotation of the query. The prototype calculation has been done on this example according to several similarity metrics. The Euclidean and hamming distances chosen among others for the similarity measures, have both selected the concept C_4 which seems a acceptable for the annotation.

4.3 Dimensionality Reduction for Visualisation of Formal Concepts

The technique presented here has not yet been linked to the disambiguation process. It illustrates our intuition that continuous semantic provided with distance fits with a high structured representation such as concept lattices. For a better understanding of this intuition, we propose to visualise formal concepts through a cartographic representation where distance between formal concepts translates the notion of semantic proximity.

We have describe previously a simple way to associate a set of formal concepts to matrices. Since the vectors associated to concepts potentially have a huge dimension, we propose to use dimensionality reduction methods on the matrix $\mathcal{M}(o, Atts)$. These methods are able to compress $\mathcal{M}(o, Atts)$ such as each vector/concept representation is reduced to two dimensions. Among these methods we have chosen *curvilinear component analysis* (CCA) [9] which can be seen as a non linear extension to *principal component analysis*. The first results of this method are depicted by the figure (4).

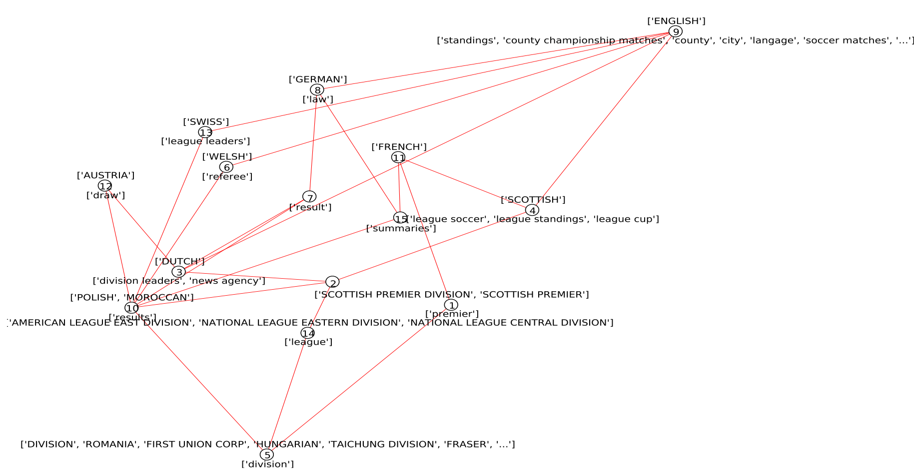


Fig. 4. Visualisation of formal concepts associated to the query *(English,* {*division*}*)* using CCA.

Reduced labelling has been used to improve the readability of the figure. In this scheme, the label for an object o is drawn above the *object concept* $\gamma(o) = (\{o\}'', \{o\}')$ while the label for an attribute a is drawn below the *attribute concept* $\mu(a) = (\{a\}', \{a\}'')$.

Our approach does not take advantage of the partial ordering between concepts that has been already computed. However, according to these figure, the general to specific ordering seems globally respected whereas it has not been taken into account for the rendering of the map : the most general and the most specific concepts occur to opposite sides of the map. The figure (4) also helps to understand where is the prototype C_4 among the other concepts resulting to the query.

5 Experiments and Evaluation

Previously, we have described an unsupervised method for conceptual annotation of NE. The evaluation of such unsupervised methods is subjective by nature since several concepts would be relevant to disambiguate a NE. In this section, we present a validation of our approach according to an existing task (supervised NE classification) that we are able to evaluate the performance. We then describe the cascade evaluation protocol [10] which considers the unsupervised conceptual annotation as a pre-processing step for a supervised NE classification task. We conclude the section with a study of the results obtained through this experiment.

5.1 CoNLL 2003 Data

The CoNLL-2003 named entity English data consists of three files : one training file (train), one development file (testa) and one test file (testb). Figure (5) gives an overview of the characteristics of the corpus.

	Articles	Sentences	Tokens	Locations	Misc	Organisations	Persons
Training corpus (train)	946	14987	203621	7140	3438	6321	6600
Development corpus (testa)	216	3466	51362	1837	922	1341	1842
Test (testb)	231	3684	46435	1668	702	1661	1617

Fig. 5. CoNLL 2003 corpus .

Our learning methods have been trained with the training and development data sets. The concept lattice obtained contains 14834 concepts for 8934 objects, 13983 attributes and 57170 relations in the formal context. The figure (6) depicts a conceptual annotation produced by our system on a CoNLL sample.

...				
eighth-seeded	JJ	I-NP	O	O
Olympic	JJ	I-NP	O	I-MISC
champion	NN	I-NP	Att52	O
Lindsay	NNP	I-NP	Obj46	I-PER
Davenport	NNP	I-NP	Obj46	I-PER
looking	VBG	I-VP	O	O
like	IN	I-PP	O	O
her	PRP	I-NP	O	O
most	RBS	I-ADVP	O	O
likely	JJ	I-ADVP	O	O
semifinal	JJ	I-NP	O	O
opponents	NNS	I-NP	O	O
.	.	O	O	O

– Att52= {champion, gold medallist, winner}
– Obj46= {Mary Pierce, Nate Miller, Kenny Harrison, Johan Museeuw, Boris Becker, Tanya Dubnicoff, Donovan Bailey, Carl Lewis, Richard Krajicek, Nathalie Lancien, Yvegeny Kafelnikov, Lindsay Davenport,Conchita Martinez,Thomas Muster}

Fig. 6. Example of conceptual annotation in the CoNLL 2003 corpus.

The Euclidean measure has been used for the prototype determination of the intent matrix $\mathcal{A}(\text{Lindsay Davenport}, \{\text{champion}\})$ and for the extent matrix $\mathcal{O}(\text{Lindsay Davenport}, \{\text{champion}\})$. It selects two concepts C_{52} and C_{46} : the intent of C_{52} provides a disambiguation of "champion" and the extent of C_{46} gives an annotation for "Lindsay Davenport".

5.2 Cascade Evaluation

In the framework of cascade evaluation [10], unsupervised learning is considered as a pre-processing step for a supervised NE classification task that we are able to evaluate. This cascade process reveals whether the conceptual annotation provides interesting enrichments to improve the supervised task on the CoNLL 2003 corpus. The protocol consists in comparing errors produced by two classifiers A and B, when they perform on the test corpus (testb), after a training step on the same training data (train + testa).

The system A is a supervised classifier trained normally on the labelled training corpus. As Ehrmann and Jacquet proposed [11], the system B provides two annotations for NE. The first is given by our unsupervised annotation system exploiting the concept lattice learned on the unlabelled training corpus. This pre-processing step provides enrichments to the initial corpus description. The system B can then benefit from these additional enrichments during the supervised learning step in order to produce the second annotation layer.

5.3 Experimental Results with Transformation-Based Learning

We have adapted the *transformation-based learning* (TBL) algorithm [12] to design a supervised NER system. The algorithm initializes the NE labels with a language model classifier (unigram), trained on the training corpus. The goal is to correct this initial classification according to the original NE labels specified in the training corpus. The next steps follow an iterative process : it corrects the initial incorrect classification by inferring a sequence of transformation rules. They are successively applied over the corpus in order to improve progressively the NE classification.

The resulting rules are instantiated from a list of extraction patterns defined manually. These patterns are able to explore co-texts features in a window of +/- 3 words : among the available features, we have considered the word, its morphosyntactic tag and the concept identifiers given by our unsupervised conceptual annotation method.

The figure (7) shows the results of the cascade evaluation. The left column indicates the performances reached by classifier A applied on the test corpus provided with morphosyntactic tagging. The right column corresponds to results obtained with the classifier B which has be used on the test corpus enriched with the conceptual annotation.

	A : TBL			B : conceptual annotation + TBL		
	Precision	Recall	$F_{\beta=1}$	Precision	Recall	$F_{\beta=1}$
Lieu	66.56%	66.19%	66.38	75.09%	65.65%	70.06
Organisation	52.22%	55.18%	53.66	61.55%	46.91%	53.24
Person	59.68%	68.62%	63.84	75.32%	57.82%	65.42
Misc.	83.58%	60.74%	70.35	85.21%	67.46%	75.30
Total	62.67%	63.61%	63.14	73.81%	59.27%	65.75

Fig. 7. Cascade evaluation results.

According to these results, the unsupervised annotation system increases the precision score to 11.14% and the $F_{\beta=1}$ (where $F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{recall})}{\beta^2\cdot\text{precision}+\text{recall}}$) measure to 2.61. However, a regression of 4.4% has been observed for recall.

6 Conclusion, Discussion and Future Work

We have presented an unsupervised method for named entity annotation, which is based on formal concept analysis. This method exploits a concept lattice structuring relations between named entities and their related lexical units, observed in text corpora. We have assumed that formal concepts are relevant units for the disambiguation of named entities. The selection of a concept for an annotation results of a query to the lattice. In addition, we have proposed a method based on dimensionality reduction for the visualisation of formal concepts. We have adapted the cascade evaluation protocol to validate the choice of concepts for annotation. It shows that a supervised named entity classifier improves its precision when it relies on the conceptual annotation produced by our unsupervised FCA-based system. Even if, our system does not reach the performances obtained by the best named entity recognizers, the first results are encouraging since some improvements are possible.

The syntactic extraction process could be improved by using a dependency parser : this could help to cover more syntactic patterns. It could also provide some additional information such as normalised forms (*e.g.* {is, was, were} → to be) or typed syntactic relations (*e.g.* subject-object, head-modifier).

The cascade evaluation framework, could compare our approach to other supervised and unsupervised classifiers : we would be particularly interested in the comparison with other FCA based classifiers [13]. At the present time, we are working on a semi-supervised lattice based classifier in which formal concepts are tagged with the NE labels (persons, locations, organisations, miscellaneous) available in the training corpus. Thus, the lattice would then be usable directly as a supervised NE classifier which would be able to produce unsupervised conceptual annotation with additional supervised labelling.

References

1. Pedersen, T.: 6. In: Unsupervised corpus-based methods for WSD. Springer (2006) 133–166
2. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations. Springer-Verlag (1999)
3. Grishman, R., Sundheim, B.: Message understanding conference 6: A brief history. In: COLING. (1996) 466–471
4. Sang, E.F.T.K., Meulder, F.D.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. CoRR **cs.CL/0306050** (2003)
5. Nazarenko, A., Zweigenbaum, P., Bouaud, J., Habert, B.: Corpus-based identification and refinement of semantic classes (1997)
6. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. Journal of Artificial Intelligence Research **24** (2005) 305–339
7. van der Merwe, D., Obiedkov, S.A., Kourie, D.G.: Addintent: A new incremental algorithm for constructing concept lattices. [15] 372–385
8. Priss, U., Old, L.J.: Modelling lexical databases with formal concept analysis. J. UCS **10**(8) (2004) 967–984
9. Hérault, J., Jausions-Picaud, C., Guérin-Dugué, A.: Curvilinear component analysis for high-dimensional data representation: I. theoretical aspects and practical use in the presence of noise. In: IWANN (2). (1999) 625–634
10. Candillier, L., Tellier, I., Torre, F., Bousquet, O.: Cascade evaluation of clustering algorithms. In Fürnkranz, J., Scheffer, T., Spiliopoulou, M., eds.: 17th European Conference on Machine Learning (ECML'2006). Volume LNAI 4212 of LNCS., Berlin, Germany, Springer Verlag (september 2006) 574–581
11. Ehrmann, M., Jacquet, G.: Vers une double annotation des entités nommées. Traitement automatique des langues **47**(3) (2006) 63–88
12. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics **21**(4) (1995) 543–565
13. Kuznetsov, S.O.: Machine learning and formal concept analysis. [15] 287–312
14. Girault, T.: Exploitation de treillis de Galois en désambiguïsation non supervisée d'entités nommées. In: 15ème conférence sur le Traitement Automatique des Langues Naturelles (TALN'08). (2008) 260–269
15. Eklund, P.W., ed.: Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings. In Eklund, P.W., ed.: ICFCA. Volume 2961 of Lecture Notes in Computer Science., Springer (2004)

An Efficient Hybrid Algorithm for Mining Frequent Closures and Generators

Laszlo Szathmary[1], Petko Valtchev[1], Amedeo Napoli[2], and Robert Godin[1]

[1] Dépt. d'Informatique UQAM, C.P. 8888,
Succ. Centre-Ville, Montréal H3C 3P8, Canada
Szathmary.L@gmail.com, valtchev.petko@uqam.ca, godin.robert@uqam.ca
[2] LORIA UMR 7503, B.P. 239, 54506 Vandœuvre-lès-Nancy Cedex, France
napoli@loria.fr

Abstract. The effective construction of many association rule bases requires the computation of both frequent closed and frequent generator itemsets (FCIs/FGs). However, these two tasks are rarely combined. Most of the existing solutions apply levelwise breadth-first traversal, though depth-first traversal, depending on data characteristics, is often superior. Hence, we address here a hybrid algorithm that combines the two different traversals. The proposed algorithm, *Eclat-Z*, extracts frequent itemsets (FIs) in a depth-first way. Then, the algorithm filters FCIs and FGs among FIs in a levelwise manner, and associates the generators to their closures. In *Eclat-Z* we present a generic technique for extending an arbitrary FI-miner algorithm in order to support the generation of minimal non-redundant association rules too. Experimental results indicate that *Eclat-Z* outperforms pure levelwise methods in most cases.

1 Introduction

The discovery of meaningful associations is a key data mining task [1]. An association miner typically proceeds in two steps: **(i)** extract all frequent patterns X of a database, and **(ii)** break each pattern X into a *premise* Y, and a *conclusion* $X \setminus Y$ parts to form a rule $Y \rightarrow X \setminus Y$. Interestingness measures, such as support and confidence, are applied to prune the set of extracted association rules. However, the number of the remaining rules may still be way too high to be practical. As a remedy, various concise representations of the family of valid association rules have been proposed [2,3,4,5,6]. A good survey can be found in [7].

Here we focus on the computation of frequent closed itemsets (FCIs) and frequent generators (FGs), which underlie the minimal non-redundant association rules ($\mathcal{MNR}$) for instance. Following [2], these are rules with the form $P \rightarrow Q \setminus P$, where $P \subset Q$, P is a *(minimal) generator* (a.k.a. key-sets or free-sets) and Q is a *closed itemset*. In other terms, in such rules the premise is minimal and the conclusion is maximal. As shown in [7], $\mathcal{MNR}$ is a *lossless*, *sound*, and *informative* representation of all valid rules. Moreover, further restrictions

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 47–58,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

can be imposed on the rules in $\mathcal{MNR}$, leading to more compact representations such as the *generic basis* or the *proper basis* (see [7] for a complete list).

From a computational point of view, constructing $\mathcal{MNR}$ or its sub-structures requires the family of frequent closed itemsets (FCIs) and their generators (FGs), and possibly the precedence order between FCIs. A few methods for extracting both FCIs and FGs have been published in the mining literature, e.g. *A-Close* [8] or *Titanic* [9]. Generators have been targeted within the concept analysis field as well [10], e.g. by *Zart* [11]. Well-known FCI/FG-miners exclusively apply levelwise strategies, although the levelwise itemset miners are knowingly outperformed by depth-first methods (e.g. *Eclat* [12], *Charm* [13], *Closet* [14]) on a broad range of dataset profiles, especially on dense ones. Hence the idea of designing a hybrid FCI/FG-miner. The algorithm that we propose called *Eclat-Z* splits the FCI/FG-mining task into three steps. First, it applies the well-known vertical algorithm *Eclat* for extracting the set of FIs. Second, it processes the FIs in a levelwise manner to filter FCIs and FGs. This is why *Eclat-Z* is said to be a hybrid algorithm. Finally, the algorithm associates FGs to their closures (FCIs) to provide the necessary starting point for the production of $\mathcal{MNR}$. Experimental results show that *Eclat-Z* outperforms two other efficient competitors, *A-Close* and *Zart*. During the design of *Eclat-Z* we had to face a challenge. The *Eclat* algorithm, due to its depth-first nature, provides the FIs in a completely unordered way. However, the levelwise post-processing steps require the FIs in ascending order by length. We managed to solve this problem with a special file indexing that proves to be efficient, generic, and gives no memory overhead at all. As we will see, the idea of *Eclat-Z* can be *generalized* and used for arbitrary FI-mining algorithm, either breadth-first or depth-first.

The main contribution of this work is a universal way of extending FI-miners for computing minimal non-redundant association rules too. We present a novel method for storing FIs in the file system if FIs are not provided in ascending order by length. Thanks to our special file indexing technique, which requires no additional memory, FIs can be sorted in a lengthwise manner. Once itemsets are available in this order, we show an original technique for filtering generators, closed itemsets, and associating generators to their closures.

The paper is organized as follows. Section 2 provides the basic concepts and essential definitions. In Section 3, we give an overview of the *Eclat* algorithm. This is followed in Section 4 with the detailed description of the *Eclat-Z* algorithm, where we also give a running example. Next, we provide experimental results in Section 5 for comparing the efficiency of *Eclat-Z* to *A-Close* and *Zart*. Finally, conclusions and future work are discussed in Section 6.

2 Basic Concepts

Consider the following 5×5 sample dataset: $\mathcal{D} = \{(1, ABDE), (2, AC), (3, ABCE), (4, BCE), (5, ABCE)\}$. Throughout the paper, we will refer to this example as **"dataset $\mathcal{D}$"**.

We consider a set of *objects* or *transactions* $\mathcal{O} = \{o_1, o_2, \ldots, o_m\}$, a set of *attributes* or *items* $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$, and a relation $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$. A set of items is called an *itemset*. Each transaction has a unique identifier (*tid*), and a set of transactions is called a *tidset*.[3] For an itemset X, we denote its corresponding tidset, often called its *image*, as $t(X)$. For instance, in dataset $\mathcal{D}$, the image of AB is 135, i.e. $t(AB) = 135$. Conversely, $i(Y)$ is the itemset corresponding to a tidset Y. The *length* of an itemset is its cardinality, whereas an itemset of length k is called a k-itemset (or a k-long itemset). The *support* of an itemset X, denoted by $supp(X)$, is the size of its image, i.e. $supp(X) = |t(X)|$. An itemset X is called *frequent*, if its support is not less than a given *minimum support* (denoted by *min_supp*), i.e. $supp(X) \geq min\_supp$. The image function induces an equivalence relation on $\wp(\mathcal{A})$: $X \cong Z$ iff $t(X) = t(Z)$ [15]. Moreover, an equivalence class has a unique maximum w.r.t. set inclusion and possibly several minima, respectively called *closed* itemset (a.k.a. *concept intents* in concept analysis [16]) and *generator* itemsets (a.k.a. *key-sets* in database theory or free-sets). The support-oriented definitions exploiting the monotony of support upon $\subseteq$ in $\wp(\mathcal{A})$ are as follows:

Definition 1 (closed itemset; generator). *An itemset X is* closed *(*generator[4]*) if it has no proper superset (subset) with the same support (respectively).*

The *closure* of an itemset X (denoted by X'' following standard FCA notation) is thus the largest itemset in the equivalence class of X. For instance, in dataset $\mathcal{D}$, the sets AB and AC are generators, and their closures are ABE and AC, respectively (i.e. the equivalence class of AC is a singleton). In our approach, we rely on the following two properties:

Property 1. A closed itemset cannot be the generator of a larger itemset.

Property 2. The closure of a frequent non-closed generator g is the smallest proper superset of g in the set of frequent closed itemsets.

An association rule r: $P_1 \rightarrow P_2$ involves two itemsets $P_1, P_2 \subseteq \mathcal{A}$, s.t. $P_1 \cap P_2 = \emptyset$, and $P_2 \neq \emptyset$. The support of a rule r is $supp(r) = supp(P_1 \cup P_2)$ and its *confidence* $conf(r) = supp(P_1 \cup P_2)/supp(P_1)$. *Frequent* rules are defined in a way similar to frequent itemsets, whereas *confident* rules play an equivalent role for the confidence measure. A *valid* rule is both frequent and confident. Finding all valid rules in a database is the target of a typical association rule mining task.

As their number may grow up to exponential, reduced sub-families of valid rules are defined, which nevertheless convey the same information (*lossless*). Associated expansion mechanisms allow for the entire family to be retrieved from the reduced ones without any non-valid rules to be mixed in (*soundness*). The minimal non-redundant association rule family ($\mathcal{MNR}$) is made of rules $P \rightarrow Q \setminus P$, where $P \subset Q$, P is a *(minimal) generator* and Q is a *closed*

[3] For convenience, we write an itemset $\{A, B, E\}$ as ABE, and a tidset {1,3,5} as 135.
[4] Generators are also called “keys” or “key itemsets”.

itemset. A more restricted family arises from the additional constraint of P and Q belonging to the same equivalence class, i.e. $P'' = Q$. It is known as the *generic basis* for exact (100% confidence) association rules [7]. Here the basis refers to the non-redundancy of the family w.r.t. a specific criterion. Inexact rule bases can also be defined by means of generators and closures, e.g. the *informative basis* [7], which further involves the inclusion order between closures.

3 Vertical Frequent Itemset Mining

The frequent itemset mining methods from the literature can be roughly split into breadth-first and depth-first miners. *Apriori*-like [1] levelwise breadth-first algorithms exploit the anti-monotony of frequent itemsets in a straightforward manner: they advance one level at a time, generating candidates for the next level and then computing their support upon the database. Depth-first algorithms, in contrast, organize the search space in a tree. Typically using a sorted representation of the itemsets, they factor out common prefixes and hence limit the computing effort. Typical depth-first FI-miners include *Eclat* [17] and *FP-growth* [18].

3.1 Common Characteristics

Eclat was the first FI-miner using a vertical encoding of the database combined with a depth-first traversal of the search space (organized in a prefix-tree) [17].

Vertical miners rely on a specific layout of the database that presents it in an item-based, instead of a transaction-based, fashion. Thus, an additional effort is required to transpose the global data matrix in a pre-processing step. However, this effort pays back since afterwards the secondary storage does not need to be accessed anymore. Indeed, the support of an itemset can be computed by explicitly constructing its tidset which in turn can be built on top of the tidsets of the individual items. Moreover, in [12], it is shown that the support of any k-itemset can be determined by intersecting the tid-lists of any two of its $(k-1)$-long subsets.

The central data structure in a vertical FI-miner is the IT-tree that represents both the search space and the final result. The IT-tree is an extended prefix-tree whose nodes are $X \times t(X)$ pairs. With respect to a classical prefix-tree or trie, in an IT-tree the itemset X provides the entire prefix from the root to the node labeled by it (and not the difference with the parent node prefix).

Example. Figure 1 presents the IT-tree of our example. Observe that the node $ABC \times 35$ for instance can be computed by combining the nodes $AB \times 135$ and $AC \times 235$. To that end, tidsets are intersected and itemsets are joined. The support of ABC is readily established to 2.

3.2 Eclat

Eclat is a plain FI-miner traversing the IT-tree in a depth-first manner in a pre-order way, from left-to-right [17,12].

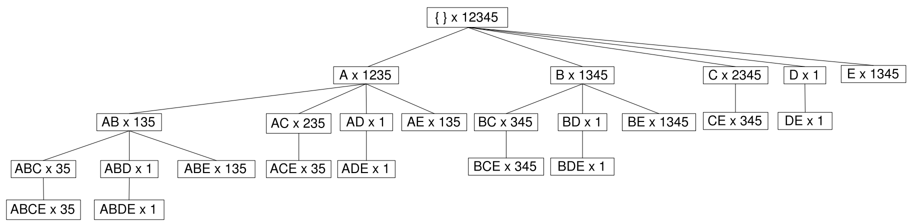


Fig. 1. IT-tree: Itemset-Tidset search tree of dataset $\mathcal{D}$

At the beginning, the IT-tree is reduced to its root (empty itemset). *Eclat* extends the root one level downwards by adding the nodes of all frequent 1-itemsets. Then, each of the new nodes is extended similarly: first, candidate descendant nodes are formed by adding to its itemset the itemset of each right sibling; second, the tidsets are computed by intersection and the supports are established; and third, the frequent itemsets are added as effective descendant nodes of the current node.

Running example. Using Figure 1, we illustrate the execution of *Eclat* on dataset $\mathcal{D}$ with $min\_supp = 1$ (20%). Initially, the IT-tree comprises only the root node whose support is 100%. Frequent items with their tidsets are then added under the root. Each of the new nodes is recursively extended, following a left-to-right order and processing the corresponding sub-trees in a pre-order fashion. For instance, the subtree of A comprises all frequent itemsets starting with A. Thus, at step two, all 2-long supersets of A are formed using the right siblings of A (frequent 1-itemsets). As AB, AC, AD, and AE are all frequent, they are added as descendant nodes under the node of A. The extend procedure is then recursively called on AB and the computation goes one level deeper in the IT-tree. When the algorithm stops, all frequent itemsets are discovered.

4 The Eclat-Z Algorithm

Eclat-Z is a hybrid algorithm that combines the vertical FI-miner *Eclat* with an original levelwise extension. *Eclat* finds all FIs that we save in the file system. Then, this file is processed in a levelwise manner, i.e. itemsets are read in *ascending* order by length, generators and closed itemsets are filtered, and finally generators are associated to their closures. In the following, we present the algorithm in detail.

4.1 Processing Itemsets in Ascending Order by Length

Sorting itemsets in ascending order by length is required for such algorithms that produce FIs in an unordered way. *Eclat*, the algorithm used as itemset mining "engine" here, is a good example of such an algorithm. Levelwise algorithms, like

Table 1. Order of frequent itemsets produced by *Eclat*

order	itemset	support
1)	ABCE	2
2)	ABC	2
3)	ABE	3
4)	AB	3
5)	ACE	2
6)	AC	3
7)	AE	3
8)	A	4

order	itemset	support
9)	BCE	3
10)	BC	3
11)	BE	4
12)	B	4
13)	CE	3
14)	C	4
15)	E	4

Apriori, represent an easier case because they produce FIs in ascending order by length. If someone wants to use such an algorithm, he can continue with the second part in Section 4.2. Here, in the first part, we present an efficient, file-system based approach to process FIs in ascending order by their length. For our example, we use dataset $\mathcal{D}$ with $min\_supp = 2$ (40%). *Eclat* produces FIs in an unordered way, as shown in Table 1.

As in practice it is impossible to keep all FIs in the main memory, we write FIs in a binary file. In main memory we have an index, called *PosIndex*, for storing file positions (Figure 2). *PosIndex* is a simple array of integers. At position k it indicates where the last k-long itemset is written in the binary file. *PosIndex* must always be kept up-to-date. On the left part of Figure 2, it is indicated how *PosIndex* changes in time between t_0 and t_{15}. The right side of the same figure shows the final state of *PosIndex*. Figure 3 shows the contents of the file. For conciseness, support values are omitted. The file structure is explained through the following examples.

Running example for storing itemsets. In our implementation of *Eclat* an IT-node is processed when we return in recursion. Thus, the first FI found by *Eclat* is $ABCE$ (see Table 1). It is a 4-itemset. The size of the *PosIndex* array is dynamically increased to size $4 + 1$ (+1, because position 0 is not used). The array is initialized: at each of its position we store -1 (time t_0). As the length of the found itemset is 4, we read the value of *PosIndex* at position 4. This value (-1), together with the itemset is written to the binary file (see Figure 3). The value that we read from *PosIndex* is a *backward pointer* that shows the file position of the previous itemset *with the same length.* As the value is -1 here, it simply means that this is the first itemset of this length. After writing $ABCE$ to the file, the 4^{th} position of *PosIndex* is updated to 0 (t_1), because the last 4-long itemset together with its backward pointer was written to position 0 in the file. ABC is written similarly, and *PosIndex* is updated (t_2). When ABE is written to the file, its backward pointer is set to 5. This value is read from *PosIndex* at position 3, since ABE is a 3-itemset. The process continues until all FIs are found. The final state of *PosIndex* is indicated on the right side of Figure 2.

	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
0	-1															
1	-1								26				38		43	45
2	-1				13		20	23			32	35		40		
3	-1		5	9		16				28						
4	-1	0														

0	-1
1	45
2	40
3	28
4	0

Fig. 2. The *PosIndex* structure. Timeline **(left)** and final state **(right)**

file positions:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	...
file contents:	-1	A	B	C	E	-1	A	B	C	5	A	B	E	-1	A	B	9	A	C	E	13	A	C	

...	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
	20	A	E	-1	A	16	B	C	E	23	B	C	32	B	E	26	B	35	C	E	38	C	43	E

Fig. 3. Contents of the file with the FIs. File positions are also indicated

Running example for reading itemsets. Figure 3 illustrates how to read k-itemsets from the file (here $k = 1$, shown in dark grey). First, we look for the last 1-itemset, which is registered in *PosIndex* (Figure 2) at position 1. The value points at position 45 in the file. Itemset E is read, and we seek to the previous 1-itemset at position 43. C is read, seek to position 38. B is read, seek to position 26. A is read, and -1 indicates that there are no more 1-itemsets. This way FIs can be processed in ascending order by length.

4.2 Finding Generators, Closures, and Associating Them

In the previous subsection, we presented the first part of the algorithm, i.e. how to get frequent itemsets in ascending order by their length, even if they are produced in an unordered way. In this subsection we continue with the second part namely how to associate generators to their closures, once FIs are available in a good order. The main block is shown in Algorithm 1. Two kinds of tables are used, namely F_i for i-long frequent, and Z_i for i-long frequent closed itemsets.

The `readTable` function is in charge of reading frequent itemsets of a given length. If such an algorithm is used that produces FIs in an unordered way, like *Eclat*, then `readTable` reads FIs from the binary file, as explained previously. The function returns FIs in an F_i table. Fields of the table are initialized: itemsets are marked as “keys” and “closed”. Of course, during the post-processing step these values may change. Frequent attributes (frequent 1-itemsets) represent a special case. If they are present in each object of the dataset, then they are not generators, because they have a smaller subset with the same support,

Algorithm 1 (Eclat-Z):

$maxItemsetLength \leftarrow$ (size of the largest FI found by the FI-miner);
$\mathcal{FG} \leftarrow \{\}$; *// global list of frequent generators*
$F_1 \leftarrow$ readTable(1); *// get frequent 1-itemsets*
for ($i \leftarrow 1$; $i < maxItemsetLength$; $i \leftarrow i + 1$) {
    $F_{i+1} \leftarrow$ readTable($i + 1$); *// get frequent* $(i + 1)$*-itemsets*
    findKeysAndClosedItemsets(F_{i+1}, F_i); *// filtering*
    $Z_i \leftarrow \{l \in F_i \mid l.\text{closed} = \text{true}\}$;
    Find-Generators(Z_i);
}
$Z_i \leftarrow \{l \in F_i \mid l.\text{closed} = \text{true}\}$;
Find-Generators(Z_i);

return $\bigcup_i Z_i$;

namely the empty set. In this case the empty set is a useful generator (w.r.t. rule generation).

The `findKeysAndClosedItemsets` procedure is in charge of filtering FCIs and FGs among FIs. The filtering procedure is based on Def. 1.

The `Find-Generators` procedure takes as input a Z_i table. The method is the following. For each frequent closed itemset z in Z_i, it finds its proper subsets in the global list $\mathcal{FG}$, registers them as generators of z, deletes them from $\mathcal{FG}$, and adds non-closed generators from F_i to $\mathcal{FG}$. Properties 1 and 2 guarantee that whenever the subsets of an FCI are looked up in the list $\mathcal{FG}$, only its generators are returned.

Running example. The execution of *Eclat-Z* on dataset $\mathcal{D}$ with $min\_supp = 2$ is illustrated in Table 2. Frequent 1-itemsets are read and stored in F_1. Since their support values are less than the total number of objects in the dataset, all of them are keys (generators). They are also marked as "closed". Then, frequent 2-itemsets are read too and stored in F_2. The algorithm compares F_2 to F_1 in order to filter non-closed and non-generator elements. The itemset BE has two subsets in F_1 with the same support, which means that BE is not a generator, and B and E are not closed (by Def. 1). The remaining closed itemsets A and C are copied from F_1 to Z_1, and their generators are determined. In the global list of frequent generators ($\mathcal{FG}$), which is still empty, they have no subsets, which means that both A and C are generators themselves. Non-closed generators of F_1 (B and E) are added to the $\mathcal{FG}$ list. Comparing F_3 to F_2, it turns out that ABE and BCE are not generators, while AB, AE, BC, and CE are not closed. The remaining closed itemsets AC and BE are copied to Z_2. The generator of AC is itself, and the generators of BE are B and E. These two generators are removed from $\mathcal{FG}$ and AB, AE, BC, and CE are added to $\mathcal{FG}$. The 4-itemset $ABCE$ is the longest FI in the example. Its generators are read from $\mathcal{FG}$. When the algorithm stops, all FCIs *with* their generators are determined (see the union

Table 2. Execution of *Eclat-Z* on dataset $\mathcal{D}$ with $min\_supp = 2$ (40%)

F_1	key?	supp	closed?
{A}	yes	4	yes
{B}	yes	4	~~yes~~
{C}	yes	4	yes
{E}	yes	4	~~yes~~

Z_1	supp	generators
{A}	4	
{C}	4	

$\mathcal{FG}_{before} = \{\}$
$\mathcal{FG}_{after} = \{B, E\}$

F_2	key?	supp	closed?
{AB}	yes	3	~~yes~~
{AC}	yes	3	yes
{AE}	yes	3	~~yes~~
{BC}	yes	3	~~yes~~
{BE}	~~yes~~	4	yes
{CE}	yes	3	~~yes~~

Z_2	supp	generators
{AC}	3	
{BE}	4	{B, E}

$\mathcal{FG}_{before} = \{B, E\}$
$\mathcal{FG}_{after} = \{AB, AE, BC, CE\}$

F_3	key?	supp	closed?
{ABC}	yes	2	~~yes~~
{ABE}	~~yes~~	3	yes
{ACE}	yes	2	~~yes~~
{BCE}	~~yes~~	3	yes

Z_3	supp	generators
{ABE}	3	{AB, AE}
{BCE}	3	{BC, CE}

$\mathcal{FG}_{before} = \{AB, AE, BC, CE\}$
$\mathcal{FG}_{after} = \{ABC, ACE\}$

F_4	key?	supp	closed?
{ABCE}	~~yes~~	2	yes

Z_4	supp	generators
{ABCE}	2	{ABC, ACE}

$\mathcal{FG}_{before} = \{ABC, ACE\}$
$\mathcal{FG}_{after} = \{\}$

of the Z_i tables in Table 2). If *Eclat-Z* leaves the generators of a closed itemset empty, it simply means that the generator is identical to the closed itemset (this is the case for A, C, and AC in the example). Recall that the support of a generator is equivalent to the support of its closure.

5 Experimental Results

We evaluated *Eclat-Z* against *Zart* [11] and *A-Close* [8]. The algorithms were implemented in Java under the Coron data mining platform [19].[5] The experiments were carried out on a bi-processor Intel Quad Core Xeon 2.33 GHz machine running under Ubuntu GNU/Linux with 4 GB RAM. For the experiments we have used the following datasets: T20I6D100K, C20D10K, and Mushrooms. The T20I6D100K[6] is a sparse dataset, constructed according to the properties of market basket data that are typical weakly correlated data. The C20D10K is a census dataset from the PUMS sample file, while the Mushrooms[7] describes mushrooms characteristics. The last two are highly correlated datasets.

[5] http://coron.loria.fr
[6] http://www.almaden.ibm.com/software/quest/Resources/
[7] http://kdd.ics.uci.edu/

Table 3. Response times of *Eclat-Z* and other statistics (response times of *Zart* and *A-Close*, number of FIs, number of FCIs, number of FGs, and the proportion of the number of FGs to the number of FIs)

min_supp	execution time (sec.)			# FIs	# FCIs	# FGs	$\frac{\#FGs}{\#FIs}$
	Eclat-Z	Zart	A-Close				
T20I6D100K							
1%	4.11	6.58	24.06	1,534	1,534	1,534	100.00%
0.75%	3.31	12.39	29.44	4,710	4,710	4,710	100.00%
0.5%	5.82	34.61	72.88	26,836	26,208	26,305	98.02%
0.25%	24.55	121.03	204.69	155,163	149,217	149,447	96.32%
C20D10K							
30%	1.07	6.27	11.27	5,319	951	967	18.18%
20%	1.71	11.32	20.77	20,239	2,519	2,671	13.20%
10%	5.17	23.99	40.70	89,883	8,777	9,331	10.38%
5%	20.24	49.29	62.64	352,611	21,213	23,051	6.54%
MUSHROOMS							
30%	0.82	2.87	5.86	2,587	425	544	21.03%
20%	3.36	7.72	11.68	53,337	1,169	1,704	3.19%
10%	37.46	46.37	29.43	600,817	4,850	7,585	1.26%
5%	368.03	391.97	50.20	4,137,547	12,789	21,391	0.52%

Table 3 contains the experimental evaluation of *Eclat-Z* against *Zart* and *A-Close*. All times reported are real, wall clock times as obtained from the Unix *time* command between input and output. We have chosen *Zart* and *A-Close* because they represent two efficient algorithms that produce exactly the same output as *Eclat-Z*. *Zart* and *A-Close* are both levelwise algorithms. *Zart* is an extension of *Pascal* [15], i.e. first it finds all FIs using pattern-counting inference, then it filters FCIs, and finally the algorithm associates FGs to their closures. *A-Close* reduces the search space to FGs only, then it calculates the closure for each generator. The way *A-Close* computes the closures of generators is quite expensive because of the huge number of intersection operations.

In the sparse dataset T20I6D100K, almost all frequent itemsets are closed and generators at the same time. It means that most equivalence classes are singletons, thus *A-Close* cannot reduce the search space significantly. Since the closure computation of *A-Close* is quite expensive, *Eclat-Z* performs much better. *Zart* and *Eclat-Z* are similar in the sense that first both algorithms extract FIs. While *Zart* is based on *Pascal*, *Eclat-Z* is based upon *Eclat*. The better performance of *Eclat-Z* is due to the better performance of its FI-miner "engine".

In datasets C20D10K and MUSHROOMS, the number of FGs is considerably less than the total number of FIs. In this case, *Zart* can take advantage of its pattern counting inference technique, and *A-Close* can benefit from its search space reduction. Despite these optimizations, *Eclat-Z* still outperforms the two algorithms in most cases. However, if the number of FGs is *much less* than the number of FIs (for instance in MUSHROOMS by $min\_supp = 5\%$), *A-Close* gives better response times.

As a conclusion we can say that *Eclat-Z* clearly outperforms its levelwise competitors on sparse datasets, and it also performs very well on dense, highly correlated datasets if the minimum support threshold is not set too low.

6 Conclusion

In this paper we presented a generic algorithm called *Eclat-Z* that identifies FCIs and their associated generators. From this output numerous concise representations of valid association rules can be readily derived.

Eclat-Z splits the FCI/FG-mining problem into three tasks: **(1)** FI-mining, **(2)** filtering FCIs and FGs, and **(3)** associating FGs to their closures (FCIs). The FI-mining part is solved by a well-known depth-first algorithm, *Eclat*. However, with *Eclat* we had to face a challenge: it produces itemsets in an unordered way. Thanks to a special file indexing technique, we managed to solve this issue in an efficient way, thus steps **(2)** and **(3)** can post-process FIs in a levelwise manner. As seen, the idea of the hybrid algorithm *Eclat-Z* can be generalized and used for *any* FI-mining algorithm, be it breadth-first or depth-first. Experimental results prove that *Eclat-Z* is highly efficient and outperforms its levelwise competitors in most cases.

The study led to a range of exciting questions that are currently investigated. *Eclat-Z* is highly efficient, but first it traverses the whole set of FIs. While in sparse datasets it causes no problem, it can be a drawback in dense datasets with very low minimum support. It would be interesting to combine the search space reduction of *A-Close* with the efficiency of *Eclat-Z*. A further challenge lies in the computation of the FCI precedence order that underlies some of the association rule bases from the literature.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proc. of the 20th Intl. Conf. on Very Large Data Bases (VLDB '94), San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1994) 487–499
2. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets. In: Proc. of the Computational Logic (CL '00). Volume 1861 of LNAI., Springer (2000) 972–986
3. Kryszkiewicz, M.: Representative Association Rules. In: Proc. of the 2nd Pacific-Asia Conf. on Research and Development in Knowledge Discovery and Data Mining (PAKDD '98), Melbourne, Australia, Springer-Verlag (1998) 198–209
4. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Closed set based discovery of small covers for association rules. In: Proc. 15emes Journees Bases de Donnees Avancees (BDA). (1999) 361–381
5. Duquenne, V.: Contextual implications between attributes and some representational properties for finite lattices. In: Beitraege zur Begriffsanalyse, B.I. Wissenschaftsverlag, Mannheim (1987) 213–239
6. Luxenburger, M.: Implications partielles dans un contexte. Mathématiques, Informatique et Sciences Humaines **113** (1991) 35–55

7. Kryszkiewicz, M.: Concise Representations of Association Rules. In: Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery. (2002) 92–109
8. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering Frequent Closed Itemsets for Association Rules. In: Proc. of the 7th Intl. Conf. on Database Theory (ICDT '99), Jerusalem, Israel (1999) 398–416
9. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing Iceberg Concept Lattices with TITANIC. Data and Knowledge Engineering **42**(2) (2002) 189–222
10. Valtchev, P., Missaoui, R., Godin, R.: Formal Concept Analysis for Knowledge Discovery and Data Mining: The New Challenges. In: Proc. of the 2nd Intl. Conf. on Formal Concept Analysis, Springer Verlag (Feb 2004) 352–371
11. Szathmary, L., Napoli, A., Kuznetsov, S.O.: ZART: A Multifunctional Itemset Mining Algorithm. In: Proc. of the 5th Intl. Conf. on Concept Lattices and Their Applications (CLA '07), Montpellier, France (Oct 2007) 26–37
12. Zaki, M.J.: Scalable Algorithms for Association Mining. IEEE Transactions on Knowledge and Data Engineering **12**(3) (2000) 372–390
13. Zaki, M.J., Hsiao, C.J.: CHARM: An Efficient Algorithm for Closed Itemset Mining. In: SIAM Intl. Conf. on Data Mining (SDM' 02). (Apr 2002) 33–43
14. Pei, J., Han, J., Mao, R.: CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. In: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. (2000) 21–30
15. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. SIGKDD Explor. Newsl. **2**(2) (2000) 66–75
16. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin/Heidelberg (1999)
17. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New Algorithms for Fast Discovery of Association Rules. In: Proc. of the 3rd Intl. Conf. on Knowledge Discovery in Databases. (August 1997) 283–286
18. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD '00), ACM Press (2000) 1–12
19. Szathmary, L.: Symbolic Data Mining Methods with the Coron Platform. PhD Thesis in Computer Science, Univ. Henri Poincaré – Nancy 1, France (Nov 2006)

Optimal Decompositions of Matrices with Grades into Binary and Graded Matrices*

Eduard Bartl[1], Radim Belohlavek[1,2], Jan Konecny[1]

[1] Dept. Systems Science and Industrial Engineering
T. J. Watson School of Engineering and Applied Science
Binghamton University–SUNY, PO Box 6000, Binghamton, NY 13902–6000, USA
ebartl1@binghamton.edu,rbelohla@binghamton.edu,jkonecn1@binghamton.edu
[2] Dept. Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic

Abstract. The paper contributes to factor analysis of relational data. We study the problem of decomposition of object-attribute matrices with grades, i.e. matrices whose entries contain degrees to which objects have attributes. The degrees are taken from a bounded partially ordered scale. Examples of such matrices are binary matrices, matrices with entries from a finite chain, or matrices with entries from the unit interval $[0, 1]$. We study the problem of decomposition of a given object-attribute matrix I with grades into an object-factor matrix A and a binary factor-attribute matrix B, with the number of factors as small as possible. We present a theorem describing optimal decompositions. The theorem shows that decompositions which use as factors particular formal concepts associated to I are optimal in that the number of factors involved is the smallest possible. Furthermore, we present an approximation algorithm for finding those decompositions and illustrative examples.

1 Introduction and Problem Setting

Problem description in brief This paper presents results on optimal decompositions of matrices with grades. Examples of such matrices are binary (or Boolean) matrices, i.e. matrices which entries are 0 or 1. Other examples are matrices which contain numbers from the unit interval $[0, 1]$ as their entries. In general we consider non-numerical matrices with entries from particular complete lattices L (binary matrices and matrices with entries from $[0, 1]$ are particular examples with $L = \{0, 1\}$ and $L = [0, 1]$, respectively).

We consider the following problem. Let L be a partially ordered scale bounded from below and above by 0 and 1 (details specified later). Given an $n \times m$ matrix I with entries from L (i.e. $I_{ij} \in L$), we want to decompose I into a product

$$I = A \circ B$$

* Supported by grant No. 1ET101370417 of GA AV ČR and by institutional support, research plan MSM 6198959214.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 59–70,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

of an $n \times k$ matrix A with entries from L (i.e. $A_{il} \in L$) and a $k \times m$ *binary matrix* B (i.e. $B_{lj} \in \{0,1\}$) with *k as small as possible.* The composition operation $\circ$ which we consider is defined by

$$(A \circ B)_{ij} = \bigvee_{l=1}^{k} A_{il} \otimes B_{lj}, \tag{1}$$

where $\otimes$ is defined by $a \otimes 1 = a$ and $a \otimes 0 = 0$. Note that if $L = \{0,1\}$ then $A \circ B$ is the well-known Boolean product of binary matrices. Note also that if we allow $A_{il} \in L$ and $B_{lj} \in L$ and if $\otimes$ is a t-norm then $\circ$ is the product of graded matrices well-known in fuzzy set theory, see e.g. [15], and that such decompositions were considered in [4, 7].

Factor analysis model For a decomposition $I = A \circ B$ given by (1), I_{ij} can be interpreted as a degree to which there is a factor l such that l applies to object i *and* l is associated to attribute j (j is a particular manifestation of l). This way, a decomposition $I = A \circ B$ provides us with a factor analysis model (see [1, 13, 16] for references on factor analysis): A relationship between objects and original attributes given by I is described using a relationship between the objects and new variables, called factors, which is given by A, and a relationship between factors and the original attributes, which is given by B. Note that we assume that B is binary, i.e. that the relationship between factors and attributes is a yes-or-no relationship. This feature distinguishes our approach from those which we considered earlier.

Needless to say, one can consider decompositions $I = A \circ B$ given by (1), in which A is binary and B arbitrary. Obviously, using $I^T = B^T \circ A^T$, one can reduce this type of decomposition to the first type (A arbitrary, B binary). Therefore, we do not consider such case.

Contribution of the paper We present a theorem regarding optimal decompositions of a given matrix I which shows that decompositions which use as factors particular formal concepts, called crisply generated concepts, are optimal in that they involve the least number of factors among all decompositions of I. Furthermore, we present an approximation algorithm for finding those decompositions and provide illustrative examples.

Related and previous work The paper is a continuation of our previous work [4, 6, 7]. In particular, in [4, 7] we considered decompositions $I = A \circ B$ given by (1), in which both A and B were arbitrary, i.e. none of them was required to be binary.

Preliminaries from fuzzy logic We use standard notions of fuzzy logic and fuzzy sets, see e.g. [2, 12, 15]. In particular, we use complete residuated lattices as structures of truth degrees. Recall that a complete residuated lattice is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice, $\langle L, \otimes, 1\rangle$ is a commutative monoid, and $\otimes$ and $\rightarrow$ satisfy so-called adjointness condition, i.e. $a \otimes b \leq c$ if and only if $a \leq b \rightarrow c$. We assume familiarity with examples and basic properties of residuated lattices. As an example, for $L = [0,1]$, $a \otimes b = \max(0, a+b-1)$, $a \to b = \min(1, 1-a+b)$, the algebra $\mathbf{L} = \langle [0,1], \wedge, \vee, \otimes, \to, 0, 1\rangle$ is a complete residuated lattice (so-called standard Łukasiewicz algebra). An L-set in a universe set U is a mapping $A : U \to L$.

2 Optimal Decompositions

2.1 Composition as $\bigvee$-superposition of matrices

We first observe that $I = A \circ B$ for $n \times k$ and $k \times m$ matrices A (graded) and B (binary) means that I is a $\bigvee$-superposition of particular rectangular matrices.

Definition 1. *Let $K_1, K_2 \subseteq L$. An $n \times m$ matrix J with entries from L is called (K_1, K_2)-rectangular iff there exist L-sets C in $\{1, \ldots, n\}$ and D in $\{1, \ldots, m\}$ with $C(i) \in K_1$ and $D(j) \in K_2$ such that $J = C \otimes D$, i.e.*

$$J_{ij} = C(i) \otimes D(j) \tag{2}$$

for $1 \le i \le n$, $1 \le j \le m$.

In particular, we need $(L, \{0,1\})$-rectangular matrices and call these just "rectangular". The term "rectangular" is inspired by the "shape" of such matrices. The following matrices are examples of $(\{0,1\}, \{0,1\})$-rectangular (J_1) and $([0,1], \{0,1\})$-rectangular (J_2) matrices:

$$J_1 = \begin{pmatrix} 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&\mathbf{1}&\mathbf{1}&\mathbf{1}&0&0 \\ 0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0 \end{pmatrix}, \quad J_2 = \begin{pmatrix} 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \\ 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \\ 0.0&0.0&\mathbf{0.5}&\mathbf{0.5}&\mathbf{0.5}&0.0&0.0 \\ 0.0&0.0&\mathbf{1.0}&\mathbf{1.0}&\mathbf{1.0}&0.0&0.0 \\ 0.0&0.0&\mathbf{0.2}&\mathbf{0.2}&\mathbf{0.2}&0.0&0.0 \\ 0.0&0.0&\mathbf{1.0}&\mathbf{1.0}&\mathbf{1.0}&0.0&0.0 \\ 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \\ 0.0&0.0&0.0&0.0&0.0&0.0&0.0 \end{pmatrix}.$$

In the above example, $J_1 = C \otimes D$ where C and D are characteristic functions of $\{3,4,5,6\}$ and $\{3,4,5\}$, respectively; $J_2 = C \otimes D$ where $C(1) = C(2) = C(7) = C(8) = 0$, $C(3) = 0.5$, $C(4) = 1.0$, $C(5) = 0.2$, $C(6) = 1.0$, and $D(1) = D(2) = D(6) = D(7) = 0$, $D(3) = D(4) = D(5) = 1$.

The role of $(L, \{0,1\})$-rectangular matrices is shown by the following theorem.

Theorem 1. *$I = A \circ B$ for $n \times k$ and $k \times m$ matrices A and B with $A_{il} \in L$ and $B_{lj} \in \{0,1\}$ iff I is a $\bigvee$-superposition of k $(L, \{0,1\})$-rectangular matrices $J_1, \ldots, J_k$, i.e. iff*

$$I = J_1 \vee J_2 \vee \cdots \vee J_k.$$

Proof. Denote by J_l the $\circ$-product $A_{\_l} \circ B_{l\_}$ of the l-th column $A_{\_l}$ of A and the l-th row $B_{l\_}$ of B, i.e. $(J_l)_{ij} = A_{il} \otimes B_{lj}$. $I = A \circ B$ means $I_{ij} = (A \circ B)_{ij}$, i.e. $I_{ij} = \bigvee_{l=1}^{k} (A_{il} \otimes B_{lj})$. Therefore, $I = J_1 \vee J_2 \vee \cdots \vee J_k$. Since B is a binary matrix, J_l are $(L, \{0,1\})$-rectangular matrices. □

Example 1. To illustrate the content of Theorem 1, consider the following decomposition $I = A \circ B$:

$$\begin{pmatrix} 0.3 & 1.0 & 0.0 & 0.0 & 0.0 & 0.3 \\ 1.0 & 1.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 1.0 & 0.9 & 1.0 & 1.0 & 0.0 & 0.8 \\ 1.0 & 0.2 & 0.0 & 0.0 & 1.0 & 0.2 \end{pmatrix} =$$

$$\begin{pmatrix} 0.3 & 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 1.0 & 0.7 \\ 0.8 & 1.0 & 0.0 & 0.9 \\ 0.2 & 0.0 & 1.0 & 0.0 \end{pmatrix} \circ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In this example $L = \{0, 0.1, \ldots, 0.9, 1\}$ and $a \otimes b = \min(a, b)$. According to Theorem 1, I is a $\bigvee$-superposition of four matrices, J_1, J_2, J_3, J_4 where J_l is a $\circ$-product of the l-th column of A and the l-th row of B, i.e.

$$\begin{pmatrix} 0.3 & 1.0 & 0.0 & 0.0 & 0.0 & 0.3 \\ 1.0 & 1.0 & 0.0 & 0.0 & 1.0 & 1.0 \\ 1.0 & 0.9 & 1.0 & 1.0 & 0.0 & 0.8 \\ 1.0 & 0.2 & 0.0 & 0.0 & 1.0 & 0.2 \end{pmatrix} =$$

$$\begin{pmatrix} 0.3 & 0.3 & 0.0 & 0.0 & 0.0 & 0.3 \\ 1.0 & 1.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.8 & 0.8 & 0.0 & 0.0 & 0.0 & 0.8 \\ 0.2 & 0.2 & 0.0 & 0.0 & 0.0 & 0.2 \end{pmatrix} \vee \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 1.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix} \vee$$

$$\begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \end{pmatrix} \vee \begin{pmatrix} 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.7 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.9 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}.$$

2.2 Formal concepts are optimal factors

Theorem 1 says that in order to find a decomposition $I = A \circ B$, we need to find a suitable set of $(L, \{0, 1\})$-rectangular matrices J_l whose $\bigvee$-superposition gives I. We now describe decompositions of I which are optimal among all possible decompositions in that the number k of factors is the smallest possible one. The decompositions use so-called crisply generated formal concepts of I [5].

Preliminaries on crisply generated formal concepts This section presents preliminaries on formal concepts of data with fuzzy attributes, particularly on crisply generated formal concepts. The reader is referred, e.g., to [3, 5] for details.

Let $X = \{1, \ldots, n\}$ and $Y = \{1, \ldots, m\}$ be sets (of objects and attributes, respectively), I be an $n \times m$ matrix with entries from a support set L of a complete residuated lattice $\mathbf{L}$. The degree $I_{xy} \in L$ is interpreted as a degree to which object x has attribute y. Consider the operators $^{\uparrow} : L^X \to L^Y$ and $^{\downarrow} : L^Y \to L^X$ defined by

$$C^{\uparrow}(y) = \bigwedge_{x \in X}(C(x) \to I_{xy}), \qquad D^{\downarrow}(x) = \bigwedge_{y \in Y}(D(y) \to I_{xy}),$$

where $\to$ is the residuum of the complete residuated lattice $\mathbf{L}$. That is, $^{\uparrow}$ assigns an L-set $C^{\uparrow}$ in Y to a given L-set C in X, and $^{\downarrow}$ assigns an L-set $D^{\downarrow}$ in X to a given L-set D in Y. $C^{\uparrow}(y)$ can verbally be described as a degree to which "for each object $x \in X$: if x is from C then x has attribute y" (note that $C^{\uparrow}(y)$ is just the degree of the last statement "for each $\cdots$" according to basic principles of first-order fuzzy logic, see [12]). Likewise, $D^{\downarrow}(x)$ is the degree to which "for each attribute $y \in Y$: if y is from D then x has attribute y" is true. If $L = \{0,1\}$, $^{\uparrow} : L^X \to L^Y$ and $^{\downarrow} : L^Y \to L^X$ coincide with the well-known concept-derivation operators of the basic setting of formal concept analysis [8, 11]. $^{\uparrow}$ and $^{\downarrow}$ form a fuzzy Galois connection [2] and the compound operators $^{\uparrow\downarrow}$ and $^{\downarrow\uparrow}$ form particular closure operators in X and Y [2]. A pair $\langle C, D\rangle$ consisting of an L-set C in X and an L-set D in Y is called a formal concept of I if $C^{\uparrow} = D$ and $D^{\downarrow} = C$. C and D are called the extent and intent of $\langle C, D\rangle$, respectively. The set of all formal concepts of I is denoted by $\mathcal{B}(X, Y, I)$. With a partial order $\leq$ defined by

$$\langle C_1, D_1\rangle \leq \langle C_2, D_2\rangle \text{ iff } C_1 \subseteq C_2 \text{ (iff } D_2 \subseteq D_1)$$

for $\langle C_1, D_1\rangle, \langle C_2, D_2\rangle \in \mathcal{B}(X, Y, I)$, $\mathcal{B}(X, Y, I)$ happens to be a complete lattice, so-called concept lattice associated to I [2,3]. Note that $C_1 \subseteq C_2$ means that C_1 is contained in C_2, i.e. for each $x \in X$, $C_1(x) \leq C_2(x)$. For $L = \{0,1\}$, $\mathcal{B}(X, Y, I)$ coincides with the ordinary concept lattice [11]. In [5], the following notion was introduced. A formal concept $\langle C, D\rangle \in \mathcal{B}(X, Y, I)$ is called *crisply generated* if there is a crisp L-set $D_c \in \{0,1\}^Y$, i.e. for each $y \in Y$: $D_c(y) = 0$ or $D_c(y) = 1$, such that $C = D_c^{\downarrow}$ (and thus $D = D_c^{\downarrow\uparrow}$). Let $\mathcal{B}_c(X, Y, I)$ denote the collection of all crisply generated formal concepts of I, i.e.

$$\mathcal{B}_c(X, Y, I) = \{\langle C, D\rangle \in \mathcal{B}(X, Y, I) \mid \text{ there is } D_c \in \{0,1\}^Y : C = D_c^{\downarrow}\}.$$

We need the following characterization of crisply generated formal concepts. For L-sets $C_1, C_2 \in L^X$ and $D_1, D_2 \in L^Y$, we put $\langle C_1, D_1\rangle \trianglelefteq \langle C_2, D_2\rangle$ if for each $x \in X$, $y \in Y$ we have $C_1(x) \leq C_2(x)$ and $D_1(y) \leq D_2(y)$.

Lemma 1 ([5]). *$\langle C, D\rangle$ is a crisply generated formal concept iff $\langle C, D\rangle$ is maximal (w.r.t. $\trianglelefteq$) such that (1) the rectangular matrix J defined by $J_{xy} = C(x) \otimes D(y)$ is contained in I (i.e. $J_{xy} \leq I_{xy}$ for all x, y) and (2) $C(x) = \bigwedge_{D(y)=1} I_{xy}$.*

Remark 1. Note that condition (2) of Lemma 1 means that for the crisp L-set $D_c \in \{0,1\}^Y$ corresponding to the 1-cut of D, which is defined by

$$D_c(y) = \begin{cases} 1 \text{ if } D(y) = 1, \\ 0 \text{ if } D(y) < 1, \end{cases} \tag{3}$$

we have $C = D_c^{\downarrow}$.

Matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ For convenience, we identify $1 \times p$ vectors with entries from L with L-sets in $\{1, \ldots, p\}$ (the l-th coordinate of the vector = the degree to which l belongs to the L-set). Given a set

$$\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\}$$

of L-sets C_l and D_l in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$, respectively, with values from L, define $n \times k$ and $k \times m$ matrices $A_{\mathcal{F}}$ and $B_{\mathcal{F}}$ by

$$(A_{\mathcal{F}})_{il} = (C_l)(i) \quad \text{and} \quad (B_{\mathcal{F}})_{lj} = (D_l)(j).$$

That is, the l-th column of $A_{\mathcal{F}}$ is the transpose of the vector corresponding to C_l and the l-th row of $B_{\mathcal{F}}$ is the vector corresponding to D_l.

For $\mathcal{F} \subseteq \mathcal{B}(X, Y, I)$, denote

$$\mathcal{F}_c = \{\langle C, D_c\rangle \mid \langle C, D\rangle \in \mathcal{F}\}.$$

Note that D_c is defined by (3). We will show that sets $\mathcal{F}_c$ corresponding to sets $\mathcal{F}$ of crisply generated formal concepts are fundamental for decompositions we are looking for.

The first theorem says that for every I, there is a decomposition $A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$ for some $\mathcal{F} \subseteq \mathcal{B}_c(X, Y, I)$.

Theorem 2 (universality). *For every I with entries from L there is $\mathcal{F} \subseteq \mathcal{B}_c(X, Y, I)$ such that $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$, i.e. I is a product of A with entries from L and B with entries from $\{0, 1\}$.*

Proof. Denote for $l \in \{1, \ldots, m\}$, $\langle C_l, D_l\rangle = \langle \{1/l\}^{\downarrow}, \{1/l\}^{\downarrow\uparrow}\rangle$. Here, $\{1/l\}$ is a singleton in $\{1, \ldots, m\}$, i.e. and L-set defined by $\{1/l\}(l) = 1$ and $\{1/l\}(j) = 0$ for $j \neq l$. $\langle C_l, D_l\rangle$ are particular crisply generated formal concepts from $\mathcal{B}(X, Y, I)$ and we have

$$I_{ij} = \bigvee_{l=1}^{m} C_l(i) \otimes D_l(j),$$

see [2]. Putting thus $\mathcal{F} = \{\langle C_l, D_l\rangle \mid l = 1, \ldots, m\}$, we get $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$. □

However, Theorem 2 and its proof yield only $|\mathcal{F}| = m$, i.e. the number $k = |\mathcal{F}|$ of factors equals the number m of attributes. In general, better decompositions may exist, i.e. those with $k < m$. The next theorem shows that the decompositions which use crisply generated formal concepts of I as factors are optimal among all decompositions of I.

Theorem 3 (optimality). *Let $I = A \circ B$ for $n \times k$ and $k \times m$ matrices A and B with $A_{il} \in L$, $B_{lj} \in \{0, 1\}$. Then there exists a set $\mathcal{F} \subseteq \mathcal{B}_c(X, Y, I)$ of crisply generated formal concepts of I such that for $\mathcal{F}_c$ we have*

$$|\mathcal{F}_c| \leq k$$

and for the $n \times |\mathcal{F}_c|$ and $|\mathcal{F}_c| \times m$ matrices $A_{\mathcal{F}_c}$ with entries from L and $B_{\mathcal{F}_c}$ with entries from $\{0, 1\}$ we have

$$I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}.$$

Proof. Sketch: Let $I = A \circ B$ for an $n \times k$ matrix A with entries from L and a $k \times m$ binary matrix B. Consider the corresponding rectangular matrices $J_1, \ldots, J_k$ of which I is a $\bigvee$-superposition according to Theorem 1. Denoting now the L-sets in $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$ corresponding to the l-th column of A and the l-th row of B by G_l and H_l, respectively, we have $J_l = G_l \otimes H_l$. We have $G_l \otimes H_l \subseteq I$ and one can check that also $H_l^{\downarrow} \otimes H_l \subseteq I$. The pair $\langle H_l^{\downarrow}, H_l\rangle$ satisfies condition (2) of Lemma 1 (see also Remark 1). Therefore, $\langle H_l^{\downarrow}, H_l\rangle$ is contained in a maximal (w.r.t. $\trianglelefteq$ defined in the paragraph preceding Lemma 1) $\langle C_l, D_l\rangle$ which is then, according to Lemma 1, a crisply generated formal concept of I. As a result, $C_l \otimes D_l \subseteq I$. Therefore, for $\mathcal{F} = \{\langle C_1, D_1\rangle, \ldots, \langle C_k, D_k\rangle\}$ we have $|\mathcal{F}| \leq k$. Because $(H_l)_j \in \{0, 1\}$ and because we may assume $H_l \subseteq D_l$, we get

$H_l \subseteq (D_l)_c$, cf. (3). So,

$$I = A \circ B = \bigvee_{l=1}^{k} G_l \otimes H_l \subseteq \bigvee_{l=1}^{k} H_l^{\downarrow} \otimes H_l$$

$$\subseteq \bigvee_{l=1}^{k} C_l \otimes (D_l)_c = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c} \subseteq \bigvee_{l=1}^{k} C_l \otimes D_l = A_{\mathcal{F}} \circ B_{\mathcal{F}} \subseteq I,$$

i.e. $A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c} = I$, finishing the proof. □

Note that using the notation from the proof of Theorem 3, two distinct $\langle G_l, H_l \rangle$'s may be contained in a single $\langle C_l, D_l \rangle$, i.e. for $\langle G_{l_1}, H_{l_1} \rangle \neq \langle G_{l_2}, H_{l_2} \rangle$ we can have $\langle C_{l_1}, D_{l_1} \rangle = \langle C_{l_2}, D_{l_2} \rangle$. As a consequence, we may have $|\mathcal{F}| < k$.

3 Algorithm

In this section, we present an approximation algorithm for computing a decomposition $I = A \circ B$ of an $n \times m$ matrix I with entries from L into an $n \times k$ matrix A with entries from L and a $k \times m$ binary matrix B with k as small as possible. Note that we do not provide the approximation factor for this algorithm.

Recall that for $L = \{0, 1\}$ (i.e. the set of grades contains just 0 and 1), our problem becomes a problem of decomposition of binary matrices. In particular, if $L = \{0, 1\}$, we are given a binary matrix I and our aim is to find a decomposition $I = A \circ B$ into an $n \times k$ binary matrix A and a $k \times m$ binary matrix B with k as small as possible. This problem is NP-hard and its decision version is NP-complete, see e.g. [17–19], and also [6].

Due to NP-hardness of a problem of decomposition of binary matrices which is a particular instance of our problem, we need to look for suitable approximation algorithms. In the following, we propose a greedy approximation algorithm inspired by the algorithms presented in [6] and [7]. Briefly, starting with empty $\mathcal{F}_c$, the algorithm selects a crisply generated concept $\langle C, D \rangle$ of I that covers a large part of I which is still uncovered. For each such selected $\langle C, D \rangle$, the corresponding $\langle C, D_c \rangle$, see (3), is added to $\mathcal{F}_c$. For determining $\langle C, D \rangle$, we use $|D \oplus j|$ which denotes the number of pairs $\langle i, j' \rangle$ of indices, for which $I_{ij'} = \left(I_{\mathcal{F}_c} \vee (D \cup \{\,^1/j\})^{\downarrow} \otimes (D \cup \{\,^1/j\})^{\downarrow\uparrow}\right)_{ij'}$. We refer to this approach as to Method 1. We also used Method 2 for which $|D \oplus j|$ takes into account also entries $\left(I_{\mathcal{F}_c} \vee (D \cup \{\,^1/j\})^{\downarrow} \otimes (D \cup \{\,^1/j\})^{\downarrow\uparrow}\right)_{ij'}$ which are close to $I_{ij'}$ but not necessarily equal (details will appear in a full version of this paper).

Note that if $L = \{0, 1\}$, our algorithm works the same way as the one from [6]. We performed several experiments with our algorithm. Due to limited scope, we present the following one. We generated 1,000 matrices I of dimension 15×15 over 5-element chain L with Łukasiewicz operations. Each matrix was generated as a product of a $15 \times k$ matrix A and a $k \times 15$ binary matrix B, so we knew the number of factors (its upper bound, in fact). Table 1 shows the numbers of factors (average value ± standard deviation) for decompositions of I obtained by our algorithm (both for Methods 1 and 2).

Algorithm 1 Find Factors

Input: I (matrix with entries from L)
Output: $\mathcal{F}_c$ (set $\mathcal{F}_c$ for which $I = A_{\mathcal{F}_c} \circ B_{\mathcal{F}_c}$)

```
set I_Fc to empty matrix ((I_Fc)_ij = 0)
while I ≠ I_Fc do
  set D to ∅
  set V to 0
  while there is j such that D(j) < 1 and |D ⊕ j| > V do
    select j such that D(j) < 1 which maximizes |D ⊕ j|
    set D to (D ∪ {1/j})^↓↑
    set V to |D ⊕ j|
  end while
  set C to D^↓
  add ⟨C, D_c⟩ to F_c
  set I_Fc to I_Fc ∨ C ⊗ D_c
end while
```

Table 1. Number of computed factors

k	no. computed factors Method 1	no. computed factors Method 2
4	5.294 ± 0.660	5.303 ± 0.712
5	7.204 ± 1.113	7.232 ± 1.063
6	8.964 ± 1.770	8.992 ± 1.688
7	10.194 ± 2.066	10.128 ± 1.990
8	11.155 ± 2.209	11.182 ± 2.067
9	11.747 ± 2.247	11.771 ± 1.878
10	12.18 ± 2.035	12.225 ± 2.054

4 Illustrative Example

In this section, we present an illustrative example regarding decompositions of a matrix with grades into a matrix with grades and a binary matrix.

In our example, we consider n users, m permissions, and a user-to-permission assignment. The assignment can be represented by an $n \times m$ matrix I with entries from a scale $L = \{0, r, w, 1\}$, with 0 representing "no permission", r and w representing "permission to read" and "permission to write", respectively, and 1 representing "full permission". We define a partial order on L such that 0 is the least element, 1 is the greatest one, and elements r and w are incomparable, see Fig. 1.

Furthermore, we need to define operations of multiplication $\otimes$. We put $x \otimes y = x \wedge y$, for all $x, y \in L$. The residuum is then determined by $\otimes$ (due to the requirement of adjointness, see Section 1) and is defined by $x \rightarrow y = 1$ for $x \leq y$, $x \rightarrow y = y$ for all $x > y$, and $r \rightarrow w = w$, $w \rightarrow r = r$.

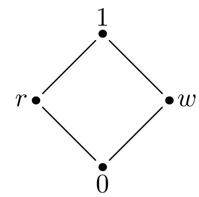
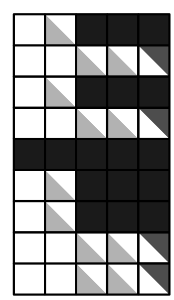


Fig. 1. Partial order on the scale of permissions

We want to decompose I into a product of $n \times k$ matrix A and $k \times m$ matrix B where A and B represent a user-to-role and a role-to-permission relationship, respectively. Therefore, the factors we want to discover are to be interpreted as roles, such as "system administrator", "standard user" or the like. Naturally, we expect A to be a binary matrix (i.e. $A_{il} \in \{0, 1\}$), assigning roles to users (a user has a given role or not), whereas B is graded matrix (i.e. $B_{lj} \in L$). In order to be consistent with previous chapters, A should be graded and B should be binary matrix. Therefore, we use well-known fact that $I = A \circ B$ is equivalent to $I^{-1} = B^{-1} \circ A^{-1}$. That is, instead of I we decompose I^{-1}.

As a particular example, we consider 9 users (or employees) and 5 file-types in some computer system (for instance, "documents", "archive files" or "system files" could be some of these types). The user-to-permission relationship is described in the table thereunder. The data can be visualized using a rectangular grid, where □, ◩, ◪, and ■ represent permissions 0, r, w, and 1, respectively:

	type_1	type_2	type_3	type_4	type_5
Alice	0	r	1	1	1
Bob	0	0	r	r	w
Charles	0	r	1	1	1
David	0	0	r	r	w
Eve	1	1	1	1	1
Frank	0	r	1	1	1
George	0	r	1	1	1
Henry	0	0	r	r	w
Isaac	0	0	r	r	w



Our aim is thus to decompose the corresponding graded matrix

$$I^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ r & 0 & r & 0 & 1 & r & r & 0 & 0 \\ 1 & r & 1 & r & 1 & 1 & 1 & r & r \\ 1 & r & 1 & r & 1 & 1 & 1 & r & r \\ 1 & w & 1 & w & 1 & 1 & 1 & w & w \end{pmatrix}.$$

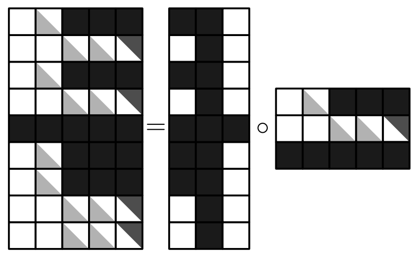
Algorithm 1 computes the following decomposition:

$$I^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ r & 0 & 1 \\ 1 & r & 1 \\ 1 & r & 1 \\ 1 & w & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

i.e.,

$$I = A \circ B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \circ \begin{pmatrix} 0 & r & 1 & 1 & 1 \\ 0 & 0 & r & r & w \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

This decomposition can be displayed as:



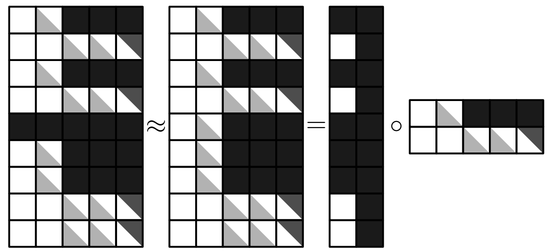
As we obtained a 9×3 binary matrix A describing a user-to-role assignment and 3×5 matrix B describing a role-to-permission assignment. Therefore, we obtained 3 factors: role_1, role_2, role_3. The first role (corresponding to the first row of matrix B) might be interpreted as "standard user", the second one (the middle row of B) as "anonymous user" ("guest"), and the third one (the last row of B) as "system administrator".

According to matrix A, we assign roles to users by:

Alice - role_1, role_2,
Bob - role_2,
Charles - role_1, role_2,
David - role_2,
Eve - all roles,
Frank - role_1, role_2,
George - role_1, role_2,
Henry - role_2,
Isaac - role_2.

Next, we compute an approximate decomposition of $I \approx A \circ B$. By this we mean that we want the entries of I to by similar to the corresponding entries of $A \circ B$ to a degree which exceeds a given similarity threshold f. In our example we set $f = 0.9$. Details regarding such similarity will be presented in a full version of this paper. Let us just note that the similarity is based on the number of matrix entries which have equal values in I and $A \circ B$. A graphical representation of an approximation decomposition computed by our algorithm depicted below.

We can see that the approximate decomposition involves the two factors corresponding to "standard user" and "anonymous user", which were involved



also in the exact decomposition. However, the factor corresponding to "system administrator" is no longer involved in the approximate decomposition. This can be seen as the result of our attempt, due to performing an approximate decomposition, to discover only a small number of factors (roles) which account for most of the data and, hence, are common. The role of "system administrator" is not common since the only user with this role is Eve.

5 Conclusions and Future Research

We presented a theorem regarding optimal decomposition of a matrix with grades into a matrix with grades and a binary matrix. Furthermore, we proposed a greedy approximation algorithm for computing such decompositions and examples illustrating such decompositions.

Further issues and future research include the following items:

- Independence of $\otimes$ and $\rightarrow$. It can be shown that the decompositions of a graded matrix into a graded and a binary matrix do not depend, in a certain sense, on the operations $\otimes$ and $\rightarrow$ on the scale L of grades. We sticked to the framework which involves $\otimes$ and $\rightarrow$ to show how the problem addressed in this paper fits into the results developed earlier. Details will be presented in the full version of this paper.
- Decompositions of matrices with grades into matrices with further constraints, different from the requirement of binarity of B.
- Approximation algorithms for approximate and exact decompositions of matrices with grades.
- Applications of the underlying factor analysis model and comparison to other models of factor analysis.
- Role of decompositions in machine learning and data mining (esp. dimensionality reduction).

References

1. Bartholomew, D. J., Knott M.: *Latent Variable Models and Factor Analysis, 2nd Ed.*, London, Arnold, 1999.
2. Belohlavek, R.: *Fuzzy Relational Systems: Foundations and Principles.* Kluwer, Academic/Plenum Publishers, New York, 2002.

3. Belohlavek, R.: Concept lattices and order in fuzzy logic. *Annals of Pure and Applied Logic* **128**(1–3)(2004), 277–298.
4. Belohlavek, R.: Optimal decompositions of matrices with grades. *IEEE Intelligent Systems 2008* (to appear).
5. Belohlavek, R., Sklenar, V., Zacpal, J.: Crisply generated fuzzy concepts. In: B. Ganter and R. Godin (Eds.): ICFCA 2005, Lecture Notes in Artificial Intelligence **3403**, pp. 268–283, Springer-Verlag, Berlin/Heidelberg, 2005.
6. Belohlavek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition (to appear). Preliminary version appeared as: On Boolean factor analysis with formal concepts as factors. SCIS & ISIS 2006, Int. Conf. Soft Computing and Intelligent Systems & Int. Symposium on Intelligent Systems, Sep 20-24, 2006, Tokyo, Japan, pp. 1054-1059.
7. Belohlavek, R., Vychodil, V.: Optimal decompositions of matrices with ordinal data (submitted).
8. Carpineto, C., Romano, G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.
9. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C.: *Introduction to Algorithms, 2nd Ed.* MIT Press, 2001.
10. Frolov, A. A., Húsek, D., Muraviev, I. P., Polyakov, P. A.: Boolean factor analysis by Hopfield-like autoassociative memory. *IEEE Transactions on Neural Networks* Vol. **18**, No. 3, May 2007, pp. 698–707.
11. Ganter, B., Wille, R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
12. Hájek, P.: *Metamathematics of Fuzzy Logic.* Kluwer, Dordrecht, 1998.
13. Harman, H. H.: *Modern Factor Analysis, 2nd Ed.* The Univ. Chicago Press, Chicago, 1970.
14. Keprt, A., Snášel, V.: Binary factor analysis with help of formal concepts. In *Proc. CLA 2004*, Ostrava, Czech Republic, 2004, pp. 90-101, ISBN 80–248–0597–9.
15. Klir, G. J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice-Hall, 1995.
16. McDonald, R. P.: *Factor Analysis and Related Methods.* Lawrence Erlbaum Associates, Inc., 1985.
17. Nau, D. S.: Specificity covering: immunological and other applications, computational complexity and other mathematical properties, and a computer program. A. M. Thesis, Technical Report CS–1976–7, Computer Sci.Dept., Duke Univ., Durham, N. C., 1976.
18. Nau, D. S., Markowsky, G., Woodbury M. A., Amos D. B.: A Mathematical Analysis of Human Leukocyte Antigen Serology. *Math. Biosciences* **40**(1978), 243–270.
19. Stockmeyer, L. J.: The set basis problem is NP-complete. IBM Research Report RC5431, Yorktown Heights, NY, 1975.
20. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (Ed.): Ordered Sets, 445–470, Reidel, Dordrecht-Boston, 1982.

Parallel Recursive Algorithm for FCA*

Petr Krajca, Jan Outrata and Vilem Vychodil

Data Analysis and Modelling Laboratory, SUNY Binghamton
Vestal Parkway E, Binghamton, NY 13902–6000, USA
petr.krajca@binghamton.edu, vychodil@binghamton.edu

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

Abstract. This paper presents a parallel algorithm for computing formal concepts. Presented is a sequential version upon which we build the parallel one. We describe the algorithm, its implementation, scalability, and provide an initial experimental evaluation of its efficiency. The algorithm is fast, memory efficient, and can be optimized so that all critical operations are reduced to low-level bit-array operations. One of the key features of the algorithm is that it avoids synchronization which has positive impacts on its speed and implementation.

1 Introduction

In this paper, we focus on extracting formal concepts, i.e. particular rectangular patterns, in binary object-attribute relational data. The input data, we are interested in, takes form of a two-dimensional data table with rows corresponding to objects, columns corresponding to attributes (features), and table entries being 1's and 0's indicating presence/absence of attributes. Tables like these represent a fundamental form of incidence data. Given a data table, we wish to find all formal concepts [9, 18] present in the table.

There are several algorithms for computing formal concepts, see [13] for an overview and comparison. Among the best known algorithms are Ganter's algorithm [8] and Lindig's algorithm [14] and their variants. Almost all algorithms proposed to date are sequential ones. Since parallel computing is recently gaining interests as hardware manufactures are shifting their focus from improving computing power by increasing clock frequencies to developing processors with multiple cores, there is a need to have scalable parallel algorithms for formal concept analysis (FCA) which can fully utilize the power of such milticore systems and deliver results faster than sequential algorithms. In this paper, we propose a parallel version of an algorithm presented in [16, 17] which is closely related to algorithm Close-by-One [12]. Our algorithm is light weight, fast, memory efficient, and can be implemented so that it uses just static linear data structures utilizing only low-level operations present in arithmetic logic units of contemporary

* Supported by grant No. 1ET101370417 of GA AV ČR and by institutional support, research plan MSM 6198959214.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 71–82,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

microchips which significantly improves the performance of its implementations. We describe the algorithm and compare its performance with the other algorithms. We also focus on scalability, i.e. the growth of algorithm's performance with respect to the growing number of processors.

Let us note that computing all formal concepts is interesting not only for FCA itself but has a wide range of applications. For instance, it has been shown in [3] that formal concepts can be used to find optimal factorization of Boolean matrices. In fact, formal concepts correspond with optimal solutions to the discrete basis problem discussed by Miettinen et al. [15]. Finding formal concepts in data tables is therefore an important task.

2 Preliminaries from FCA

In this section we recall basic notions of the formal concept analysis. More details can be found in monographs [9] and [5].

Let $X = \{0, 1, \ldots, m\}$ and $Y = \{0, 1, \ldots, n\}$ be our sets of objects and attributes, respectively. A formal context is a triplet $\langle X, Y, I\rangle$ where $I \subseteq X \times Y$, i.e. I is a binary relation between X and Y, $\langle x, y\rangle \in I$ meaning that object x has attribute y. As usual, we consider a couple of concept-forming operators [9] ${}^{\uparrow}\colon 2^X \to 2^Y$ and ${}^{\downarrow}\colon 2^Y \to 2^X$ defined, for each $A \subseteq X$ and $B \subseteq Y$, by

$$A^{\uparrow} = \{y \in Y \mid \text{for each } x \in A\colon \langle x, y\rangle \in I\}, \tag{1}$$

$$B^{\downarrow} = \{x \in X \mid \text{for each } y \in B\colon \langle x, y\rangle \in I\}. \tag{2}$$

By definition (1), $A^{\uparrow}$ is the set of all attributes shared by all objects from A and, by (2), $B^{\downarrow}$ is the set of all objects sharing all attributes from B. Operators ${}^{\uparrow}\colon 2^X \to 2^Y$ and ${}^{\downarrow}\colon 2^Y \to 2^X$ defined by (1) and (2) form the so-called Galois connection [9]. A formal concept (in $\langle X, Y, I\rangle$) is any couple $\langle A, B\rangle \in 2^X \times 2^Y$ such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. If $\langle A, B\rangle$ is a formal concept then A and B will be called the extent and intent of that concept, respectively. The subconcept-superconcept hierarchy $\leq$ is defined as $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (or, iff $B_2 \subseteq B_1$, both the ways are equivalent), see [5, 9] for details.

Remark 1. There is a useful view of formal concepts which is often neglected in literature. Namely, formal concepts in $\langle X, Y, I\rangle$ correspond to maximal rectangles in $\langle X, Y, I\rangle$. In a more detail, any $\langle A, B\rangle \in 2^X \times 2^Y$ such that $A \times B \subseteq I$ shall be called a rectangle in I. Rectangle $\langle A, B\rangle$ in I is a maximal one if, for each rectangle $\langle A', B'\rangle$ in I such that $A \times B \subseteq A' \times B'$, we have $A = A'$ and $B = B'$. Now, it is easily seen that $\langle A, B\rangle \in 2^X \times 2^Y$ is a maximal rectangle in I iff $A^{\uparrow} = B$ and $B^{\downarrow} = A$, i.e. maximal rectangles = formal concepts.

3 Computing Closures

Here we describe a procedure common to both the sequential and parallel versions of our algorithm. It generates a new concept from an existing one by enlarging its intent and shrinking its extent (at the same time).

Procedure ComputeClosure($\langle A, B\rangle, y$)

```
for i from 0 upto m do
│ set C[i] to 0;
end
for j from 0 upto n do
│ set D[j] to 1;
end
foreach i in A ∩ rows[y] do
│ set C[i] to 1;
│ for j from 0 upto n do
│ │ if table[i, j] = 0 then
│ │ │ set D[j] to 0;
│ │ end
│ end
end
return ⟨C, D⟩
```

Representation of the Input Data For the sake of efficiency, we represent each $\langle X, Y, I\rangle$ two ways. First, by a two-dimensional array, denoted *table*, which corresponds with I in the usual sense. That is, the array *table* is filled with 1s and 0s so that $table[i, j] = 1$ iff $\langle i, j\rangle \in I$ and $table[i, j] = 0$ iff $\langle i, j\rangle \notin I$.

The second representation of the data is an array of ordered lists of objects. For each attribute $y \in Y$, we let $rows[y]$ be a list of all objects having the attribute y. Thus, $rows[y]$ contains $x \in X$ iff $\langle x, y\rangle \in I$. In addition to that, the numbers of rows contained in $rows[y]$ will be ordered in the ascending order (this is for the sake of efficiency). For instance, $rows[y] = (2, 4, 7)$ means that the only objects from X having y in I are the objects 2, 4, and 7. The two-dimensional array *table* and the array of lists *rows* will be used by the subsequent algorithms.

All the algorithms we are going to describe will use sets of objects and attributes represented by their characteristic arrays. That is, in case of attributes, a subset $B \subseteq Y = \{0, 1, \ldots, n\}$ will be represented by an $(n + 1)$-element linear array b of 1s and 0s such $b[k] = 1$ iff $k \in B$ (and $b[k] = 0$ iff $k \notin B$). By a slight abuse of notation, we will identify B with b and write $B[k] = 1$ to denote $k \in B$.

Description of the Algorithm If $\langle A, B\rangle$ is a formal concept then due to the monotony of ${}^{\downarrow\uparrow}$, all the formal concepts whose intents are strictly greater than B can be written as $\big\langle (B \cup C)^{\downarrow}, (B \cup C)^{\downarrow\uparrow}\big\rangle$, where $C \subseteq Y$ is a set of attributes such that there is at least one attribute $y \in Y$ such that $y \in C$ and $y \notin B$. In particular, if we consider $C = \{y\} \subseteq Y$ such that $y \notin B$, then

$$\big\langle (B \cup \{y\})^{\downarrow}, (B \cup \{y\})^{\downarrow\uparrow}\big\rangle \tag{3}$$

is a formal concept such that $(B \cup \{y\})^{\downarrow} \subset A$ and $B \subset (B \cup \{y\})^{\downarrow\uparrow}$. This is important from the computational point of view because if we want to compute

$(B \cup \{y\})^{\downarrow}$, it suffices to go exactly through all objects in A having attribute y:

$$(B \cup \{y\})^{\downarrow} = \{x \in A \,|\, \langle x, y\rangle \in I\} = A \cap \{y\}^{\downarrow}. \tag{4}$$

The common attributes of objects from (4) form the intent of (3). We have just outlined the idea behind our algorithm which generates formal concept (3) given formal concept $\langle A, B\rangle$ and attribute $y \in Y$ which does not belong to B. The corresponding procedure will be called ComputeClosure. It accepts a formal concept $\langle A, B\rangle$ and an attribute $y \notin B$ and produces a new formal concept $\langle C, D\rangle$ which equals to (3). We can show that the algorithm is sound, see [16].

Remark 2. We have used two representations of the input data to establish desired efficiency of computing new formal concepts, i.e. the redundancy in representation is a trade-off for efficiency. The two-dimensional array representation is used to determine which attributes are not present in the intent of the newly computed formal concept (see lines 7–14 of ComputeClosure). The second representation is used to skip rows in which y does not appear. Such rows do not contribute to the closure $(B \cup \{y\})^{\downarrow\uparrow}$, i.e. they can be disregarded. Our representation is most efficient for mid-size data sets (hundreds of attributes + thousands of objects) stored in RAM.

4 Sequential Algorithm

The previous section described how we can efficiently compute a new formal concept (3) given an initial formal concept $\langle A, B\rangle$. In this section we present a simplified version of our sequential algorithm for computing formal concepts [16, 17] which is suitable for parallelization. The main idea behind this algorithm is the same as in case of the algorithm Close-by-One proposed by Kuznetsov in [12].

Listing Formal Concepts in a Unique Order The core of our algorithm is a recursive procedure GenerateFrom which lists all formal concepts using a depth-first search through the space of all formal concepts. The procedure starts with an initial formal concept $\langle \emptyset^{\downarrow}, \emptyset^{\downarrow\uparrow}\rangle$. During the search, the procedure first generates a new formal concept R by adding attributes to the intent of the current formal concept, i.e. it applies the procedure described in ComputeClosure. Then, it is checked whether R has already been found. If not, it processes R (e.g., prints it on the screen), and proceeds with generating further formal concepts resulting from R by adding attributes to its intent, i.e. here GenerateFrom recursively calls itself with R being the current formal concept.

The key issue here is to have a quick procedure testing whether a newly generated formal concept has been generated before. We generate the formal concepts in a unique order which ensures that each formal concept is processed exactly once. The principle is the following. Let $\langle A, B\rangle$ be a formal concept, $y \in Y$ such that $y \notin B$. Put $D = (B \cup \{y\})^{\downarrow\uparrow}$, i.e. the new formal concept is $\langle (B \cup \{y\})^{\downarrow}, D\rangle$, see (3). Once D is computed using ComputeClosure, we check whether

$$D \cap \{0, 1, \ldots, y-1\} = B \cap \{0, 1, \ldots, y-1\} \tag{5}$$

Procedure GenerateFrom($\langle A, B\rangle, y$)

```
1  process B (e.g., print B on screen);
2  if B = Y or y > n then
3  |  return
4  end
5  for j from y upto n do
6  |  if B[j] = 0 then
7  |  |  set ⟨C, D⟩ to ComputeClosure(⟨A, B⟩, j);
8  |  |  set skip to false;
9  |  |  for k from 0 upto j − 1 do
10 |  |  |  if D[k] ≠ B[k] then
11 |  |  |  |  set skip to true;
12 |  |  |  |  break for loop;
13 |  |  |  end
14 |  |  end
15 |  |  if skip = false then
16 |  |  |  GenerateFrom(⟨C, D⟩, j + 1);
17 |  |  end
18 |  end
19 end
20 return
```

is true. Note that the "$\supseteq$"-part of (5) is trivial. Moreover, (5) is true iff D agrees with B on the attributes $0, 1, \ldots, y-1$. In other words, (5) is true iff, for each $i \in \{0, 1, \ldots, y-1\}$: $i \in D$ iff $i \in B$. Thus, condition (5) expresses the fact that the closure D of $B \cup \{y\}$ does not contain any new attributes which are "before y". Condition (5) will be used to check whether we should process D. If (5) will be false, we will not process D because due to the depth-first search method, D has already been processed.

Description of the Algorithm The algorithm is represented by a procedure GenerateFrom that accepts two arguments. First, a formal concept $\langle A, B\rangle$ represented by characteristic vectors of objects A and attributes B covered by the concept. Second, an attribute y which is the first attribute to be added to B. $\langle A, B\rangle$ serves as an initial concept from which we start generating other formal concepts. After its invocation, GenerateFrom proceeds as follows:

– It processes the formal concept $\langle A, B\rangle$ (e.g., it prints A and B on screen).
– Then, the procedure checks whether B contains all the attributes from Y, i.e. whether B represents the greatest intent, in which case we exit current branch of recursion (lines 2–4).
– The main loop (lines 5–20) iterates over all remaining attributes, starting with the attribute y. In the body of the main loop (lines 6–18), j denotes the current attribute which we are about to add to B. The if-condition at line 6 checks whether j is already present in B. If so, we proceed with another attribute. If j is not present in B, we try to generate new intent from $B \cup \{j\}$ (lines 7–17).

– At line 7, we compute a new formal concept denoted $\langle C, D \rangle$. The loop between lines 9–14 checks whether B and D satisfy condition (5) for y being j. A flag *skip* is initially set to false (line 8). The flag is reset to true iff there is $k < j$ such that B and D disagree on k.
– If *skip* is false, i.e. if D and B agree on all attributes up to $j - 1$, we make a recursive call of the procedure GenerateFrom to compute descendant intents of D, starting with the next attribute $j + 1$ (line 16).

In order to compute all the formal concepts, we invoke GenerateFrom with $\langle \emptyset^{\downarrow}, \emptyset^{\downarrow\uparrow} \rangle$ and $y = 0$ as its arguments. Then, after finitely many steps, the algorithm produces all formal concepts, each of them exactly once. The soundness of the algorithm is proved in [16], cf. also [12].

Relationship to Other Sequential Algorithms Conceptually, GenerateFrom is the same algorithm as Close-by-One proposed by Kuznetsov [12] although there are some technical differences. GenerateFrom can be seen as simpler version of Close-by-One since we are not interested in the order of generated concepts. On the other hand, we utilize ComputeClosure which results to a much better performance. The algorithm is similar to Lindig's algorithm [13, 14] in that it performs a depth-first search through the search space of all formal concepts. The key difference between our algorithm and that proposed by Lindig [14] and its variants is the way how we test that new formal concept has already been found. Lindig's algorithm and its variants use additional data structures to store intents of found formal concepts. Thus, after a new formal concept is computed, Lindig's algorithm looks up for the concept in a data structure, typically a search tree or a hashing table. Our algorithm uses similar idea as Ganter's algorithm [8] to ensure that no concept is generated multiple times, see (5). Compared to Ganter's algorithm, the number of concepts which are computed multiple times and "dropped" is much lower, see [16].

5 Parallel Algorithm

The sequential version of our algorithm, described in previous section, lists all formal concepts using a depth-first search through the space of all formal concepts. Consider a calling tree of the recursive procedure GenerateFrom. The parallel version consists in modification of GenerateFrom so that subtrees of the calling tree are executed simultaneously by independent processes. The problem to solve is, given a process, which subtree(s) will be executed in the process, or, put in other words, how to distribute computed formal concepts among the processes.

Computing Formal Concepts in More Processes In the following we describe our approach for computing formal concepts in a given fixed number P of separate processes running in parallel. In the approach, processes are executing subtrees (of the calling tree of GenerateFrom) containing, in the root node, a call of GenerateFrom for a formal concept generated by a predefined number of attributes. The number of attributes, denoted by L, is a second parameter of the parallel algorithm. The parameter has an impact on the distribution of computed formal concepts among the processes, see Remark 3 on page 9.

The algorithm, consisting in modification of GenerateFrom, first simulates original sequential GenerateFrom until it reaches the recursion level at which formal concepts generated by $0 < L \leq n$ attributes are to be processed. The initial recursion halts at level which equals L, counting recursion levels from 0 upwards. The formal concepts generated by L attributes, i.e. formal concepts $\langle C, D\rangle = \langle\{y_0, \ldots, y_{L-1}\}^{\downarrow}, \{y_0, \ldots, y_{L-1}\}^{\downarrow\uparrow}\rangle$ such that $y_i \in Y$, are stored in a queue instead of being processed. For each of the P processes there is exactly one queue and the selection of the queue to which we store $\langle C, D\rangle$ is the key point of the algorithm. In fact, by selecting a queue we select a process which will list all formal concepts descendant to $\langle C, D\rangle$. The optimal selection method should distribute all formal concepts to processes equally. This is, however, very hard to achieve since we do not know the distribution of formal concepts in the search space of all formal concepts until we actually compute them all. In the present version of the algorithm we select process r, where r is the total number of stored formal concepts so far modulo the number P of processes.

After filling up the queues, the modified procedure then forks itself into P processes (or, alternatively, runs the following in $P - 1$ new processes too), and in each process the original sequential GenerateFrom is called for each formal concept in the queue of the respective process. This will list all the remaining descendant formal concepts, in parallel.

Description of the Algorithm The algorithm is represented by a procedure ParallelGenerateFrom, the modification of GenerateFrom which accepts one additional argument: the recursion level counter l, which is used to recognize the recursion level L at which formal concepts generated by L attributes are to be stored in a queue rather than processed. After its invocation, ParallelGenerateFrom proceeds as follows:

- Until it reaches the recursion level $L > 0$, the procedure simulates original GenerateFrom (lines 6–24). The code is identical, with two exceptions: first, instead of exiting at line 8 it skips to the point where original GenerateFrom ends and, second, upon each recursive call of itself it increases the recursion level counter l (line 21). In this step it (sequentially) processes all formal concepts generated by up to $L - 1$ attributes.
- When recursion level counter l is equal to L, i.e. the procedure is about to process formal concept $\langle A, B\rangle$ generated by L attributes, it (instead of processing $\langle A, B\rangle$) stores $\langle A, B\rangle$ and y (the attribute to be added to B) to queue *queue*$[r]$ of selected process r and exits current branch of recursion (lines 2–4). In this step, all formal concepts generated by L attributes are stored in the queues.
- Notice that when ParallelGenerateFrom exits a branch of recursion at line 4, the execution continues at line 22 because line 21 is the only place where ParallelGenerateFrom is recursively called. Therefore, it continues at line

Procedure ParallelGenerateFrom($\langle A, B\rangle, y, l$)

```
1  if l = L then
2  |  select r from 0 to P − 1 (e.g. r = (∑_{s=0}^{P−1} queue[s])  mod P);
3  |  store (⟨A, B⟩, y) to queue[r];
4  |  return
5  end
6  process B (e.g., print B on screen);
7  if B = Y or y > n then
8  |  goto line 25;
9  end
10 for j from y upto n do
11 |  if B[j] = 0 then
12 |  |  set ⟨C, D⟩ to ComputeClosure(⟨A, B⟩, j);
13 |  |  set skip to false;
14 |  |  for k from 0 upto j − 1 do
15 |  |  |  if D[k] ≠ B[k] then
16 |  |  |  |  set skip to true;
17 |  |  |  |  break for loop;
18 |  |  |  end
19 |  |  end
20 |  |  if skip = false then
21 |  |  |  ParallelGenerateFrom(⟨C, D⟩, j + 1, l + 1);
22 |  |  end
23 |  end
24 end
25 if l = 0 then
26 |  for r from 1 upto P − 1 do
27 |  |  new process
28 |  |  |  while set (⟨C, D⟩, j) to load from queue[r] do
29 |  |  |  |  GenerateFrom(⟨C, D⟩, j);
30 |  |  |  end
31 |  |  end
32 |  end
33 |  while set (⟨C, D⟩, j) to load from queue[0] do
34 |  |  GenerateFrom(⟨C, D⟩, j);
35 |  end
36 end
37 return
```

25 after exiting the loop between line 10–24. Here, it either exits the current branch of recursion (if $l \neq 0$) or continues if the top recursion level ($l = 0$) has been reached (i.e., no more branches of recursion are on the call stack).

– On the top recursion level ($l = 0$), it runs new $P - 1$ processes running in parallel (lines 26, 27) and the last step is performed by the new processes too.
– Finally, still on the top recursion level only, in each process, it calls original GenerateFrom for each formal concept $\langle C, D\rangle$ and attribute j in the queue

of the respective process (lines 28–30 and 33–35). That means, all formal concepts generated by L or more attributes are processed in separate processes running in parallel.

In order to compute all the formal concepts, we invoke ParallelGenerateFrom with $\langle \emptyset^{\downarrow}, \emptyset^{\downarrow\uparrow} \rangle$, $y = 0$ and $l = 0$ as its arguments. Then, after finitely many steps, the algorithm produces all formal concepts, each of them exactly once. The soundness of the algorithm follows directly from the soundness of the sequential version [12, 16] and the fact that processes compute predefined disjoint sub-collections of all formal concepts. This also means that the processes do not interfere with each other and hence the algorithm needs no synchronization. We postpone the proof to the full version of the paper. The parallelization also does not increase the overall theoretical complexity of the algorithm which is the same as for the sequential version.

Remark 3. Note that the parameter L, in addition to the process selection method, also determines the number of formal concepts computed by each process. If $L = 1$, most of the formal concepts (formal concepts descendant to a formal concept generated by a single attribute) are computed by one or two processes. With increasing L, formal concepts are distributed to processes more equally. On the other hand, however, with increasing L more formal concepts are computed sequentially and less in parallel. From our experimentation it seems a good trade-off value is already $L = 2$, where almost all formal concepts (for $n \gg L$) are computed in parallel and are distributed to processes nearly optimally. This will be further discussed in Section 6.

Remark 4. There have been several approaches to parallel algorithms in FCA. For instance, [7] proposes a parallelization of Ganter's algorithm by decomposing the set of all concepts into non-overlapping subsets which are computed simultaneously. Another parallelization of Ganter's algorithm is presented in [2]. The basic idea in [2] is that the lexicographically ordered power set 2^Y is split into p intervals of the same length (p indicates a number of processes). Then, each of the p intervals is executed by an independent process using a serial version of Ganter's algorithm. A different approach is shown, e.g., in [11] where the algorithm is based on dividing the input data into disjoin fragments which are then computed by independent processes. A detailed comparison of the algorithms in terms of their efficiency and scalability is beyond the scope of this paper and will be a subject of future investigation.

6 Experimental Evaluation

We have run several experiments to compare the algorithm with other algorithms for computing formal concepts. In the experiments, we have used Ganter's [8], Lindig's [14] and Berry's [4] algorithms and were interested in the performance of the algorithms measured by the running time. Furthermore, we have run several experiments to compare algorithm performances in dependence on number of

dataset	mushroom	tic-tac-toe	Debian tags	anonymous web
size	8124 × 119	958 × 29	14315 × 475	32710 × 295
density	19 %	34 %	< 1 %	1 %
our (1 CPU)	6.543	0.092	12.746	65.221
our (2 CPUs)	3.541	0.047	7.710	33.364
our (4 CPUs)	2.343	0.035	4.545	18.520
our (8 CPUs)	1.393	0.029	3.043	11.466
Ganter's	834.409	2.158	1720.827	10039.733
Lindig's	5271.988	14.530	2639.670	13422.643
Berry's	934.507	5.783	1531.944	3615.078

Fig. 1. Performance for selected datasets (seconds)

used CPUs. For the sake of comparison, we have implemented all the algorithms in ANSI C. The experiments were done on otherwise idle 64-bit x86_64 hardware with 8 independent processors (dual processor workstation with Quad-core Intel Xeon Processor E5345, 2.33 GHz, 12 GB RAM).

Note that even the serial version of our algorithm significantly outperforms the most commonly used algorithms for FCA. A detailed comparison can be found in [16]. In this section, we focus primarily on the scalability of our algorithm, i.e., we focus on the speed improvement with growing number of hardware processors.

Our first experiment compares our algorithm with various FCA algorithms using several data tables from the UCI Machine Learning Repository [1], UCI Knowledge Discovery in Databases Archive [10], and our dataset describing packages in the Debian GNU/Linux [6]. The results, along with the information on size and density (percentage of 1s) of used data sets, are depicted in Figure 1. First four rows contain computation times measured in seconds in case of our algorithm which has been run on 1 (sequential version), 2, 4, and 8 hardware processors. From all the graphs and tables we can see that our algorithm (significantly) outperforms all the other algorithms.

We now focus on the scalability of the algorithm, i.e., ability to decrease running time using multiple CPUs (or more precisely CPU cores). We have used selected data sets and various randomly generated data tables. Fig. 2 (left) contains results for selected datasets while Fig. 2 (right) contains results for randomly generated tables with 10000 objects and 5 % density of 1's. By a *relative speedup* which is shown on y-axes in the graphs, we mean the theoretical speedup given by number of hardware processors (e.g., if we have 4 processors, the execution can be 4 times faster). Therefore, the relative speedup is a ratio of running time using a single CPU (the sequential version of the algorithm) and running time using multiple CPU cores. Note that the theoretical maximum of speedup is equal to the number of used CPUs but real speedup is always smaller due to certain overhead caused by managing of multiple threads of computation. Nevertheless, from the point of view of the speedup, we can see from the experiments

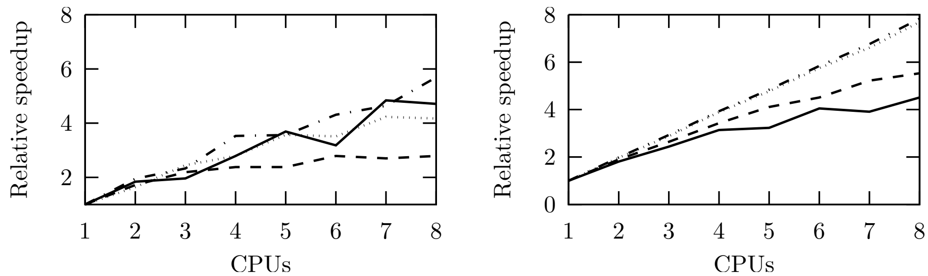


Fig. 2. Relative speedup dependent on various data tables (solid line—mushrooms, dashed line—tic-tac-toe, dotted line—Debian tags, dot-and-dashed line—annonymous web) and used CPU cores (on the left); relative speedup dependent on number of attributes (solid line—50 attributes, dashed line—100 attributes, dotted line—150 attributes, dot-and-dashed line—200 attributes) and used CPU cores measured using randomly generated contexts with 10000 objects and 5 % density (on the right).

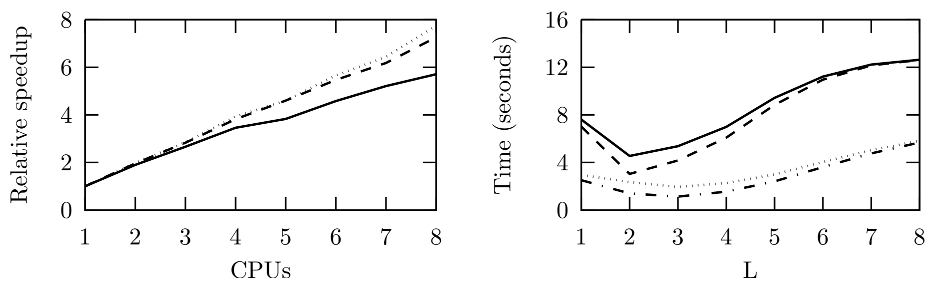


Fig. 3. Relative speedup dependent on density of 1's (solid line—5 %, dashed line—10 %, dotted line—20 %) and used CPU cores (on the left); running time dependent on the argument L (the solid line is for the Debian tags data table and 4 CPUs used, the dashed line is for the Debian tags data table and 8 CPUs used, the dotted lines is for the mushrooms data table and 4 CPUs used and dot-and-dashed lines is for the mushrooms data table and 8 CPUs used) (on the right).

that with growing number of attributes, the real speedup of the algorithm is near its theoretical limits.

In next experiment, that is depicted in Fig. 3 (left), we were focusing on the impact of density of 1's. That is, we have generated data tables with various densities and observed the impact on the scalability. We have used data tables of size 100×10000. Finally, Fig. 3 (right) illustrates the influence of parameter L on various data tables and amounts of CPU cores. The experiments indicate that good choice is $L \in \{2, 3\}$, see Remark 3.

7 Conclusions

We have introduced a parallel algorithm for computing formal concepts in object-attribute data tables. The parallel algorithm is an extension of the serial algorithm we have proposed in [16]. The algorithm consists of a procedure for computing closures and a recursive procedure for computing formal concepts. The main feature of the recursive procedure is that it simulates the sequential one up to a point where the procedure forks into multiple processes and each process computes a disjoint set of formal concepts. Due to our design of the algorithm, there is no need for synchronization which significantly improves efficiency of the algorithm. We have shown that the algorithm is scalable. With growing numbers of CPUs, the speedup of the computation given by increasing number of CPUs is near its theoretical limit. The future research will focus on further refinements of the algorithm and comparison with other approaches.

References

1. Asuncion A., Newman D. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2007.
2. Baklouti F., Levy G.: A distributed version of the Ganter algorithm for general Galois Lattices. In: Belohlavek R., Snasel V. (Eds.): *Proc. CLA 2005*, pp. 207–221.
3. Belohlavek R., Vychodil V. On boolean factor analysis with formal concept as factors. *Proceedings of SCIS & ISIS 2006,* pp. 1054–1059, 2006. Tokyo, Japan: Tokyo Institute of Technology.
4. Berry A., Bordat J.-P., Sigayret A. A local approach to concept generation. *Annals of Mathematics and Artificial Intelligence*, **49**(2007), 117–136.
5. Carpineto C., Romano G. *Concept data analysis. Theory and applications.* J. Wiley, 2004.
6. *DAMOL Dataset Repository* (in preparation).
7. Fu H., Mephu Nguifo E.: A Parallel Algorithm to Generate Formal Concepts for Large Data. *ICFCA 2004, LNCS* **2961**, pp. 394–401.
8. Ganter B. *Two basic algorithms in concept analysis.* (Technical Report FB4-Preprint No. 831). TH Darmstadt, 1984.
9. Ganter B., Wille R. *Formal concept analysis. Mathematical foundations.* Berlin: Springer, 1999.
10. Hettich S., Bay S. D.: The UCI KDD Archive University of California, Irvine, School of Information and Computer Sciences, 1999.
11. Kengue J. F. D., Valtchev P., Djamégni C. T.: A Parallel Algorithm for Lattice Construction. *ICFCA 2005, LNCS* **3403**, pp. 249–264.
12. Kuznetsov S.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. *PKDD 1999*, pp. 384–391.
13. Kuznetsov S., Obiedkov S. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Int.*, **14**(2002), 189–216.
14. Lindig C. Fast concept analysis. *Working with Conceptual Structures–
–Contributions to ICCS 2000,* pp. 152–161, 2000. Aachen: Shaker Verlag.
15. Miettinen P., Mielikäinen T., Gionis A., Das G., Mannila H. The discrete basis problem. *PKDD*, pp. 335–346, 2006. Springer.
16. Outrata J., Vychodil V. Fast algorithm for computing maximal rectangles from object-attribute relational data (submitted).
17. Vychodil V.: A new algorithm for computing formal concepts. In: Trappl R. (Ed.): *Cybernetics and Systems 2008: Proc. 19th EMCSR*, 2008, pp. 15–21.
18. Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts. *Ordered Sets*, pp. 445–470, 1982. Dordrecht-Boston.

Proto-fuzzy Concepts, their Retrieval and Usage

Ondrej Krídlo and Stanislav Krajči

University of Pavol Jozef Šafárik, Košice, Slovakia

Abstract. The aim of this paper is to define so-called proto-fuzzy concepts, as a base for generating different types of one-sided fuzzy concept lattices. Fuzzy formal context is a triple of a set of objects, a set of attributes and a fuzzy binary relation over a complete residuated lattice, which determines the degree of membership of each attribute to each object. A proto-fuzzy concept is a triple of a subset of objects, a subset of attributes and a value as the best common degree of membership of all pairs of objects and attributes from the above-mentioned sets to the fuzzy binary relation. Then the proto-fuzzy concepts will be found with a help of cuts and projections to the object-values or attribute-values plains of our fuzzy-context.

1 Introduction and motivation

Let us have a group of schoolmates of a secondary grammar school and their studying results of ten subjects as it is shown in the table below. Names of subjects are in the table as abbreviations (Ma – Math, Sl – Slovak language, Ph – Physics, Ge – Geography, Bi – Biology, Gr – German, En – English, Ch – Chemistry, Ae – Aesthetics, Hi – History). Abbreviations of names of students are in the table.

Table 1. Example of fuzzy formal context.

		Ma	Sl	Ph	Ge	Bi	Gr	En	Ch	Ae	Hi
F	Fred	1	1	1	3	2	1	2	2	1	2
J	Joey	3	1	2	1	1	1	1	3	1	1
A	Alice	3	2	3	1	1	1	1	3	2	2
N	Nancy	4	2	4	3	2	2	1	2	3	2
M	Mary	1	1	1	1	1	1	1	1	1	1
E	Eve	1	1	1	1	1	1	1	1	1	1
L	Lucy	1	3	1	2	2	2	2	1	2	2
D	David	2	3	4	3	4	1	1	2	2	2
P	Peter	2	1	2	1	1	2	2	3	1	2
T	Tom	1	3	2	2	2	2	2	3	1	2

The table is a concrete example of fuzzy formal context. Students represent objects, subjects represent attributes and corresponding valuations represent

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 83–95,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

values assigned to every object–attribute pair by fuzzy binary relation over the set $\{1, 2, 3, 4, 5\}$ (1 – best, ..., 5 – worst). Goal is to find groups of students similar by their studying results of all subjects, or to find subsets of subjects similar by results of all students. In other words to find pairs of classical subset of objects or attributes and fuzzy subset of attributes or objects. Similarity is determined by fuzzy subsets. Those pairs are called one-sided fuzzy concepts ([1]).

The starting point of this paper is to define so-called *proto-fuzzy concepts*, triples made of a subset of objects, a subset of attributes and a value from the set of degrees of membership forming fuzzy binary relation, which is not exceeding for any object-attribute pair of cartesian product of object and attribute subsets meant above. Every element of the triple is "maximal" opposite to other two elements. Proto-fuzzy concepts can be taken as a "base structure unit" of one-sided fuzzy concepts. If values in the table are taken as columns tall as degree of membership of subsistent object–attribute pair to fuzzy binary relation, then proto-fuzzy concepts could be taken as a maximal "sub-blocks" of satisfying triples object-attribute-value of the 3D block representing fuzzy context. Examples of some proto-fuzzy concepts of the example will be shown in section 3.

2 Basic definitions

Definition 1. A formal context *is a triple* $\langle \mathcal{O}, \mathcal{A}, \mathcal{R} \rangle$ *consists of two sets* $\mathcal{O}$*, the set of* objects*, and* $\mathcal{A}$*, the set of* attributes*, and a relation* $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{A}$*.*

Definition 2. A fuzzy formal context *is a triple* $\langle \mathcal{O}, \mathcal{A}, r \rangle$ *consists of two sets* $\mathcal{O}$*, the set of* objects*, and* $\mathcal{A}$*, the set of* attributes*, and* r *is fuzzy subset of* $\mathcal{O} \times \mathcal{A}$*, mapping from* $\mathcal{O} \times \mathcal{A}$ *to* L*, where* L *is a lattice.*

In the sense of simplicity of an idea "fuzzy" will be used instead of L-fuzzy.

Definition 3. *For every* $l \in L$ *define mappings* $\uparrow_l$: $\mathcal{P}(\mathcal{O}) \to \mathcal{P}(\mathcal{A})$ *and* $\downarrow_l$: $\mathcal{P}(\mathcal{A}) \to \mathcal{P}(\mathcal{O})$*: For every subset* $O \subseteq \mathcal{O}$ *put*

$$\uparrow_l (O) = \{a \in \mathcal{A} : (\forall o \in O) r(o, a) \geq l\}$$

and for all $A \subseteq \mathcal{A}$ *put*

$$\downarrow_l (A) = \{o \in \mathcal{O} : (\forall a \in A) r(o, a) \geq l\}.$$

Lemma 1. *For all* $l \in L$ *the pair* $(\uparrow_l, \downarrow_l)$ *forms a Galois connection between the power-set lattices* $\mathcal{P}(\mathcal{O})$ *and* $\mathcal{P}(\mathcal{A})$*.*

Definition 4. *Let* $\langle \mathcal{O}, \mathcal{A}, r \rangle$ *be a fuzzy context. A pair* $\langle O, A \rangle$ *is called an* l-concept *iff* $\uparrow_l (O) = A$*, and* $\downarrow_l (A) = O$*, hence the pair is a concept in a classical context* $\langle \mathcal{O}, \mathcal{A}, \mathcal{R}_l \rangle$*, that*

$$\mathcal{R}_l = \{(o, a) \in \mathcal{O} \times \mathcal{A} : r(o, a) \geq l\}.$$

Context $\langle \mathcal{O}, \mathcal{A}, \mathcal{R}_l \rangle$ *is called an* l-cut*. The set of all concepts in an* l*-cut will be denoted* $\mathcal{K}_l$*.*

Table 2. 2-cut.

2	Ma	Sl	Ph	Ge	Bi	Gr	En	Ch	Ae	Hi
F	•	•	•		•	•	•	•	•	•
J		•	•	•	•	•	•		•	•
A		•		•	•	•	•		•	•
N		•			•	•	•	•		•
E	•	•	•	•	•	•	•	•	•	•
M	•	•	•	•	•	•	•	•	•	•
L	•		•	•	•	•	•	•	•	•
D	•					•	•	•	•	•
P	•	•	•	•	•	•	•		•	•
T	•		•	•	•	•	•		•	•

In our example the l-cut means a look at the level of success for the value l. So the l-cut gives an Yes/No answer for the question: *Is the result of each student in each subject at least of the value l?* For example a concept $\langle O, A\rangle$ from $\mathcal{K}_2$ represents the group O of students, that every subject of the set A is fulfilled at least in the value 2.

By exploring all l-cuts for such $l \in L$, it can be seen that some l-concepts are equal for different $l \in L$. But information that Eve and Mary are successful in all subjects for the value 2 is not complete and not as useful as information that they are successful for 1. This information is not complete, "closed".

Two interesting properties will be shown in following lemmas and theorems. It will be a continuation of the knowledge of the paper [3], where some properties of cuts was shown.

Lemma 2. *Let $l_1, l_2 \in L$ that $l_1 \leq l_2$. $\uparrow_{l_1}(O) \supseteq \uparrow_{l_2}(O)$ for every $O \subseteq \mathcal{O}$ and $\downarrow_{l_1}(A) \supseteq \downarrow_{l_2}(A)$ for every $A \subseteq \mathcal{A}$.*

Proof. The proof will be shown for $\uparrow$. The proof for $\downarrow$ is likewise.

If $l_1 \leq l_2$ then

$$\{a \in \mathcal{A} : (\forall o \in O) r(o, a) \geq l_1\} \supseteq \{a \in \mathcal{A} : (\forall o \in O) r(o, a) \geq l_2\}.$$

Hence $\uparrow_{l_1}(O) \supseteq \uparrow_{l_2}(O)$. □

Lemma 3. *Let $O \subseteq \mathcal{O}$, $A \subseteq \mathcal{A}$ and $l_1, l_2 \in L$. Then $\uparrow_{l_1}(O) \cap \uparrow_{l_2}(O) = \uparrow_{l_1 \vee l_2}(O)$ and $\downarrow_{l_1}(A) \cap \downarrow_{l_2}(A) = \downarrow_{l_1 \vee l_2}(A)$.*

Proof. If $a \in \uparrow_{l_1}(O) \cap \uparrow_{l_2}(O)$ then for all $o \in O$ is $r(o, a) \geq l_1$ and $r(o, a) \geq l_1$. It follows from above that for every $o \in O$ is $r(o, a) \geq l_1 \vee l_2$ and so $a \in \uparrow_{l_1 \vee l_2}(O)$. Hence $\uparrow_{l_1}(O) \cap \uparrow_{l_2}(O) \subseteq \uparrow_{l_1 \vee l_2}(O)$. The lemma 6 implies that $\uparrow_{l_1 \vee l_2}(O) \subseteq \uparrow_{l_1}(O)$ and $\uparrow_{l_1 \vee l_2}(O) \subseteq \uparrow_{l_2}(O)$. It implies that $\uparrow_{l_1 \vee l_2}(O) \subseteq \uparrow_{l_1}(O) \cap \uparrow_{l_2}(O)$. From the both inclusions implies that $\uparrow_{l_1 \vee l_2}(O) = \uparrow_{l_1}(O) \cap \uparrow_{l_2}(O)$. The proof for $\downarrow$ is likewise. □

Theorem 1. *Let $l_1, l_2 \in L$ and $\langle O, A\rangle \in \mathcal{K}_{l_1} \cap \mathcal{K}_{l_2}$. Then for all $l \in L$, if $l_1 \leq l \leq l_2$ then $\langle O, A\rangle \in \mathcal{K}_l$.*

Proof. The lemma 6 and $\langle O, A\rangle \in \mathcal{K}_{l_1} \cap \mathcal{K}_{l_2}$ implies that
$A = \uparrow_{l_1}(O) \supseteq \uparrow_l(O) \supseteq \uparrow_{l_2}(O) = A$,
$O = \downarrow_{l_1}(A) \supseteq \downarrow_l(A) \supseteq \downarrow_{l_2}(A) = O$
Hence $\uparrow_l(O) = A$ and $\downarrow_l(A) = O$, which implies $\langle O, A\rangle \in \mathcal{K}_l$. □

Theorem 2. *Let $l_1, l_2 \in L$ and $\langle O, A\rangle \in \mathcal{K}_{l_1} \cap \mathcal{K}_{l_2}$. Then $\langle O, A\rangle \in \mathcal{K}_{l_1 \vee l_2}$.*

Proof. The lemma 7 implies
$\uparrow_{l_1 \vee l_2}(O) = \uparrow_{l_1}(O) \cap \uparrow_{l_2}(O) = A \cap A = A$,,
$\downarrow_{l_1 \vee l_2}(A) = \downarrow_{l_1}(A) \cap \downarrow_{l_2}(A) = O \cap O = O$.
Hence $\langle O, A\rangle \in \mathcal{K}_{l_1 \vee l_2}$. □

3 Proto-fuzzy concepts and their usage

Definition 5. *Triples $\langle O, A, l\rangle \in \mathcal{P}(\mathcal{O}) \times \mathcal{P}(\mathcal{A}) \times L$ such that $\langle O, A\rangle \in \bigcup_{k \in L} \mathcal{K}_k$ and $l = \sup\{k \in L : \langle O, A\rangle \in \mathcal{K}_k\}$ will be called* proto-fuzzy concepts. *The set of all proto-fuzzy concepts will be denoted $\mathcal{K}^P$.*

For our example will proto-fuzzy concept $\langle O, A, l\rangle$ means the group of students O, whose best common result of all subjects from the set A is l. In the following tables are some proto-fuzzy concepts of our example.

{F, J, A, N, M, E, L, D, P, T}	{F, J, A, P, E, M}	{F, M, E, L}
{Sl, Ge, Gr, En, Ch, Ae, Hi}	{Sl, Bi, Ae , Gr, En , Hi}	{Ma, Ph}
3	2	1

The set of all proto-fuzzy concepts will be used for creating one-sided fuzzy concepts with help of mappings defined below. Mappings will determine which side will be fuzzy.

Definition 6. *Let $O \subseteq \mathcal{O}$ be an arbitrary set of objects. The set*

$$\mathcal{K}^P_O = \{\langle A, l\rangle \in \mathcal{P}(\mathcal{A}) \times L : (\exists B \supseteq O)\langle B, A, l\rangle \in \mathcal{K}^P\}$$

will be called the contraction *of the set of proto-fuzzy concepts subsistent to the set O.*

Definition 7. *Define mappings*

$$\Uparrow : 2^{\mathcal{O}} \to L^{\mathcal{A}},$$

$$\Downarrow : L^{\mathcal{A}} \to 2^{\mathcal{O}}$$

in the following way: For every subset O of objects and for every fuzzy-subsets of attributes put

$$\Uparrow(O)(a) = \sup\{l \in L : (\exists\langle A, l\rangle \in \mathcal{K}^P_O) a \in A\}$$

$$\Downarrow(\widetilde{A}) = \bigcup\{O \subseteq \mathcal{O} : (\forall a \in \mathcal{A})(\exists\langle A, l\rangle \in \mathcal{K}^P_O) a \in A \;\&\; l \geq \widetilde{A}(a)\}.$$

Lemma 4. *Let O and A are arbitrary subsets of objects and attributes respectively, and l be an arbitrary value of L such that for every object o of the set O and for every attribute a of the set A, $\mathcal{R}(o,a) \geq l$. Then there exist subsets $\overline{O} \supseteq O$, $\overline{A} \supseteq A$ and value $k \in L$ such that $k \geq l$ and $\langle \overline{O}, \overline{A}, k \rangle \in \mathcal{K}^P$.*

Proof. It is given that $(\forall o \in O)(\forall a \in A) r(o,a) \geq l$. Take

$$\overline{A} = \uparrow_l (O) = \{a \in \mathcal{A} : (\forall o \in O) r(o,a) \geq l\} \supseteq A.$$

Then

$$\overline{O} = \downarrow_l (\overline{A}) = \downarrow_l (\uparrow_l (O))$$

and from the fact that for every $l \in L$ the pair $(\uparrow_l, \downarrow_l)$ forms a Galois connection, it implies that $\downarrow_l (\uparrow_l (O)) \supseteq O$ and hence $\langle \overline{O}, \overline{A} \rangle \in \mathcal{K}_l$. If

$$k = \sup\{m \in L : \langle \overline{O}, \overline{A} \rangle \in \mathcal{K}_m\}$$

the theorem 9 implies that $\langle \overline{O}, \overline{A} \rangle \in \mathcal{K}_k$ and so

$$\langle \overline{O}, \overline{A}, k \rangle \in \mathcal{K}^P.$$

□

Lemma 5. *Let $l \in L$, $O_1, O_2 \subseteq \mathcal{O}$ and $\langle A_1, l_1 \rangle \in \mathcal{K}^P_{O_1}$, $\langle A_2, l_2 \rangle \in \mathcal{K}^P_{O_2}$ that $A_1 \cap A_2 \neq \emptyset$ and $l_1 \wedge l_2 \geq l$. Then exists $\langle A, k \rangle \in \mathcal{K}^P_{O_1 \cup O_2}$ that $A \supseteq A_1 \cap A_2$ and $k \geq l$.*

Proof. $\langle A_1, l_1 \rangle \in \mathcal{K}^P_{O_1}$ it means that

$$(\forall o \in O_1)(\forall a \in A_1) r(o,a) \geq l_1.$$

$\langle A_2, l_2 \rangle \in \mathcal{K}^P_{O_2}$ it means that

$$(\forall o \in O_2)(\forall a \in A_2) r(o,a) \geq l_2.$$

Hence

$$(\forall o \in O_1 \cup O_2)(\forall a \in A_1 \cap A_2) r(o,a) \geq l_1 \wedge l_2 \geq l.$$

The lemma 13 implies that

$$(\exists O \supseteq O_1 \cup O_2)(\exists A \supseteq A_1 \cap A_2)(\exists k \in L : k \geq l) \langle O, A, k \rangle \in \mathcal{K}^P$$

hence

$$\langle A, k \rangle \in \mathcal{K}^P_{O_1 \cup O_2}.$$

□

Lemma 6. *Let $O \subseteq \mathcal{O}$, $\langle A_1, l_1 \rangle$, $\langle A_2, l_2 \rangle \in \mathcal{K}^P_O$ such that $A_1 \cap A_2 \neq \emptyset$. Then there exist $A \subseteq \mathcal{A}$ and $l \geq l_1 \vee l_2$ such that $\langle A, l \rangle \in \mathcal{K}^P_O$.*

Proof. For all $o \in O$ and for all $a \in A_1 \cap A_2$ is

$$r(o,a) \geq l_1 \text{ and } r(o,a) \geq l_2.$$

Hence

$$r(o,a) \geq l_1 \vee l_2.$$

From above and lemma 13 implies that there exist

$$(\exists B \supseteq O)(\exists A \supseteq A_1 \cap A_2)(\exists l \in L : l \geq l_1 \vee l_2)\langle B, A, k\rangle \in \mathcal{K}^P.$$

Hence

$$\langle A, l\rangle \in \mathcal{K}^P_O.$$

□

Lemma 7. *Let O_1, O_2 be an arbitrary subsets of the set of objects such that $O_1 \subseteq O_2$. Then $\mathcal{K}^P_{O_1} \supseteq \mathcal{K}^P_{O_2}$.*

Proof. Because of $O_1 \subseteq O_2$ is

$$\{\langle A_1, l_1\rangle \in \mathcal{P}(A) \times L : (\exists B_1 \supseteq O_1)\langle B_1, A_1, l_1\rangle \in \mathcal{K}^P\} \supseteq$$

$$\supseteq \{\langle A_2, l_2\rangle \in \mathcal{P}(A) \times L : (\exists B_2 \supseteq O_2)\langle B_2, A_2, l_2\rangle \in \mathcal{K}^P\}.$$

Hence

$$\mathcal{K}^P_{O_1} \supseteq \mathcal{K}^P_{O_2}.$$

□

Theorem 3. *The pair of mappings $(\Uparrow, \Downarrow)$ forms a Galois connection between the power-set lattice $\mathcal{P}(\mathcal{O})$ and the fuzzy power-set lattice $\mathcal{F}(\mathcal{A})$.*

Proof. For every set O, the subset of the set of objects and the fuzzy set $\widetilde{A}$, the fuzzy-subset of the set of attributes, have to be proven that O is the subset of $\Downarrow(\widetilde{A})$ if, and only if $\widetilde{A}$ is the fuzzy-subset of $\Uparrow(O)$.
$\boxed{\Rightarrow}$

$$O \subseteq \Downarrow(\widetilde{A}) = \bigcup\{B \subseteq \mathcal{O} : (\forall b \in \mathcal{A})(\exists \langle A, l\rangle \in \mathcal{K}^P_B) b \in A \;\&\; l \geq \widetilde{A}(b)\}.$$

Let $a \in \mathcal{A}$ be an arbitrary attribute. The lemma 14 implies that there exists $A_a \subseteq \mathcal{A}$, $l_a \in L$ such that $a \in A_a$, $l_a \geq \widetilde{A}(a)$ and

$$\langle A_a, l_a\rangle \in \mathcal{K}^P_{\Downarrow(\widetilde{A})}.$$

$O \subseteq \Downarrow(\widetilde{A})$ implies that $\mathcal{K}^P_O \supseteq \mathcal{K}^P_{\Downarrow(\widetilde{A})}$. Hence $\langle A_a, l_a\rangle \in \mathcal{K}^P_O$. So

$$\widetilde{A}(a) \leq l_a \leq \sup\{l \in L : (\exists \langle A, l\rangle \in \mathcal{K}^P_O) a \in A\} = \Uparrow(O)(a).$$

Because of arbitrarity of attribute a and from unequality above implies that $\widetilde{A}$ is the fuzzy-subset of $\Uparrow(O)$.

$\boxed{\Leftarrow}$ Let $a \in \mathcal{A}$ be an arbitrary attribute. Denote

$$l_a = \Uparrow(O)(a) = \sup\{l \in L : (\exists \langle A, l\rangle \in \mathcal{K}_O^P) a \in A\}.$$

The proposition implies that for every $a \in \mathcal{A}$, $\widetilde{A}(a) \leq l_a$. The lemma 15 implies that there exists $A_a \subseteq \mathcal{A}$ such that $\langle A_a, l_a\rangle \in \mathcal{K}_O^P$, and that implies

$$O \in \{B \subseteq \mathcal{O} : (\forall b \in \mathcal{A})(\exists \langle A, l\rangle \in \mathcal{K}_B^P) a \in A \ \&\ l \geq \widetilde{A}(b)\}$$

hence

$$O \subseteq \bigcup\{B \subseteq \mathcal{O} : (\forall b \in \mathcal{A})(\exists \langle A, l\rangle \in \mathcal{K}_B^P) a \in A \ \&\ l \geq \widetilde{A}(b)\} = \Downarrow(\widetilde{A}).$$

So the set O is subset of $\Downarrow(\widetilde{A})$. □

For the case of object fuzzy side will be used mappings:

$$\Uparrow : 2^{\mathcal{A}} \to L^{\mathcal{O}},$$

$$\Downarrow : L^{\mathcal{O}} \to 2^{\mathcal{A}}.$$

Let $\widetilde{O}$ be a fuzzy subset of objects and $A \subseteq \mathcal{A}$ is subset of attributes.

$$\Uparrow(A)(o) = \sup\{l \in L : (\exists \langle O, l\rangle \in \mathcal{K}_A^P) o \in O\}$$

$$\Downarrow(\widetilde{O}) = \bigcup\{T \subseteq \mathcal{A} : (\forall o \in \mathcal{O})(\exists \langle O, l\rangle \in \mathcal{K}_T^P) o \in O \ \&\ l \geq \widetilde{O}(a)\},$$

where

$$\mathcal{K}_A^P = \{\langle O, l\rangle : (\exists T \supseteq A)\langle O, T, l\rangle \in \mathcal{K}^P\}.$$

Example 1. For example take the fuzzy-subset of the set of attributes,

$$\widetilde{A} = \{(\text{Ma},1), (\text{Sl},3), (\text{Ph},1), (\text{Ge},3), (\text{Bi},4), (\text{Gr},2), (\text{En},2), (\text{Ch},2), (\text{Ae},4), (\text{Hi},4)\}.$$

In the table below are some proto-fuzzy concepts which contains students whose results satisfy to $\widetilde{A}$. Hence $\Downarrow(\widetilde{A})$ ={F,L,M,E}. Elements of $K^P_{\Downarrow(\widetilde{A})}$ are shown in the next table. Hence

$$\Uparrow(\Downarrow(\widetilde{A})) =$$

$$= \{(\text{Ma},1),(\text{Sj},3),(\text{Ph},1),(\text{Ge},3),(\text{Bi},2),(\text{Gr},2),(\text{En},2),(\text{Ch},2),(\text{Ae},2),(\text{Hi},2)\}$$

.

Table 3. Some of proto-fuzzy concepts which satisfy to $\widetilde{A}$

{M, E}	{Ma,Sl,Ph,Ge,Bi,Gr,En,Ch,Ae,Hi}	1
{M, E, F}	{Ma, Sl, Ph, Gr, Ae, Hi}	1
{M, E, L}	{Ma, Ph, Ch}	1
{M, E, F, L}	{Ma, Ph}	1
{M, E, F}	{Ma,Sl,Ph,Bi,Ch,Ae,En,Gr,Hi}	2
{M, E, L}	{Ma,Ph,Ge,Bi,Ch,Ae,En,Gr,Hi}	2
{M, E, F, L}	{Ma,Ph,Bi,Ch,Ae,En,Gr,Hi}	2

Table 4. Elements of the $K^P_{\Downarrow(\widetilde{A})} = K^P_{\{\mathrm{M,E,F,L}\}}$

{Ma,Sl,Ph,Ge,Bi,Gr,En,Ch,Ae,Hi}	3
{Ma, Ch, Ae, Gr, En, Hi}	2
{Ma, Ph, Bi, Ch, Ae, Gr, En, Hi}	2
{Ma,Ph,Bi,Gr,En,Ae,Hi}	2
{Ae,Gr,En,Hi}	2
{Ph,Bi,Gr,En,Ae,Hi}	2
{Bi,Gr,En,Ae,Hi}	2
{Bi,Gr,En,Ae,Hi}	2
{Ch,Gr,En,Hi}	2
{Ma,Gr,En,Ae,Hi}	2
{Ma,Ph}	1

4 Retrieval of proto-fuzzy concepts

Proto-fuzzy concepts will be retrieved with a help of cuts and "pessimistic sights" to object-value or attribute-value plains.

Definition 8. *Define new binary relations*

$$\mathcal{R}_{\mathcal{A}} = \{(o,l) \in \mathcal{O} \times L : (\forall a \in \mathcal{A}) r(o,a) \geq l\}$$

and

$$\mathcal{R}_{\mathcal{O}} = \{(a,l) \in \mathcal{A} \times L : (\forall o \in \mathcal{O}) r(o,a) \geq l\}.$$

The formal context $\langle \mathcal{O}, L, \mathcal{R}_{\mathcal{A}} \rangle$ *will be called* object–value sight *and the formal context* $\langle \mathcal{A}, L, \mathcal{R}_{\mathcal{O}} \rangle$ *will be called* attribute–value sight.

Table 5. Object–value and attribute–value sight

	1	2	3	4	5
Fred			•	•	•
Joey			•	•	•
Alice			•	•	•
Nancy				•	•
Mary	•	•	•	•	•
Eve	•	•	•	•	•
Lucy			•	•	•
David				•	•
Peter			•	•	•
Tom			•	•	•

	1	2	3	4	5
Math				•	•
Slovak language			•	•	•
Physics				•	•
Geography			•	•	•
Biology				•	•
German language		•	•	•	•
English language		•	•	•	•
Chemistry			•	•	•
Aesthetics			•	•	•
History		•	•	•	•

Definition 9. *Define new mappings*

$$\uparrow_{\mathcal{A}} : 2^{\mathcal{O}} \to L \text{ and } \downarrow_{\mathcal{A}} : L \to 2^{\mathcal{O}},$$

$$\uparrow_{\mathcal{O}} : 2^{\mathcal{A}} \to L \text{ and } \downarrow_{\mathcal{O}} : L \to 2^{\mathcal{A}}.$$

For every $O \subseteq \mathcal{O}$, $A \subseteq \mathcal{A}$ *and* $l \in L$ *put*

$$\uparrow_{\mathcal{A}} (O) = \inf\{\sup\{l \in L : (o,l) \in \mathcal{R}_{\mathcal{A}}\} : o \in O\}$$

$$\downarrow_{\mathcal{A}} (l) = \{o \in \mathcal{O} : (o,l) \in \mathcal{R}_{\mathcal{A}}\}$$

$$\uparrow_{\mathcal{O}} (A) = \inf\{\sup\{l \in L : (a,l) \in \mathcal{R}_{\mathcal{O}}\} : a \in A\}$$

$$\downarrow_{\mathcal{O}} (l) = \{a \in \mathcal{A} : (a,l) \in \mathcal{R}_{\mathcal{O}}\}.$$

Theorem 4. *Pairs of mappings $(\uparrow_{\mathcal{A}}, \downarrow_{\mathcal{A}})$ and $(\uparrow_{\mathcal{O}}, \downarrow_{\mathcal{O}})$ form Galois connections between the power-set lattice $\mathcal{P}(O)$ or $\mathcal{P}(A)$ and the lattice of values L.*

Proof. The proof will be shown only for first pair. The proof for second pair is likewise.

1. Let $O_1 \subseteq O_2 \subseteq \mathcal{O}$. It follows from an inclusion above that

$$\{\sup\{l \in L : (o,l) \in \mathcal{R}_{\mathcal{A}}\} : o \in O_1\} \subseteq$$

$$\subseteq \{\sup\{l \in L : (o,l) \in \mathcal{R}_{\mathcal{A}}\} : o \in O_2\}$$

and from a properties of infimum

$$\inf\{\sup\{l \in L : (o,l) \in \mathcal{R}_{\mathcal{A}}\} : o \in O_1\} \geq$$

$$\geq \inf\{\sup\{l \in L : (o,l) \in \mathcal{R}_{\mathcal{A}}\} : o \in O_2\}.$$

Hence

$$\uparrow_{\mathcal{A}} (O_1) \geq \uparrow_{\mathcal{A}} (O_2).$$

2. Let $l_1, l_2 \in L$. If $l_1 \leq l_2$ then

$$\{o \in \mathcal{O} : (o, l_1) \in \mathcal{R}_{\mathcal{A}}\} \supseteq \{o \in \mathcal{O} : (o, l_2) \in \mathcal{R}_{\mathcal{A}}\}.$$

Hence

$$\downarrow_{\mathcal{A}} (l_1) \supseteq \downarrow_{\mathcal{A}} (l_2).$$

3. Let $O \subseteq \mathcal{O}$. Denote

$$s_o = \sup\{l \in L : (o,l) \in \mathcal{R}_{\mathcal{A}}\},$$

for arbitrary object $o \in O$. From definition of $\uparrow_{\mathcal{A}}$

$$\uparrow_{\mathcal{A}} (O) = \inf\{s_b : b \in O\} \leq s_o$$

and from property 2 implies

$$\downarrow_{\mathcal{A}} (\uparrow_{\mathcal{A}} (O)) \supseteq \downarrow_{\mathcal{A}} (s_o) = \{b \in \mathcal{O} : (b, s_o) \in \mathcal{R}_{\mathcal{A}}\}.$$

Arbitrarity of o implies that

$$\downarrow_{\mathcal{A}} (\uparrow_{\mathcal{A}} (O)) \supseteq \bigcup_{o \in O} \{b \in \mathcal{O} : (b, s_o) \in \mathcal{R}_{\mathcal{A}}\} \supseteq O.$$

4. Let $l \in L$ be an arbitrary value. Denote $s_o = \sup\{k \in L : (o,k) \in \mathcal{R}_{\mathcal{A}}\}$. For all

$$o \in \downarrow_{\mathcal{A}} (l) = \{b \in \mathcal{O} : (b,l) \in \mathcal{R}_{\mathcal{A}}\}$$

is $s_o \geq l$. Hence

$$\uparrow_{\mathcal{A}} (\downarrow_{\mathcal{A}} (l)) = \inf\{s_b : b \in \downarrow_{\mathcal{A}} (l)\} \geq l.$$

□

Definition 10. *The pair $\langle O, l\rangle$ is called* $\mathcal{A}$-concept *of the object–value sight $\langle \mathcal{O}, L, \mathcal{R}_{\mathcal{A}}\rangle$ iff $\uparrow_{\mathcal{A}} (O) = l$ and $\downarrow_{\mathcal{A}} (l) = O$. The set of all $\mathcal{A}$-concepts will be denoted $\mathcal{K}_{\mathcal{A}}$.*

Definition 11. *The pair $\langle A, l\rangle$ is called* $\mathcal{O}$-concept *of the attribute–value sight $\langle \mathcal{A}, L, \mathcal{R}_{\mathcal{O}}\rangle$ iff $\uparrow_{\mathcal{O}} (A) = l$ and $\downarrow_{\mathcal{O}} (l) = A$. The set of all $\mathcal{O}$-concepts will be denoted $\mathcal{K}_{\mathcal{O}}$.*

It can be defined an object–value sight for every subset of attributes or attribute–value sight for every subset of objects, but their usage for this paper wasn't necessary.

Theorem 5. *Let $l \in L$, $A_1, A_2 \subseteq \mathcal{A}$, $O_1, O_2 \subseteq \mathcal{O}$ such that $\langle \mathcal{O}, A_1, l\rangle, \langle O_1, \mathcal{A}, l\rangle \in \mathcal{K}^P$ and $\langle O_2, A_2\rangle \in \mathcal{K}_l$ for context $\langle \mathcal{O} \setminus O_1, \mathcal{A} \setminus A_1, \mathcal{R}_l\rangle$. Then*

$$\langle O_1 \cup O_2, A_1 \cup A_2, l\rangle, \in \mathcal{K}^P.$$

Proof. It will be shown that $A_1 \cup A_2 =\downarrow_l (O_1 \cup O_2)$ and $O_1 \cup O_2 =\uparrow_l (A_1 \cup A_2)$.

If $a \in A_1$ then for all $o \in \mathcal{O}$ is $(o, a) \in \mathcal{R}_l$.

If $a \in A_2$ then for all $o \in O_1 \cup O_2$ is $(o, a) \in \mathcal{R}_l$.

If $a \in A_1 \cup A_2$ then for all $o \in (\mathcal{O} \cap (O_1 \cup O_2)) = O_1 \cup O_2$ is $(o, a) \in \mathcal{R}_l$.

Hence $A_1 \cup A_2 \subseteq\uparrow_l (O_1 \cup O_2)$.

The opposite inclusion will be shown by contradiction. Let us assume $a \in\uparrow_l (O_1 \cup O_2)$ and $a \notin A_1 \cup A_2$.

From $a \in\uparrow_l (O_1 \cup O_2)$ implies that for all $o \in O_1 \cup O_2 \supseteq O_2$ is $(o, a) \in \mathcal{R}_l$. From $a \notin A_1 \cup A_1$ implies that $a \in (\mathcal{A} \setminus (A_1 \cup A_1)) = ((\mathcal{A} \setminus A_1) \setminus A_2)$. It is the contradiction to precondition $\langle O_2, A_2\rangle \in \mathcal{K}_l$ for context $\langle \mathcal{O} \setminus O_1, \mathcal{A} \setminus A_1, \mathcal{R}_l\rangle$.

The second equality can be shown likewise. □

Subcontexts from the theorem will be called *auxiliary subcontexts of l-cut*. Concepts of sights will be retrieved with a help of mappings $\uparrow_{\mathcal{A}}, \downarrow_{\mathcal{A}}, \uparrow_{\mathcal{O}}$ and $\downarrow_{\mathcal{O}}$. It's good to know that $\langle O, l\rangle \in \mathcal{K}_{\mathcal{A}}$ then $\langle O, \mathcal{A}, l\rangle \in \mathcal{K}^P$, because of $\mathcal{A}$ is closed. Denote $\mathcal{A}$ as the set of all subjects and $\mathcal{O}$ as the group of all students from our example. Hence

$$\langle \mathcal{O}, \mathcal{A}, 4\rangle \in \mathcal{K}^P,$$

$$\langle \mathcal{O} \setminus \{\text{N,D}\}, \mathcal{A}, 3\rangle, \langle \mathcal{O}, \mathcal{A} \setminus \{\text{Ma,Ph,Bi}\}, 3\rangle \in \mathcal{K}^P,$$

$$\langle \{\text{M,E}\}, \mathcal{A}, 1\rangle, \langle \mathcal{O}, \{\text{Gr,En,Hi}\}, 2\rangle \in \mathcal{K}^P,$$

Let us create auxiliary subcontexts of 3-cut, 2-cut and 1-cut.

Table 6. Auxiliary subcontexts of 3-cut

3	Ma	Ph	Bi
N			•
D	•		

There are only two concepts in the auxiliary subcontext of 3-cut, $\langle\{N\},\{Bi\}\rangle$ and $\langle\{D\},\{Ma\}\rangle$. The theorem 23 implies that
$\langle\mathcal{O}\setminus\{\mathrm{N}\},\mathcal{A}\setminus\{Ph,Bi\},3\rangle$, $\langle\mathcal{O}\setminus\{D\},\mathcal{A}\setminus\{\mathrm{Ma,Ph}\},3\rangle\in\mathcal{K}^P$.

Table 7. Auxiliary subcontext of 2-cut

2	Ma	Sl	Ph	Ge	Bi	Ch	Ae
F	•	•	•		•	•	•
J		•	•	•	•		•
A		•		•	•		•
N		•			•	•	
L	•		•	•	•	•	•
D	•					•	•
P	•	•	•	•	•		•
T	•		•	•	•		•

Because of the convexity of l-concepts, we can omit Eve and Mary from the set of students for auxiliary subcontext of 2-cut. And for input of theorem 23 for degree 2 can be used proto-fuzzy concepts $\langle\{\mathrm{M,E}\},\mathcal{A},1\rangle,\langle\mathcal{O},\{\mathrm{Gr,En,Hi}\},2\rangle\in\mathcal{K}^P$.

Table 8. Auxiliary subcontext of 1-cut

1	Ma	Sl	Ph	Ge	Bi	Gr	En	Ch	Ae	Hi
F	•	•	•			•			•	
J		•		•	•	•	•		•	•
A				•	•	•	•			
N							•			
L	•		•					•		
D						•	•			
P		•		•	•				•	
T	•								•	

5 Conclusion

Conceptual scaling and theory of triadic contexts will be the object of our future work and study. We will try to algoritmize outline process.

We are grateful for precious comments of our colleagues RNDr. Peter Eliaš PhD. and RNDr. Jozef Pócs.

Paper was created with support of grant 1/3129/06 Slovak grant agency VEGA.

References

1. Krajči, S.: Cluster Based Efficient Generation Of Fuzzy Concepts Neural Network World 5/03 521–530
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st. Springer-Verlag New York, Inc., 1997
3. Snášel, V., Ďuráková, D., Krajči, S., Vojtáš, P.: Merging Concept Lattices of α-cuts Of Fuzzy Contexts Contributions To General Algebra 14 Proceedings of the Olomouc Conference 2002 (AAA 64) and the Postdam Conference 2003 (AAA 65) Verlag Johanes Heyn, Klagenfurt 2004
4. Bělohlávek, R.: Lattices Generated By Binary Fuzzy Relations Tatra Mountains Mathematical Publications 16 (1999),11-19

Application of the Formal Concept Analysis in Evaluation of Results of ANEWS Questionnaire and Physical Activity of the Czech Regional Centers*

Jiří Zacpal[1], Erik Sigmund[1], Josef Mitáš[2], Vladimír Sklenář[2]

[1] Dept. Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic
{vladimir.sklenar,jiri.zacpal}@upol.cz

[2] Centre for Kinanthropology Research, Palacky University, Olomouc
tr. Miru 115, CZ-771 11 Olomouc, Czech Republic
{erik.sigmund,josef.mitas}@upol.cz

Abstract. Formal concept analysis is a method of exploratory data analysis that aims at the extraction of natural clusters from object-attribute data tables. The clusters, called formal concepts, can be similar to human-perceived concepts in a traditional sense and can be partially ordered by a subconcept-superconcept hierarchy. The hierarchical structure of formal concepts (so-called concept lattice) represents structured information obtained automatically from the input data table. The goal of this paper is to describe a method of evaluation of ANEWS questionnaire by Formal concept analysis. We describe a method adjustment of questionnaire by scaling to classical formal context. After that we separate some attributes to groups and make so-called "aggregate atributes". This way we make modified formal context and calculate formal concept lattice. We define term "characteristic function" for every concept. This is function, which for given extent or intent return a real number, which characterized this concept and is important for evaluation. Our method is illustrated on ANEWS questionnaire and measured steps in randomized sample of 15-65 years-old inhabitants of the Czech regional centers.

1 Introduction and problem setting

Questionnaires are being used in many areas of human activities. The aim is to reveal patterns of behavior and various kinds of dependencies among variables being surveyed. Descriptive statistics and statistical hypotheses testing are among the tools traditionally used for evaluation of questionnaires. A practical disadvantage of the traditional statistical approaches is the need to formulate

* Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, by institutional support, research plan MSM 6198959214, and by grant No. 6198959221 'Physical activity and inactivity of inhabitants of the Czech Republic in the context of behavioral changes' of the MSM.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 97–108,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

hypotheses to be tested. Without any prior structured view on the data contained in the questionnaires, formulation of relevant hypotheses is a difficult task. Another disadvantage of traditional statistical approaches is the limitation regarding what statistics can tell about data and how statistical summaries can be understood by experts in the field of inquiry who are not experts in statistics.

This paper presents results on evaluation of ANEWS questionnaire and physical activity of the czech regional centers. The paper is a continuation of previous studies regarding the IPAQ questionnaire, see [5]. At the beginning of our study, there was a need for an alternative means of evaluation of questionnaires formulated by experts (domain experts) from the Faculty of Physical Culture of the Palacky University, Olomouc, who are involved in a world-wide project of monitoring physical activities in today's population. The experts struggled with classical statistical techniques and were looking for alternative methods of evaluation of the questionnaires. It turned out that basic methods of formal concept analysis (FCA) [10] are quite useful for the domain experts. Putting briefly, a concept lattice and its parts provide the experts with an easy-to-understand hierarchical view on the data.

In terms of FCA, the basic idea is the following. The objects are the individuals (or their groups) being surveyed in the questionnaires, the attributes correspond to the variables being monitored by the questionnaires. The corresponding concept lattice or its parts reveals to the domain expert the groups in dependence on the attributes and the expert can see various dependencies between attributes, how large the groups are etc. Therefore, the concept lattice provides the expert with a first insight into the data. Such an insight is crucial. Very often, this insight is what the expert needs to see. Furthermore, based on this insight, the expert can pursue more detailed inquiries including those based on classical statistical techniques.

Recent study focuses on considering groups of individuals as objects. The present study is based on the idea that some questions are closely related. It's useful to group those attributes which resulted from scaling of the questions into one attribute. Thus we would obtain a more comprehensive view of the questionnaire. This idea made us create so-called "aggregate attributes".

The advantage of taking groups and the relative frequencies instead of individuals and original attributes is conciseness of the description provided by the resulting concept lattice which is what the experts asked for. The disadvantage, as with any other method which involves aggregation and summarization, is loss of information. We present our method, experimental results, as well as a brief description of the software tool we used.

2 Questionnaire adjustment

Each questionnaire consists of questions to which the respondents choose an answer from a multiple choice. From the perspective of FCA the group of respondents can be understood as a set of objects and individual questions as attributes. The respondents answers then create binary relation between the set of objects and the attributes. The answers do not have to be necessarily bi-valent (yes-no). Multiple-value type of answers (age, number of steps,) can appear here. Due to this, a suitable scale needs to be applied to transfer the multiple-value type of answers into bivalent forms. The result of this process is a context $\langle X, Y, I \rangle$, where X is the set of objects – respondents, Y is the set of attributes – adjusted answers from the questionnaire and I is the binary relation between X and Y, where $(x, y) \in I$ means that respondent x answered yes to question y.

Another adjustment of the questionnaire is based on the idea that some questions are closely related. For example question: "The streets in my neighborhood do not have many cul-de-sacs (dead-end streets)" is closely related with question: "The distance between intersections in my neighborhood is usually short (100 yards or less; the length of a football field or less)", because the questions are related to conditions for walking. Was it not more useful than to group those attributes which resulted from scaling of the questions into one attribute? Thus we would obtain a more comprehensive view of the questionnaire. This idea made us create so-called "aggregate attributes".

Firstly, an expert needs to decide which questions can be grouped into an "aggregate attributes". Then, we replace all the attributes which were formed through scaling with "aggregate attributes" using the following procedure. We calculate the weighted mean of individual attributes and we scale this mean.

Formally: There is number n of questions in the questionnaire which we want to cluster into the "aggregate attributes". Through scaling of these questions $\sum_{i=1}^{n} m_i$ of attributes was created, where m_i is the number of attributes which was formed through scaling of i-question. The weighted mean for the object x, is calculated according to the formula:

$$v(x) = \sum_{i=1}^{n} \sigma_i \sum_{j=1}^{m_i} \omega_{ij} I(x, a_{ij})$$

where

σ_i is weight of question i

ω_{ij} is weight of attribute j which was formed through scaling of question i

a_{ij} is attribute which was formed through scaling of question i, $j \in m_i$

I is binary relation between X and Y_1, which is the original set of attributes from which we remove all the attributes which we have grouped into aggregate attributes and then we add the aggregate attributes into it.

Value $v(x) \in \langle 0, 1 \rangle$. We create 5 aggregate attributes according to these rules:

$\langle x$,NAME-very low$\rangle \in I_1$ iff $v(x) \in \langle 0, 0.2 \rangle$

$\langle x$,NAME-low$\rangle \in I_1$ iff $v(x) \in (0.2, 0.4\rangle$

$\langle x$,NAME-moderate$\rangle \in I_1$ iff $v(x) \in (0.4, 0.6\rangle$

$\langle x$,NAME-high$\rangle \in I_1$ iff $v(x) \in (0.6, 0.8\rangle$

$\langle x$,NAME-very high$\rangle \in I_1$ iff $v(x) \in (0.8, 1\rangle$,

where NAME is the name of the group of attributes which we grouped. Using these aggregate attributes, we replace all the grouped attributes. This way a formal context $\langle X, Y_1, I_1 \rangle$ is created, where Y_1 is the original set of attributes from which we remove all the attributes which we have grouped into aggregate attributes and then we add the aggregate attributes into it. $\langle x, y \rangle \in I_1$ if y is not aggregate attribute and for aggregate attributes the above rules are applied.

Example 1. For better understanding we provide an example. There are questions (G1-G3) in the questionnaire which concern Streets in my neighborhood. The expert states the values in individual weights: $\sigma_{G1} = 0.4$, $\sigma_{G2} = 0.4$ and $\sigma_{G3} = 0.2$ To all questions, the respondents could choose these answers: 1 - strongly disagree, 2 - somewhat disagree, 3 - somewhat agree, 4 - strongly agree. The value of weights is stated in Tab. 1. They created 5 "aggregate attributes": Street-very low, Street-low, Street-moderate, Street-high, Street-very high (the classification of streets depending on their suitability for walking). If respondent x answers the questions this way: G1 - 3, G2 - 1, G3 - 2, will be $v(x) = 0.4 \cdot 0.75 + 0.4 \cdot 0.5 + 0.2 \cdot 0,5 = 0.6$ and then $\langle x,\text{Street-moderate} \rangle \in I_1$.

Table 1. Weights ω_{ij} from example 1.

questions	answers			
	1	2	3	4
G1 - absence of cul-de-sac (dead-end streets)	0.25	0.5	0.75	1
G2 - short distance between intersections	0.25	0.5	0.75	1
G3 - alternative routes for getting from place to place	0.25	0.5	0.75	1

Typically, such a formal context contains many objects and a manageable number of attributes. The corresponding concept lattice is too large for an expert to comprehend. In addition, the expert might not be interested in the formal concepts from this concept lattice. Rather, the expert might want to consider aggregates of the individual respondents as objects in the formal context with the aggregates defined by having the same attributes on a set S of attributes specified by an expert, such as those regarding age, sex, etc., with S being a subset of the set Y of all attributes. Attributes from S will be called characteristic attributes.

The aggregates we consider are equivalence classes of individual respondents. For respondents $x_1, x_2 \in X$, put

$$x_1 \equiv_S x_2 \text{ if and only if } \{x_1\}^{\uparrow} \cap S = \{x_2\}^{\uparrow} \cap S.$$

Clearly, $\equiv_S$ is an equivalence relation on X and $x_1 \equiv_S x_2$ means that x_1 and x_2 have the same attributes from S, i.e. are indistinguishable by the attributes from S. We call the classes $[x]_{\equiv_S}$ of $\equiv_S$ aggregate objects and denote, furthermore,

- by X_1 the set of all classes of $\equiv_S$, i.e. $X_1 = X/\equiv_S$, by Y_2 the set of those attributes from Y_1 not included in S, i.e. $Y_2 = Y_1 - S$.

Now, for each class $[x]_{\equiv_S}$ from X_1 and each attribute $y \in Y_2$, we consider the relative frequency of objects in having attribute y and denote it by $I_2([x]_{\equiv_S}, y)$ or simply by $I_2(x, y)$. That is, we put

$$I_2(x, y) = \frac{|\{x_1 \in [x]_{\equiv_S} \,:\, x_1 \text{ has } y\}|}{|[x]_{\equiv_S}|}$$

We can consider I_2 a fuzzy relation which will indeed be the case in this study. Namely, we will consider a particular concept lattice associated to $\langle X_1, Y_2, I_2 \rangle$, called a lattice of crisply generated fuzzy concepts. For technical reasons, we round the degrees assigned by I_1 to those from the scale $\{0, 0.01, \ldots, 0.99, 1\}$.

More details on this method are described in the article [5].

3 Characteristic concept function

With Formal Concept Analysis we can find concepts whose intent include attributes interesting for our way of evaluation. Extents of these concepts contain some number of respondents. Often we are not interesting in attributes of individual respondent. Only values that characterize all respondents in the concept extent as a whole are interesting for concept evaluation. Arithmetic mean of the value with more than two-valued attribute is possible example of such value. We will use the term "characteristic function" for function that returns such value for given extent.

4 Questionnaire analysis

The ANEWS questionnaire (Neighborhood Environment Walkability Scale - Abbreviated) includes 54 questions in total. They were answered by 662 respondents. Using the method described above, we created 8 aggregate attributes, from which we created 40 attributes using scaling (8x5). Next to these attributes, the context involves other attributes of demographic data: gender (2 attributes), BMI (4), age (5), smoking (2), driver (2), orgPA (4), Steps5bigger2 (2) - attribute indicated, whether the respondents shows more steps during week than at weekend, Steps (4) - see Tab. 2.

Table 2. Scale for value Steps

attribute	steps per week
Steps-low	less then 5 999
Steps-moderate	6 000-9 999
Steps-high	10 000-13 999
Steps-very high	more than 14 000

Thus we obtained a formal context which includes 662 objects and 65 attributes. For another adjustment of formal context, aggregate objects are applied. We used Sex-male, Sex-female and steps (Steps-low, Steps-moderate, Steps-high a Steps-very high) as characteristic attributes. The obtained formal fuzzy context includes 8 objects and 59 attributes. Using it, we created corresponding fuzzy conceptual lattice. When studying the lattice, we tried to examine what influence the environment (characterized by aggregate attributes) has on the number of steps in respondents. We studied males and females separately. The Tab. 3 shows the corresponding concepts for male and Tab. 4 for female. We state only the aggregate attributes in the levels of very high (VH) and high (H). It is possible to compare also the other levels (moderate, low a very low), but we were interested mainly in the positive influence of the environment on steps.

Table 3. Degree of some attributes in concepts, which extents consist of aggregate objects SexMale an Steps-low, Steps-moderate, Steps-high, Steps-very high. Aggregate objects: L - Steps-low, M - Steps-moderate, H - Steps-high, VH - Steps-very high.

attribute	extent		
	L,M,H,VH	VH	L
BuildingsFlat-very high	0	0.01	0
BuildingsFlat-high	0.18	0.18	0.23
BuildingsHouse-very high	0	0.06	0
BuildingsHouse-high	0.34	0.42	0.46
Distance-very high	0.01	0.01	0.08
Distance-high	0.15	0.26	0.15
Neighbourhood-very high	0.08	0.14	0.08
Neighbourhood-high	0.23	0.28	0.23
Safety-very high	0.38	0.53	0.53
Safety-high	0.39	0.39	0.46
Service-very high	0.38	0.53	0.38
Servie-high	0.35	0.39	0.46
Street-very high	0.38	0.57	0.38
Street-high	0.26	0.26	0.46
Walking-very high	0.15	0.24	0.15
Walking-high	0.43	0.45	0.76

The levels of correspondence express minimal number of respondents in percentage, which show the given attribute. Based on the comparison of the concepts, we can see that great difference between respondents who show high number of steps (VH) and low number of steps (L) on a day, are apparent mainly in the Street-very high attribute. It is apparent that the type of street is closely associated with the number of steps. Due to this we focused on the aggregate attribute Street. We formed a formal context of attributes which were parts of the aggregate attribute Street. These attributes are formed from questions ClosedStreet (The streets in my neighborhood do not have many cul-de-sacs (dead-end streets)), ShortDistance (The distance between intersections in my neighborhood is usually short (100 yards or less; the length of a football field or

Table 4. Degree of some attributes in concepts, which extents consist of aggregate objects SexFemale an Steps-low, Steps-moderate, Steps-high, Steps-very high. Aggregate objects: L - Steps-low, M - Steps-moderate, H - Steps-high, VH - Steps-very high.

attribute	extent		
	L,M,H,VH	VH	L
BuildingsFlat-very high	0	0	0
BuildingsFlat-high	0.10	0.19	0.10
BuildingsHouse-very high	0	0.05	0
BuildingsHouse-high	0.34	0.34	0.52
Distance-very high	0.03	0.03	0.10
Distance-high	0.21	0.26	0.21
Neighbourhood-very high	0.05	0.09	0.05
Neighbourhood-high	0.10	0.33	0.10
Safety-very high	0.44	0.47	0.52
Safety-high	0.34	0.45	0.36
Service-very high	0.47	0.57	0.47
Servie-high	0.30	0.30	0.42
Street-very high	0.34	0.48	0.36
Street-high	0.33	0.35	0.42
Walking-very high	0.21	0.21	0.26
Walking-high	0.41	0.56	0.57

less) and MoreWays (There are many alternative routes for getting to one place in my neighborhood. (I don't have to go the same way every time). Each question can be answered in values 1 to 4. Using scaling we obtained a context which was formed by 662 objects (respondents) and 36 (25 - demographic attributes, 12 - attributes of environment) attributes. We used the method of aggregate objects. As characteristic attributes, we used gender (Sex-male, Sex-female) and steps (Steps-low, Steps-moderate, Steps-high a Steps-very high). A formal fuzzy context was thus created which included 6 objects and 33 attributes. We formed a corresponding fuzzy conceptual lattice. Examining the lattice we were trying to identify whether any question from the aggregate attribute Street has greater influence on the number of steps in respondents. We studied males (Tab. 5) and females (Tab. 6) separately.

The levels of correspondence express minimal number of respondents in percentage, which show the given attribute. Based on the comparison of the concepts, we can see that great difference between respondents who show high number of steps (VH) and low number of steps (L) on a day, are apparent mainly in the MoreWays attribute (here we are interested primarily in the value 4 of the answer – strongly agree). It is apparent that the variety of walking routes, when I do not have to take just a one way, are attractive and motivating for walking and cycling.

Another possibility of the questionnaire analysis is using so-called characteristic function of the concept. In this case, we define it as arithmetic mean of number of steps for respondents – objects, for 7 days in the extent of the concept.

Table 5. Degree of some attributes in concepts, which extents consist of aggregate objects SexMale an Steps-low, Steps-moderate, Steps-high, Steps-very high. Aggregate objects: L - Steps-low, M - Steps-moderate, H - Steps-high, VH - Steps-very high.

attribute	extent		
	L,M,H,VH	VH	L
ClosedStreets-1	0.03	0.06	0.08
ClosedStreets-2	0.10	0.10	0.15
ClosedStreets-3	0.28	0.28	0.46
ClosedStreets-4	0.31	0.54	0.31
MoreWays-1	0	0.03	0
MoreWays-2	0.09	0.09	0.15
MoreWays-3	0.38	0.38	0.62
MoreWays-4	0.23	0.49	0.23
ShortCross-1	0.09	0.10	0.08
ShortCross-2	0.21	0.22	0.23
ShortCross-3	0.35	0.40	0.54
ShortCross-4	0.15	0.27	0.15

Table 6. Degree of some attributes in concepts, which extents consist of aggregate objects SexMale an Steps-low, Steps-moderate, Steps-high, Steps-very high. Aggregate objects: L - Steps-low, M - Steps-moderate, H - Steps-high, VH - Steps-very high

attribute	extent		
	L,M,H,VH	VH	L
ClosedStreets-1	0.05	0.09	0.11
ClosedStreets-2	0.13	0.15	0.21
ClosedStreets-3	0.27	0.29	0.32
ClosedStreets-4	0.37	0.47	0.37
MoreWays-1	0.03	0.03	0.05
MoreWays-2	0.10	0.10	0.21
MoreWays-3	0.42	0.40	0.42
MoreWays-4	0.32	0.46	0.32
ShortCross-1	0.10	0.13	0.11
ShortCross-2	0.16	0.18	0.16
ShortCross-3	0.33	0.40	0.37
ShortCross-4	0.23	0.27	0.37

We used a formal context with aggregate attributes. We wanted to examine what influence service availability has on the value of characteristic function (aggregated attribute Service-very high, Service-high, Service-moderate, Service-low and Service-very low). The values of the characteristic function for individual concepts are shown in Tab. 7.

Table 7. Value of characteristic concept function

intent	avarage of steps	number of objects
Sex-male	12198	278
Sex-male, Distance-very high	8934	8
Sex-male, Distance-high	13226	60
Sex-male, Distance-moderate	12193	138
Sex-male, Distance-low	11707	69
Sex-male, Distance-very low	11871	3
Sex-female	11907	384
Sex-female, Distance-very high	10574	21
Sex-female, Distance-high	12019	110
Sex-female, Distance-moderate	12318	180
Sex-female, Distance-low	11095	69
Sex-female, Distance-very low	11408	4

Using this type of analysis, we can replace the classification of steps according to clear limits set in advance (Tab. 2) with one more concrete value. Along with the value of the arithmetic mean, we have to consider the number of objects to which the arithmetic mean is related (Tab. 5). Tab. 5 shows that in groups of men, it is apparent that longer distance to services (Distance-moderate, Distance-high and Distance-very high) is closely associated with higher number of steps per day. In women, the difference is not so apparent. Services (shops, restaurants, offices, banks, etc.) are an important part of everyday life, therefore further distance from the place of living does not impede women and men in accessing them.

5 Software tool

We used a software tool which is developed in the Department of Computer Science at Palacky University, Olomouc, to create the fuzzy contexts and to browse the corresponding fuzzy concept lattice. This software tool supports the whole process of the processing and evaluation of IPAQ questionnaire. The basic overview of functions that are supported and their succession is shown in Fig. 1.

The processing of the questionnaire consists of the following steps.

- **Reading data**. IPAQ questionnaire is recorded in the form of an MS Excel file. The columns of this file contain respondents' answers to individual

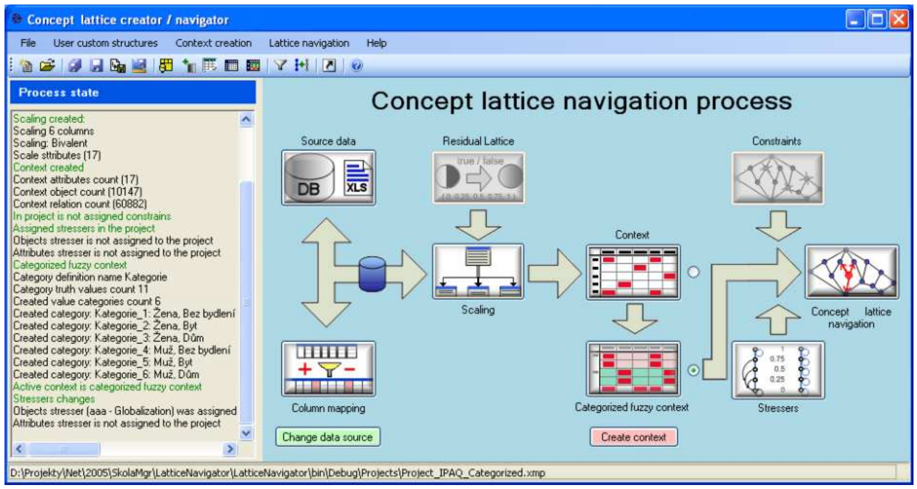


Fig. 1. Base screen of application

questions. The software tool allows to specify which columns are included in the processing.

– **Scaling**. The answers to some questions may be in the form of many-valued attributes. For example, the values in the column Age may be in the interval from 18 to 69. Due to this fact it is necessary to transform the original file to the form in which each column contains only 0 or 1. This process is called conceptual scaling [10]. Our software tool allows one to specify the bivalent attributes and the scale for each column in data source file.
– **Creation of aggregate objects**. The tool allows to interactively specify the set of characteristic attributes. The user also chooses parameters regarding the structure of truth degrees.

A fuzzy context is created after these steps. A user can then explore the associated fuzzy concept lattice and its concepts. Our software tool does not create the whole concept lattice. Instead, it supports an interactive navigation in the concept lattice. It shows the information related to the current concept and its direct neighbors. A user selects next steps by choosing an ancestor or successor of the current concept. He/she can move from a more general concept to a more special concept and vice versa. He/she can also specify the content of the extent or the intent and move to the appropriate concept. We can see the user's screen in Fig. 2.

The navigation in the concept lattice needs the calculation of the current concept and its neighbors only. This calculation is relatively fast and does not depend on the size of the whole concept lattice. Due to this fact the navigation proceeds on-line and the user can modify the course of navigation interactively, based on information gained. The user can also specify additional constraints to be satisfied by formal concepts which are to be presented to him/her.

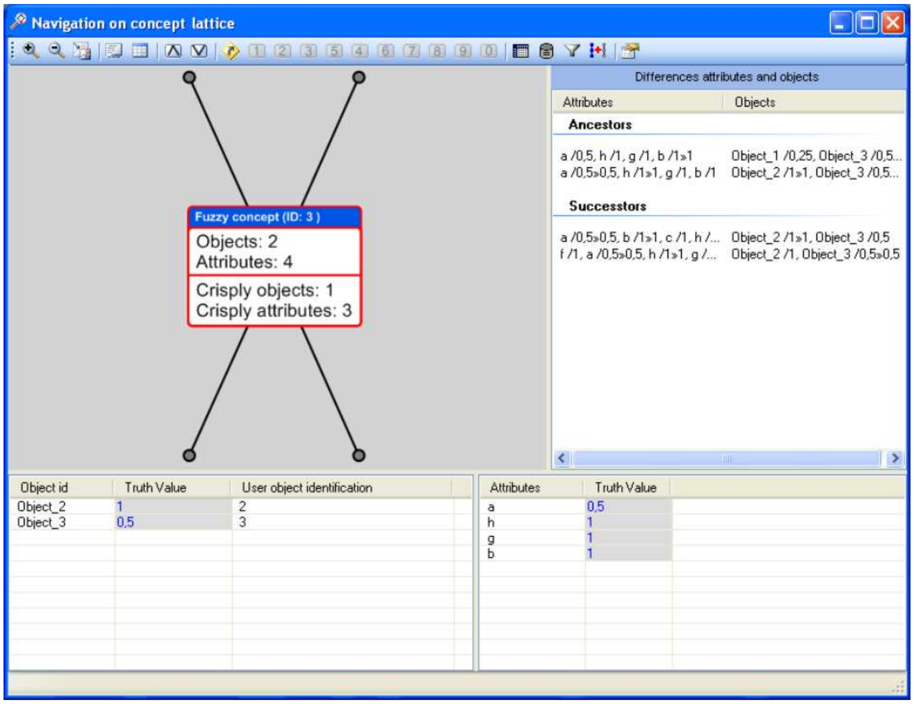


Fig. 2. Navigation in fuzzy concept lattice

6 Conclusions

Our paper described a method of analysis of a questionnaire which comprises number of questions which can be grouped based on their relation in meaning. Such an approach allows for a more global assessment of the data. We have applied this method to the ANEWS questionnaire. We can conclude that environment which is physical activity friendly and stimulating in Czech cities can be on the basis of the data and number of steps per day characterized by availability of services in short distances, walking friendly streets (walkability and cleanness of streets, no cul-de-sacs) and by nice environment in residential areas.

References

1. Bauman, A., Chey, T., Bowles, H., Smith, B., Meron, D., Ainsworth, B., Jones, D. A., Craig, C., Cameron, C., Sjostrom, M., Hagestromer, M., Frome,l K., Mitas, J. *et al.*: *International physical activity prevalence estimates: Results from the International Prevalence Study in 20 Countries*, Medicine and Science in Sports and Excercise (in press).
2. Belohlavek, R.: *Fuzzy Relational Systems: Foundations and Principles*. Kluwer, Academic/Plenum Publishers, New York, 2002.

3. Belohlavek, R.: Concept lattices and order in fuzzy logic. *Annals of Pure and Applied Logic* **128**(1–3)(2004), 277–298.
4. Belohlavek, R., Sklenar, V., Zacpal, J.: Crisply generated fuzzy concepts. ICFCA 2005, Int. Conf. Formal Concept Analysis, *LNAI* 3403, pp. 268–283, Springer-Verlag, Berlin/Heidelberg.
5. Belohlavek, R., Sklenar, V., Zacpal, J., Sigmund, E.: Evaluation of questionnaires supported by formal concept analysis. CLA 2007, Strany: 96–108, University of Montpellier II.
6. Belohlavek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: FUZZ-IEEE 2005, The IEEE International Conference on Fuzzy Systems, May 22–25, Reno (Nevada, USA), pp. 663–668.
7. Belohlavek, R., Vychodil, V.: What is a fuzzy concept lattice? In: Proc. CLA 2005, 3rd Int. Conference on Concept Lattices and Their Applications, September 7–9, Olomouc, Czech Republic, pp. 34–45.
8. Ben Yahia, S., Jaoua, A.: Discovering knowledge from fuzzy concept lattice. In: Kandel A., Last M., Bunke H. (Ed.): *Data Mining and Computational Intelligence*, pp. 167–190, Physica-Verlag.
9. Carpineto, C., Romano, G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.
10. Ganter, B., Wille, R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
11. Hájek, P.: *Metamathematics of Fuzzy Logic.* Kluwer, Dordrecht, 1998.
12. Klir, G. J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice-Hall, 1995.
13. Krajči, S.: A generalized concept lattice. *Logic J. of IGPL* **13**, 543–550.
14. Pollandt, S.: *Fuzzy Begriffe.* Springer-Verlag, Berlin/Heidelberg, 1997.
15. Sklenář, V., Zacpal, J., Sigmund, E.: *Evaluation of IPAQ questionnaire by FCA*, CLA 2005, pp. 60–69, ISBN: 80–248–0863–3, Palacky University, Olomouc, 2005.
16. Thomas, J. R., Nelson, J. K., Silverman, S. J.: *Research Methods in Physical Activity.* Human Kinetic, Champaign, 2005.
17. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (Ed.): Ordered Sets, 445–470, Reidel, Dordrecht-Boston, 1982.

A Model-driven Engineering Based RCA Process for Bi-level Models Elements / Meta-elements: Application to Description Logics

X. Dolques[1], J.-R. Falleri[1], M. Huchard[1], and C. Nebut[1]

LIRMM, CNRS and Université de Montpellier 2,
161, rue Ada, 34392 Montpellier cedex 5, France
{dolques, falleri, huchard, nebut}@lirmm.fr

Abstract. Relational Concept Analysis (RCA) facilitates the discovery of new abstractions in data descriptions including relations. A model driven approach for RCA implementation makes possible to deal with most input data (models) simply by configuring the transformation for the chosen input data type (metamodel). Until now, we only applied this approach to one-level models (mainly class models). In this paper we study RCA applied to bi-levels models, which mix elements and meta-elements (class-instance models, e.g. OWL models). We propose a model hybridisation approach to tackle the encoding problems and we provide a case study showing the results obtained on OWL models.

1 Introduction

Programs and models are easier to understand and maintain when they are organised using abstractions. Relational Concept Analysis (RCA) is one of the existing approaches to automatically detect such abstractions. RCA is an extension of Formal Concept Analysis (FCA) taking into account the relations linking the analysed entities.

To apply RCA, the analysed entities have first to be encoded into contexts containing information on the attributes of the entities and on the relations linking the entities. Then RCA is applied and builds concept lattices containing the discovered entities, that are then decoded towards the initial language for the entities.

A model-driven engineering (MDE) based approach has been proposed [1,2] to provide a generic mecanism for the encoding and decoding part of the process, that has just to be configured to be adapted to a given language. In this approach, to apply RCA to a given model m, two inputs are needed (in addition to m): the metamodel for m (that can be seen as the structural definition of the language in which m is written), and the configuration making precise which meta-elements of this metamodel have to be taken into account during the RCA process (for example: names of elements, roles of associations, etc).

Until now, such an RCA-MDE process has been successfully applied to class models. The contribution of this paper is to study its application to bi-level

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 109–120,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

models where entities and meta-entities co-exist [3]. Such models are frequently found in cases when we want to represent in the same model a concept and an instance of it, for example in UML instance diagrams [4], RDFS resources [5] or ODM ontologies [6]. We focus in this paper on applying RCA-MDE to individuals in the sense of description logics. The idea is thus to look for abstractions among individuals, each individual being typed by a class. The main issue is to deal with two levels of abstraction: the level of individuals (classically named M0) and the one of classes (M1). The presence of two levels complexifies the application of the RCA-MDE process, as we will show in this paper. After providing background on our approach, we detail two ways to apply RCA-MDE on bi-level models: a naive one, directly inspired from mono-level models, and a more elaborated one giving more relevant results, and that is based on an automated hybridation of the input model and metamodel. We explain how this process is implemented in our RCA-MDE platform, and provide results on two real-world ontologies.

2 Background : Relational Concept Analysis and description logics

Relational Concept Analysis (RCA) Relational Concept Analysis [7] is one of the extensions of Formal Concept Analysis [8] that considers links between objects in the concept construction. Connections can be made with other FCA-based proposals for dealing with relational descriptions or complex structures including [9,10,11,12] to mention just a few. RCA uses a natural representation of data in the form of tables that constitute a relational context family. Some of these tables represent objects of several categories described by binary attributes (*formal contexts*) while the other tables represent relations between objects from the categories (*relational contexts*). We illustrate RCA with an example including a single formal context (K_{nature}, see Fig.1) and two relational contexts, R_{eat} and R_{live}, shown in Figures 3 and 4.

	Thing	Plant	Place	Animal
berry		X		
mountain			X	
sheep				X
lichen		X		
wolf				X
rabbit				X
bear				X
herb		X		

Fig. 1. Formal context K_{nature}

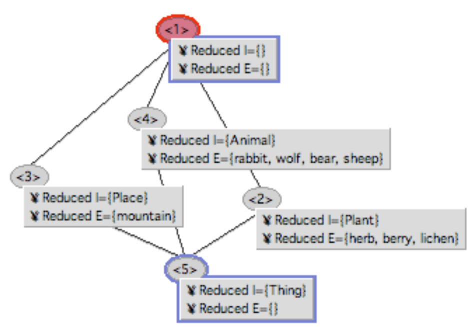


Fig. 2. Concept Lattice L_{nature}

	berry	sheep	lichen	rabbit	herb
berry					
mountain					
sheep					X
lichen					
wolf		X		X	
rabbit					X
bear	X	X	X		
herb					

Fig. 3. Relational context R_{eat}

	mountain
berry	
mountain	
sheep	X
lichen	
wolf	X
rabbit	X
bear	X
herb	

Fig. 4. Relational context R_{live}

Definition 1 (Relational Context Family (RCF)). *A Relational Context Family $\mathcal{R}$ is a pair (K, R). K is a set of formal contexts $K_i = (O_i, A_i, I_i)$, R is a set of relational contexts $R_j = (O_k, O_l, I_j)$ (O_k and O_l are the object sets of the contexts K_k and K_l of K).*

New abstractions emerge iterating the two following steps. The first step is classical concept lattice construction. In second step, formal contexts are concatenated with relational contexts enhanced by concepts created in previous lattice construction.

Initialisation step. Lattices are built at this step using FCA. For each formal context K_i, a lattice $\mathcal{L}_i^0$ is created (in our example, it is shown in Fig. 2).

Step n+1. For each relational context $R_j = (O_k, O_l, I_j)$, an enhanced relational context $R_j^s = (O_k, A, I)$ is created. A is the concept set of the lattice $\mathcal{L}_l^n$ (created at step n). In the case of R_{eat}, we obtain $R_{eats}^s = (O_{nature}, L_{nature}, I)$. I contains the set of pairs (o, a) s.t. $S(R(o), Extent(a))$ is *true*, where S is a *scaling* operator. We consider here two scaling operators: $S_\exists(R(o), Extent(a))$, which is *true* iff $\exists x \in R(o), x \in Extent(a)$, and $S_{\forall\exists}(R(o), Extent(a))$, which is *true* iff $\forall x \in R(o), x \in Extent(a) \wedge \exists x \in R(o), x \in Extent(a)$.

We give a first example using $S_\exists$ to compute R_{eat}^s. Initialisation step allows us to discover the abstraction represented by the concept C_2 (plants). As we have $(berry) \in R_{eat}(bear)$ with $berry \in Extent(C_2)$, $(bear, C_2) \in I$. For similar reasons, $(rabbit, C_2) \in I$. This highlights the fact that bears and rabbits eat at least one kind of plant.

Now we examine a computation based on the scaling operator $S_{\forall\exists}$. As $(sheep) \in R_{eat}(bear)$ and $sheep \notin Extent(C_2)$, now $(bear, C_2) \notin I$. Reversely, since $R_{eat}(rabbit) = \{herb\} \subseteq Extent(C_2)$, we still have $(rabbit, C_2) \in I$. This indicates that rabbits only eat plants, while bears do not only eat plants: but also sheep.

Applying FCA to $K_k \cup \{R_j^s = (O_k, A, I)\}$ creates new concepts that are added to $\mathcal{L}_k^n$ to obtain $\mathcal{L}_k^{n+1}$. For example, still using the scaling operator $S_{\forall\exists}$ on the concatenation of K_{nature}, R_{eat}^s and R_{live}^s (Fig. 5), we obtain the concept

$(\{sheep, rabbit\}, \{Animal, eat : C1, eat : C2, live : C1, live : C3\})$. This concept represents objects that are animals, eat only plants and live only in places (here mountain due to the very restricted example). As the process goes on, more and more complex information on relational structuring emerges. The process stops when lattices at step n are equivalent to those at step $n-1$ *i.e.* when no new concept appear.

	K_{nature}				R^s_{eat}					R^s_{live}				
	Thing	Plant	Place	Animal	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
berry		X												
mountain			X											
sheep				X	X	X				X		X		
lichen		X												
wolf				X	X			X		X		X		
rabbit				X	X	X				X		X		
bear				X	X					X		X		
herb		X												

Fig. 5. Context K_{nature} concatenated with enhanced relational contexts R^s_{eat} and R^s_{live}.

Description logics Description logics [13] allow knowledge representation with *Concepts*, *Individuals* and *Roles*. Concepts, included in the terminological box (or TBox), are primitive ones (like $Plant$, $Animal$), constants ($\top$, $\bot$) or defined using several constructors, such as negation ($\neg$), disjunction ($\sqcup$), or conjunction ($\sqcap$). Here we are especially interested in constructors composed with roles: universal role quantification ($\forall R.C$, where R is a role and C is a concept) and existential quantification ($\exists R.C$). $\mathcal{FL}^{-}\mathcal{E}$ is the description logic we will consider in the paper. If *eat* is a role, the concept $Herbivorous$ can be defined with expression $Herbivorous := Animal \sqcap \exists eat.\top \sqcap \forall eat.Plant$, since an herbivorous is an animal which eats at least one thing and which only eats plants. Assertion box (or ABox) contains instanciations. An individual (for example *herb*) is defined by its type (for example $Plant(herb)$), which is a concept of the TBox, while a role is defined by a set of individual pairs like $eat(rabbit, herb)$.

3 Adapting RCA-MDE to bi-level models

RCA in a model-driven engineering approach Lessons learned from previous prototypes [14,15,16] highlighted the need to easily encode data from a large range of models (UML class models in several UML versions, component models, description logics, etc.) into relational context families as well as to parameterise RCA application. The RCA-MDE approach proposes a generic solution to these issues [1,2]. An overview is given in Figure 6.

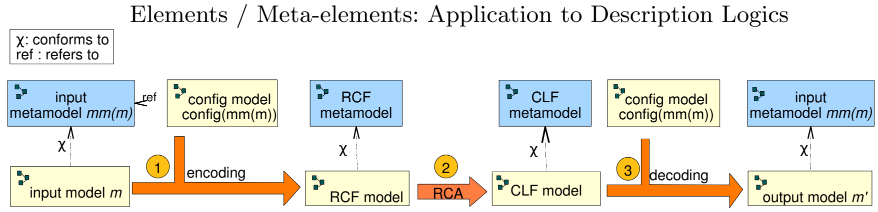


Fig. 6. RCA-MDE approach

To apply RCA to a model m, one just need to provide the corresponding metamodel *mm(m)* and the RCA configuration *config(mm(m))* for this metamodel. *config(mm(m))* defines the meta-elements of *mm(m)* which are considered for analysis. Converting the model into a relational context family is done in two steps. In the first step, a formal context is created for each meta-class indicated in *config(mm(m))*. Binary attributes are the meta-class attributes mentioned in *config(mm(m))*. A specialization/generalization link can be specified for a given meta-class. This link allows to compute the inherited relational attributes. In the second step, a relational context is created for every meta-relation indicated in *config(mm(m))*.

Until now, RCA-MDE has been applied to models owning entities of a single level, i.e. that do not mix entities and meta-entities. In this paper, we show how to apply RCA-MDE on bi-levels models. Such models can be found in models representing instances like ODM [6] that allows to represent ontologies, or UML instance diagrams [4]. We illustrate the approach with description logics : we aim at refactoring models owning individuals based on classes. Those models are composed of a TBox and an ABox (TAB models). We show in this section why a naive adaption based on the one applied for mono-level models is surprisingly not well-suited, and propose an original way to correctly apply it.

Naive adaptation, based on the mono-level modeling To apply RCA to a TAB model, the first (naive) adaptation consists in providing to RCA-MDE a metamodel where coexist: classes, indivuals, relations, and instances of relations, as illustrated at the right of Figure 7. Note that we work with a simplified metamodel of description logics. The model of animals is thus an instance of this metamodel, an excerpt is given at the left of Figure 7, where we only see the animals that live in the moutain.

We also need to provide a configuration model for this metamodel. All the entities of the metamodel (*Class*, *Individual*, *Instance of Relation* and *Relation*) correspond to a formal context. The inter-entity links give rise to relational contexts, and for each relational context, the scaling operator is chosen and defined in the configuration model. We take into account the following associations:

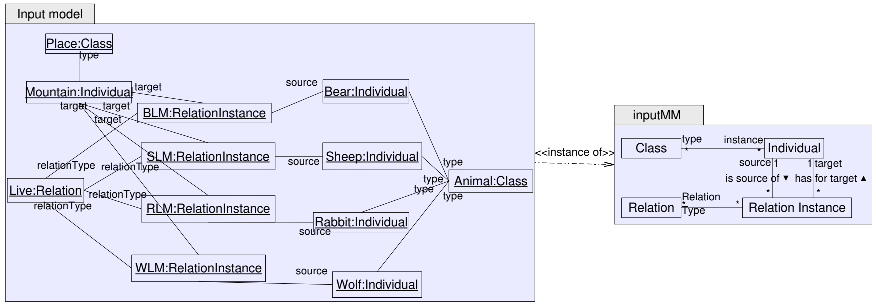


Fig. 7. Naive adaptation for the model of animals.

	ir0	ir1	ir2	ir3	ir4	ir5	ir6	ir7	ir8	ir9	ir10
berry											
mountain											
sheep	X	X									
lichen											
wolf			X	X	X						
rabbit						X	X				
bear								X	X	X	X
herb											

	berry	mountain	sheep	lichen	wolf	rabbit	bear	herb
ir0								X
ir1		X						
ir2						X		
ir3			X					
ir4		X						
ir5								X
ir6		X						
ir7				X				
ir8	X							
ir9			X					
ir10		X						

	eat	live
ir0	X	
ir1		X
ir2	X	
ir3	X	
ir4		X
ir5	X	
ir6		X
ir7	X	
ir8	X	
ir9	X	
ir10		X

Fig. 8. Relational contexts of relations *is source of* (lhs), *has for target* (center) and *relation type* (rhs).

– *type*[1] linking *Individual* to *Class*. The associated scaling operator is $S_{\exists}$. The relational context R_{type} will thus be created, associated to the $S_{\exists}$ operator;
– *is source of* linking *Individual* to *Relation Instance* (it leads to generate the $R_{issourceof}$ context). We chose here the $S_{\forall\exists}$ operator;
– *has for target* linking *Relation Instance* to *Individual* (it leads to generate the $R_{hasfortarget}$ context). We chose here the $S_{\exists}$ operator;
– *relation type* linking *Relation Instance* to *Relation* (it leads to generate the $R_{typerelation}$ context). We chose here the $S_{\exists}$ operator.

We thus have a single formal context for all the instances of relations. The relation context *relation type* represents the links between those instances of relations and the relations (see Figure 8). Yet if abstractions can be found with this configuration, others cannot, that could however be discovered applying RCA in a classical way, i.e. filling the contexts without taking into account the way input data are modeled. The problems arise when using a scaling operator different from $S_{\exists}$. For example, let us refer to animals linked to their habitation and their diet. Using RCA, we hope obvious abstractions to appear with operator $S_{\forall\exists}$, e.g. herbivorous (animals such that, whatever they eat, it is plant), carnivorous (animals such that whatever they eat, it is animal) and omnivorous (animals eating both vegetal and animal food). However, all the relation instances (links) are

[1] *type* is in fact a role of this association.

represented by the same metaclass in the metamodel and they belong to a single formal context. As a first consequence we cannot apply different scaling operators to the relations (e.g. $S_{\forall\exists}$ for *eat* and $S_{\exists}$ for *live*). As a second consequence, some concepts built by original RCA (on relational context family like in Section 2) cannot be found. While original RCA builds a concept including *sheep* and *rabbit* (because all *eat* links end at *herb* which is included in C_2 extent), naive modeling of RCA-MDE cannot (because all links ends are not included in a non trivial concept extent : *eat* links go towards *herb* while *live* links go towards *mountain*).

The source of the problem comes from the following causes. First, our approach, due to a genericity matter, creates the formal contexts from the elements of a metamodel. Second, we use a model with two levels of abstraction, and the instanciation relation allowing to go from one level to the other is defined in the metamodel by a simple association. In our case, it clearly appears that an instanciation relation exists between the relations and the instances of relations, but the relations do not belong to the metamodel, thus it is not possible to create a formal context for each relation. We thus propose in the next section a more complex yet more adequate solution.

Adaptation for bi-levels models: hybridisation of the input metamodel with the input model We propose to *promote* a part of the model, i.e. to move the relations in the metamodel, in the form of relation of model, and to transform the relation *relation type* into an instanciation relation. We thus hybridise a part of the input model with a part of the input metamodel, as illustrated in Figure 9. The input model is then also modified as shown in the right of Figure 9. The hybridisation transformation aims at deleting the reification of the relations in a model (by the concept *Instance of relation*) since it is not relevant to look for new abstractions.

The configuration model follows the same idea as the transformation: we only take into account two entities of the metamodel: *Class* and *Individual*, and the following relations:

- *type* linking *Individual* to *Class*. We associate it the scaling operator $S_{\exists}$. The relational context R_{type} will thus be created, associated to the operator $S_{\exists}$;
- *live* linking *Individual* to *Individual*, with a scaling operator (e.g. $S_{\forall\exists}$).
- *eat* linking *Individual* to *Individual*, with a scaling operator (e.g. $S_{\forall\exists}$).

This solution does not imply to modify the RCA-MDE process, we only modify input models. Those modifications result in a relational context of each relation of the input model (in our example: exactly the contexts of Figures 3 and 4). Reading the tables is easier, because the instances of relations are represented by a table per relation, and this table owns the source and the target of the instance of relation.

Hybridising the model has then for consequence to obtain back the model elements leveling while still keeping the same information on the input model ; and the relations of the input model that were semantically relations on instances become actual relations on instances. The hybrid model with its metamodel is then a mono-level model on which the RCA-MDE process can be applied in a

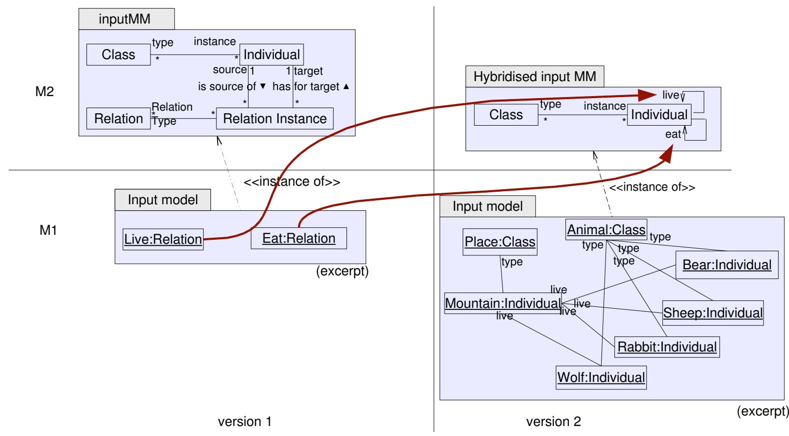


Fig. 9. Hybridisation of the metamodel.

classical way. The new definitions of concepts in the context of the individuals will depend on a single relation. In our example, we will be able to define a concept to which *sheep* belongs and which gathers all the individuals that only eat plants, independently from the other types of relations like *live*. This modeling corresponds to what can be obtained with a classical application of RCA.

4 Platform and experimentations

Platform The platform implementing RCA-MDE uses the modeling framework EMF [17]. It is based on the meta-modeling language Ecore to read models and their metamodels.

To represent description logics models, we use the OWL language (Web Ontology Language, [18]). OWL is intensively used in Web technologies as knowledge representation language, and many modeling tools exist for OWL, as well as many OWL models. Our metamodel is included, renaming the entities, in the one proposed in the Eclipse plugin [19] that allows to handle OWL models with EMF. In this way, a TAB model can easily be translated into an OWL model, renaming `Class` into `OWLClass`, etc.

To adapt the RCA-MDE platform to TAB models, keeping in mind that we want the process to remain generic (adaptable to other input data), we have to define the hybridisation transformation of the OWL metamodel for a given TAB model (see Figure 10), and a configuration making explicit which meta-elements have to be taken into account by the RCA. The hybridisation transformation is fully automated, it is written in Java and uses EMF to handle models. It takes as input the OWL metamodel, a TAB model and the configuration model and generates both the hybridised metamodel, the new TAB model conform to the

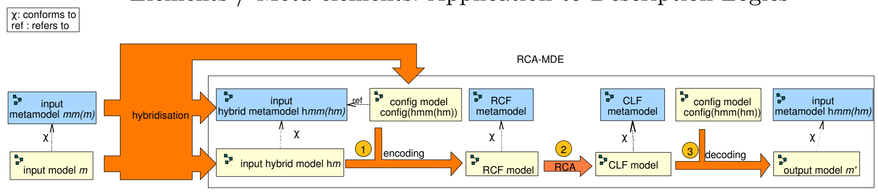


Fig. 10. Illustration of the metamodel hybridisation in RCA-MDE.

hybridised metamodel, and the corresponding configuration. As shown in Figure 9, the hybridisation transformation lists all the relations and puts them up a level of abstraction higher, so that they appear in the hybridised metamodel. This transformation only modifies the *relations* and the *instances of relation*. The transformation also impacts the configuration model, since it refers to a given metamodel. The metamodel being hybridised, the configuration must also be hybridised. The obtained configuration can then be modified by the final user, in particular for the choice of the scaling operators.

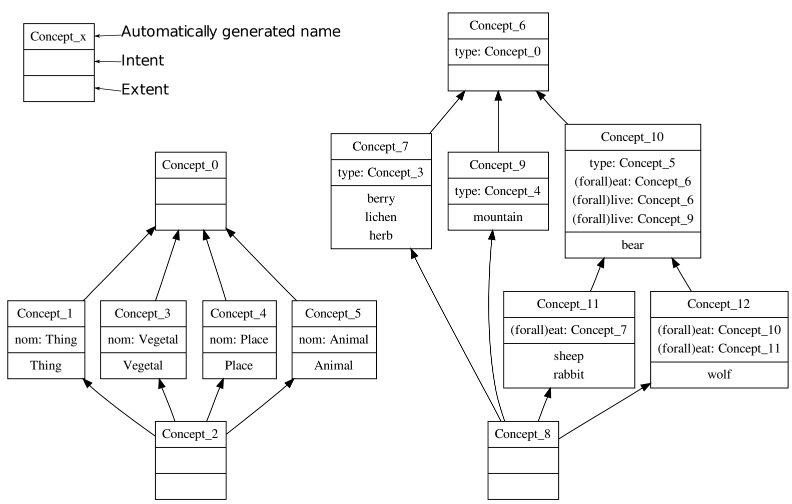


Fig. 11. Lattices obtained applying RCA-MDE on the example of animals. The class lattice is on the lhs, the Individual lattice on the rhs.

Figure 11 shows, using Hasse diagrams, the lattices obtained applying RCA-MDE on the example of animals. The lattice we focus on is the one obtained from the context of the *Individuals*. We see that concepts have been created to group the elements of same type: *Concept_7* groups the elements of type *vegetal*, *Concept_9* groups those of type *place* and *Concept_10* groups those of type *animal*. *Concept_11* groups animals that eat only vegetals. *Concept_12* groups the animals that eat only animals, in particular animals of *Concept_11*. From the intent of the new concepts, we can infer in an automatic way a logical formula defining them.

The platform resulting from this work allows us to analyse bi-level models in an automatic way: as well as for the analysis of mono-level models, we just need to configure the RCA with a configuration model, so we kept the generic approach. Our experiments showed us that the configuration for bi-level models is slightly more complex since it frequently handles several scaling operators.

Table 1. Models studied. The Ontology *Model* is about surface water and water quality model , the Ontology *Autos* is about Automobiles and their equipment. We take into account the number of *Individuals*, *Classes*, *Object Properties* and their instances we called *Links* in this table. We compare the number of Concepts obtained using FCA and RCA in the lattice from *Individual* Context.

Ontology name	Individuals	Classes	Object Properties	Links	FCA concepts	RCA concepts
Model[2]	114	20	8	273	31	65
Autos[3]	321	91	13	375	82	175

We have tested our approach of hybridisation on some real ontologies in OWL format (Tab.1). We compare the result[4] obtained using Formal Concept Analysis (we take the results obtained after the first iteration of the RCA process) to those obtained using Relational Concept Analysis. In the following we will concentrate on the lattice from *Individual* context, as the lattice from the *Class* lattice cannot produce new concepts (there is no relation in the metamodel with *Class* elements as source).

On the *Model* Ontology, FCA would generate 31 concepts. Nearly each class is transformed in one concept, except for 3 classes which have the same extent. Our approach generated 55 concepts using a $S_{\exists}$ operator of scaling and 65 concepts using a $S_{\exists}$ and a $S_{\forall\exists}$ operator of scaling. From the concepts obtained by FCA, 7 of them have a more precise intent with RCA.

We note that concept intents here describe the presence of relation between individuals more than a value of the relation. This is caused by the fact that a lot of classes do not have subclasses.

The use of $S_{\forall\exists}$ scaling in addition with $S_{\exists}$ scaling gives us a way to know the most specialized type for the target of a relation. To deduce the type of a relation target, you need to know the target type of all the instances of this relation.

[2] http://loki.cae.drexel.edu/~wbs/ontology/model.htm
[3] http://gaia.isti.cnr.it/~straccia/Teaching/IS/2007/Exercises/autos.owl
[4] all the lattices can be found at http://www.lirmm.fr/~dolques/publications/data/cla08

Elements / Meta-elements: Application to Description Logics

The RCA approach applied on bi-level models produces a different type of result from applied on mono-level. On bi-level models we take a set of elements, and the process produces new subsets depending on the properties of the elements. On the examples we studied in OWL, the partitioning depends on the type of the individuals, and the Object Property (name of relations in OWL vocabulary) of which they are sources. For instance FCA can produce a concept that groups the individuals of type *ModelingSystem* and that are linked by the Object Property *hasModel* to another individual. RCA shows us that the individual of this concept are all linked by the ObjectProperty *hasModel* to Individuals of type *Model* which all have an Individual of type *Organisation* as a developer (concept `NMWithAv_Dev_Org_ModelDim`).

This kind of result could help to complete under-specified ontologies in an approach by-example, by showing which kind of data is missing in individual description and by restricting the domain and range of an Object Property.

5 Conclusion

Until now, building abstractions using an RCA process and an MDE paradigm has been applied to models with a single level of abstraction, i.e. models that do not mix entities and meta-entities. In this paper, we have studied how to use an RCA-MDE approach for bi-level models, where co-exist meta-entities and their instances. We based our work on the example of description logics: we focused on models mixing individuals and links between individuals with the classes typing the individuals and the relations defining the links. We have shown that a direct application of the approach does not give the expected results, and proposed a more complex solution giving results similar to those obtained with an application of RCA with a manual encoding of data. This solution is based on the promotion of the instances of relations from the input model up to the input metamodel.

In this paper, we worked with description logics with minimal expressivity; in particular we do not take into account language elements existing in OWL like the specialization relation that could be applied to classes, or the constraints that could be added on the sources and targets of the relations. Future work will consist in dealing with more complex description logics, as well as in integrating the obtained results in the original model. We also plan to make the hybridisation generic instead of specific to our description logics metamodel, in order to validate our approach with other kinds of bi-level models such as UML models with classes and instances.

References

1. Arévalo, G., Falleri, J.R., Huchard, M., Nebut, C.: Building abstractions in class models: Formal concept analysis in a model-driven approach. In: Proc. of the MoDELS'06 conference. (2006) 513–527
2. Falleri, J.R., Huchard, M., Nebut, C., Arévalo, G.: Use of Model Driven Engineering in Building Generic FCA/RCA Tools. In Diatta, J., Eklund, P., Liquière,

M., eds.: Proc. of CLA'07: Fifth International Conference on Concept Lattices and Their Applications. (2007) 225–236
3. Dolques, X., Falleri, J.R., Huchard, M., Nebut, C.: Adaptation d'un processus de construction d'abstractions basé idm à des modèles bi-niveaux éléments / méta-éléments : Application aux logiques de description. In: LMO Conference. (2008)
4. OMG: UML 2.0 superstructure. Technical report, Object Management Group (2004)
5. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. Technical report, W3C (2004)
6. OMG: Ontology definition metamodel. Technical report, Object Management Group (2007)
7. Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Relational concept discovery in structured datasets. Ann. Math. Artif. Intell. **49**(1-4) (2007) 39–76
8. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations. Springer, Berlin (1999)
9. Priss, U.: Classification of meronymy by methods of relational concept analysis. In: Online Proceedings of the 1996 Midwest Artificial Intelligence Conf., Bloomington, Indiana. (1996)
10. Prediger, S., Wille, R.: The lattice of concept graphs of a relationally scaled context. In: Proc. of the 7th Intl. Conf. on Conceptual Structures (ICCS'99), Springer (1999) 401–414
11. Ganter, B., Kuznetsov, S.: Pattern structures and their projections. In Delugach, H., Stumme, G., eds.: Conceptual Structures: Broadening the Base, Proc. of the 9th Intl. Conf. on Conceptual Structures (ICCS'01), Stanford, CA. Volume 2120 of LNCS., Springer (2001) 129–142
12. Ferré, S., Ridoux, O., Sigonneau, B.: Arbitrary relations in formal concept analysis and logical information systems. In Dau, F., Mugnier, M.L., Stumme, G., eds.: ICCS. Volume 3596 of Lecture Notes in Computer Science., Springer (2005) 166–180
13. Baader, F., Nutt, W.: Basic description logics. In Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003) 43–95
14. Valtchev, P., Grosser, D., Roume, C., Hacene, M.R.: Galicia: an open platform for lattices. In B. Ganter, A.d.M., ed.: Using Conceptual Structures: Contributions to ICCS'03, Aachen (DE), Shaker Verlag (2003) 241–254
15. Dao, M., Huchard, M., Hacene, M.R., Roume, C., Valtchev, P.: Towards Practical Tools for Mining Abstractions in UML Models. In Manolopoulos, Y., Filipe, J., Constantopoulos, P., Cordeiro, J., eds.: ICEIS (3). (2006) 276–283
16. Seuring, P.: Design and implementation of a UML model refactoring tool. Master's thesis, Hasso-Plattner-Institute for Software Systems Engineering at the University of Postdam (2005) http://www.lirmm.fr/~huchard/Documents/Papiers/PhilippSeuringMasterThesis.pdf.
17. Budinsky, F., Grose, T., Steinberg, D., Ellersick, R., Merks, E., Brodsky, S.: Eclipse Modeling Framework: a developer's guide. Addison-Wesley Professional (2003)
18. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language Overview. Technical report, W3C (2004)
19. EODM: site du projet EODM, http://www.eclipse.org/modeling/mdt/?project=eodm

Extending Attribute Exploration by Means of Boolean Derivatives

J. Antonio Alonso-Jiménez[1], Gonzalo A. Aranda-Corral[1], Joaquín Borrego-Díaz[1], and M. Magdalena Fernández-Lebrón[2], M. José Hidalgo-Doblado[1]

[1]Departamento de Ciencias de la Computación e Inteligencia Artificial
[2]Departamento de Matemática Aplicada I
E.T.S. Ingeniería Informática, Universidad de Sevilla, Avda. Reina Mercedes s.n.
41012-Sevilla, Spain

Abstract. We present a translation of problems of Formal Context Analysis into ideals problems in $\mathbb{F}_2[\mathbf{x}]$ through the Boolean derivatives. The Boolean derivatives are introduced as a kind of operators on propositional formulas which provide a complete calculus. They are useful to refine stem basis as well as for extending attribute exploration.

1 Introduction

Attribute exploration (cf. [3]) is a family of interactive procedures for Knowledge Acquisition (KA) in Formal Concept Analysis (FCA), whose goal is to build a knowledge base of the attributes we are working with. The procedures used in FCA have nice computer implementations, existing even generalizations for the management of the background information. Sometimes attribute exploration is hard or tedious to apply. Thus, it may advisable to use automated tools. Many Computer Algebra Systems (CAS) provide tools for working with discrete data, for example, Gröbner basis. Since it is possible to translate entailment problems into ideal problems in finite fields, Gröbner basis is a powerful tool for reasoning in propositional logic [8, 9, 2].

Our aim is to extend the framework of attribute exploration through the introduction of *Boolean derivatives* and the assistance of a CAS. The CAS that we will use CoCoA (`http://cocoa.dima.unige.it/`), very well suited for our pourposes because of its easy management of Gröbner basis and related tools. The paper is organized as follows. The next section reviews the relationship between propositional logic and the ring $\mathbb{F}_2[\mathbf{x}]$, as well as the basics of FCA. In the third section the *Boolean derivatives* are introduced, as well as a complete polynomial calculus based on them. An algebraic characterization of sensitivity for implications in FCA is given in forth section. In fifth and sixth sections new versions of attribute exploration are introduced, and in section 7 an application to graph theory is given. We conclude with some remarks about future work.

2 Background

We assume that the reader is familiar with propositional logic and polynomial algebra on positive characteristics. We setup a propositional language $PV =$

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 121–132,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

$\{p_1, \ldots, p_n\}$, $PForm$ will denote the set of propositional formulas, and $var(F)$ denotes the set of variables of the propositional formula F.

The ring in which we are working is $\mathbb{F}_2[\mathbf{x}]$ (where $\mathbf{x} = x_1, \ldots, x_n$). A key ideal is $\mathbb{I}_2 := (x_1 + x_1^2, \ldots, x_n + x_n^2)$. To clarify our proposition, let fix an identification $p_i \mapsto x_i$ (or $p \mapsto x_p$) between PV and the set of indeterminates.

Given $\alpha = (\alpha_1, \ldots \alpha_n) \in \mathbb{N}^n$, let us define $|\alpha| := \max\{\alpha_1, \ldots, \alpha_n\}$, and $sg(\alpha) := (\delta_1, \ldots, \delta_n)$, where δ_i is 0 if $\alpha_i = 0$ and 1 otherwise. The *degree of* $a(\mathbf{x}) \in \mathbb{F}_2[\mathbf{x}]$, is $\deg_\infty(a(\mathbf{x}))$:=max$\{|\alpha| \,:\, \mathbf{x}^\alpha$ is a monomial of $a\}$, and $\deg_i(a(\mathbf{x}))$ is the degree w.r.t. x_i. If $\deg_\infty(a(\mathbf{x})) \leq 1$, $a(\mathbf{x})$ is called a *polynomial formula*.

Three maps represent the relationship between propositional logic and $\mathbb{F}_2[\mathbf{x}]$:

- $\Phi : \mathbb{F}_2[\mathbf{x}] \to \mathbb{F}_2[\mathbf{x}]$ is defined by $\Phi(\sum_{\alpha \in I} \mathbf{x}^\alpha) := \sum_{\alpha \in I} \mathbf{x}^{sg(\alpha)}$.
- The map $P : PForm \to \mathbb{F}_2[\mathbf{x}]$ is defined by the following equations
  - $P(\bot) = 0, P(p_i) = x_i, P(\neg F) = 1 + P(F)$
  - $P(F_1 \wedge F_2) = P(F_1) \cdot P(F_2)$ and $P(F_1 \vee F_2) = P(F_1) + P(F_2) + P(F_1)P(F_2)$
  - $P(F_1 \to F_2) = 1 + P(F_1) + P(F_1)P(F_2)$, and
  - $P(F_1 \leftrightarrow F_2) = 1 + P(F_1) + P(F_2)$
- $\Theta : \mathbb{F}_2[\mathbf{x}] \to PForm$ is defined by
  - $\Theta(0) = \bot,\ \Theta(1) = \top,\ \Theta(x_i) = p_i$,
  - $\Theta(a \cdot b) = \Theta(a) \wedge \Theta(b)$, and $\Theta(a + b) = \neg(\Theta(a) \leftrightarrow \Theta(b))$.

We have that $\Theta(P(F)) \equiv F$ and $P(\Theta(a)) = a$. Since we shall frequently be applying $\Phi \circ P$, we define the *polynomial projection* as $\pi := \Phi \circ P$.

Regarding valuations and polynomials, the key fact is that, if $v : PV \to \{0, 1\}$ is a valutation with $v(p_i) = \delta_i$, then for every $F \in PForm$,

$$v(F) = P(F)(\delta_1, \ldots \delta_n)$$

The behaviour of the ideals of $\mathbb{F}_2[\mathbf{x}]$ is well known: If $A \subseteq (\mathbb{F}_2)^n$, then $V(I(A)) = A$, and for every $\mathfrak{I} \in Ideals(\mathbb{F}_2[\mathbf{x}])$, $I(V(\mathfrak{I})) = \mathfrak{I} + \mathbb{I}_2$. Therefore $F \equiv F'$ if and only if $P(F) = P(F') \pmod{\mathbb{I}_2}$ which is also equivalent to $\Phi \circ P(F) = \Phi \circ P(F')$. The following theorem states the main relationship between propositional logic and $\mathbb{F}_2[\mathbf{x}]$:

Theorem 1. *The following conditions are equivalent:*
(1) $\{F_1, \ldots, F_m\} \models G$.
(2) $1 + P(G) \in (1 + P(F_1), \ldots, 1 + P(F_n)) + \mathbb{I}_2$.
(3) $\texttt{NF}(1 + P(G), \texttt{GB}\,[(1 + P(F_1), \ldots, 1 + P(F_m)) + \mathbb{I}_2]) = 0$.
(where `GB` *denotes* Gröbner basis*) and* `NF` *denotes* normal form*.*

In the rest of this section we succintly present some elements of FCA we use, although we assume that the reader knows the basic principles of this theory (the fundamental reference is [3]). We represent a *formal context* as $M = (O, A, I)$, which consists of two sets, O (the *objects*) and A (the *attributes*) and a relation $I \subseteq O \times A$. Finite contexts can be represented by a 1-0-table (representing I as a Boolean function on $O \times A$). The main goal in FCA is the computation of the *concept lattice* associated to the context.

Basic logical expressions in FCA are *implication between attributes*, that is, pair of sets of attributes written as $Y_1 \to Y_2$. Truth with respect to $M = (O, A, I)$ is defined as follows. A subset $T \subseteq A$ *respects* $Y_1 \to Y_2$ if $Y_1 \not\subseteq T$ or $Y_2 \subseteq T$. We say that $Y_1 \to Y_2$ *holds* in M ($M \models Y_1 \to Y_2$) if for all $o \in O$, the set $\{o\}'$ respects $Y_1 \to Y_2$. In that case we say that $Y_1 \to Y_2$ is an *implication* of M.

From a propositional logic viewpoint, $Y_1 \to Y_2$ is the formula $\bigwedge Y_1 \to \bigwedge Y_2$, so it is equivalent to a set of Horn clauses (implications with a singleton as right-hand side). On the other hand, the definition of truth can be extended: Given $Y \subseteq A$, define $\neg Y := Y \to \bot$, and it holds in the context if for all $o \in O$, $Y \not\subseteq \{o\}'$. Given a formula written with $\{\to, \bot\}$, $M \models F$ can be defined in the natural way. Since this set of connectives is functionally complete, truth definition can be extended to $PForm$.

Definition 1. *Let $\mathcal{L}$ be a set of implications and L an implication of M.*
- L follows from $\mathcal{L}$ *($\mathcal{L} \models L$) if each subset of A respecting $\mathcal{L}$ also respects L.*
- $\mathcal{L}$ *is* closed *if every implication following from $\mathcal{L}$ is already in $\mathcal{L}$.*
- $\mathcal{L}$ *is* complete *if every implication of the context follows from $\mathcal{L}$.*
- $\mathcal{L}$ *is* non-redundant *if for each $L \in \mathcal{L}$, $\mathcal{L} \setminus \{L\} \not\models L$.*
- $\mathcal{L}$ *is a* stem basis *for M if it is complete and non-redundant.*

For every context we can obtain a stem basis from the pseudo-intents:

Theorem 2. *[7] The set $\mathcal{L} = \{Y \to Y'' \ : \ Y$ is a pseudointent$\}$ is a stem basis.*

Actually one can choose $Y \to Y'' \setminus Y$ instead of $Y \to Y'$, so we will assume, by default, that for every implication $Y_1 \to Y_2$ belonging to a stem basis Y_1 and Y_2 are disjoint. Such a basis for the example of figure 5 (left) is $\mathcal{L} = \{\emptyset \to N, \{N, A\} \to \{Mo\}, \{N, Le\} \to \{Mo\}\}$.

The called *Amstrong rules* facilitates implicational reasoning:

$$R1: \ \frac{}{X \to X} \qquad R2: \frac{X \to Y}{X \cup Z \to Y} \qquad R3: \ \frac{X \to Y, \ Y \cup Z \to W}{X \cup Z \to W}$$

It has that A set of implications $\mathcal{L}$ is closed if and only if the set is closed by Amstrong rules [1]. A consequence of Amstrong result is that, if $\vdash_A$ denotes the proof notion associated to Amstrong rules, stem basis are $\vdash_A$-complete, that is:

Theorem 3. *Let $\mathcal{L}$ be a stem basis for M, and L an implication. Then $M \models L$ if and only if $\mathcal{L} \vdash_A L$*

The computing of stem basis may be expensive if the set of objects is large. Even it is possible we do not have the complete context M, or it has a potentially infinite set of objects. *Attribute exploration* is an interactive procedure designed to obtain a stem basis starting with a set H of *good examples* generating the subcontext

$$M \upharpoonright_H := (H, A, I \cap (H \times A))$$

One expects that a stem basis associated to $M \upharpoonright_H$ is also a stem basis for the complete context. To guarantee it, we proceed as follows. Assume that $\mathcal{L} =$

1. *Compute pseudo-intent*: Find X a pseudo-intent for $M\upharpoonright_H$.
2. *Soundness of the new implication*: Ask to the user $X \stackrel{?}{\to} X''$ (the operators $'$ are w.r.t. the subcontext). The user must react:
   - Confirming the suggested implication (adding it to $\mathcal{L}$), or
   - giving o (a counterexample) such that $\{o\}'$ does not respect the implication. This is added to H, and the implication is discarded.

Fig. 1. Attribute exploration

$\{L_1, \ldots, L_k\}$ is a partial set of implications accepted as true, built from pseudo-intents of $M\upharpoonright_H$[1]. Attribute exploration consists in a loop of the two steps shown in fig. 1, and it stops when no new pseudointent is found (see [4] for variants).

3 Boolean derivatives and non-clausal theorem proving

We introduce an operator on propositional formulas as a translation of the usual derivation on $\mathbb{F}_2[\mathbf{x}]$. In this section we review its basic properties (from [2]).

Recall that a derivation on a ring R is a map $d : R \to R$ verifying that $d(a+b) = d(a) + d(b)$ and $d(a \cdot b) = d(a) \cdot b + a \cdot d(b)$

Definition 2. *A map $\partial : PForm \to PForm$ is a* Boolean derivation *if there exists a derivation d on the ring $\mathbb{F}_2[\mathbf{x}]$ such that $\partial = \Theta \circ d \circ \pi$*

If the derivation on $\mathbb{F}_2[\mathbf{x}]$ is $d = \frac{\partial}{\partial x_p}$, we denote ∂ as $\frac{\partial}{\partial p}$. It has that:

$$\frac{\partial}{\partial p} F \equiv \neg(F\{p/\neg p\} \leftrightarrow F)$$

Thus, the value of $\frac{\partial}{\partial p}F$ with respect to a valuation does not depend on p. Therefore, we can apply valuations on $PV \setminus \{p\}$ to this formula.

Definition 3. *The* independence rule *(or ∂-rule) on polynomial formulas is*

$$\partial_x(a_1, a_2) : \quad \frac{a_1,\ a_2}{1 + \Phi\left[(1 + a_1 \cdot a_2)(1 + a_1 \cdot \frac{\partial}{\partial x} a_2 + a_2 \cdot \frac{\partial}{\partial x} a_1 + \frac{\partial}{\partial x} a_1 \cdot \frac{\partial}{\partial x} a_2)\right]}$$

In order to simplify the notation, if $a_i = b_i + x_p \cdot c_i$, with $\deg_{x_p}(b_i) = \deg_{x_p}(c_i) = 0$ $(i = 1, 2)$,. Then we can rewrite the values as:

$$\partial_{x_p}(a_1, a_2) : \quad \frac{b_1 + x_p \cdot c_1,\ b_2 + x_p \cdot c_2}{\Phi\left[1 + (1 + b_1 \cdot b_2)[1 + (b_1 + c_1)(b_2 + c_2)]\right]}$$

The rule is symmetric and generalizes resolution of non-tautological polynomial clauses (see lemma 1). For formulas the rule is translated as

$$\partial_p(F_1, F_2) := \Theta(\partial_{x_p}(\pi(F_1), \pi(F_2))).$$

[1] Pseudointents are generated in lexicographic order. This way previously computed pseudointents are preserved by augmentations of H. See th. 27 in [3].

It naturally induces a concept of proof, $\vdash_\partial$. A $\vdash_\partial$-refutation is a proof of $\bot$. In [2] the soundness and the refutational completeness of $\vdash_{partial}$ has been proved

Theorem 4. *[2] Let $v : PV \setminus \{p\} \to \{0,1\}$. The following conditions are equivalent:*

1. *$v \models \partial_p(F_1, F_2)$.*
2. *There exists an extension of v to PV is a model of $\{F_1, F_2\}$.*

For example, $\partial_{x_1}(x_1(1+x_2), x_1(1+x_2)) = 1 + x_2$. So the valuation v s.t. $v(\neg p_2) = 1$ is the only one that we can extend to a model of $p_1 \wedge \neg p_2$. When $\partial_p(\pi(F_1), \pi(F_2)) = 1$, every partial valuation is extendable to a model of $\{F_1, F_2\}$.

Theorem 5. *[2] If Γ is inconsistent then $\Gamma \vdash_\partial \bot$.*

Let be $\partial_p[\Gamma]$ defined as $\partial_p[\Gamma] := \{\partial_p(F, G) \ : \ F, G \in \Gamma\}$.

Given $Q = \{q_1, \ldots, q_k\} \subseteq PV$ the operator $\partial_Q := \partial_{q_1} \circ \cdots \circ \partial_{q_k}$ is well defined modulo logical equivalence (by corollary 4, for every $p, q \in PV$, $\partial_p \circ \partial_q[\Gamma] \equiv \partial_q \circ \partial_p[\Gamma]$). A consequence of corollary 4 and theorem 5 is that entailment can be located on variables of the goal;

Corollary 1. $\Gamma \models F \quad \Longleftrightarrow \quad \partial_{PV\setminus var(F)}[\Gamma] \models F$

We can define an explicit equivalent expression for ∂_p when it is applied to implications. To simplify, suppose that the right-side of implications is a singleton.

Lemma 1. *Let $C_i \equiv \bigwedge Y_1^i \to \bigwedge Y_2^i$ be a implications ($i = 1, 2$, $Y_1^i \cap Y_2^j = \emptyset$), and Γ be a set of implications. Let $\partial_p^c(C_1, C_2)$ be the symmetric operator*

$$\partial_p^c(C_1, C_2) := \begin{cases} \{C_1, C_2\} & p \notin var(C_1) \cup var(C_2) \\ \{C_2\} & p \in Y_1^1,\ p \notin var(C_2) \\ \{\bigwedge Y_1^1 \to \bigwedge (Y_2^1 \setminus \{p\}), C_2\} & p \in Y_2^1,\ p \notin var(C_2) \\ \{\top\} & p \in (Y_1^1 \cap Y_1^2) \cup (Y_2^1 \cap Y_2^2) \\ \{Resolvent_p(C_1, C_2)\} & p \in Y_1^1 \cap Y_2^2 \end{cases}$$

If $\partial_p^c[\Gamma] := \bigcup \{\partial_p^c(C_1, C_2) \ : \ C_1, C_2 \in \Gamma\}$, then $\partial_Q^c[\Gamma] \equiv \partial_Q[\Gamma]$ ($Q \subseteq PV$).

4 Algebraic characterization of sensitive implications

We shall provide an algebraic treatment for implications on a fixed $M = (O, A, I)$. It is well know that every set $X \subseteq (\mathbb{F}_2)^n$ is an algebraic set; that is, there exists $a_X \in \mathbb{F}_2[\mathbf{x}]$ such that $V(a_X) = X$. If $|A| = n$, M is identified with a subset $X(M)$ of $(\mathbb{F}_2)^n$ (each object identified with the 1-0 expresion of its intent). Let $a_M \in \mathbb{F}_2[\mathbf{x}]$ denote a polynomial formula such that $V(a_M) = X(M)$. Since $IV(a_M) = (a_M) + \mathbb{I}_2$, the coordinate ring of M is

$$\mathbb{F}_2[\mathbf{x}]/_{I(X(M))} \cong (\mathbb{F}_2[\mathbf{x}]/_{(a_M)})/_{\mathbb{I}_2}$$

One might also use an ideal J_X such that $V(J_X) = X$, if it is better to work with them (for example using CoCoA's command `IdealsofPoints`). Thus we can assume that $\mathbb{I}_2 \subseteq J_M$. We choose a_M only to simplify the proofs.

Also, each $o \in O$ defines a valuation v_o defined by: $v_o(p_i) = 1$ iff oIp_i.

Proposition 1. *Let $F \in PForm$ and let $\mathcal{L}$ be a stem basis. The following conditions are equivalent:*
(1) $M \models F$.
(2) $1 + \pi(F) \in (a_M) + \mathbb{I}_2$.
Moreover, if F is an implication, they are also equivalent to
(3) $\{P(L) \ : \ L \in \mathcal{L}\} \cup \{1 + \pi(F)\} \vdash_\partial 0$.
(4) $\partial^c_{PV \setminus var(F)}[\mathcal{L}] \models F$

Proof

(1) $\Longleftrightarrow$ (2): If $M \models F$, then $V(a_M) \subseteq V(1 + \pi(F))$. Thus, $IV(1 + \pi(F)) \subseteq IV(a_M)$ hence $1 + \pi(F) \in (a_M) + \mathbb{I}_2$. The converse is similar.
If F is an implication and $M \models F$, then $\mathcal{L} \models F$. Therefore $\mathcal{L} \cup \{\neg F\}$ is inconsistent so by completeness, $\mathcal{L} \cup \{\neg F\} \vdash_\partial \perp$ hence we have (3). The converse is true by soundness. (4) is equivalent to $\mathcal{L} \models F$ by lemma 1.

We now deal with the problem of redundant arguments in implications. In the worst case, the recognizing of redundancy requires a complete exploration of intents. An argument is redundant if it is not *sensitive*:

Definition 4. *A formula F is* sensitive *in p w.r.t. a formal context M if* $M \not\models F\{p/\neg p\} \leftrightarrow F$. *We say that F is sensitive w.r.t. M (or simply sensitive, if M is fixed) if F is sensitive in all its variables.*

Thus, F is not sensitive in p iff $M \models \neg \frac{\partial}{\partial p} F$. In this case, there exists G with var(G) =var$(F) \setminus \{p\}$ such that $M \models F \leftrightarrow G$ (e.g. $F\{p/\perp\}$).

Sensitive implications (also called *proper implications*) have several advantages over implications obtained from pseudo-intents (see [10]). In attribute exploration, sensitivity analysis is justified: it is possible that implications are based on a nonrepresentative set of examples, and thus they can be refined, basically giving witnesses of the role of the arguments in the implication, or making them more precise, removing redundant arguments:

Lemma 2. *Let $L = Y_1 \to Y_2$ be an implication. If $M \models L$ and L is not sensitive in $p \in Y_1$, then $M \models Y_1 \setminus \{p\} \to Y_2$. If $p \in Y_2$, then $M \models \neg Y_1$.*

By default, sensitivity analysis for implications will be always restricted to attributes in the left-hand side.

Proposition 2. *Let $p \in$var(F). The following conditions are equivalent:*
(1) F is sensitive in p w.r.t. M.
(2) $\frac{\partial}{\partial x_p}\pi(F) \neq 0$ *in the coordinate ring of M.*

Proof. (1) $\Longrightarrow$ (2): Assume $v_o \not\models F \leftrightarrow F\{p/\neg p\}$ for some $o \in O$. Then $v_o \models \frac{\partial}{\partial p}F$, so $V(a_M) \not\subseteq V(\pi(\frac{\partial}{\partial p}F)) = V(\frac{\partial}{\partial x_p}\pi(F))$, hence $\frac{\partial}{\partial x_p}\pi(F) \notin (a_M) + \mathbb{I}_2$.
(2) $\Longrightarrow$ (1): If $\frac{\partial}{\partial x_p}\pi(F) \notin (a_M) + \mathbb{I}_2$, then $O = V(a_M) \not\subseteq V(\pi(\frac{\partial}{\partial p}F))$. Therefore there exists $o \in O$ such that $v_o \models \frac{\partial}{\partial p}F$. Thus F is sensitive in p.

(3) *Sensitivity test*: If the implication has not been discarded, test whether the implication is sensitive in all its arguments w. r. t. the actual set H (using lemma 2 if necessary). If it is not sensitive in some of them, the user must to react:
- Adding a new example o to H, witness of the sensitivity (that is, he/she thinks that it is sensitive), or
- eliminating the attribute of the implication (it accepts it is not sensitive), changing the accepted implication by the refined one.

Fig. 2. Sensitivity test to add to algorithm of fig. 1

One can recursively apply the above criteria (w.r.t. an order on PV) to obtain sensitive implications. If $\mathcal{L}$ is a stem basis and $\mathcal{L}'$ is the refinement obtained, since Amstrong's rule R2 states $Y_1 \setminus \{Y\} \to Y_2 \models Y_1 \to Y_2$, one has that $\mathcal{L}' \models \mathcal{L}$. Thus $\mathcal{L}'$ is also a complete set of implications. The set $\mathcal{L}'$ has an advantage over other sets of proper implications (e.g. [10]) that it directly works on Duquenne-Guigues basis so it does not need an specific algorithm to build it.

5 Variants of attribute exploration

We shall propose new steps for attribute exploration. All of them are investigated with the translation into polynomials in mind. Although in the exposition we do not explicitely use polynomials -the results and their proofs are more readable in logical form- in practice they will be useful.

The attribute exploration can be extended by adding a sensitivity test w.r.t H (shown in fig. 2). Note that addition of a new object follows the formula

$$a_{H\cup\{(\delta_1,\ldots,\delta_n)\}} = \Phi(a_H \cdot (1 + \Pi_{i=1}^n (x_i + \delta_i + 1)))$$

For the running example, the implication $N \wedge A \to Mo$ is obtained and considered as sound. In this case, $a_M = x_1x_2x_4 + x_1x_2 + x_1x_3 + x_1x_4 + x_1 + 1$. A Gröbner basis for $a_M + \mathbb{I}_2$ is

$$\{x_4^2 + x_4, x_3^2 + x_3, x_2^4 + x_2, x_3x_4 + x_3, x_2x_4 + x_2 + x_3 + x_4, x_2x_3 + x_2, x_1 + 1\}$$

It verifies (with CoCoA) that $\frac{\partial}{\partial x_1}\pi(N \wedge A \to Mo) = x_2(1 + x_3) \in (a_M) + \mathbb{I}_2$

We think that is not really sensitive in N (every live being needs water), so we accept $A \to Mo$, which is now sensitive. Reasoning similarly with the other one, it obtains $\{N,\ A \to Mo, L \to Mo\}$, a stem basis of sensitive implications.

Sensitivity test can be also added when background knowledge exists. In this case, we deal with hard problems as consistency checking or entailment. It starts with H and a background knowledge Γ for $M\restriction_H$, that is, $M\restriction_H \models \Gamma$. Or, in algebraic terms, $V(a_H) \subseteq V(\{1 + \pi(F)\ :\ F \in \Gamma\})$. The step to add is in given figure 3. Condition $(*)$ means

$$1 + \frac{\partial}{\partial x_p}\pi(L) \notin (\{1 + \pi(F)\ :\ F \in \Gamma\}) + \mathbb{I}_2$$

(3) *Sensitivity test*: Test whether the implication is sensitive in all its arguments with the actual set H and the background knowledge Γ, that is, whether for each attribute p in the antecedent of L

$$\Gamma \not\models L \leftrightarrow L\{p/\neg p\} \qquad (*)$$

If it is not sensitive in some of them, the user must to react:

- **Giving** $o \in O$, a witness of the sensitivity (which is added to H), that is

$$v_o \models \{\frac{\partial}{\partial p} L\} \cup \Gamma \qquad (**)$$

- **or else** removing the attribute of the implication (he/she accepts it is not sensitive).

Fig. 3. Sensitivity test with background knowledge

Regarding to the existence of an object for $(**)$, if the user does not know one, but believes that it really exists, a model search program may be used to give an anonymous object. Test $(**)$ can be fairly translated into algebraic terms.

6 Attribute exploration with new attributes

Now we propose how to extend the context by adding new attributes. Formally, one starts with M_0, a subcontext with partial set of attributes,

$$M_0 = (H, A_0, I \cap (H \times A_0)), \qquad \text{with } A_0 \subsetneq A$$

Assume that, at some stage, full extents for a set H' of objects are introduced, with the aim of expanding the new attributes to initial objects of M_0 (see fig. 5). The user only knows -about the new attributes- a background knowledge Γ, relating old and new attributes. Since it seems not advisable to add many arguments at once (to facilitate the answers of tests), Γ will be relatively small.

It is important to observe that $\Delta = \mathcal{L} \cup \Gamma$, where $\mathcal{L}$ is the partial set of implications, may be inconsistent with ontological commitments implicitely or unconsciously accepted for the old attributes; that is, it may be false for M_0, whenever the extents of H were expanded to the full attribute set. Thus one needs an *expandability test* for objects of H (to simplify assume that the new attributes are $\{p_{k+1}, \ldots, p_n\}$):

For each $o \equiv (\delta_1, \ldots, \delta_k)$ *of* H*, is there* $(\delta_{k+1}, \ldots \delta_n) \in \{0,1\}^{n-k}$ *such that* $\{p_j \ : \ \delta_j = 1 \ \wedge \ j \in \{1, \ldots, n\}\}$ *respects* Δ*?*

Theorem 6. *Let* M *be an expansion of* M_0 *to the complete attribute set, with the same set of objects. If* Γ *is a stem basis (respectively a background knowledge) for* M*, then* $\partial^c_{\{p_{k+1},\ldots,p_n\}}[\Gamma]$ *is a complete set of implications (respectively* $\partial_{\{p_{k+1},\ldots,p_n\}}[\Gamma]$ *is a background knowledge) for* M_0*.*

Proof. Assume that Γ is a stem basis. Let L be an implication in the language $A \setminus \{p_{k+1}, \ldots, p_n\}$. If $M \models L$, then $\Gamma \models L$. By corollary 1, $\partial_p[\Gamma] \models L$ so by lemma 1, $\partial^c_p[\Gamma] \models L$ holds. If Γ is a background knowledge, the result is a straightforward consequence of corollary 1.

(3) *Expansion test*: If implication has not been discarded, test whether the set of implications plus background knowledge is extendable to H.
  - **If it is extendable**, the user shall proceed:
    - Confirming the suggested implication, or
    - giving $o \in O$ such that $\{o\}'$ does not respect the implication. This is added to H', and the implication is discarded.
  - **Else**, it must revise the background knowledge, or to discard the implication

Fig. 4. Aditional step for exploration with new attributes

	Need water	Aquatic	Mobility	Legs
Cat	1	0	1	1
Leech	1	1	1	0
Frog	1	1	1	1
Maize	1	0	0	0
Fish	1	1	1	0

	Need water	Aquatic	Mobility	Legs	Land
Cat	1	0	1	1	?
Leech	1	1	1	0	?
Frog	1	1	1	1	?
Maize	1	0	0	0	?
Fish	1	1	1	0	0
Dog	1	0	1	1	1
Bean	1	0	0	0	1

Fig. 5. Extension of the context on live beings with new attributes

Corollary 2. *Let $\Delta \subseteq PForm$. The following conditions are equivalent:*
(1) Every extension of objects of H can be expanded to the full attribute set, consistently with Δ.
(2) $\{1 + \pi(F) \ : \ F \in \partial_{\{p_{k+1},\dots,p_n\}}[\Delta]\} \subseteq (a_H) + \mathbb{I}_2$

Assume now that it has previously certified that Δ is expandable to objects of H, and let L be a new implication. If $\Delta \cup \{L\}$ can be consistently extended to H, but the user thinks that it is not true, in a first stage the user is required to give a counterexample for L by completing the extention of some object of H (in this way it bounds the set of new examples), or, if he/she does not know which, a new example. Summarizing, the new step is shown in figure 4.

For example, suppose that we decide to add a new attribute, *to live in land* (La). Some complete extensions are given (figure 5). We only know as background knowledge that aquatic live beings do not live in land, and we consider the implication *every live being with legs and mobility lives in land*, that is

$$\Delta = \{A \to \neg La,\ Le \wedge Mo \to La\}$$

In this case, $\pi[\Delta] = \{1 + x_A x_{La},\ 1 + x_{Le}x_{Mo} + x_{Le}x_{Mo}x_{La}\}$.

The set H can not be consistently expanded to a model of Δ, because $\partial_{\{x_{La}\}}[\pi[\Delta]] = \{1 + x_A + x_A x_{Le} x_M, 1\}$ and $x_A + x_A x_{Le} x_M \notin (a_H) + \mathbb{I}_2$.

6.1 A final remark: defining the new attributes

We now see how to extend the above procedure *for learning* the new attribute. We suppose we have a stem basis consistent with old information; and, in a second stage, we wish to find a definition of the new attribute w.r.t the old ones (if the user thinks it is possible). The next theorem states a solution, which is an adaptation of predicate completion procedure (sect. 6.2 in [5]). That is, we are considering the stem basis is a complete knowledge base for the attribute.

Theorem 7. *Let M_0 as in section 6 with $A_0 = A \setminus \{p\}$. Assume that $\mathcal{L}$ is a stem basis, built by attribute exploration with expansion test. Let*

$$\Omega = \{Y_1 \subseteq A_0 \;:\; \textit{there exists } Y \subseteq A_0 \textit{ s.t. } \; Y_1 \to Y \cup \{p\} \in \mathcal{L}\}$$

If M_c is the expansion of M_0 to A by defining the intent w.r.t. $\{p\}$ by

$$p \in \{o\}' \quad \Longleftrightarrow \quad v_o(\bigvee_{Y \in \Omega} \bigwedge Y) = 1$$

then $\mathcal{L}$ is a stem basis for M_c.

Since $M_c \models p \leftrightarrow \bigvee_{Y\in\Omega} \bigwedge Y$, M_c is model of completion formula for p. Thus, the intent of each object o is expanded to p_n by the polynomial

$$v_o(p_n) := \pi(\bigvee_{Y\in\Omega} \bigwedge Y)(v_o(p_1), \ldots, v_o(p_{n-1}))$$

7 An application: discovering tree notion in graph theory

We shall investigate the relationship among several properties on graphs (with three or more nodes), comparing stem basis produced by classical attribute exploration with the result of the new methods. The properties are: *acyclic*, *connected*, *2-connected* (if one edge of the graph is removed, the induced subgraph is connected), *geodetic* (for every two nodes there exists only one shortest path), *bipartite* (it can be partionated the set of nodes in two sets such that every edge joins a node of each set), *nonseparable* (connected and, if one removes a node, the resulting graph remains connected), and *planar*.

We begun (classical) attribute exploration with the two first objects of figure 6 (left). For this we used ConExp, and the result is the formal context of fig. 6 (left. K_5 is the complete graph with five nodes, and K_{33} is the complete bipartite graph with two sets of tree nodes each one as partition). The stem basis is:
$$\begin{cases} L_1 : t \to a, b, c, g, p & L_2 : n \to c, d \quad L_3 : g \to c \quad L_4 : d \to c \\ L_5 : b, c, g, p \to a, t & L_6 : b, c, d, g \to n \quad L_7 : a \to b, p \quad L_8 : a, c, b, p \to g, t \end{cases}$$

One might apply completion procedure on *tree*, obtaining a (messy) definition,

(Bipartite ∧ Connected ∧ Geodetic ∧ Planar) ∨ (Acyclic ∧ Connected ∧ Bipartite ∧ Planar)

Even it is not evident that the first conjuction defines a tree; it is necessary to know the fact that every geodetic and bipartite graph is acyclic. For this context, the ideal generated is

$J_M = (g + b + t + 1, c + d + t, a + d + 1, t^2 + t, pt + t, nt, bt + t, dt, p^2 + p, np + n + p + 1, dp + d + p + 1, n^2 + n, dn + n, b^2 + b, db + d + b + 1, d^2 + d)$.

The first interesting sensitivity analysis is on L_5 ($\pi(L_5) = 1 + bcgp(1 + at)$):

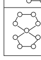
	Acyclic (a)	Connected (c)	2-connected (d)	Geodetic (g)	Bipartite (b)	Nonseparable (n)	Planar (p)	*Tree (t)*	rad.-minimal (r)
	1	1	0	1	1	0	1	1	1
	0	1	1	0	1	1	1	0	0
K_5	0	1	1	1	0	1	0	0	0
	1	0	0	0	1	0	1	0	0
	0	1	1	1	0	1	1	0	0
K_{33}	0	1	1	0	1	1	0	0	0
	0	1	1	0	1	0	1	0	0
	0	1	1	1	0	0	1	0	0

Fig. 6. Formal context on graphs, and the extension obtained for *radius-minimal*

- $\frac{\partial}{\partial b}\pi(L_5) \notin J_M$, thus is sensitive in b, hence preserve the implication.
- $\frac{\partial}{\partial c}\pi(L_5) \in J_M$, hence is not sensitive (in this case, we see that $g \to c$ holds in graphs), hence we redefine $L_5 := b, g \to a, t$.
- $\frac{\partial}{\partial g}\pi(L_5) \notin J_M$, thus it is sensitive in g, so preserve g finishing the analysis.

Other cases (L_6 and L_8) are similarly treated. The resultant is

$$\mathcal{L} = \begin{cases} t \to a, c, g, b, p & n \to c, d & g \to c & d \to c \\ b, g \to a, t, p & a \to b, p & a, c \to g, t & d, t \to n \end{cases}$$

The completion of *tree* from this basis is

$$Tree \leftrightarrow (\text{Bipartite} \wedge \text{Geodetic}) \vee (\text{Acyclic} \wedge \text{Connected})$$

It easy to see that the first conjunction is equivalent to the second one, the original definition of *tree*.

Our next aim is to expand our set of attributes with a new one, *radius-minimal* (denoted as variable by r). The *distance* of two nodes of a graph is the length of a shortest path between them. The *eccentricity* of a node v is the distance to a node farthest from v. The *radius* of a graph G, $r(G)$, is the minimum eccentricity of the nodes. Finally, a graph is called *radius-minimal* if $r(G - e) > r(G)$ for every edge in G. We used the method shown in section 6; the objects of fig. 6 suffices for it.

The exploration starts with the first two objects of figure 6, knowing that the first one is *radius-minimal* and the second one is not. Also we have the background knowledge $\{\neg c \to \neg r\}$. The procedure gives the basis

$$\mathcal{L} = \begin{cases} L_1 : r \to a, c, g, b, p, t & L_2 : t \to a, c, g, b, p, r & L_3 : n \to c, d \\ L_4 : g \to c & L_5 : d \to c & L_6 : c, g, b \to a, p, t, r \\ L_7 : a \to b, p & L_8 : a, c, b, p \to g, t, r & L_9 : a, c, d, g, b, p, t, r \to n \end{cases}$$

After testing sensitivity, three implications are refined, producing:

$$L_6 : g, b \to a, p, t, r \quad L_8 : a, c \to g, t, r \quad L_9 : d, r \to n$$

and the rest remains. Thus completion for r is

$$\text{Radius-minimal} \leftrightarrow \text{Tree} \vee (\text{Geodetic} \wedge \text{Bipartite}) \vee (\text{Acyclic} \wedge \text{Connected})$$

The last two conjunctions are equivalent to *Tree*, so we conclude that *Radius-minimal* and *Tree* are equivalent. Actually, this result is proved in [6]. Thus we take $v_o(r) := t$ to extend the attribute r for objects.

8 Conclusions and Future work

We present a framework for solving problems of FCA with the assistance of a CAS. We are confident that the tools described here may be useful to facilitate knowledge processing. As mentioned in previous sections, the complexity of some subproblems involved in the improvements of attribute exploration may restrict the method to projects of modest size, if a CAS as CoCoA is not used.

Throughout the paper we remarked some works related with the tools used here. The future work is the extension to many-valued logics and their applications [9].

References

1. Amstrong, W. W.. Dependency structures of data base relationships. *Proc. of IFIP Congress*, Geneva, 580-583 (1974).
2. Borrego-Díaz, J., Fernández-Lebrón, M.: Theoretical foundations of a specialised polynomial-based calculus for computing conservative retractions in propositional logic, *to appear* (2008).
3. Ganter, B., Wille, R.: *Formal Concepts Analysis. Mathematical Foundations* Springer, Berlin (1999).
4. Ganter, B., Attribute exploration with background knowledge. *Theoretical Computer Science* **217** 215–233 (1999).
5. Genesereth, M., Nilsson N., *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, Los Altos (1987).
6. Gliviak, F.: *On radially critical graphs* in Recents Advances in Graph Theory, Proc. Sympos. Academia Praha, 207–221 (1975),
7. Guigues, J.-L., Duquenne, V.: Familles minimales d' implications informatives resultant d'un tableau de donnees binaires. *Math. Sci. Humaines* **95**, 5–18 (1986).
8. Kapur, D., Narendran, P., An equational approach to theorem proving in first-order predicate calculus, *Proc. 9 Int. Joint Conf. on Artificial Intelligence (IJCAI'85)*, 1146-1153.
9. Laita, L. M., Roanes-Lozano, E., de Ledesma, L., Alonso-Jiménez, J. A.: A computer algebra approach to verification and deduction in many-valued knowledge systems. *Soft Computing* **3**, 7–19 (1999).
10. Taouil, R., Bastide, Y.. Computing Proper Implications. *Proc. of Workshop on Concept Lattices-based Theory, Methods and Tools for Knowledge Discovery in Databases* (E. Mephu et al. eds.), 49-61 (2001).

FCA Software Interoperability

Uta Priss

Napier University, School of Computing,
u.priss@napier.ac.uk
www.upriss.org.uk

Abstract. This paper discusses FCA software interoperability from a variety of angles: because the central FCA structures, formal contexts and concept lattices, can be represented in non-FCA software, interoperability with such software is of relevance. The non-FCA software in question is spreadsheet, relational database, graph and vector graphics software. The simplest approach to interoperability consists of providing file format conversion tools, such as FcaStone, which is therefore also discussed in this paper. Interoperability can be hindered by social factors, i.e. if the FCA researchers do not want to use non-FCA software. This issue is investigated with respect to software-derived graph layouts of lattice diagrams. An experiment that compares different software-derived lattice diagram layouts is conducted and leads to a surprising result.

1 Introduction

There appears to be some controversy among Formal Concept Analysis (FCA) researchers in how far FCA software should interoperate with other software. Some researchers complain about the lack of interoperability among FCA tools and the lack of connections between FCA and non-FCA applications. For example, formal contexts might be presentable in spreadsheet and relational database software whereas lattice diagrams might be edited in graph and vector graphics software. Other researchers express the view that the quality of FCA will be diminished if non-FCA approaches are applied, for example, with respect to non-FCA graph layout algorithms. This paper discusses different aspects of FCA interoperability and in particular investigates the use of non-FCA algorithms for graph layouts. A small experiment is conducted by deriving layouts of five well-known examples of formal contexts using FCA and non-FCA graph software. The experiment has a surprising result.

Section 2 of this paper provides a brief overview of the interoperability discussions in the FCA community. Section 3 discusses the relationship between some FCA and non-FCA tools. Section 4 describes the FCA file format conversion software FcaStone. Section 5 compares graph layouts derived with different FCA tools.

2 The FCA interoperability discussion

This section provides a brief overview of the discussion of interoperability in the FCA community and in the slightly broader conceptual structures (CS) community. In past

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 133–144,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

years, several ICCS authors expressed disappointment with the lack of progress in CS research with respect to applications and software (Chein & Genest (2000) and Keeler & Pfeiffer (2006)). Members of the ICCS community tend to agree that CS ideas are in principle extremely relevant to modern information representation tasks and, for example, the Semantic Web, but for some reason CS research has not been able to influence mainstream research communities (Rudolph, Krötzsch & Hitzler, 2007). In particular the software that is currently available for conceptual graphs (CG) and FCA does not reach the full potential of CS research and is not yet of commercial quality. Several suggestions have been made by the CS community to improve the situation. Keeler & Pfeiffer (2006) suggest to employ a pragmatic methodology for tool development using a "game" metaphor. Rudolph et al. (2007) suggest to establish connections with larger existing related communities (for instance, the Semantic Web community). Others have organised CS tool interoperability workshops[1] and challenges[2].

Dobrev (2006) presents an overview of interoperability issues of CG tools. He argues that although limited data exchange between CG tools is possible at the syntactic level using the standard exchange format, exchange at a semantic level, which incorporates contextual and background knowledge is not yet possible. In contrast to the CG community which has an ISO approved standard for Common Logic[3], there is no similar standard for FCA. The rest of this paper is only concerned with FCA software, not with the broader field of CS software. Tilley (2004) provides an overview of FCA software as described in FCA research papers. Interoperability between FCA software is low. Each software has different storage formats and different input/output options, which are not necessarily compatible with other software. Most of the FCA software appears to be at a somewhat "prototypical" stage and not of the same quality as commercial software. Although there is an overlap of features between different FCA software tools, certain features are only available in certain software. Thus in order to use all features that are currently implemented, a user would need to download and install several different tools and then try to figure out how to export data from one tool so that it can be incorporated into other tools, which is not always possible. In theory, it should be easy to convert between the different XML formats, but in practice, all of the current FCA XML formats have a completely different semantics. Because of the lack of interoperability among the tools, new developments, such as newly discovered faster algorithms, have to be implemented separately by the developers of each tool. There is no plug-in architecture that would allow algorithms to be easily incorporated into different tools.

3 Interoperability with non-FCA software

This section argues that FCA software shares a number of features with non-FCA software. More specifically, software for representing and operating on formal contexts shares features with database and spreadsheet software. Software for displaying and

[1] http://www.kde.cs.uni-kassel.de/ws/cs-tiw2008

[2] https://skyhawk.cs.uah.edu/concept/index.php/ICCS_Challenge

[3] http://cl.tamu.edu/

editing concept lattices shares features with vector graphics and graph drawing software. The difference between graph drawing and vector graphics software is that vector graphics is more general. Graphs consist of nodes and edges. Graph editors normally provide graph layout algorithms. The connection between a node and its edges is usually fixed, so that clicking on a node and moving it around will move the connected edges with that node. Vector graphics editors, on the other hand, can be used for any sort of graphics (not just nodes and edges). Although vector graphics editors usually have some grouping mechanism that allows to create complex objects which can be moved around and edited as a whole, it is not always possible to connect edges to nodes in such a manner. While vector graphics editors can represent graphs and provide many editing features, they often do not provide the specific editing features that more specialised graph editors have. Both graph and vector graphics software is of interest to FCA, but because of the differences between them, not all FCA features can be represented with such software.

It should be noted that the discussion in this section focuses on software, not mathematical modelling. Thus some of the mathematical aspects, such as the difference between abstract lattices, Hasse diagrams and general graphs are ignored if they are not immediately relevant for what is implemented in software tools. Furthermore, the list of FCA features that is discussed is not complete and depends on the current state of the art of FCA research and software technology.

Fig. 1 lists FCA features which are currently provided by FCA software. The context-related features are grouped into features that are shared with spreadsheet and database software. Spreadsheet software allows to create cross tables in which data can be entered, rows can be permuted and display parameters (font, colour, etc) can be changed. Relational database software also allows to store and edit objects, attributes and their relationships (crosses). But tables in relational databases need not be binary relations; databases are more akin to power context families. Before lattices can be drawn, users need to build binary contexts from the data in the database. FCA software should interoperate with spreadsheet and database software. Of course, not all context features are provided by spreadsheets and databases. Thus, although data can be imported from spreadsheets and databases, such software is not suitable as a sole interface for formal contexts.

With respect to displaying concept lattices, both vector graphics software and graph editing software have many features that are commonly used to modify lattice diagrams. Several FCA tools allow for lattice diagrams to be exported in SVG (scalable vector graphics) format. If minor edits are required that are not supported by the FCA software, it is possible to create a lattice using the FCA software and then to export the diagram and use a vector graphics program for further editing. For example, the graph layout algorithms, the manner in which the objects and attributes are displayed and so on could be implemented as options that the user chooses when exporting a diagram. Graph editing software has two important features that are not necessarily available in vector graphics software: the availability of graph layout algorithms and the feature of clicking on a node to move it in a manner that the attached edges stay attached. Modern vector graphics editors, such as Inkscape[4] and Dia support this to some degree. But because

[4] The URLs for all tools mentioned in this paper can be found on the last page of this paper.

there is no universally accepted graph representation format and Inkscape and Dia have their own formats, it is difficult for FCA software to export the lattice diagrams in formats that preserve sufficient information and can be read by graphics software. Older vector graphics editors (such as xfig) tend not to have graph functionality and are thus not as suitable for lattice editing.

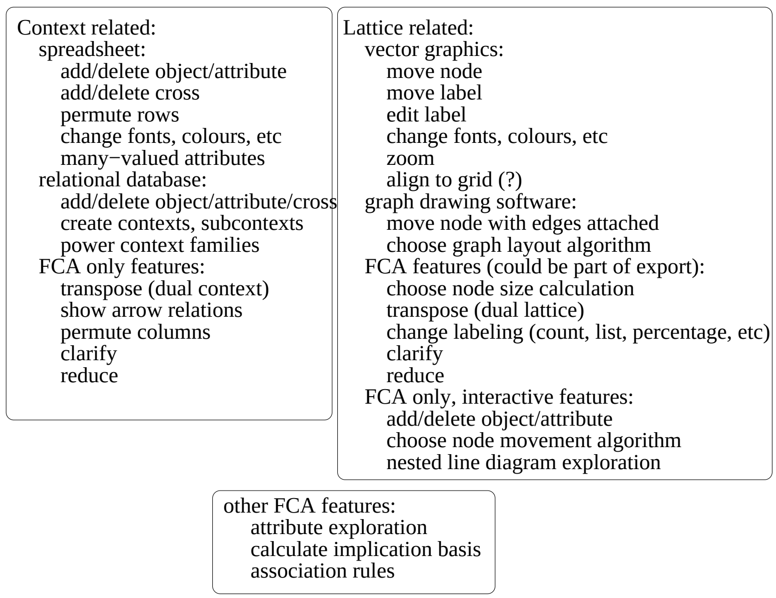


Fig. 1. Tasks for FCA software

Although vector graphics and graph editors provide means for adding and deleting, such features may not be consistent with the FCA features for adding or deleting objects, attributes and concepts. It can be a danger that inexperienced users might modify a diagram using the editor's add/delete features in such a manner that the diagram is no longer a lattice. Experienced FCA users might miss the ability to choose "node movement algorithms", i.e. the ability to move a whole filter or ideal in a lattice by dragging a node. It seems unlikely that current vector graphics and graph editors have such functionality. But this would be an opportunity for FCA developers to communicate with graphics editor developers. Maybe it would be possible to add such functionality to the editors. The Dia software, for example, supports different application modes (e.g. ER diagrams, flow charts). Maybe it would be possible to add an FCA mode to that program. Maybe the developers in the vector graphics communities would also be interested in layout algorithms that have been developed by FCA researchers. This might be a good opportunity for collaboration.

Complex FCA features, such as the exploration of nested line diagrams will maybe never be supported by traditional vector graphics editors. But Priss (2008a) discusses nested line diagrams as a means of "faceting", as used in library and information science. Several software tools for manipulating facets exists. Thus there could be some overlap in technology between software for faceted classification and FCA software. Other FCA features, such as association rules and implications are shared with data mining approaches. There could be opportunities for interoperability for FCA software in that area as well.

The simplest means of interoperability for FCA software with non-FCA software is to allow the import and export in compatible formats. With respect to spreadsheets and databases, FCA software should support comma-separated value files and with respect to vector graphics, the SVG format should be supported as an export option. It would be convenient to also allow input from graphics formats, but that is a difficult challenge because a lattice graph can be encoded in many different ways.

The question of whether non-FCA graph layout algorithms are useful for FCA software will be discussed in more detail further below. One obvious advantage for using external graph layout algorithms is that it eases the burden on the FCA programmers. The first popular non-FCA graph layout program that was used by FCA software was probably Graphplace (Eijndhoven, 1994), which converts a binary relation into a coordinate representation in a postscript format. A more modern program which implements many different graph layout algorithms and all kinds of features is Graphviz. The "directed graph" option in Graphviz provides layouts for lattices in a top-down manner. Graphviz also converts into many other graph, raster and vector graphics formats. Thus if FCA software exports lattices in a Graphviz format, then all these other formats are automatically accessible as well. The FCA tools Colibri and FcaStone make use of Graphviz.

4 FCAStone: FCA file format conversion software

A simple approach for allowing FCA software to interoperate with non-FCA software is by providing means for converting between the file formats of the different tools. FcaStone (named in analogy to "Rosetta Stone") is a command-line utility that converts between the file formats of commonly-used FCA tools (such as ToscanaJ, ConExp, Galicia, Colibri[5]) and between FCA formats and other graph and vector graphics formats. The main purpose of FcaStone is to improve the interoperability between FCA, graph editing and vector graphics software. Because it is a command-line tool, FcaStone can easily be incorporated into server-side web applications, which generate concept lattices on demand. FcaStone is open-source software and available for download from Sourceforge. FcaStone is written in an interpreted language (Perl) and thus platform-independent. FcaStone does not intend to compete with or replace the Java-based tools (ToscanaJ, ConExp, Galicia, etc) but instead to provide a different type of functionality, which is aimed more at server-side applications and conversion. FcaStone does not have a graphical user interface (GUI).

The emphasis of FcaStone is on converting file formats, but FcaStone can also convert formal contexts into lattices. It uses the Graphviz software to calculate the graph

[5] The URLs for all tools mentioned in this paper are listed at the end of the paper.

layouts. Graphviz is open-source graph visualisation software, which contains several graph layout algorithms. In this respect, FcaStone is similar to the Colibri software, which also relies on Graphviz for lattice layouts. Because Graphviz provides a large number of file conversion options, FcaStone only needs to produce a single format (called "dot format") which can then be further converted by Graphviz into a large number of other formats.

It is somewhat difficult to produce concept lattice diagrams in a graph format, because the dual labelling of nodes with objects and attributes is not easily supported in non-FCA graph formats. Priss (2008b) discusses how lattices can be represented using Graphviz's format. Another problem is that Graphviz's layout of lattices produces curved lines, which is not usually accepted in the FCA community. Thus, some FCA researchers may not approve of using FcaStone and Graphviz to produce lattice diagrams. We argue that FcaStone's diagrams are produced without manual editing. There are applications where manual editing of lattices is not feasible, for example, if the lattice diagrams are produced on-line as a response to user queries. An advantage of Graphviz's layouts is that they can be generated in an overlapping-free manner. Out of the three open-source FCA tools, ToscanaJ, ConExp, and Galicia, only Galicia produces lattices which are overlapping-free (see the next section). If it was possible to export the lattice layouts from Galicia, FcaStone could use such layouts instead of Graphviz layouts. But as far as we know the graph coordinates cannot be exported in Galicia. More details about FcaStone and the formats it supports can be found in Priss (2008b).

5 Graph layout for lattices

The previous sections have highlighted different aspects of FCA interoperability with non-FCA software. This section concentrates on comparing graph layouts produced by different tools. Five of Rudolf Wille's (the founder of FCA) well known examples of formal contexts have been selected. The five examples are fairly randomly chosen from an overview lecture given by Wille at the 2007 KPP conference[6]. The first example, "digits" was originally published in Stahl & Wille (1986). The "bodies of water" and the "live in water" examples were published in Wille (1984) and the "tea ladies" and the "lattice properties" examples were published in Wille (1992). The background of these lattices shall not be discussed in this paper because we are only interested in the representation of the line diagrams of these lattices. All five examples have reasonably complex line diagrams.

It is not the aim of the experiment conducted here to rank FCA software with respect to the "quality" of their diagrams. All FCA tools that were used here have different purposes. For example, the diagrams produced by Siena (part of the ToscanaJ suite) are intended for manual editing. Siena's initial layout contains many overlapping nodes. But because the initial layout contains many parallel edges, it only requires a few nodes to be moved manually in order to obtain a diagram that preserves the parallel edges. In general, there is some disagreement among researchers as to what diagrams should look like, whether they should have parallel edges, symmetries or whether the nodes

[6] http://www.fbi.h-da.de/kpp2007.html

should be arranged on levels. This paper does not intend to provide any judgement on these issues.

This paper is only interested in what we call “graphical similarity” of line diagrams. First, we define the “position” of a node in a line diagram as follows: if the nodes are arranged in levels starting from the top, then the position of a node refers to the level it is on and the distance it has from the side. Position “0,0” is the top node. Position “1,0” refers to the nodes that are the furthest to the left and right among all neighbours of the top node, and so on. This could either be one node, if the top has only one lower neighbour, or two nodes. Two line diagrams A and B are called “graphically similar” if a) they contain the same number of edge crossings and b) a node that is in the same “position” in A and B has the same number of upper and lower neighbours in A as in B.

Wille uses a particular method for drawing line diagrams, called the “geometric method” (Ganter & Wille, 1999). This paper intends to test whether any of the default lattices produced by commonly used FCA tools produce lattices that are similar to the lattice layouts that Wille derived with his geometric method. Again, it should be stressed that this is not intended as a value judgement with respect to the quality of these diagrams. But since some users may want to produce layouts that are similar to Wille’s it would be useful if software existed that produced such layouts on demand.

We conducted the following experiment. We derived the default layouts of the five formal contexts in ConExp, Galicia, Siena (part of ToscanaJ) and using the Graphviz layout of FcaStone. The three FCA tools were chosen because they are open-source and freely available. Furthermore, we manually reproduced Wille’s layouts. For the automatically derived layouts, we allowed ourselves only to change font sizes and node sizes. None of the nodes were moved. The production of the pictures was helped by the FcaStone software because with this software it took only seconds to convert the formal contexts into formats that can be read by the different tools. We apologise that the fonts in the pictures are too small to read. Only the layouts matter for this paper. We have provided a website[7] where researchers can find larger scale pictures and the contexts in “cxt” format so that this experiment can be reproduced and be extended to other FCA tools.

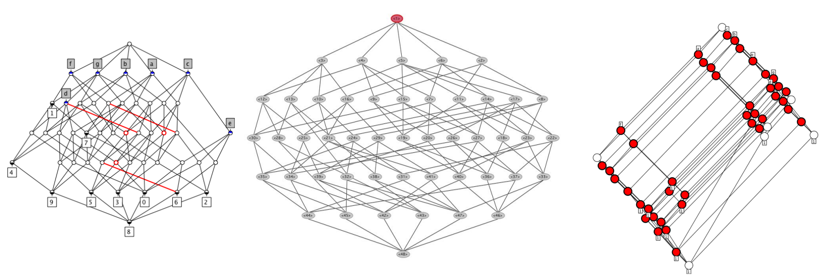


Fig. 2. The “digits” example: ConExp, Galicia, Siena

[7] `http://www.upriss.org.uk/fca/examples.html`

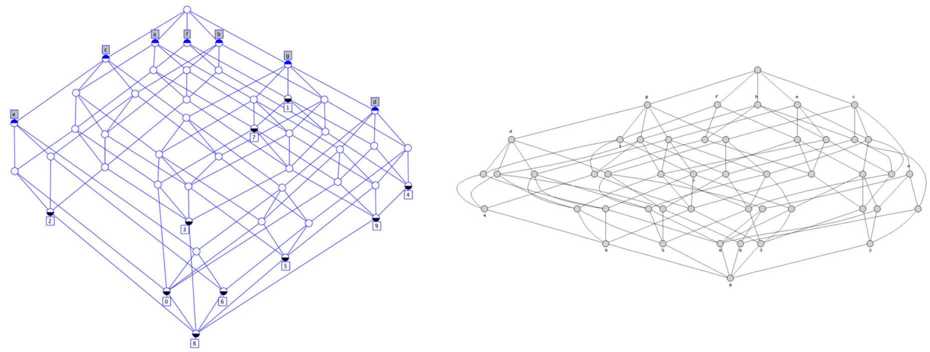


Fig. 3. The “digits” example: Wille’s layout, Graphviz

In our opinion, the result is surprising. The layouts that are produced by Graphviz are from an FCA view very unconventional because the edges are not parallel and in many cases even curved. Nevertheless, across all five examples, using our definition of “graphical similarity”, the lattices produced by Graphviz are similar to Wille’s layouts. It seems to us that if an algorithm was found that started with the Graphviz layouts and then straightened the edges and looked for parallel edges, it might be possible to automatically produce Wille-style layouts. In our opinion, this little experiment is an argument for increased interoperability between FCA and non-FCA tools. Even if non-FCA tools produce something that initially does not look appropriate (such as the curved edges in the Graphviz diagrams), it may ultimately have a functionality that is useful for FCA purposes. Only if FCA tools interoperate with non-FCA tools, it is possible to explore such features.

In the “digits” example, Graphviz’s and Wille’s layout are graphically similar because they are almost mirror images of each other. In ConExp and Galicia, the nodes are more permuted and not in the same positions. In all examples, Siena is difficult to see because of the strong degree in overlap. In the “bodies of water” example, ConExp’s, Wille’s and Graphviz’s layouts are similar and are different from Galicia and Siena. In the “lattice properties” example, both Wille’s and Graphviz’s layout have 7 edge crossings, ConExp has 6, Galicia has more. In the “live in water” example, Wille’s, Graphviz’s and ConExp’s layouts differ by a few switched nodes and by one edge crossing. Galicia is very different and has more edge crossings. In the “tea ladies” example, the nodes neighbouring the top node are roughly (but not exactly) in the same positions in Graphviz’s and Wille’s layouts, but not in ConExp and Galicia.

6 Conclusion

This paper analyses FCA software interoperability from a variety of angles. It is argued that interoperability with non-FCA software can be challenging because non-FCA applications have entirely different aims and purposes. But there can be benefits. For example, it appears that the graph layouts provided by a non-FCA tool are in some sense

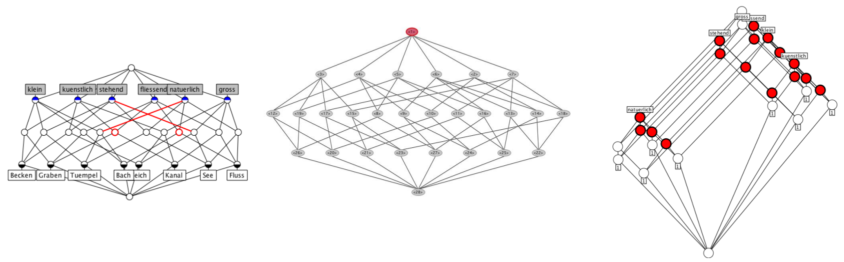


Fig. 4. The "bodies of water" example: ConExp, Galicia, Siena

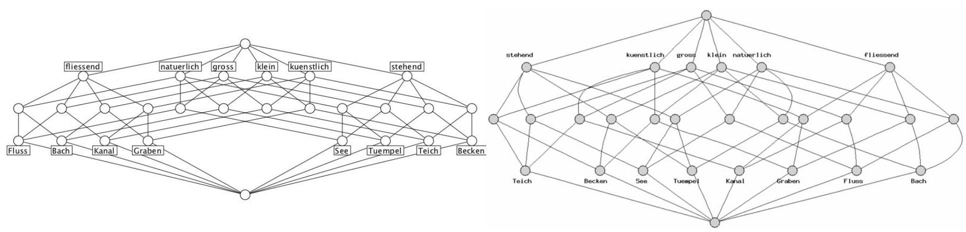


Fig. 5. The "bodies of water" example: Wille's layout, Graphviz

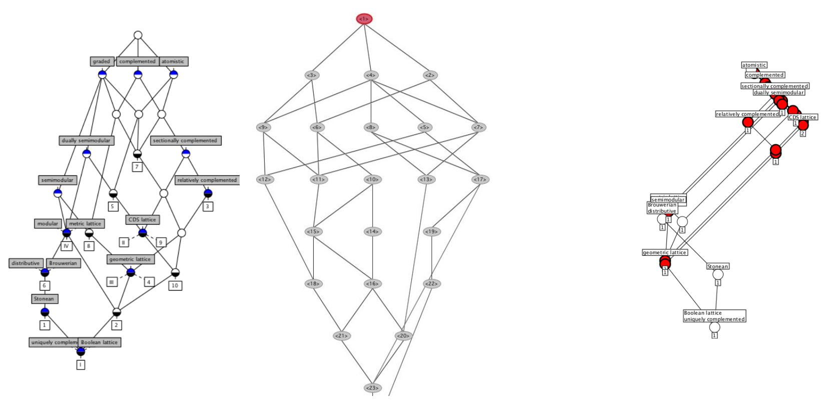


Fig. 6. The "lattice properties" example: ConExp, Galicia, Siena

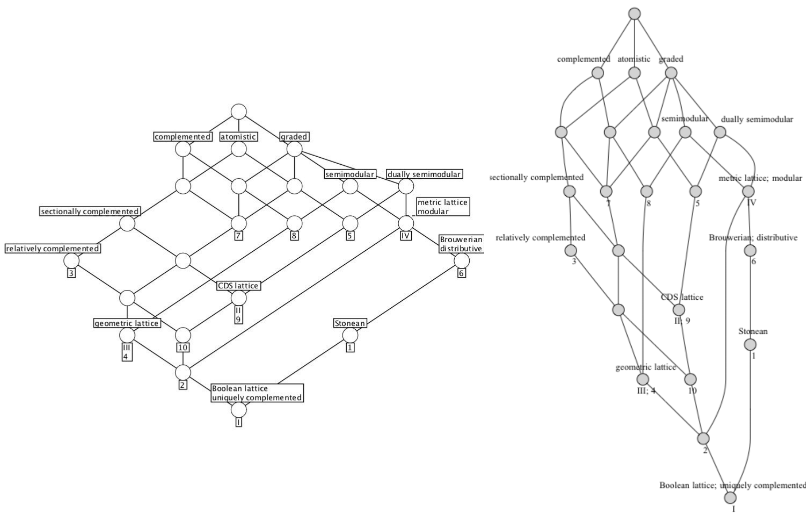


Fig. 7. The “lattice properties” example: Wille’s layout, Graphviz

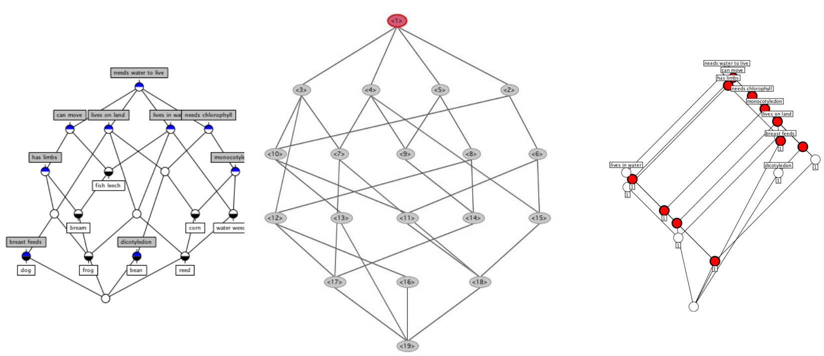


Fig. 8. The “live in water” example: ConExp, Galicia, Siena

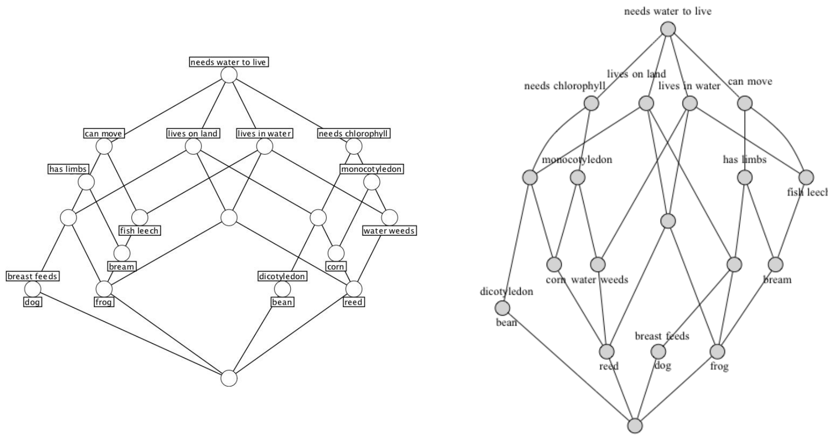


Fig. 9. The "live in water" example: Wille's layout, Graphviz

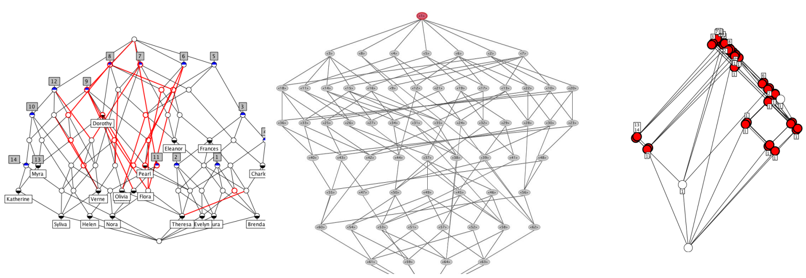


Fig. 10. The "tea ladies" example: ConExp, Galicia, Siena

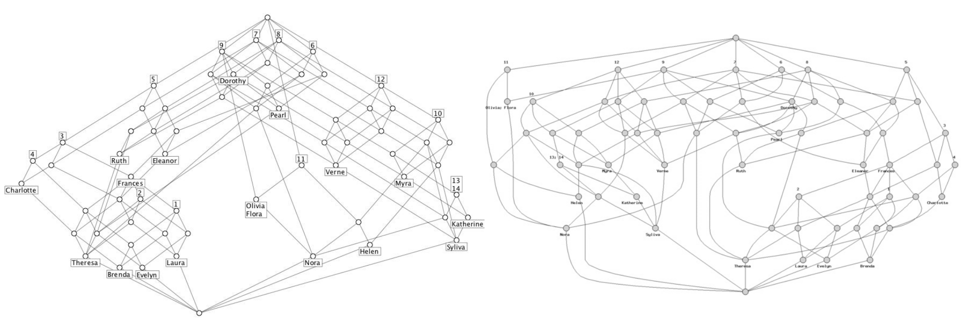


Fig. 11. The "tea ladies" example: Wille's layout, Graphviz

similar to manually derived layouts from researchers in the FCA community. Thus combining FCA software with non-FCA software can provide new insights and inspirations.

URLs for the Tools mentioned in this paper

1. Colibri: `http://www.st.cs.uni-sb.de/~lindig/#colibri`
2. ConExp: `http://sourceforge.net/projects/conexp`
3. Dia: `http://live.gnome.org/Dia`
4. FcaStone: `http://fcastone.sourceforge.net`
5. fca.sty: `http://www.math.tu-dresden.de/ganter/fca`
6. Galicia: `http://www.iro.umontreal.ca/~galicia`
7. Graphviz: `http://www.graphviz.org`
8. Inkscape: `http://www.inkscape.org`
9. ToscanaJ: `http://toscanaj.sourceforge.net`
10. Tockit (related to ToscanaJ): `http://tockit.sourceforge.net`
11. Xfig: `http://www.xfig.org`

References

1. Chein, M.; Genest, D. (2000). *CGs Applications: Where Are We 7 Years After the First ICCS?* In: Ganter; Mineau (eds.): Lecture Notes in Artificial Intelligence 1876, Springer, p. 127-139.
2. Dobrev, P. (2006). *CG Tools Interoperability and the Semantic Web Challenges.* Contributions to ICCS 2006, 14th International Conference on Conceptual Structures, Aalborg University Press.
3. Eijndhoven, Jos van (1994). *Graphplace - a graph layouter*. Software, Eindhoven University of Technology, The Netherlands. Available via anonymous ftp from several sites.
4. Ganter, Bernhard; Wille, Rudolf (1999). *Formal Concept Analysis.* Mathematical Foundations. Springer Verlag.
5. Keeler, M.; Pfeiffer, H. (2006). *Building a Pragmatic Methodology for KR Tool Research and Development.* In: Schaerfe, Hitzler, Ohrstrom (eds.), Conceptual Structures: Inspiration and Application, Proceedings of the 14th International Conference on Conceptual Structures, ICCS'06, Springer Verlag, LNAI 4068, p. 314-330.
6. Priss, Uta (2008a). *Facet-like Structures in Computer Science.* Axiomathes, Vol 14, Springer-Verlag.
7. Priss, Uta (2008b). *FcaStone - FCA file format conversion and interoperability software.* Conceptual Structures Tool Interoperability Workshop (CS-TIW).
8. Rudolph, S.; Krötzsch, M.; Hitzler, P. (2007) *Quo Vadis, CS? On the (non)-impact of Conceptual Structures on the Semantic Web.* In: Priss, Polovina, Hill (eds.), Proceedings of the 15th International Conference on Conceptual Structures, ICCS'07, Springer Verlag, LNAI 4604, p. 464-467.
9. Stahl, J.; Wille, R. (1986). *Preconcepts and set representation of contexts.* In: Gaul & Schader (eds): Classification as a tool of research.
10. Tilley, Thomas (2004). *Tool Support for FCA*. In: Eklund (ed.), Concept Lattices: Second International Conference on Formal Concept Analysis, Springer Verlag, LNCS 2961, p. 104-111.
11. Wille, Rudolf (1984). *Liniendiagramme hierarchischer Begriffssysteme.* Studien zur Klassifikation. Indeks Verlag.
12. Wille, Rudolf (1992). *Concept Lattices and Conceptual Knowledge Systems.* Computers Math. Applic., 23, 6-9, p 493-515.

GARM: Generalized Association Rule Mining

T. Hamrouni[1,2], S. Ben Yahia[1] and E. Mephu Nguifo[2]

[1] Department of Computer Science, Faculty of Sciences of Tunis, Tunis, Tunisia.
{tarek.hamrouni, sadok.benyahia}@fst.rnu.tn
[2] CRIL-CNRS, IUT de Lens, Lens, France.
{hamrouni, mephu}@cril.univ-artois.fr

Abstract. A thorough scrutiny of the literature dedicated to association rule mining highlights that a determined effort focused so far on mining the co-occurrence relations between items, *i.e.*, conjunctive patterns. In this respect, disjunctive patterns presenting knowledge about complementary occurring items were neglected in the literature. Nevertheless, recently a growing number of works is shedding light on their importance for the sake of providing a richer knowledge for users. For this purpose, we propose in this paper a new tool, called GARM, aiming at building a partially ordered structure amongst some particular disjunctive patterns, namely the disjunctive closed ones. Starting from this structure, deriving generalized association rules, *i.e.*, those offering conjunctive, disjunctive and negative connectors between items, becomes straightforward. Our experimental study put the focus on the mining performances as well as the quantitative aspect and proved the utility of the proposed approach.
Keywords: Data mining, disjunctive closed pattern, frequent essential pattern, disjunctive support, equivalence class, partially ordered structure, generalized association rules.

1 Introduction and Motivations

Association rule mining is a fundamental topic in Data mining [1]. It has been extensively investigated since its inception. Its key idea consists in looking for causal relationships between sets of items, commonly called *itemsets*, where the presence of some items suggests that others follow from them. A typical example of a successful application of association rules is the market basket analysis, where the discovered rules can lead to important marketing and management strategic decisions. Recently, mining association rules was extended to various pattern classes like sequential patterns, graphs, etc. Nevertheless, the main moan that can be addressed to the contributions related to association rules is their focus on co-occurrences between items [2], probably as a heritage of the market basket analysis framework. Indeed, almost all related works neglect the other kinds of relations, like mutually exclusive occurrences [3], that can also bring information of worth interest for users.

In this paper, we propose a new tool, called GARM [1], covering the whole process allowing the extraction of generalized association rules. These latter generalize classical rules – positive rules – to offer disjunctive and negative connectors between items,

[1] GARM is the acronym of generalized association rule miner.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 145–156,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

in addition to the conjunctive one [4]. Our tool includes a first component making it possible extracting a concise representation of frequent patterns based on disjunctive patterns. Thanks to a second component, these latter will be partially structured *w.r.t.* set inclusion. Once the partially ordered structure obtained, generalized association rules can be easily derived thanks to the last component of our tool.

Noteworthily, extracting an exact concise representation of frequent patterns in the first component of the process makes it possible to exactly derive the different supports of each frequent pattern. This will make us able to compute the exact values of quality measures. Indeed, it was shown in [5] that almost all interestingness measures for association rules are expressed depending on the support of the rule and those of its associated premise and conclusion. In addition, using disjunctive patterns – in particular closed and essential patterns [6] – will provide an interesting starting point towards mining association rules conveying complementary occurrences between items, rather than co-occurrences. Indeed, these latter relationships – co-occurrences within literals [2] – were explored in-depth in the literature through association rules having conjunction of literals, called *literalsets*, in premise and conclusion. This leads to what is commonly known as positive and negative association rules. While disjunctive association rules only have recently begin to grasp the interest of researchers.

In general, generalized association rules are useful in many applications. In particular, disjunctive association rules – having disjunction of items either in premise or in conclusion – were considered for two main purposes: On the one hand, they were used as an intermediate step for defining some concise representations for frequent patterns [1]. On the other hand, they were exploited to provide users with new forms of association rules [7, 8]. For example, the added-value of such association rules has been recently highlighted in [2]. It is however important to note that generalized association rules can be considered as particular GUHA rules [9].

Note that we restrict ourselves in this work to disjunctive closed patterns whose smallest seeds, *i.e.* essential patterns, are frequent with respect to a minimum conjunctive support threshold. This is argued by the fact that we aim at retaining the spirit of association rule mining where this threshold, as well as the confidence-based one, is used to dramatically limit the number of extracted association rules. In addition, the use of a partially ordered structure will make it possible to select representative subsets of rules to be extracted. This nucleus of rules will be of paramount help for avoiding to overwhelm users by highly-sized rule lists.

The remainder of the paper is organized as follows. The next section discusses the related work. Section 3 recalls the key notions used throughout this paper. The structural properties of the disjunctive search space are explored in Section 4, followed by a detailed description of the GARM tool having for purpose to offer a complete process for the extraction of generalized association rules in Section 5. Experimental results focusing on the mining time as well as the quantitative aspect are reported and discussed in Section 6. Section 7 concludes the paper and points out future works.

[2] A literal is an item or the negation of an item.

2 Related Work

Contributions related to association rule mining mainly concentrated on the classical rule form, namely that presenting conjunction of items in both premise and conclusion parts. In this respect, many concise representations for such rules were proposed in the literature [10]. Recently, some works focused on introducing negative items. Nevertheless, the majority of items are not present in each transaction leading to explosive amounts of association rules with negation. Thus, existing approaches have tried to address this problem through the use of additional background information about the data, incorporating attribute correlations, and additional rule interestingness measures, etc. Here we will mainly detail the reduced number of related works on association rules relying on the disjunctive connector within items.

Some works [7, 8] were interested in using the disjunction connector within the association rule mining issue to define what is called *generalized association rules*. These rules grasped the interest of many researchers since they offer wealthier types of knowledge in many applications. In addition to the inclusive disjunction operator, *i.e.*, the operator $\vee$, Nanavati *et al.* in [8] were also interested in the exclusive disjunction operator, denoted $\oplus$. The authors hence proposed two kinds of rules which are the simple disjunctive rules and the generalized disjunctive ones. Simple disjunctive rules are those having either the premise or the conclusion (*i.e.*, not simultaneously both) composed by a disjunction of items. This disjunction can be inclusive (the simultaneous occurrence of items is possible) or exclusive (two distinct items cannot occur together). On the other hand, generalized disjunctive rules are disjunctive rules whose premises or conclusions contain a conjunction of disjunctions. These disjunctions can either be inclusive or exclusive. In [7], the author mainly focuses on getting out association rules having conclusions containing mutually exclusive items, *i.e.*, the presence of one of them leads to the absence of the others, what is expressed in [8] using the operator $\oplus$. Other forms of generalized association rules were also described in [11]. In [12], Shima *et al.* extract what they called *disjunctive closed rules*. In their work, a disjunctive closed rule simply stands for a clause under the disjunctive normal form (DNF) such that its disjuncts are constituted by frequent closed patterns. Elble *et al.* used disjunctive rules to handle numerical attributes by considering disjunctions between intervals [13]. This latter work extends other ones taking also into account categorical attributes (see [13] for references). Finally, it is worth noting that the disjunction connector has also been used to define some concise representations of frequent patterns through the so-called *disjunctive rule* (see for example [1] for references).

3 Basic Concepts

In this section, we briefly sketch the key notions that will be of use throughout the paper.

Definition 1. *An extraction context is a triplet* $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ *where* $\mathcal{O}$ *and* $\mathcal{I}$ *are, respectively, a finite set of objects* (*or transactions*) *and items* (*or attributes*)*, and* $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I}$ *is a binary relation between the objects and items. A couple* $(o, i) \in \mathcal{R}$ *denotes that the object* $o \in \mathcal{O}$ *contains the item* $i \in \mathcal{I}$*.*

Example 1. *We will consider in the remainder a context that consists of transactions* (1, *AB*), (2, *ACD*), (3, *CDE*), (4, *DEF*), (5, *ABCDE*), *and* (6, *ABC*) [3].

Definition 2. (SUPPORTS OF A PATTERN) *Let* $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ *be a context and* I *be a pattern. We mainly distinguish three kinds of supports related to* I*:*

$$Supp(\wedge I) = |\{o \in \mathcal{O} \mid (\forall\, i \in I, (o,i) \in \mathcal{R})\}|$$
$$Supp(\vee I) = |\{o \in \mathcal{O} \mid (\exists\, i \in I, (o,i) \in \mathcal{R})\}|$$
$$Supp(\overline{I}) = |\{o \in \mathcal{O} \mid (\forall\, i \in I, (o,i) \notin \mathcal{R})\}|$$

Roughly speaking, the semantics of the aforementioned supports is as follows:

- $Supp(\wedge I)$ is the number of objects containing all items of I.
- $Supp(\vee I)$ is the number of objects containing at least one item of I.
- $Supp(\overline{I})$ is the number of objects that do not contain any item of I.

Note also that $Supp(\vee I)$ and $Supp(\overline{I})$ are two complementary quantities *w.r.t.* $|\mathcal{O}|$ in the sense that: $Supp(\vee I) + Supp(\overline{I}) = |\mathcal{O}|$.

Example 2. *Consider our running context. We have* $Supp(\wedge\, CDE) = |\{3, 5\}| = 2$, $Supp(\vee\, CDE) = |\{2, 3, 4, 5, 6\}| = 5$ *and* $Supp(\overline{CDE}) = |\{1\}| = 1$.

Hereafter, $Supp(\wedge I)$ will simply be denoted $Supp(I)$. In addition, if there is no risk of confusion, the *conjunctive support* will simply be called *support*. A pattern I is said to be *frequent* if $Supp(I)$ is greater than or equal to a minimum support threshold, denoted *minsupp*. Since the set of frequent patterns is an order ideal, the set of items $\mathcal{I}$ will be considered as only containing frequent items. Lemma 1 states that conjunctive supports can be derived starting from disjunctive ones.

Lemma 1. *[14] Let* $I \subseteq \mathcal{I}$*. The following equalities hold:*

$$Supp(I) = \sum_{\emptyset \subset I' \subseteq I} (-1)^{|I'|-1} Supp(\vee I')$$

4 Structural Properties of the Disjunctive Search Space

In this section, we will characterize disjunctive patterns through the associated equivalence classes induced by the following closure operator:

Definition 3. *Let* $\mathcal{K} = (\mathcal{O}, \mathcal{I}, \mathcal{R})$ *be an extraction context. The disjunctive closure operator* h *is defined as follows [6]:*

$h : \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{I})$

$\quad I \mapsto h(I) = \{i \in \mathcal{I} \mid (\forall\, o \in \mathcal{O})\ ((o,i) \in \mathcal{R}) \Rightarrow (\exists\, i_1 \in I)((o, i_1) \in \mathcal{R})\}$.

The disjunctive closure $h(I)$ of a pattern I is equal to the maximal set of items which *only* appear in the transactions that contain at least an item of I. The closure operator h induces an equivalence relation on the power-set of $\mathcal{I}$, which partitions it into so-called *disjunctive equivalence classes*. In each class, all the elements have the same disjunctive support. The smallest incomparable elements, *w.r.t.* set inclusion, of a disjunctive equivalence class are called essential patterns, while the disjunctive closed pattern is the largest one [6]. These particular patterns are defined as follows.

[3] We use a separator-free form for the sets, *e.g.*, *ABC* stands for the set of items {A, B, C}.

Definition 4.
• *A pattern* $I \subseteq \mathcal{I}$ *is a* ***disjunctive closed pattern*** *if* $I = h(I)$ *or, equivalently, Supp*$(\vee I)$ $< min\{$*Supp*$(\vee I') \mid I' \subseteq \mathcal{I}$ *s.t.* $I \subset I'\}$.
• *A pattern* $I \subseteq \mathcal{I}$ *is an* ***essential pattern*** *if* $\forall\ I' \subset I$, $I \nsubseteq h(I')$ *or, equivalently, Supp*$(\vee I) > max\{$*Supp*$(\vee I') \mid I' \subseteq \mathcal{I}$ *s.t.* $I' \subset I\}$.

Example 3. *Consider our running context. The pattern* CDEF *is disjunctively closed, while* BE *is not, since Supp*$(\vee$ BE$)$ = *Supp*$(\vee$ BEF$)$. *On the other hand, the pattern* AC *is essential, while* DE *is not, since Supp*$(\vee$ DE$)$ = *Supp*$(\vee$ D$)$.

In the remainder, $\mathcal{FEP}_{\mathcal{K}}$ [4] denotes the set of frequent essential patterns associated to a given context $\mathcal{K}$ and a fixed *minsupp* value. The associated set of disjunctive closure will further be denoted $\mathcal{EDCP}_{\mathcal{K}}$ [5]. This latter set is hence equal to $\{h(I) \mid I \in \mathcal{FEP}_{\mathcal{K}}\}$.

To establish the link with conjunctive equivalence class – gathering patterns having the same Galois closure [15] – we notice that essential patterns (*resp.* disjunctive closed patterns) are equivalent to minimal generators *aka* free-sets (*resp.* closed patterns) (see [1] for references). These latter patterns were at the basis of the main concise representations of association rules that were proposed in the literature [10]. This clearly motivates the use of their correspondences within the disjunctive search space.

5 Detailed Description of the GARM Tool

As mentioned in the first section, the GARM tool is composed of three complementary components which are as follows: (i) Extracting an exact concise representation of frequent patterns based on disjunctive closed patterns and frequent essential ones. (ii) Building a partially ordered structure *w.r.t.* set inclusion within disjunctive closed patterns. Each one of these latter will be accompanied by its set of frequent essential patterns. (iii) Deriving generalized association rules from the built structure.

5.1 Extracting a New Concise Representation based on Disjunctive Patterns

Our representation is based on the sets $\mathcal{FEP}_{\mathcal{K}}$ and $\mathcal{EDCP}_{\mathcal{K}}$, as stated by Theorem 1.

Theorem 1. *The set* $\mathcal{EDCP}_{\mathcal{K}} \cup \mathcal{FEP}_{\mathcal{K}}$ *is an exact concise representation of the set of frequent patterns* $\mathcal{FP}_{\mathcal{K}}$ *[16].*

Example 4. *Figure 1* (*Left*) *lists the set of disjunctive closed patterns associated to the running context. For each closed pattern, its associated disjunctive support and frequent essential patterns, for minsupp* = $\mathbf{1}$*, are also given.*

This representation will be denoted $\mathcal{DSSR}_{\mathcal{K}}$ [6]. It is extracted thanks to an adaptation of our DCPR_MINER [7] algorithm [17], what constitutes the first component of the

[4] Stands for frequent essential patterns.
[5] Stands for essential disjunctive closed patterns.
[6] Stands for disjunctive search space-based representation.
[7] DCPR_MINER is the acronym of disjunctive closed pattern-based representation miner.

$\mathcal{EDCP}_{\mathcal{K}}$	**Disj. Supp.**	$\mathcal{FEP}_{\mathcal{K}}$
B	3	B
C	4	C
F	1	F
AB	4	A
EF	3	E
ABC	5	AC, BC
BEF	5	BE
DEF	4	D
CDEF	5	CD, CE
ABCDEF	6	AD, AE, BD, BCE

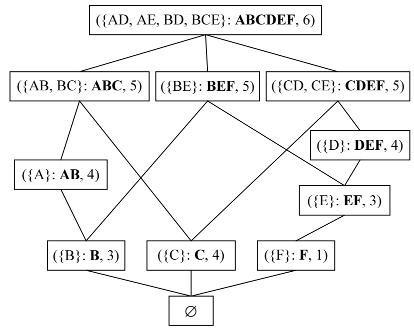


Fig. 1. (**Left**) The set $\mathcal{EDCP}_{\mathcal{K}}$ and the associated disjunctive support and frequent essential patterns for *minsupp* = 1. (**Right**) The equivalence classes partially ordered *w.r.t.* set inclusion.

GARM tool. Starting from $\mathcal{DSSR}_{\mathcal{K}}$, the conjunctive and negative supports of frequent patterns can thus be deduced using disjunctive supports. This representation also allows the derivation of the support of each literalset whose positive variation is based on a frequent pattern. This is carried out using the following formula [4]: *Supp*$(x_1 \wedge x_2 \wedge \ldots \wedge x_n \wedge \overline{y_1} \wedge \overline{y_2} \wedge \ldots \wedge \overline{y_m})$ = $\sum_{S \subseteq \{y_1, \ldots, y_m\}} (-1)^{|S|}$*Supp*$(x_1 \wedge x_2 \wedge \ldots \wedge x_n \wedge S)$, such that its positive variation, namely $\{x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m\}$, belongs to $\mathcal{FP}_{\mathcal{K}}$.

5.2 Building the Partially Ordered Structure

In this section, we will propose a new algorithm, called POSB [8], for partially sorting disjunctive closed patterns *w.r.t.* set inclusion. The POSB algorithm hence takes as input the representation $\mathcal{DSSR}_{\mathcal{K}}$ *s.t.* to each disjunctive closed pattern is associated its set of frequent essential patterns and disjunctive support. A node in the partially ordered structure will be associated to each disjunctive closed pattern. The pseudo-code of POSB is shown by Algorithm 1. Our algorithm inherits two main optimizations used in the algorithm proposed by Valtchev *et al.* [18], namely the sorting of disjunctive closed patterns, and the use of a border. Indeed, the set of disjunctive closed patterns $\mathcal{EDCP}_{\mathcal{K}}$ is sorted *w.r.t.* the increasing pattern size. Since closures of equal size cannot be comparable, this sorting avoids unnecessary comparisons. In addition, it makes possible that the closure f under treatment be of the largest size *w.r.t.* already treated ones. Thus, it suffices to find its lower cover among the nodes inserted in the structure. This lower cover is composed by those closures which are *immediately covered* by f.

On the other hand, the border $\mathcal{B}$ is an anti-chain *w.r.t.* set inclusion containing maximal closures among those already treated. In fact, the Valtchev *et al.* algorithm constructs the Hasse diagram representing the subset-superset relationship among concepts in the Galois lattice. It begins at the top of the lattice and then recursively identifies the lower neighbors of each concept. Nevertheless, it is not directly adapted to our situation. Indeed, although the intersection of two disjunctive closed patterns is obviously

[8] POSB is the acronym of partially ordered structure builder.

Algorithm 1: POSB

```
Input: The set 𝓔𝓓𝓒𝓟_𝓚 of disjunctive closed patterns.
Output: The disjunctive closed patterns ordered by set inclusion.
Begin
  𝓑 := ∅ ;
  Foreach (f ∈ 𝓔𝓓𝓒𝓟_𝓚) do
    Prohibited_List = ∅;
    Foreach (b ∈ 𝓑) do
      inter := b ∩ f;
      If (inter = b) then
        LOWER_COVER_INSERTION(f, b);
        𝓑 := 𝓑\ b;
      Else If (inter ≠ ∅) then
        LOWER_COVER_MANAGEMENT(f, b);
    𝓑 := 𝓑 ∪ f ;
End
```

a disjunctive closed pattern, this latter does not necessarily belong to $\mathcal{EDCP}_{\mathcal{K}}$. This is due to the fact that it could have all its essential patterns infrequent and, hence, has been already pruned. On its side, the proposed algorithm in [18] relies on the fact that the intersection of two concepts was already treated and it suffices to locate the corresponding node within the Hasse diagram.

In Algorithm 1, disjunctive closed patterns are inserted one at a time to a structure which is only partially finished to obtain at the end the entire one. Let f be the current disjunctive closed pattern to be inserted in the partially ordered structure. f will be compared to the elements of the border $\mathcal{B}$. If an element $b \in \mathcal{B}$ is included in f, then it is an element of its lower cover. A link between the node representing b and that representing f will be constructed thanks to the LOWER_COVER_INSERTION procedure (*cf.* Algorithm 2). The element b will then be deleted from the border. If b is not included in f but their intersection is not empty, then the LOWER_COVER_MANAGEMENT procedure will identify the common immediate predecessors of b and f (*cf.* Algorithm 3). Finally, f will be added to the border. It is important to note that in the LOWER_COVER_MANAGEMENT procedure, a prohibited list is associated to each disjunctive closed pattern to be inserted in the partially ordered structure. Indeed, when updating the precedence link between disjunctive closed patterns, a node can be visited more than once since it can be an immediate predecessor of many other nodes. This list will avoid such useless treatments by only allowing the visit of nodes that do not belong to it.

Example 5. *The associated structure to our running context is given by Figure 1* (*Right*).

5.3 Deriving Generalized Association Rules

Once the partially ordered structure built, deriving (subsets) generalized association rules can be easily done. An association rule R: $X \Rightarrow Y$ based on a pattern Z, denoted Z*-based rule*, is such that $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathcal{I}$ and $Y = \{y_1, y_2, \ldots, y_m\} \subseteq \mathcal{I}$ be two patterns, $X \cap Y = \emptyset$, and $X \cup Y = Z$. An association rule is usually considered as interesting *w.r.t.* two statistical measures, namely the support and the confidence. The formulae of these measures for an arbitrary rule are as follows:

Algorithm 2: LOWER_COVER_INSERTION

```
Input: A disjunctive closure f, and an element pred to be inserted in its lower cover.
Output: The updated lower cover of f.
Begin
    Foreach (l ∈ Lower_Cover(f)) do
        inter := l ∩ pred;
        If (inter = pred) then
            return;
        Else If (inter = l) then
            Lower_Cover(f) := Lower_Cover(f) \ l;
    Lower_Cover(f) := Lower_Cover(f) ∪ pred;
End
```

Algorithm 3: LOWER_COVER_MANAGEMENT

```
Input: A disjunctive closed pattern f, and an element b of the border B.
Output: The updated lower cover of f.
Begin
    Foreach (pred_b ∈ Lower_Cover(b)) do
        If (pred_b ∉ Prohibited_List) then
            inter := pred_b ∩ f;
            If (inter = pred_b) then
                LOWER_COVER_INSERTION(f, pred_b);
            Else If (inter ≠ ∅) then
                LOWER_COVER_MANAGEMENT(f, pred_b);
            Prohibited_List := Prohibited_List ∪ pred_b;
End
```

$$\textit{Supp}(X \Rightarrow Y) = \textit{Supp}(X \wedge Y), \text{ and, } \textit{Conf}(X \Rightarrow Y) = \frac{\textit{Supp}(X \wedge Y)}{\textit{Supp}(X)}$$

A rule is said to be *exact* if its confidence is equal to $\mathbf{1}$. Otherwise, it is said to be *approximate*. In addition, it is said to be *interesting* or *valid* if its support and confidence values are greater than or equal to their respective minimum thresholds *minsupp* and *minconf*. It is clear that whenever we are able to evaluate *Supp*$(X \Rightarrow Y)$, the derivation of the confidence value will be straightforward.

Let us now adapt the association rule framework to our context. As shown in Subsection 5.1, the $\mathcal{DSSR}_{\mathcal{K}}$ representation allows deriving the disjunctive, conjunctive and negative supports of each set of positive and negative items whose positive variation is based on a frequent pattern. In the sequel, we present an overview of the process by which we retrieve generalized association rules and evaluate their associated supports through traversing the partially ordered structure. Rules can be classified according to the number of nodes required for their extraction. We then distinguish two cases:

1. **An intra-node rule**: it requires a unique node and highlight relationships between a frequent essential pattern and its disjunctive closure f (here $Z = f$).
2. **An inter-nodes rule**: it is extracted using two nodes N_1 and N_2 *s.t.* the associated disjunctive closure of N_1, denoted f_1, is one of the immediate predecessors of that of N_2, denoted f_2. Let e_1 be a frequent essential pattern of f_1. An inter-nodes rule describes relationships between either f_1 and f_2 or e_1 and f_2 (here $Z = f_2$).

Both kinds of rules – intra-node and inter-nodes – can be either exact or approximate.

Different forms of generalized association rules can be extracted starting from our representation (*cf.* [16] for a detailed description). To limit the number of possible extracted rule forms, we mainly focus here on the following ones:

1. **Form 1**: *disjunction of items in premise and conclusion* $\vee X \Rightarrow \vee Y$: *Supp*$(\vee X \Rightarrow \vee Y)$ = *Supp*$(\vee X \wedge \vee Y)$ = *Supp*$(\vee X)$ + *Supp*$(\vee Y)$ - *Supp*$((\vee X) \vee (\vee Y))$ = *Supp*$(\vee X)$ + *Supp*$(\vee Y)$ - *Supp*$(\vee Z)$,
2. **Form 2**: *negation of items in premise and conclusion* $\overline{X} \Rightarrow \overline{Y}$: *Supp*$(\overline{X} \Rightarrow \overline{Y})$ = *Supp*$(\overline{X} \wedge \overline{Y})$ = *Supp*$(\overline{((\vee X) \vee (\vee Y))})$ = *Supp*$(\overline{Z})$ = $|\mathcal{O}|$ - *Supp*$(\vee Z)$,
3. **Form 3**: *disjunction of items in premise and negation of items in conclusion* $\vee X \Rightarrow \overline{Y}$: *Supp*$(\vee X \Rightarrow \overline{Y})$ = *Supp*$(\vee X \wedge \overline{Y})$ = *Supp*$((\vee X) \vee (\vee Y))$ - *Supp*$(\vee Y)$ = *Supp*$(\vee Z)$ - *Supp*$(\vee Y)$, and,
4. **Form 4**: *negation of items in premise and disjunction of items in conclusion* $\overline{X} \Rightarrow \vee Y$: *Supp*$(\overline{X} \Rightarrow \vee Y)$ = *Supp*$(\overline{X} \wedge \vee Y)$ = *Supp*$((\vee X) \vee (\vee Y))$ - *Supp*$(\vee X)$ = *Supp*$(\vee Z)$ - *Supp*$(\vee X)$,

where either X or Y is a frequent essential pattern or a disjunctive closed one, and Z = $X \cup Y$ is a disjunctive closed pattern (as described above). For each rule, the support of Z is known. It is the same for either X or Y since one of them is assumed to be a frequent essential pattern or a disjunctive closed pattern. For the sake of simplicity, we assume in the remainder that X is a frequent essential pattern or a disjunctive closed pattern. Since Y = $Z \backslash X$, then Y does not necessarily belong to $\mathcal{DSSR}_{\mathcal{K}}$ and, may even not be a frequent pattern. Nevertheless, its disjunctive support is required to evaluate that of the associated rule. To this end, we bound the support of Y using a lower bound, denoted *lb_Supp*, and an upper bound, denoted *ub_Supp*, computed as follows:

- ***lb_Supp***$(\vee Y)$ = $max\{$*Supp*$(\vee e) \mid e \in \mathcal{FEP}_{\mathcal{K}}$ and $e \subseteq Y\}$,
- ***ub_Supp***$(\vee Y)$ = $min\{$*Supp*$(\vee f) \mid f \in \mathcal{EDCP}_{\mathcal{K}}$ and $Y \subseteq f\}$.

In this respect, if Y is encompassed between a frequent essential pattern and its disjunctive closure, then *lb_Supp*$(\vee Y)$ = *ub_Supp*$(\vee Y)$. Hence, the support and confidence of the associated rule will be exactly computed. Otherwise, these latter measures will be bounded by a minimal and a maximal possible value using the bounds associated to Y. Such rules, further denoted *approximated* rules, are defined as follows:

Definition 5. *An association rule is said to be approximated if it has either its support or its confidence not exactly determined.*

Then, only valid rules having minimum possible values of support and confidence greater than or equal to *minsupp* and *minconf*, respectively, will be retained. Note that an approximated rule is different from an approximate rule in the sense that the latter has its support and confidence exactly computed (with a confidence not equal to $\mathbf{1}$), what is not the case of the former. In this respect, approximated rules were shown to convey interesting knowledge in the case of positive rules (see for example [19]).

Noteworthily, the bounds *lb_Supp*$(\vee Y)$ and *ub_Supp*$(\vee Y)$ always exist. Indeed, on the one hand, since the set of items $\mathcal{I}$ is pruned *w.r.t. minsupp*, then Y will be composed of frequent items even if it is infrequent. These items obviously belong to $\mathcal{FEP}_{\mathcal{K}}$, what ensures the existence of the lower bound. On the other hand, Y is covered by at least a disjunctive closed pattern, namely Z, what ensures the existence of the upper bound.

Example 6. *Let minsupp* = $\mathbf{1}$ *and let minconf* = $\mathbf{0.7}$. *Consider the intra-node rule* R_1 *of **Form 1** based on the disjunctive closed pattern ABCDEF and its frequent essential pattern BCE:* $\vee$ *BCE* $\Rightarrow \vee$ *ADF. Supp*(R_1) = *Supp*($\vee$ *BCE*) + *Supp*($\vee$ *ADF*) - *Supp*($\vee$ *ABCDEF*) = *Supp*($\vee$ *ADF*) (*since* $h(BCE)$ = *ABCDEF*). *Since ADF* $\notin \mathcal{DSSR}_{\mathcal{K}}$, *we need to evaluate its support. Since AD* $\subseteq$ *ADF* $\subseteq h(AD)$ = *ABCDEF* (*cf. Figure 1* (*Left*)), *then lb_Supp*($\vee$ *ADF*) = *ub_Supp*($\vee$ *ADF*) = $\mathbf{6}$. *Hence, Supp*(R_1) = $\mathbf{6}$ *and Conf*(R_1) = $\mathbf{1}$. R_1 *is hence a valid rule. Now, consider the inter-nodes rule* R_2 *of **Form 1** based on ABCDEF and one of its immediate predecessors, namely ABC* (*cf. Figure 1* (*Right*)): $\vee$ *ABC* $\Rightarrow \vee$ *DEF. In this case, DEF* $\in \mathcal{EDCP}_{\mathcal{K}}$. *Hence, Supp*$(\mathrm{R}_2)$ = *Supp*($\vee$ *ABC*) + *Supp*($\vee$ *DEF*) - *Supp*($\vee$ *ABCDEF*) = $\mathbf{5}$ + $\mathbf{4}$ - $\mathbf{6}$ = $\mathbf{3}$, *and Conf*(R_2) = $\mathbf{0.6}$. *Here, we took* X = *ABC. If we set* Y = *ABC, then the associated rule* R_3 = $\vee$ *DEF* $\Rightarrow \vee$ *ABC will have the same support than* R_2. *Nevertheless, its confidence is equal to* $\mathbf{0.75}$. *Hence,* R_3 *is a valid rule while* R_2 *is not.*

6 Experimental Results

Our experiments [9] focused on the mining time as well as the number of extracted valid rules *w.r.t.* their associated type, *i.e.*, exact, approximate or approximated. They were carried out on a PC equipped with a Pentium (R) having 3GHz as clock frequency and 1.75GB of main memory, running the GNU/Linux distribution Fedora Core 7 (with 2GB of swap memory). The compiler gcc 4.1.2 is used to generate the executable code starting from our C++ implementation.

Table 1. Mining time of generalized association rules on benchmark contexts.

Context	*minsupp* (%)	Component 1	Component 2	Component 3	Total time
CONNECT	*80.00*	2.1530	0.0068	0.0380	**2.1978**
	60.00	2.2807	0.0402	0.1618	**2.4827**
	40.00	2.5571	1.0443	0.9813	**4.5827**
PUMSB	*90.00*	3.1875	0.0403	0.1015	**3.3293**
	80.00	3.1581	2.9364	1.9693	**8.0638**
	70.00	3.6630	19.5460	8.7276	**31.9366**
KOSARAK	*0.90*	12.4551	0.1645	0.2239	**12.8435**
	0.70	16.2936	0.6825	0.3794	**17.3555**
	0.50	26.4491	5.6164	0.8738	**32.9393**
RETAIL	*2.00*	0.8471	0.0039	0.0135	**0.8645**
	1.00	1.0803	0.0113	0.0334	**1.1250**
	0.50	2.3909	0.1127	0.1331	**2.6367**

In the proposed experiments, the *minconf* value is set to the relative minimum support value, *i.e.*, $\frac{minsupp}{|\mathcal{O}|}$. Table 1 presents the mining time in seconds of the three components of GARM. This table shows the efficiency of our tool towards extracting generalized associated rules. Indeed, even for low *minsupp* values, GARM remains very fast. In this respect, the time consumed by each component, *w.r.t.* the total time,

[9] Test contexts are available at: *http://fimi.cs.helsinki.fi/data.*

Table 2. Number of extracted generalized association rules on benchmark contexts.

Context	*minsupp* (%)	Exact	Approximate	Approximated	Total number
CONNECT	*80.00*	620	316	152	**1, 088**
	60.00	1, 533	1, 337	354	**3, 224**
	40.00	3, 319	5, 813	3, 130	**12, 262**
PUMSB	*90.00*	566	1, 322	730	**2, 618**
	80.00	4, 376	13, 426	5, 002	**22, 804**
	70.00	9, 409	26, 747	14, 870	**51, 026**
KOSARAK	*0.90*	0	7, 586	0	**7, 586**
	0.70	0	13, 046	0	**13, 046**
	0.50	0	29, 648	0	**29, 648**
RETAIL	*2.00*	0	464	0	**464**
	1.00	0	1, 160	0	**1, 160**
	0.50	0	4, 622	0	**4, 622**

closely depends on the context characteristics. Nevertheless, the second and third components are in general faster than the first one. On the other hand, Table 2 highlights that the number of extracted rules closely depends on the context density. Indeed, the higher the value of this latter, the larger the associated equivalence classes are, and the greater the number of frequent essential patterns and closed ones is. This fact augments the number of rules even for high *minsupp* values for dense contexts. Interestingly enough, the number of exact and approximated rules for RETAIL and KOSARAK is equal to $\mathbf{0}$ for the tested *minsupp* values. This is due to the fact that for both contexts, each essential pattern is equal to its disjunctive closure what is not the case for the CONNECT and PUMSB contexts. Please note that the mining time and the number of extracted rules when *minconf* varies is omitted here, due to space limitations.

7 Conclusion and Perspectives

In this paper, we presented a complete tool, called GARM, allowing the extraction of generalized association rules. Our tool is composed of three components. The first consists in extracting a concise representation of frequent patterns based on disjunctive closed ones. The second component aimed at partially ordering these closure *w.r.t.* set inclusion. Once the structure built, extracting subsets of generalized association rules becomes a straightforward task thanks to the last component. Carried out experiments proved the effectiveness of the proposed tool. It is also important to mention that our GARM tool is easily adaptable to the case where the input is composed by conjunctive (closed) patterns instead of disjunctive ones.

Other avenues for future work mainly address the following points: First, a detailed comparison of our approach to the general GUHA approach [9] will be carried out. Second, the relationships between the various rule forms will be studied. The purpose is to only retain a lossless subset of rules while being able to derive the remaining redundant ones. Adequate axiomatic systems need thus to be set up.

Acknowledgments: We would like to thank anonymous reviewers for their helpful comments and suggestions. We are also grateful to Mrs. Nassima Ben Younes for fruitful discussions and help in the implementation of the tool. This work is supported by the French-Tunisian project CMCU-Utique 05G1412.

References

1. Ceglar, A., Roddick, J.F.: Association mining. ACM Computing Surveys, **volume 38(2)** (2006)
2. Steinbach, M., Kumar, V.: Generalizing the notion of confidence. Knowledge and Information Systems, **volume 12(3)** (2007) 279–299
3. Tzanis, G., Berberidis, C.: Mining for mutually exclusive items in transaction databases. International Journal of Data Warehousing and Mining, **volume 3(3)** (2007) 45–59
4. Toivonen, H.: Discovering of frequent patterns in large data collections. PhD thesis, University of Helsinki, Helsinki, Finland (1996)
5. Hébert, C., Crémilleux, B.: A unified view of objective interestingness measures. In: Proceedings of the 5th International Conference Machine Learning and Data Mining in Pattern Recognition, Springer-Verlag, LNCS, volume 4571. (2007) 533–547
6. Hamrouni, T., Denden, I., Ben Yahia, S., Mephu Nguifo, E.: A new concise representation of frequent patterns through disjunctive search space. In: Proceedings of the 5th International Conference on Concept Lattices and their Applications. (2007) 50–61
7. Kim, H.D.: Complementary occurrence and disjunctive rules for market basket analysis in data mining. In: Proceedings of the 2nd IASTED International Conference Information and Knowledge Sharing. (2003) 155–157
8. Nanavati, A.A., Chitrapura, K.P., Joshi, S., Krishnapuram, R.: Mining generalised disjunctive association rules. In: Proceedings of the 10th International Conference on Information and Knowledge Management. (2001) 482–489
9. Hájek, P., Havránek, T.: Mechanizing Hypothesis Formation: Mathematical Foundations for a General Theory. Springer-Verlag (1978)
10. Kryszkiewicz, M.: Concise representations of association rules. In: Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining, Springer-Verlag, LNCS, volume 2447. (2002) 92–109
11. Grün, G.A.: New forms of association rules. Technical Report TR 1998-15, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada (1998)
12. Shima, Y., Hirata, K., Harao, M., Yokoyama, S., Matsuoka, K., Izumi, T.: Extracting disjunctive closed rules from MRSA data. In: Proceedings of the 1st International Conference on Complex Medical Engineering. (2005) 321–325
13. Elble, J., Heeren, C., Pitt, L.: Optimized disjunctive association rules via sampling. In: Proceedings of the 3rd IEEE International Conference on Data Mining. (2003) 43–50
14. Galambos, J., Simonelli, I.: Bonferroni-type inequalities with applications. Springer (2000)
15. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
16. Hamrouni, T., Denden, I., Ben Yahia, S., Mephu Nguifo, E.: Exploring the disjunctive search space towards discovering new exact concise representations for frequent patterns. Technical report, CRIL-CNRS of Lens, Lens, France (2007)
17. Denden, I., Hamrouni, T., Ben Yahia, S.: Efficient exploration of the disjunctive lattice towards extracting concise representations of frequent patterns. To appear in the Proceedings of the 9th African Conference on Research in Computer Science and Applied Mathematics (in French). (2008)
18. Valtchev, P., Missaoui, R., Lebrun, P.: A fast algorithm for building the Hasse diagram of a Galois lattice. In: Proceedings of the Conference on Combinatorics, Computer Science and Applications. (2000) 293–306
19. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: A condensed representation of Boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery **volume 7(1)** (2003) 5–22

Concept-based Recommendations for Internet Advertisement

Dmitry I. Ignatov and Sergei O. Kuznetsov

Higher School of Economics, Department of Applied Mathematics
Kirpichnaya 33/5, Moscow 105679, Russia
{dignatov, skuznetsov}@hse.ru

Abstract. The problem of detecting terms that can be interesting to the advertiser is considered. If a company has already bought some advertising terms which describe certain services, it is reasonable to find out the terms bought by competing companies. A part of them can be recommended as future advertising terms to the company. The goal of this work is to propose better interpretable recommendations based on FCA and association rules.

1 Introduction

Contextual Internet advertising is a form of e-commerce. The largest revenues of the major players at this market, like search systems, are obtained from the so-called search sensitive advertisement, i.e, advertisement in a sense close to user queries. Here we consider the problem of detecting terms that can be interesting to an advertiser. Assume that a company F has already bought some advertising terms which describe certain services. As a rule, there are already competing companies at the market, therefore it is reasonable to find terms bought by them. These terms can be compared to those bought by F and part of them can be recommended as future advertising terms to F. The goal of this work is to propose well-interpretable recommendations based on FCA. The rest of the paper is organized as follows: First we recall main definitions from FCA and rule mining. Then we consider experimental data and the problem statement. Afterwards, we propose morphology-based and ontology-based metarules that can be derived without experimental data. We conclude the paper with experiments and their discussion.

2 Main definitions

First, we recall some basic notions from Formal Concept Analysis (FCA) [1]. Let G and M be sets, called the set of objects and attributes, respectively, and let I be a relation $I \subseteq G \times M$: for $g \in G,\ m \in M$, gIm holds iff the object g has the attribute m. The triple $K = (G, M, I)$ is called a *(formal) context*. If $A \subseteq G$, $B \subseteq M$ are arbitrary subsets, then the *Galois connection* is given by the following *derivation operators*:

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 157–166,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

$$A' \stackrel{\text{def}}{=} \{m \in M \mid gIm \text{ for all } g \in A\},$$
$$B' \stackrel{\text{def}}{=} \{g \in G \mid gIm \text{ for all } m \in B\}.$$

If we have several contexts derivative operator of a context (G, M, I) denoted by $(.)^I$.

The pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *(formal) concept (of the context K)* with *extent* A and *intent* B (in this case we have also $A'' = A$ and $B'' = B$). For $B, D \subseteq M$ the *implication* $B \to D$ holds if $B' \subseteq D'$.

In data mining applications, an element of M is called an *item* and a subset of M is called an *itemset*.

The *support* of a subset of attributes (an itemset) $P \subseteq M$ is defined as $supp(P) = |P'|$. An itemset is *frequent* if its support is not less than a given *minimum support* (denoted by min_supp). An itemset P is closed if there exists no proper superset with the same support. The closure of an itemset P (denoted by P'') is the largest superset of P with the same support. The task of frequent itemset mining consists of generating all (closed) itemsets (with their supports) with supports greater than or equal to a specified min_supp. An association rule is an expression of the form $I_1 \to I_2$, where I_1 and I_2 are arbitrary itemsets $(I_1, I_2 \subseteq A)$, $I_1 \cap I_2 = \emptyset$ and $I_2 \neq \emptyset$. The left side, I_1 is called *antecedent*, the right side, I_2 is called *consequent*. The support of an association rule $r : I_1 \to I_2$ [1] is defined as: $supp(r) = supp(I_1 \cup I_2)$. The *confidence* of an association rule r: $I_1 \to I_2$ is defined as the conditional probability that an object has itemset I_2, given that it has itemset I_1: $conf(r) = supp(I_1 \cup I_2)/supp(I_1)$. An association rule r with $conf(r) = 100\%$ is an *exact* association rule (or implication [1]), otherwise it is an *approximate* association rule. An association rule is *valid* if $supp(r) \geq min\_supp$ and $conf(r) \geq min\_conf$. An itemset P is a generator if it has no proper subset $Q (Q \subset P)$ with the same support. Let FCI be the set of frequent closed itemsets and let FG be the set of frequent generators. The *informative basis* for approximate association rules: $\mathcal{IB} = \{r : g \to (f \setminus g) | f \in FCI \ \wedge \ g \in FG \ \wedge \ g'' \subset f\}$.

3 Initial Data and Problem Statement

For experimentation we used data of US Overture [2], which were first transformed in the standard context form. In the resulting context $K = (G, M, I)$ objects from G stay for advertising companies (advertisers) and attributes from M stay for advertising terms (bids), gIm means that advertiser g bought term m. In the context $|G| = 2000$, $|M| = 3000$, $|I| = 92345$.

In our context, the number of attributes per object is bounded as follows: $13 \leq |g'| \leq 947$. For objects per attribute we have $18 \leq |m'| \leq 159$. From

[1] In this paper we use absolute values, but the support of an association rule r is also often defined as $supp(r) = supp(I_1 \cup I_2)/|O|$.

this context one had to compute formal concepts of the form (advertisers, bids) that represent market sectors. Formal concepts of this form can be further used for recommendation to the companies on the market, which did not buy bids contained in the intent of the concept. In other words, empty cell (g, m) of the context can be considered as a recommendation to advertiser g to buy bid m, if this advertiser bought other bids contained in the intent of any concept. This can also be represented as association rules of the form "If an advertiser bought bid a, then one can recommend this advertiser to buy term b" See [3] for the use of association rules in recommendation systems.

We consider the following context: $\mathbb{K}_{FT} = (F, T, I_{FT})$, where F is the set of advertising firms (companies), T is the set of advertising terms, or phrases, $fI_{FT}t$ means that firm $f \in F$ bought advertising term $t \in T$.

For constructing recommendations we used the following approaches and tools:

1. D-miner algorithm for detecting large market sectors as concepts;
2. Coron system for constructing association rules;
3. Construction of association metarules using morphological analysis;
4. Construction of association metarules using ontologies (thematic catalogs).

4 Standard approach to rule mining

4.1 Detecting large market sectors with D-miner.

D-miner is a freely available tool [4], [5] which constructs the set of concepts satisfying given constraints on sizes of extents and intents (icebergs and dual icebergs). D-miner takes as input a context and two parameters: minimal admissible extent and intent sizes and outputs a "band" of the concept lattice: all concepts satisfying constraints given by parameter values ($|intent| \geq m$ and $|extent| \geq n$, where $m, n \in \mathbb{N}$, see table 1).

Table 1. D-miner results.

Minimal extent size	Minimal intent size	Number of concepts
0	0	8 950 740
10	10	3 030 335
15	10	759 963
15	15	150 983
15	20	14 226
20	15	661
20	16	53
20	20	0

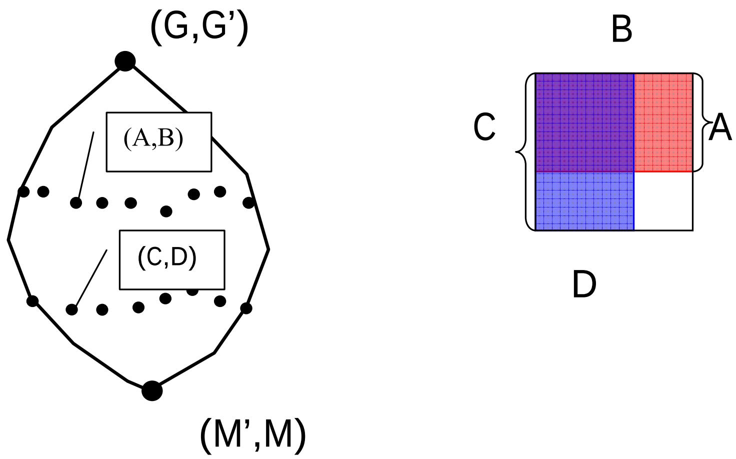


Fig. 1. A concept lattice and its band output by D-miner.

We give examples of intents of formal concepts for the case $|L| = 53$, where L is a concept lattice.

Hosting market.

{ affordable hosting web, business hosting web, cheap hosting, cheap hosting site web, cheap hosting web, company hosting web, cost hosting low web, discount hosting web, domain hosting, hosting internet, hosting page web, hosting service, hosting services web, hosting site web, hosting web }.

Hotel market.

{ angeles hotel los, atlanta hotel, baltimore hotel, dallas hotel, denver hotel, hotel chicago, diego hotel san, francisco hotel san, hotel houston, hotel miami, hotel new orleans, hotel new york, hotel orlando, hotel philadelphia, hotel seattle, hotel vancouver}

Distance communication market.

{ call distance long, calling distance long, calling distance long plan, carrier distance long, cheap distance long, company distance long, company distance long phone, discount distance long, distance long, cheap calling distance long, distance long phone, distance long phone rate, distance long plan, distance long provider, distance long rate, distance long service }

Weight loss drug market.

{ adipex buy, adipex online, adipex order, adipex prescription, buy didrex, buy ionamin, ionamin purchase, buy phentermine, didrex online, ionamin online, ionamin order, online order phentermine, online phentermine, order phentermine, phentermine prescription, phentermine purchase }

4.2 Recommendations based on association rules.

Using the Coron system (see [6]) we construct the informative basis of association rules [7]. We have chosen the informative basis, since it proposes a compact and effective way of representing the whole set of association rules. The results are given in table 2.

Table 2. Properties of informative basis.

min_supp	max_supp	min_conf	max_conf	number of rules
30	86	0,9	1	101 391
30	109	0,8	1	144 043

Here are some examples of association rules.

– $\{evitamin\} \rightarrow \{cvitamin\}$ supp=31 [1.55%]; conf=0.861 [86.11%]
– $\{gift\ graduation\} \rightarrow \{anniversary\ gift\}$, supp=41 [2.05%]; conf=0.820 [82.00%];

The value $supp = 31$ of the first rule means that 31 companies bought phrases "e vitamin" and "c vitamin". The value $conf = 0.861$ means that 86,1% companies that bought the phrase "e vitamin" also bought the phrase "c vitamin".

To make recommendations for each particular company one may use an approach proposed in [3]. For company f we find all association rules, the antecedent of which contain all the phrases bought by the company, then we construct the set T_u of unique advertising phrases not bought by the company f before. Then we order these phrases by decreasing of confidence of the rules where the phrases occur in the consequences. If buying a phrase is predicted by several rules (i.e., the phrase is in the consequences of several rules), we take the largest confidence.

5 Mining metarules

5.1 Morphology-based Metarules

Each attribute of our context is either a word or a phrase. Obviously, synonymous phrases are related to same market sectors. The advertisers companies have usually thematic catalogs composed by experts, however due to the huge number of advertising terms manual composition of catalogs is a difficult task. Here we propose a morphological approach for detecting similar bids.

Let t be an advertising phrase consisting of several words (here we disregard the word sequence): $t = \{w_1, w_2, \ldots, w_n\}$. A stem is the root or roots of a word, together with any derivational affixes, to which inflectional affixes are added [8]. The stem of word w_i is denoted by $s_i = stem(w_i)$ and the set of stems of words of the phrase t is denoted by $stem(t) = \bigcup_i stem(w_i)$, where $w_i \in t$. Consider the formal context $\mathbb{K}_{TS} = (T, S, I_{TS})$, where T is the set of all phrases and S is the set of all stems of phrases from T, i.e. $S = \bigcup_i stem(t_i)$. Then tIs denotes that the set of stems of phrase t contains s.

In this context we construct rules of the form $t \rightarrow s_i^{I_{TS}}$ for all $t \in T$, where $(.)^{I_{ts}}$ denotes the prime operator in the context K_{TS}. Then the a of the context $\mathbb{K}_{TS}$ (we call it a metarule, because it is not based on experimental data, but on implicit knowledge resided in natural language constructions) corresponds to $t \xrightarrow{FT} s_i^{I_{TS}}$, an association rule of the context $\mathbb{K}_{FT} = (F, T, I_{FT})$. If the values of support and confidence of this rule in context $\mathbb{K}_{FT}$ do not exceed certain thresholds, then the association rules constructed from the context $\mathbb{K}_{FT}$ are considered not very interesting.

Table 3. A toy example of context $\mathbb{K}_{FT}$ for "long distance calling" market.

firm \ phrase	call distance long	calling distance long	calling distance long plan	carrier distance long	cheap distance long
f_1	x		x		x
f_2		x	x	x	
f_3				x	x
f_4		x	x		x
f_5	x	x		x	x

Table 4. A toy example of context $\mathbb{K}_{TS}$ for "long distance calling" market.

phrase \ stem	call	carrier	cheap	distanc	long	plan
call distance long	x			x	x	
calling distance long	x			x	x	
calling distance long plan	x			x	x	x
carrier distance long		x		x	x	
cheap distance long			x	x	x	

Metarules of the following forms seem also to be reasonable. First, one can look for rules of the form $t \xrightarrow{FT} \bigcup_i s_i^{I_{TS}}$, i.e., rules, the consequent of which contain all terms containing at least one word with the stem common to a word in the antecedent term. Obviously, constructing rules of this type may result in the fusion of phrases related to different market sectors, e.g. "black jack" and "black coat". Second, we considered rules of the form $t \xrightarrow{FT} (\bigcup_i s_i)^{I_{TS}}$, i.e., rules with the consequent with the set of stems being the same as the set of stems of the antecedent. Third, we also propose to consider metarules of the form $t_1 \xrightarrow{FT} t_2$, where $t_2^{I_{TS}} \subseteq t_1^{I_{TS}}$. These are rules with the consequent being sets of stems that contain the set of stems of the antecedent.

Example of metarules.

- $t \xrightarrow{FT} s_i^{I_{TS}}$
  $\{last\ minute\ vacation\} \rightarrow \{last\ minute\ travel\}$
  Supp= 19 Conf= 0,90
- $t \xrightarrow{FT} \bigcup_i s_i^{I_{TS}}$
  $\{mail\ order\ phentermine\} \rightarrow \{adipex\ online\ order, adipex\ order, \ldots,$
  $phentermine\ prescription, phentermine\ purchase, phentermine\ sale\}$
  Supp= 19 Conf= 0,95
- $t \xrightarrow{FT} (\bigcup_i s_i)^{I_{TS}}$
  $\{distance\ long\ phone\} \rightarrow \{call\ distance\ long\ phone, \ldots,$
  $carrier\ distance\ long\ phone, distance\ long\ phone\ rate, distance\ long\ phone\ service\}$
  Supp= 37 Conf= 0,88
- $t_1 \xrightarrow{FT} t_2, \quad t_2^{I_{TS}} \subseteq t_1^{I_{TS}}$
  $\{ink\ jet\} \rightarrow \{ink\}$, Supp= 14 Conf= 0,7

5.2 Constructing ontologies and ontology-based metarules.

Here we use simple tree-like ontologies, where the closeness to the root of a tree defines generality of ontology concepts, which are advertisement phrases. For example, we use a manually constructed WordNet-like ontologies of market sectors. In our ontology of the pharmaceutical market the concept "pharmaceutical product" is more general than that of "vitamin." We introduce two operators acting on the set of advertising words T. *Generalization operator* $g_i(.) : T \rightarrow T$ takes a concept to a more general concept i levels higher in the generality order. *Neighborhood operator* $n(.) : T \rightarrow T$ takes a concept to the set of sibling concepts.

Now we define two types of metarules for ontology: a generalization rule $t \rightarrow g_i(t)$ and a neighborhood rule $t \rightarrow n(t)$. These rules can also be considered as association rules of the context $\mathbb{K}_{FT} = (F, T, I_{FT})$, which allows one to understand which of them are good supported by data.
Examples of metarules for pharmaceutical market.
Rule of the form $t \rightarrow n(t)$, where $t =$ "$B\_VITAMIN''$.

$\{B\_VITAMIN\} \rightarrow \{B\_COMPLEX\_VITAMIN, B12\_VITAMIN, C\_VITAMIN, \ldots$
$D\_VITAMIN, DISCOUNT\_VITAMIN, E\_VITAMIN, MINERAL\_VITAMIN, \ldots$
$MULTI\_VITAMIN, SUPPLEMENT\_VITAMIN, VITAMIN\}$
Rules of the form $t \rightarrow g_1(t)$, where $t =$ "$B\_VITAMIN''$, $g_1(t) =$ "$VITAMINS''$.
$\{B\_VITAMIN\} \rightarrow \{VITAMINS\}$.

6 Experimental Validation

For validation of association rules we used an adapted version of cross-validation. The training set was randomly divided into 10 parts, 9 of which were taken as the training set and the remaining part was used as a test set. By $A \xrightarrow{tr} B$ we denote an association rule generated on a training context. The confidence of this association rule measured on the test set, i.e.,

$$conf(A \xrightarrow{test} B) = \frac{|A^{I_{test}} \cap B^{I_{test}}|}{|A^{I_{test}}|}$$

shows the relative amount of companies that bought phrase B having bought phrase A.

We constructed 10 sets of association rules for 10 different training sets 1800 companies each (with $min\_supp = 1,5\%$ and $min\_conf = 90\%$. The aggregated quality measure of the obtained rules is the average confidence:

$$average\_conf(Rules_i) = \frac{\sum\limits_{A \to B \in Rules} conf(A \xrightarrow{test} B)}{|Rules_i|},$$

where $Rules_i$ is the set of association rules obtained on the i-th training set. We also considered rules with $min\_conf \geq 0.5$ and computed averaged confidence, which was again averaged over 10 cases, $average\_conf = \frac{\sum\limits_{i=1}^{n} average\_conf(Rules_i)}{n}$.

Table 5. Results of cross-validation for association rules.

	Number of rules	Number of rules with sup > 0	average_conf	Number of rules with min_conf=0.5	average_conf (min_conf=0.5)
1	147170	73025	0,77	65556	0,84
2	69028	68709	0,93	68495	0,93
3	89332	89245	0,95	88952	0,95
4	107036	93078	0,84	86144	0,90
5	152455	126275	0,82	113008	0,90
6	117174	114314	0,89	111739	0,91
7	131590	129826	0,95	128951	0,96
8	134728	120987	0,96	106155	0,97
9	101346	67873	0,72	52715	0,92
10	108994	107790	0,93	106155	0,94
means	115885	99112	0,87	92787	0,92

The confidence of rules averaged over the test set is almost the same as the min_conf for the training set, i.e., $(0,9 - 0,87)/0,9 \approx 0,03$.

We used confidence measure also for validation of metarules. Support does not have much importance here, since we do not look for large markets or mostly sellable phrases, but stable dependencies of purchases. So, we considered only rules with confidence larger than 0.8 (or 0.9). Confidence and support for metarules are computed for the context $\mathbb{K}_{FT} = (F, T, I_{FT})$. We present the values of confidence and support in the tables for morphology-based metarules.

Table 6. Average support and average confidence for morphology-based metarules.

Rule type	Average supp	Average conf	Number of rules
$t \xrightarrow{FT} s_i^{I_{TS}}$	6	0,26	2389
$t \xrightarrow{FT} \bigcup_i s_i^{I_{TS}}$	6	0,24	456
$t \xrightarrow{FT} (\bigcup_i s_i)^{I_{TS}}$	12	0,40	1095
$t \xrightarrow{FT} t_i$, such that $t_i^{I_{TS}} \subseteq t^{I_{TS}}$	15	0,49	7409
$t \xrightarrow{FT} \bigcup_i t_i$, such that $t_i^{I_{TS}} \subseteq t^{I_{TS}}$	11	0,36	2006

We set the minimal support 0,5 and compute the number of rules of each group for which this threshold is exceeded. Table 5 shows that average_conf of these metarules is actually much higher (about 0,9).

Table 7. Average supp and conf for morphological metarules for $min\_conf = 0,5$.

Rule types	Average supp	Average value of conf	Number of rules
$t \xrightarrow{FT} s_i^{I_{TS}}$	15	0,64	454
$t \xrightarrow{FT} \bigcup_i s_i^{I_{TS}}$	15	0,63	75
$t \xrightarrow{FT} (\bigcup_i s_i)^{I_{TS}}$	18	0,67	393
$t \xrightarrow{FT} t_i$ such that $t_i^{I_{TS}} \subseteq t^{I_{TS}}$	21	0,70	3922
$t \xrightarrow{FT} \bigcup_i t_i$ such that $t_i^{I_{TS}} \subseteq t^{I_{TS}}$	20	0,69	673

From tables 6 and 7one can easily see that most confident and supported rules are of the form $t \xrightarrow{FT} \bigcup_i t_i$. Note that the use of morphology is completely automated and allows one to find highly plausible metarules without data on purchases. The rules with low support and confidence may be tested against recommendation systems such as Google AdWords, which uses the frequency of queries for synonyms. For validation of ontological rules we used Google service AdWords. 90% of recommendations (words) were contained in the list of synonyms output by AdWords.

7 Conclusion and further work

The obtained results show that a part of dependencies in databases for purchases of advertisement phrases may be detected automatically, with the use of standard means of computer linguistics. Along with methods of data mining, these approaches allows one to improve recommendations and propose good means of ranking, which is very important for making Top-N recommendations. Another advantage of the approach consists in the possibility of detecting related advertisement phrases not given directly in data. Results of FCA-based biclusterization show the possibility of detecting relatively large advertisement markets (with more than 20 participants) given by companies and advertising phrases. To improve the proposed approach we plan to use well-developed ontologies like WordNet for constructing ontology-based metarules.

8 Acknowledgements

This work was supported by the Scientific Foundation of Russian State University Higher School of Economics as a part of project 08-04-0022.

References

1. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin/Heidelberg (1999)
2. Zhukov, L.E.: Spectral clustering of large advertiser datasets. Technical report, Overture R&D (2004)
3. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: ACM Conference on Electronic Commerce. (2000) 158–167
4. Besson, J., Robardet, C., Boulicaut, J.F., Rome, S.: Constraint-based bi-set mining for biologically relevant pattern discovery in microarray data. Intelligent Data Analysis journal **9**(1) (2005) 59–82
5. Besson, J., Robardet, C., Boulicaut, J.F.: Constraint-based mining of formal concepts in transactional data. In Dai, H., Srikant, R., Zhang, C., eds.: PAKDD. Volume 3056 of Lecture Notes in Computer Science., Springer (2004) 615–624
6. Szathmary, L., Napoli, A.: CORON: A Framework for Levelwise Itemset Mining Algorithms. In: Suppl. Proc. of ICFCA '05, Lens, France. (2005) 110–113
7. Szathmary, L., Napoli, A., Kuznetsov, S.O.: ZART : A Multifunctional Itemset Mining Algorithm. Research Report 00001271, LORIA (2005)
8. David, C.: A dictionary of linguistics and phonetics. third edn. Oxford: Blackwell Publishers (1991)

The Mathematical in Music Thinking

Rudolf Wille and Renate Wille-Henning

Technische Universität Darmstadt, Fachbereich Mathematik,
wille@mathematik.tu-darmstadt.de

Abstract. *"The Mathematical in music thinking"* is based on Heidegger's understanding of "the Mathematical" as the basic assumption of the knowledge of the things. Heidegger's ideas are combined with Peirce's classification of sciences, in particular, to distinguish between the Mathematical from the less abstract logical thinking and the more abstract mathematical thinking. The aim of this paper is to make understandable the role of the Mathematical in music. The paper concentrates on three domains: the rhythmic of music, the doctrine of music forms, and the theory of tonal systems. The theoretical argumentations are assisted by musical examples: the Adagio of Mozart's string quartet C major (KV 465), the second movement of Webern's Symphony op.21, and a cadence illustrating the problem of the harmony of second degree.

1 Music Thinking and The Mathematical

"Musica est exercitium arithmeticae occultum animi" (*"Music is a hidden arithmetical exercise of the soul"*) - this statement was written by the philosopher, mathematician, and scientist Gottfried Wilhelm Leibniz on April 17, 1712, in a letter to the mathematician and diplomat Christian von Goldbach. Leibniz referred with his statement to the astonishing *phenomenon of the correspondence between musical tones and numbers* which has been already demonstrated by the pythagoreans on their monochord. This phenomenon has been extensively described by the German musicologist Martin Vogel in his book *"Die Lehre von den Tonbeziehungen"*; there he writes: "Each interval used in music corresponds to a certain numerical proportion and, since each melody and each harmonic connection can be composed by numerically described intervals, each composition can finally be understood and analytically recognized as an arrangement of uniquely determined relations of numbers" ([18], p.9).

If one wants to comprehensively understand the role of mathematics in music thinking, then the numerical relations in music compositions pointed out by Vogel do not suffice. In particular, the numerical relations cannot suitably grasp the more extended set semantics basic for modern mathematics. For our theme we use the understanding of *"the Mathematical"* which Martin Heidegger worked out in his 1935/36 lecture on "Basic Questions of Metaphysics" (published in [11]). For Heidegger "the Mathematical" is not derivable out of mathematics, but mathematics itself is at the time a historically, socially, and culturally determined formation abstracted from the Mathematical. Heidegger deduced his

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 167–180,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

understanding of "the Mathematical" from the ancient Greeks: $\tau\grave{\alpha}\ \mu\alpha\vartheta\acute{\eta}\mu\alpha\tau\alpha$ means "the learnable". Learning the learnable is a kind of "taking", by which the taker takes only such things which, strictly speaking, he already has. According to Heidegger it follows: "$\tau\grave{\alpha}\ \mu\alpha\vartheta\acute{\eta}\mu\alpha\tau\alpha$, the *Mathematical*, is what of the things we actually already know, which we therefore do not first take out of the things, but which we already bring with us in a certain way" ([11], p.57); or phrased in another way: "The Mathematical is that basic position to the things by which we take on the things according to that which the things have already been given to us. The Mathematical is therefore the basic assumption of the knowledge of the things" ([11], p.58). For Heidegger this makes clear the central significance of the Mathematical for modern thinking, because *"a will of reformation and self-foundation of the knowledge form as such"* lies in the character of the Mathematical as distinctive conception ([11], p.75).

But how can we recognize the Mathematical? A promising approach is to abstract logical forms of thinking to mathematical forms of thinking which gives rise to rich mathematical theory developments retroacting, in particular, the logical forms and in this way enriching also the logical thinking (cf. [23]). To capture the Mathematical in music thinking, it suggests itself to identify first of all *the logical in music*, for instance in a manner as articulated by the musicologist Hans-Peter Reineke in referring to musical hearing; he writes: "Certain regulatives in musical hearing constitute and preserve *music as a logical being* that must sound plausibly out of itself if it shall be accepted" [17]. During the ending 18th century the term "musical logic" was linked to the idea "that music is an art which is autonomous, resting in itself, and submitted only to its own law of form; in particular, its right to exist needs not to be justified extra-musically" ([3], p.66). But, inspite of numerous efforts (here, first of all, the musicologist Hugo Riemann has to be named), a musical logic has never been really established in musicology. Nevertheless, to identify the Mathematical in music thinking, the connection between logical and mathematical thinking shall be discussed more extensively.

The philosopher and scientist Charles Sanders Peirce has convincingly described the connection between logical and mathematical thinking in the frame of his philosophy of science. In his classification of sciences from 1903 ([16], 258ff.), in which he ordered the sciences by the degree of their abstractness, mathematics as the most abstract science of all sciences is positioned at the most abstract level. As the only hypothetical science, mathematics has the task to develop a cosmos of forms of potential realities. All other sciences, under which philosophy is the most abstract, relate to actual realities. According to Peirce's classification, philosophy partitions into phenomenology, normative science, and metaphysics while normative science divides further into esthetics, ethics, and logic. Musicology has to be classified - such as history - under the descriptive science. In Peirce's classification the sciences are ordered in a manner that each science

– refers, according to its general principles, exclusively to the sciences which are more abstract than itself, and

– makes use of examples and specific facts elaborated by sciences which are less abstract than then the considered science.

For instance, logic as the third part of normative science is supposed to refer to ethics, esthetics, phenomenology, and mathematics concerning its general principles, and gains its actually real contents from metaphysics and the special sciences, particularly also from musicology. On the other hand, musicology can benefit from the manifoldness of the forms of logical and mathematical thinking.

As already pointed out, Heidegger does not view *"the Mathematical"* as part of mathematics, but views mathematics as an abstraction of the Mathematical, respectively. Thus, it seems very likely to locate the Mathematical within the phenomenology which is the initial part of philosophy in Peirce's classification of sciences ([16], p.258ff.). According to Peirce, the general task of phenomenology is to investigate the universal qualities of the phenomenons in their immediate character. Heidegger's conceptions of thingness can be understood as such universal qualities of phenomenons. This becomes more clear by the following determination of the nature of the Mathematical which has been summarized by Heidegger in his book [11] on p.71f:

1. The Mathematical is a conception of *thingness* leaping virtually over its things.
2. This conception determines what the *things* are considered for, as what they and how they should be acknowledged in advance.
3. The conception of the Mathematical is an *axiomatic anticipation* in the nature of the things tracing out how each thing and each relationship between those things are formed.
4. This formation offers the *scale* for delimiting the domain which embraces in future all things of such nature.
5. The axiomatically determined domain now demands for the things belonging to it an *accessibility* suitable alone for the axiomatically predetermined things.

For getting a better understanding of Heidegger's conception of the Mathematical, it might be helpful to discuss Heidegger's summary with respect to an example. Let us choose the space in which we live. Our understanding of the space is quite supported by our experiences with the bodies in the space so that we can rephrase Heidegger's five statements concerning the space of bodies as follows:

1. The conception of *space* leaping over its bodies is a model of the Mathematical (which has been abstracted mathematically to the real vector space).
2. This conception determines what the *bodies* are considered for, as what they and how they should be acknowledged in advance (which can be supported by representing the bodies mathematically using bounded connected subsets of the real vector space).
3. The conception of the Mathematical is an *axiomatic anticipation* in the nature of the bodies tracing out how each body and each relationship between

those bodies are formed (which become mathematically descriptive by algebraic terms).
4. This formation offers the *scale* for delimiting the domain which embraces in future all bodies of such nature (in particular, this allows to measure bodies mathematically).
5. The axiomatically determined spacial domain now demands for the bodies belonging to it an *accessibility* suitable alone for the axiomatically predetermined bodies (which can be mathematically abstracted within the axiomatically defined real vector space).

Let us record for this paper that the Mathematical as part of phenomenology is less abstract than mathematics, but is more abstract than logic, the third subpart of normative science. For investigating the Mathematical in music thinking, it is important to understand the relationships between Mathematical and logical thinking. Peirce convincingly explains the close connection between logical and mathematical thinking in his Cambridge Conferences Lectures from 1898, which have only completely be published, with 100 pages introduction and commentary, in 1992 under the title "Reasoning and the Logics of Things" [15]. Without pointing in details to Peirce's explanations, it shall be attempted in the following to demonstrate an analogous connection between the forms of music thinking and the forms of the Mathematical with its abstractions in mathematics thinking. The manifoldness of music thinking, in which we would have to investigate the Mathematical, cannot exhaustively be discussed in this contribution. Therefore we shall concentrate on forms of thinking about the rhythmic of music, the doctrine of music forms, and the theory of tone systems.

2 The Mathematical in the Rhythmic of Music

By Riemann's Music Encyclopaedia, *rhythm* has to be understood as an autonomous principle of form and order which is characterized on the one hand by regularity and relationship to a fixed tempo, on the other hand by grouping, subdivision, and alternation. In this conceptual characterization, first of all

- the *"uniformity of parts"* ,
- the *"succession of parts"*, and
- the *"distinctness of parts"*

have entered in music thinking as basic forms of thinking of the Mathematical. In the case of rhythmic, these forms of thought become forms of mathematics if uniformity, succession, and distinctness of rhythm-parts are defined in the sense of an *established semantics of mathematics*. The metric fixation of rhythmic in musical notation may definitely be understood as such a semantically abstracting mathematization. However the musical interpretations usually liberate from the rigid mathematical structure by their agogics and accentuations. Therefore the Mathematical does not disappear by mathematizing the rhythmic, but keeps preserved in its autonomous independence.

Fig. 1. The Adagio of Mozart's string quartet C major (KV 465)

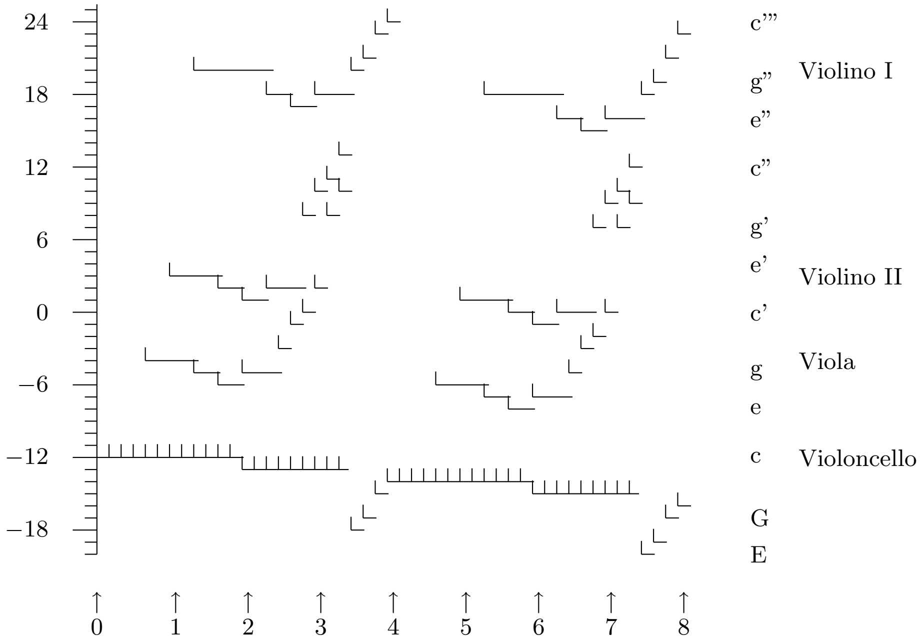


Fig. 2. A mathematical representation of the first eight bars of the Adagio of Mozart's string quartet C major (KV 465)

The interplay between the Mathematical in music and the mathematization of music shall be demonstrated here by the Adagio of Mozart's string quartet C major (KV 465). The score of the Adagio - presented in Fig. 1 - shows that the Adagio consists only of 22 bars in which astonishingly many dissonances occur, but which finally leads to the light C major clearness of the following Allegro.

The result of a mathematization of the first eight bars of the Adagio is shown in Fig. 2. The presented mathematical structure shall be considered as embedded into a two-dimensional real vector space. The part of its vertical axis from -20 to 25 is visible on the left of the diagram (the numbers -18, -12, -6, 0, 6, 12, 18, 24 shall help to identify the integer locations on the vertical axis). There is a one-to-one correspondence between the integers of the vertical axis from -20 to 25 and the tones of the chromatic scale from E to $c'''\#$. Fig. 2 indicates the part of this correspondence which horizontally links the numbers $-20 < -17 < -12 < -8 < -5 < 0 < 4 < 7 < 12 < 16 < 19 < 24$ to the tones of the C major triad $E < G < c < e < g < c' < e' < g' < c'' < e'' < g'' < c'''\#$

The location of the integers 0, 1, ... , 8 on the (imaginary) horizontal axis are indicated by the numbers on the bottom of the diagram (the smallest unit for the horizontal numbers is one sixth, in numerals: 1/6). The horizontal straight line segments on the right of the vertical axis, closed on the left end and open on the right end, represent the sounds of the four instruments with their pitches, respectively (the pitch of such a line segment is determined by the height of the line segment measured by the vertical axis). The small vertical line segments on the right of the vertical axis and the line segment between the points (0,-12) and (0,-11) indicate the beginning of the sound belonging to the horizontal line segment connected at the bottom of that small vertical line segment. The union of all those line segments can be divided into four disjoint subsets corresponding exactly to the four instruments Violino I, Violino II, Viola, and Violoncello (notice that the representations of the sounds beginning at the points (18/6,2), (19/6,8), (20/6,10) and (42/6,0), (43/6,6), (44/6,8) belong to the Viola subset, but not to the Violino II subset). Thus, the structure of those four subsets determines the mathematical representation of the first eight bars of the Adagio. It is not difficult to extend this representation to a mathematical representation of the whole Adagio.

Although the discussed mathematical description of the tones of the Adagio by their pitch, length, location, and instrument are in one-to-one correspondence to the notes of the score presented in Fig. 1, there are more signatures in the score concerning tempo, loudness, crescendo, and bows which are not mathematized. Above all the expressive interpretations of a score by rhythm, agogics, accentuations etc. are far away from a meaningful mathematization. That, in particular, the rhythm evades any mathematical description becomes clear by the following quotation: "The rhythm comprises the order, division, and meaningful arrangement of the time development of sound events. In spite of the tendency, created by the rhythm, to return to the same or the similar, the rhythm should not be confused with the metre and beat because just the vivid differences of the courses of time make possible the musical manifoldness of the rhythms which first of all appear through graded durations of sounds and accents, but also through melodic movements, changing sounds and tone colours, changes of tempo and loudness, phrasing and articulation" ([2], p.656).

3 The Mathematical in the Doctrine of Music Forms

According to the Composer György Ligeti: "The combination of association, abstraction, remembrance, and prevision let only actually achieve the suggestiveness which makes possible the conception of a musical form" ([1], p.9). Without the principle of order, the musical forms would be neither communicable nor apperceivable. Clear orders and relationships are a criterion of its conceivability and indispensible assumption for its understanding. The smallest units of musical sense are the so-called *"motives"* which are understood as the smallest meaningful elements of musical compositions. Motives join up with their own transformations and other motives to larger parts which might be again only parts of a larger whole (cf. [1], p.16f).

For understanding the Mathematical in music forms, it might be helpful to analyse the multitude of music forms in the Adagio presented in Fig. 1 (cf. ([14], p.446). As a whole the music form is an *introduction* to the Allegro, the first movement of the C major string quartet (KV 465). The introduction divides into two parts each of which has 11 bars; the *first part* is polyphonic, the *second* *part* is homophonic. The violoncello starts the Adagio with eighth notes repeated through all the eleven bars of the first part, interrupted only by a *four notes motive* chromatically ascending at the end of the fourth bar and the eighth bar, respectively. After the first four eigth notes of the violoncello the other three instruments present a *theme* which divides into two motives each of which consisting of four notes, where the viola starts at the end of the first bar, the violino II one quarter note later, and the violino I again one quarter note later. The first chord of the four instruments combining the notes c - g - e'♭ - a" contains the two surprising dissonances g - a" and e'♭ - a" and allowed in the following further dissonances until the second motive occurs in combing consonant chords. Starting from the fifth bar, the first four bars are repeated always a major note downwards. The last three bars of the first part of the Adagio function as a bridge to the second part in which the four instruments play the same role between each other in diminishing the motives.

The example shows that the *mathematization of music forms* can use in addition to the descriptive dimensions pitch, length, location, and instrument also the dimension *"music form"*. In our example Fig. 1 we can consider as music forms the whole Adagio, the disjunctive two parts of the Adagio which cover the Adagio, smaller meaningful parts such as periods, themes, phrases, motives, scales, harmonies, chords, tones etc. Many of those music forms of the Adagio can be mathematically represented by a subset of the two-dimensional vector space sketched in Fig. 2; for instance:

- the first motive of the theme presented first for the viola,
- the first theme presented first for the viola,
- the first motive of the theme presented first for the violino II,
- the first theme presented first for the violino II,
- the first motive of the theme presented first for the violino I,
- the first theme presented first for the violino I,
- the first motive of the theme presented secondly for the viola,
- the first theme presented first for the viola,
- the first motive of the theme presented secondly for the violino II,
- the first theme presented first for the violino II,
- the first motive of the theme presented secondly for the violino I,
- the first theme presented first for the violino I,
- the first four tone motive ending with B presented for the violoncello,
- the second four tone motive ending with B presented for the violoncello.

The mathematical description of music forms may extend the mathematization of structures determined by the dimensions of pitch, length, location, and instrument as, for example, presented in Fig. 2. Nevertheless, the expressive interpretations of musical scores are still not in reach to be completely mathematized. Thus, there is still quite a distance between the Mathematical and the more abstract mathematization, but further attempts of diminishing the distance can be elaborated of which two approaches shall be briefly mentioned.

In the doctrine of music forms, symmetries play a special role for which the form of thinking *"equality of parts as expression of a whole"* (cf. [19]) can be assumed to belong to the Mathematical. This phenomenological form of thinking finds its abstraction in mathematics by the mathematical concepts of *symmetry transformation*" and *"symmetry group*", respectively. A direct correspondence between the phenomenological and the mathematical form of thinking regarding compositions is almost only given by strong canons. But if one weakens the mathematical concept of symmetry transformation to a concept of partial symmetry transformation, then considerably more correspondencies could be identified.

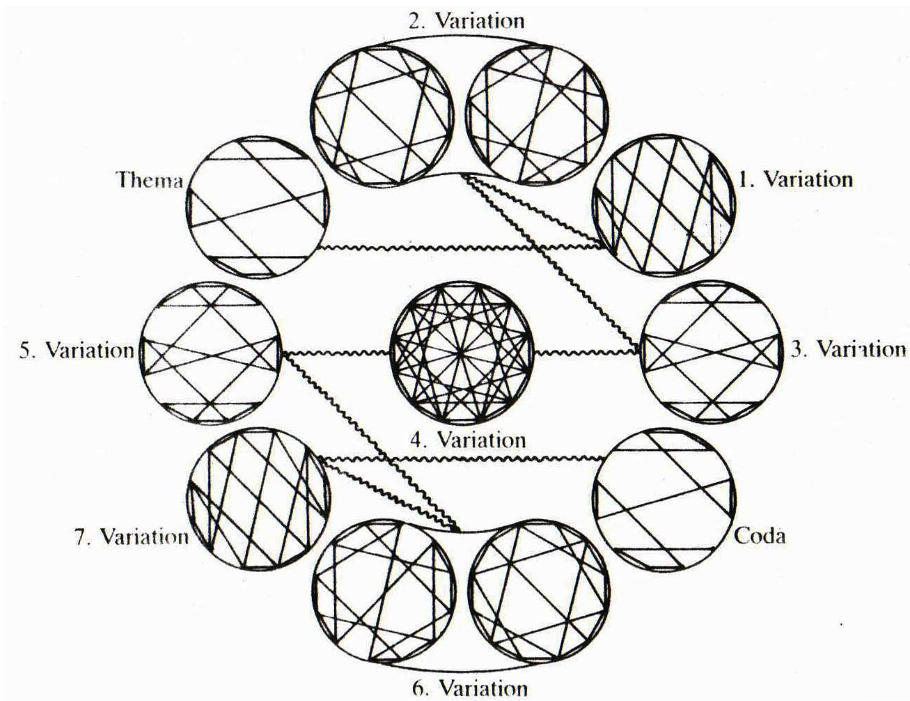


Fig. 3. The symmetry structure of the second movement of Anton Webern's Symphony op. 21

As another generalization of the mathematical form of symmetry, the twelve-tone music used more general symmetries which view octave tones to be structurally identified. The example shown in Fig. 3 represents twelve tone rows by a sequence of eleven straight sections on a circle. Each circle presents at least one symmetry and all circles together are arranged in such a way that a 180° rotation maps the total picture onto itself. Musically this indicates that the total symphony is a transposition of its retrogression.

The composer Fred Lerdahl and the linguist Ray Jackendoff have elaborated a much more far-reaching approach to formally grasping forms of music which was published in their book *"A generative theory of tonal music*" [13]. For this, they developed a generative grammar of music, which was inspired by Chomsky's linguistic transformation grammar, but developed purely within music thinking.

As fundamental components of the musical understanding of a composition they considered grouping structures of subunits of the composition. For these grouping structures the form of thinking *"division of a whole into subunits"* can be assumed to belong to the Mathematical and abstracted to a mathematical structure of a weighted ordered set. Lerdahl and Jackendoff impressively demonstrate their theory by many examples, as fore instance by the beginning of Mozart's Symphony G minor, KV 550.

4 The Mathematical in the Theory of Tonal Systems

Tonal systems, which serve as foundation of music thinking, rest thoroughly on different forms of thinking of the Mathematical:

- Behind the *tonal system of the equal-tempered keyboard*, there is the form of thinking of a musical scale consisting of 7 white keys with the steps whole-whole-half-whole-whole-whole-half which are completed by 5 black keys to a musical scale with 12 half steps.
- The *tonal systems of musical instruments with finger-board* suggest a form of thinking which relates to finger positions; for example, the player of a violin thinks especially which finger has to be placed on which string in which position .
- The *tonal system of the names of tones* obtains its form by the names of the 12 octave tones c - c# - d - e♭ - e - f - f# - g - a♭ - a - b♭ - b which are rising by half-tone steps; adding # or ♭ to a tone name yields the name of a tone which is a half-tone higher or lower, respectively.
- The *tonal system of the standard notation* is founded on the form of the 5+5-line system with additional ledger lines, in which the tones are represented by note-heads with and without accidentals on and between the lines; the tone distances describable in this way are multiples of half-tone steps.
- The *harmonic tone system* extends the form of the tone system of tone names by adding integer exponents to the tone names; a tone name t^z represents a tone which is z-many syntonic commas higher or lower than the tone t^0, respectively (syntonic comma := 4 fifth − 2 octaves − 1 major third; multiplicatively, the syntonic comma is the frequency ratio 81 : 80 obtained by computing $((3:2)^4 : (2:1)^2) : (5:4))$ where the frequency ratio 3 : 2 represents the fifth, the ratio 2 : 1 the octave and the ratio 5 : 4 the major third).

Here only the *harmonic tone system* shall be further discussed. In Fig. 4, this system is represented by a tone net in just intonation which is freely generated by the perfect fifth 3 : 2 and the perfect major third 5 : 4 (modulo the octave 2 : 1). Leonhard Euler was the first who published such a tone net which he named *speculum musicum* [6]. Following Euler's idea, realizations of the harmonic tone system on musical instruments have been approached again and again (for an overview about those attempts see [18]). In particular, the

e^{-2}	b^{-2}	$f\#^{-2}$	$c\#^{-2}$	$g\#^{-2}$	$d\#^{-2}$	$a\#^{-2}$	$e\#^{-2}$	$b\#^{-2}$
c^{-1}	g^{-1}	d^{-1}	a^{-1}	e^{-1}	b^{-1}	$f\#^{-1}$	$c\#^{-1}$	$g\#^{-1}$
$a\flat^{0}$	$e\flat^{0}$	$b\flat^{0}$	f^{0}	c^{0}	g^{0}	d^{0}	a^{0}	e^{0}
$f\flat^{+1}$	$c\flat^{+1}$	$g\flat^{+1}$	$d\flat^{+1}$	$a\flat^{+1}$	$e\flat^{+1}$	$b\flat^{+1}$	f^{+1}	c^{+1}
$d\flat\flat^{+2}$	$a\flat\flat^{+2}$	$e\flat\flat^{+2}$	$b\flat\flat^{+2}$	$f\flat^{+2}$	$c\flat^{+2}$	$g\flat^{+2}$	$d\flat^{+2}$	$a\flat^{+2}$

Fig. 4. The tone net of the harmonic tone system

instrument MUTABOR should be mentioned which even allows to realize arbitrary mutating pitches of tones in just intonation, but also in any other form of intonation (see [7], [22]).

Although performing music pieces in just intonation is an ideal for many music ensembles (for instance for a string quartet), there are problems of being consistent with the intonation. This shall be briefly explained by the so-called *Problem of the Harmony of Second Degree* illustrated in the harmonic tone system shown in Fig. 5 (cf. [21], p.197f). The figure represents a musical cadence formed by five perfect triads starting with the major triad $\underline{c}^0$ and ending with the major triad $\underline{c}^{-1}$. More precisely,

- the major triad $\underline{c}^0$ meets the major triad $\underline{f}^0$ in the note c,
- the major triad $\underline{f}^0$ meets the minor triad $\underline{d}^{-1}$ in the notes f^0 and a^{-1},
- the minor triad $\underline{d}^{-1}$ meets the major triad $\underline{g}^{-1}$ in the note d^{-1}, and
- the major triad $\underline{g}^{-1}$ meets the major triad $\underline{c}^{-1}$ in the note g^{-1}.

Playing a cadence as described above, musicians usually have the tendency to end with the same chord as they started with, i.e. with the major triad $\underline{c}^0$. Then, of course, they have to modify the pitches in between, but still to produce perfect triads. Cadences with such intonations defy convincing mathematization so that it would be interesting to find out how much *the Mathematical* could contribute to overcome those vaguenesses.

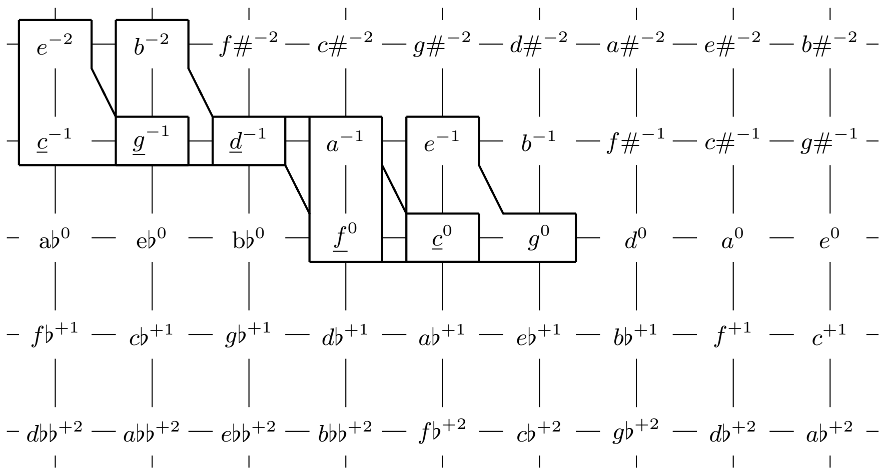


Fig. 5. A musical cadence leading from the major triad $\underline{c}^0$ in four steps via the major triad $\underline{f}^0$, minor triad $\underline{d}^{-1}$, and the major triad $\underline{g}^{-1}$ to the major triad $\underline{c}^{-1}$

5 Semantic Logic in Music Thinking and Its Semantology

A basic question is how to support our understanding of the Mathematical in music. Since the Mathematical is more abstract than logic which itself is more abstract than music, the study of logic in music may particularly contribute to a better understanding of the Mathematical in music thinking.

It is common sense that humans may be affected by music so that it reaches human feelings, emotions, and thought. Humans can even be deeply moved by music, particularly by its musical senses and meanings which may be represented by semantic structures in music (cf. [25]). Now, such structures could be abstracted to semantic structures in logic. For instance, the chords of a well-tempered piano can be abstracted to a logic structure which represents the possible interactions and relationships between those chords.

The result of all such abstractions has been named by the musicologist C. Dahlhaus *"musical logic"* which he characterized by the compositional, technical and esthetic moments which made the automation of instruments possible. Dahlhaus saw the musical logic closely related to the idea of the "language character" of music. That music is presented as sounding discourse, as development of musical thought, is the justification of its esthetic claim, that music is there to be heard for the sake of itself (see [3], p.105f). The richness of this understanding of musical logic is an important assumption for a better understanding of the Mathematical in music thinking.

To obtain even more insights into the Mathematical in music thinking, a further development of the recently introduced *"semantology of music"* could be helpful (cf. [25]). In particular, its philosophic-logical level is basic for the analysis of the Mathematical because philosophical concepts with their objects, their attributes, and their relationships" are highly abstract, but still deduced from actual realities (cf. [10], [5]). The supportive mathematical level is already elaborated to a great extent by methods of *Formal Concept Analysis* (see [8], [9], [24]).

References

1. Altmann,G.: Musikalische Formenlehre. Schott Musik International, Mainz 2001.
2. Der Brockhaus Musik - Personen, Epochen, Sachbegriffe. 2. Auflage. Brockhaus,F.A., Mannheim-Leipzig 2001.
3. Dahlhaus,C.: Die Idee der absoluten Musik. Bärenreiter Verlag, Kassel 1978.
4. Eggebrecht, H. H.: *Musikverstehen.* Piper, München 1995.
5. Eklund, P., Wille, R.: Semantology as basis for Conceptual Knowledge Processing. In: Kuznetsov, S. O., Schmidt, St. (eds.): *Formal Concept Analysis. ICFCA 2007.* LNAI **4390**. Springer, Heidelberg 2007, 18–38.
6. Euler, L.: De harmoniae veris principiis per speculum musicum repraesentatis. In: Leonhardi Euleri Opera Omnia. Serie III, Band 1. Leipzig, Berlin 1926, 568–586.
7. Ganter, B., Henkel, H., Wille, R.: MUTABOR - Ein rechnergesteuertes Musikinstrument zur Untersuchung von Stimmungen. *Acustica* 57 (1985), 284–289.
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg 1999.
9. Ganter, B., Stumme, G., Wille, R. (Eds.): Formal Concept Analysis: foundations and applications. State-of-the-Art Survey. LNAI **3626**. Springer, Heidelberg 2005.
10. Gehring, P., Wille, R.: Semantology: basic methods for knowledge representations. In: H. Schärfe, Pascal Hitzler, Peter Øhrstrøm (eds.): *Conceptual structures: inspiration and application.* LNAI **4068**. Springer, Heidelberg 2006, 215–228.
11. Heidegger, M.: Die Frage nach dem Ding. 3. Aufl. Max Niemeyer Verlag, Tübingen 1987.
12. Kant, I.: Logic. Dover, Mineola 1988.
13. Lerdahl, F., Jackendoff, R.: A generative theory of tonal music. The MIT Press, Cambridge Mass. 1983.
14. Leopold, S. (Hrsg.): Mozart Handbuch. Bärenreiter, Kassel 2005.
15. Peirce, Ch. S.: Reasoning and the logic of things. Edited by K.L. Ketner, with an introduction by K. L. Ketner and H. Putnam. Havard Univ. Press, Cambridge 1992.
16. Peirce, Ch. S.: The three normative sciences. In: *The Essential Peirce. Selected philosophical writings.* Vol.2, edited by the Peirce Edition Project. Indiana University Press, Bloomington 1998.
17. Reinecke, H.-P.: Über die Eigengesetzlichkeit des musikalischen Hörens und die Grenzen der naturwissenschaftlichen Akustik. In: B. Dopheide (Hrsg.): Musikhören. Wissenschaftliche Buchgesellschaft, Darmstadt 1975, 223–241.
18. Vogel, M.: Die Lehre von den Tonbeziehungen. Verlag systematische Musikwissenschaft, Bonn 1975.
19. Wille, R.: Symmetrien in der Musik - für ein Zusammenspiel von Musik und Mathematik. *Neue Zeitschrift für Musik* **143** (1982), 12–19.

20. Wille, R.: Musiktheorie und Mathematik. In: H. Götze, R. Wille (Hrsg.): *Musik und Mathematik - Salzburger Musikgespräch 1984 unter Vorsitz von Herbert von Karajan*. Springer-Verlag, Heidelberg 1985, 4–31.
21. Wille, R.: Triadic concept graphs. In: M.-L. Mugnier, M. Chein (eds.): Conceptual structures: theory, tools and applications. ICCS 1998. LNAI **1433**. Springer, Heidelberg 1998, 195–208.
22. Wille, R.: Eulers Speculum Musicum und das Instrument MUTABOR. *DMV-Mitteilungen*, Heft 4/2000, 9–12.
23. Wille, R.: Mensch und Mathematik: Logisches und mathematisches Denken. In: Lengnink, K., Prediger, S., Siebel, F. (Hrsg.): Mathematik und Mensch: Sichtweisen der Allgemeinen Mathematik. Verlag Allgemeine Wissenschaft, Mühltal 2001, 141–160.
24. Wille, R.: Methods of Conceptual Knowledge Processing. In: R. Missaoui, J. Schmid (eds.): *Formal Concept Analysis. ICFCA 2006.* LNAI **4068**. Springer, Heidelberg 2006, 1–29.
25. Wille, R., Wille-Henning, R.: Towards a semantology of music. In: Priss, U., Polovina, S., Hill, R. (eds.): Formal Concept Analysis. ICFCA 2007. LNAI **4604**. Springer, Heidelberg 2007, 303–312.

An Application of Formal Concept Analysis to Neural Decoding

Dominik Endres[1], Peter Földiák[1], and Uta Priss[2]

[1]School of Psychology, University of St. Andrews, {dme2,pf2}@st-andrews.ac.uk
[2]School of Computing, Napier University, u.priss@napier.ac.uk

Abstract. This paper proposes a novel application of Formal Concept Analysis (FCA) to neural decoding: the semantic relationships between the neural representations of large sets of stimuli are explored using concept lattices. In particular, the effects of neural code sparsity are modelled using the lattices. An exact Bayesian approach is employed to construct the formal context needed by FCA. This method is explained using an example of neurophysiological data from the high-level visual cortical area STSa. Prominent features of the resulting concept lattices are discussed, including indications for a product-of-experts code in real neurons.

1 Introduction

Mammalian brains consist of billions of neurons, each capable of independent electrical activity. From an information-theoretic perspective, the patterns of activation of these neurons can be understood as the codewords comprising the neural code. The neural code describes which pattern of activity corresponds to what information item. We are interested in the (high-level) visual system, where such items may indicate the presence of a stimulus object or the value of some stimulus attribute, assuming that each time this item is represented the neural activity pattern will be the same or at least similar. *Neural decoding* is the attempt to reconstruct the stimulus from the observed pattern of activation in a given population of neurons [1,2,3,4]. Popular decoding quality measures, such as Fisher's linear discriminant [5] or mutual information [6] capture how accurately a stimulus can be determined from a neural activity pattern (e.g., [4]). While these measures are certainly useful, they provide little information about the structure of the neural code, which is what we are concerned with here. Furthermore, we would also like to elucidate how this structure relates to the represented information items, i.e. we are interested in the semantic aspects of the neural code.

To explore the relationship between the representations of related items, Földiák [7] demonstrated that a sparse neural code can be interpreted as a graph (a kind of "semantic net"). Each codeword can then be represented as a set of active units (a subset of all units). The codewords can now be partially ordered under set inclusion: codeword $A \leq$ codeword B iff the set of active neurons

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 181–192,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

of A is a subset of the active neurons of B. This ordering relation is capable of capturing semantic relationships between the represented information items. There is a duality between the information items and the sets representing them: a more general class corresponds to a smaller subset of active neurons, and more specific items are represented by larger sets [7]. Additionally, storing codewords as sets is especially efficient for sparse codes, i.e. codes with a low activity ratio (i.e. few active units in each codeword). These findings by Foldiak [7] did not employ the terminology and tools of Formal Concept Analysis (FCA) [8,9]. But because this duality is a Galois connection, it is of interest to apply FCA to such data. The resulting concept lattices are an interesting representation of the relationships implicit in the code.

We would also like to be able to represent how the relationship between sets of active neurons translates into the corresponding relationship between the encoded stimuli. In our application, the stimuli are the formal objects, and the neurons are the formal attributes. The FCA approach exploits the duality of extensional and intensional descriptions and allows to visually explore the data in lattice diagrams. FCA has shown to be useful for data exploration and knowledge discovery in numerous applications in a variety of fields [10,11].

This paper does not include an introduction to FCA because FCA is well described in the literature (e.g., [9]). We use the phrase *reduced labelling* to refer to line diagrams of concept lattices which have labels only attached to object concepts and attribute concepts. As a reminder, an object concept is the smallest (w.r.t. the conceptual ordering in a concept lattice) concept of whose extent the object is a member. Analogously, an attribute concept is the largest concept of whose intent the attribute is a member. *Full labelling* refers to line diagrams of concept lattices where concepts are depicted with their full extent and intent.

We provide more details on sparse coding in section 2 and demonstrate how the sparseness (or denseness) of the neural code affects the structure of the concept lattice in section 3. Section 4 describes the generative classifier model which we use to build the formal context from the responses of neurons in the high-level visual cortex of monkeys. Finally, we discuss the concept lattices so obtained in section 5.

2 Sparse coding

One feature of neural codes which has attracted a considerable amount of interest is its *sparseness*. As detailed in [12], sparse coding is to be distinguished from local and dense distributed coding. At one extreme of low average activity ratio are local codes, in which each item is represented by a separate neuron or a small set of neurons. This way there is no overlap between the representations of any two items in the sense that no neuron takes part in the representation of more than one item. An analogy might be the coding of characters on a computer keyboard (without the Shift and Control keys), where each key encodes a single character. It should be noted that locality of coding does not necessarily imply that only one neuron encodes an item, it only says that the neurons are highly selective, corresponding to single significant items of the environment (e.g. a "grandmother cell" - a hypothetical neuron that has the exact and only purpose to be activated when someone sees, hears or thinks about their grandmother).

The other extreme (approximate average activity ratio of 0.5) corresponds to dense, or *holographic* coding. Here, an information item is represented by the combination of activities of all neurons. For N binary neurons this implies a representational capacity of 2^N. Given the billions of neurons in a human brain, 2^N is beyond astronomical. As the number of neurons in the brain (or even just in a single cortical area, such as primary visual cortex) is substantially higher than the number of receptor cells (e.g. in the retina), the representational capacity of a dense code in the brain is much greater than what we can experience in a lifetime (the factor of the number of moments in a lifetime adds the requirement of only about 40 extra neurons). Therefore the greatest part of this capacity is redundant.

Sparse codes (small average activity ratio) are a favourable compromise between dense and local codes ([13]). The small representational capacity of local codes can be remedied with a modest fraction of active units per pattern because representational capacity grows exponentially with average activity ratio (for small average activity ratios). Thus, distinct items are much less likely to interfere when represented simultaneously. Furthermore, it is much more likely that a single layer network can learn to generate a target output if the input has a sparse representation. This is due to the higher proportion of mappings being implementable by a linear discriminant function. Learning in single layer networks is therefore simpler, faster and substantially more plausible in terms of biological implementation. By controlling sparseness, the amount of redundancy necessary for fault tolerance can be chosen. With the right choice of code, a relatively small amount of redundancy can lead to highly fault-tolerant decoding. For instance, the failure of a small number of units may not make the representation undecodable. Moreover, a sparse distributed code can represent values at higher accuracy than a local code. Such distributed coding is often referred to as coarse coding.

3 Concept lattices of local, sparse and dense codes

In the case of a binary neural code, the sparseness of a codeword is inversely related to the fraction of active neurons. The average sparseness across all codewords is the sparseness of the code [13,12]. Sparse codes, i.e. codes where this fraction is low, are found interesting for a variety of reasons: they offer a good compromise between encoding capacity, ease of decoding and robustness [14]; they seem to be employed in the mammalian visual processing system [15]; and they are well suited to representing the visual environment we live in [16,17]. It is also possible to define sparseness for graded or even continuous-valued responses (see e.g. [18,4,12]). To study what structural effects different levels of sparseness would have on a neural code, we generated random codes, i.e. each of 10 stimuli was associated with randomly drawn responses of 10 neurons, subject to the constraints that the code be perfectly decodable and that the sparseness of each codeword was equal to the sparseness of the code. Fig.1 shows the contexts (represented as cross-tables) and the concept lattices of a local code (activity ratio 0.1), a sparse code (activity ratio 0.2) and a dense code (activity ratio 0.5). In a local code, the response patters to different stimuli have no overlapping activations, hence the lattice representing this code is an anti-chain with top and bottom element added. Each concept in the anti-chain introduces (at least) one stimulus and (at least) one neuron. In contrast, a dense code results in a larger number of concepts which introduce neither a stimulus nor a neuron. The lattice of the dense code also contains substantially longer chains between the top and bottom nodes than in the case of sparse and local codes.

	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10
s1	×									
s2			×							
s3		×								
s4						×				
s5									×	
s6					×					
s7				×						
s8							×			
s9								×		
s10										×

	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10
s1					×				×	
s2			×			×				
s3		×					×			
s4				×						×
s5	×								×	
s6			×		×					
s7		×		×						
s8			×						×	
s9		×				×				
s10	×							×		

	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10
s1	×		×	×			×			×
s2	×	×		×			×		×	
s3		×	×	×				×		×
s4	×		×			×			×	×
s5		×			×	×	×			×
s6		×	×	×			×	×		
s7	×		×		×	×			×	
s8	×		×		×				×	×
s9	×	×			×	×		×		
s10			×	×	×		×			×

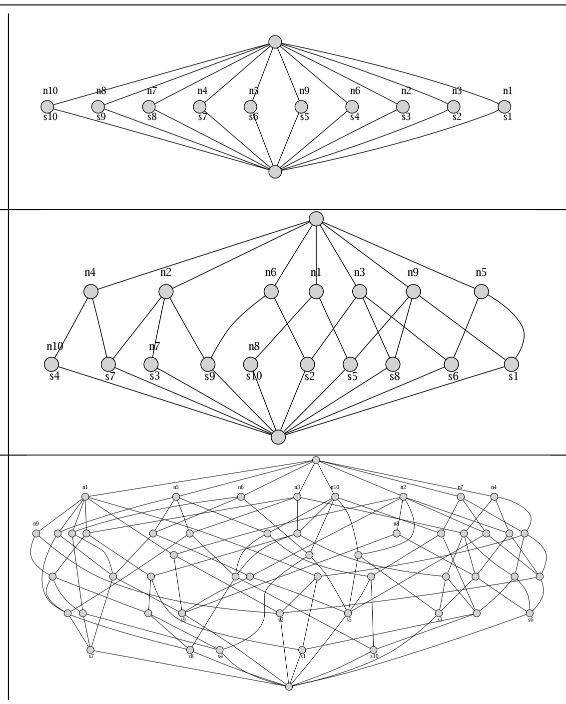


Fig. 1. Contexts (represented as cross-tables) and concept lattices for a local, sparse and dense random neural code. Each context was built out of the responses of 10 (hypothetical) neurons (n1, ..., n10) to 10 stimuli (s1, ..., s10). Each node represents a concept.

The most obvious differences between the lattices is the total number of concepts. A dense code, even for a small number of stimuli, will give rise to a large number of concepts, because the neuron sets representing the stimuli are very probably going to have non-empty intersections. These intersections are potentially the intents of concepts which are larger than those concepts that introduce the stimuli. Hence, the latter are found towards the bottom of the lattice. This implies that they have large intents, which is of course a consequence of the density of the code. Determining these intents thus requires the observation of a large number of neurons, which is unappealing from a decoding perspective. The local code does not have this drawback, but is hampered by a small encoding capacity (maximal number of concepts with non-empty extents): the concept lattice in fig.1 is the largest one which can be constructed for a local code comprised of 10 binary neurons. Which of the above structures is most appropriate depends on the conceptual structure of the environment to be encoded and the appropriate sparseness that can be selected.

4 Building a formal context from responses of high-level visual neurons

To explore whether FCA is a suitable tool for interpreting real neural codes, we constructed formal contexts from the responses of high-level visual cortical cells in area STSa (part of the temporal lobe) of monkeys. Characterising the responses of these cells is a difficult task. They exhibit complex nonlinearities and invariances which make it impossible to apply linear techniques, such as reverse correlation [19], that were shown to be useful in understanding the responses of neurons in early visual areas [20,21]. The concept lattices obtained by FCA might enable us to display and browse these invariances: if the response of a subset of cells indicates the presence of an invariant feature in a stimulus, then all stimuli having this feature should form the extent of a concept whose intent is given by the responding cells.

4.1 Physiological data

The data were obtained through [22], where the experimental details can be found. Briefly, spike trains were obtained from single neurons within the upper and lower banks of the superior temporal sulcus (STSa) of an awake and behaving monkey (*Macaca mulatta*) via standard extracellular recording techniques [23]. During the recordings, the monkey had to perform a fixation task. This area contains cells which are responsive to faces. Extracellular voltage fluctuations were measured, and the stereotypical action potentials (i.e. 'spikes') of the neuron were detected and their temporal sequence was recorded resulting in a 'spike train'. These spike trains were turned into distinct samples, each of which contained the spikes from -300 ms before to 600 ms after the stimulus onset with a temporal resolution of 1 ms. The stimulus set consisted of 1704 images, containing colour and black and white views of human and monkey head and body, animals, fruits, natural outdoor scenes, abstract drawings and cartoons. Stimuli were presented for 55 ms each without inter-stimulus gaps in random sequences. While this rapid serial visual presentation (RSVP) paradigm complicates the task of extracting stimulus-related information from the spike trains, it has the advantage of allowing for the testing of a large number of stimuli. A given cell was tested on a subset of 600 or 1200 of these stimuli, each stimulus was presented between 1-15 times.

The data were previously analysed with respect to the stimulus selectivity of individual cells only. Previous neural population decoding studies were aimed at identifying stimulus lables (e.g. [2,3]) only. This paper presents the first analysis of the semantic structure of neural data with FCA.

4.2 Bayesian thresholding

In order to apply FCA, we extracted a binary attribute from the raw spike trains. We could use many-valued attributes to describe the neural response, but we will employ a simple binary thresholding as a starting point. This binary attribute should be as informative about the stimulus as possible, to allow for the construction of meaningful concepts. We do this by Bayesian thresholding, as detailed below. This procedure also avails us of a null hypothesis $H_0 =$"the responses contain no information about the stimuli".

A standard way of obtaining binary responses from neurons is thresholding the spike count within a certain time window. This is a relatively straightforward task, if the stimuli are presented well separated in time and a large number of trials per stimulus are available. Then latencies and response offsets are often clearly discernible and thus choosing the time window is not too difficult. However, under RSVP conditions with few trial per stimulus, response separation becomes more tricky, as the responses to subsequent stimuli will tend to follow each other without an intermediate return to baseline activity. Moreover, neural responses tend to be rather noisy. We will therefore employ a simplified version of the generative Bayesian Bin classification algorithm (BBCa) [24], which was shown to perform well on RSVP data [25].

BBCa was designed for the purpose of inferring stimulus labels g from a continuous-valued, scalar measure z of a neural response. The range of z is divided into a number of contiguous bins. Within each bin, the observation model for the g is a Bernoulli scheme with G types and with a Dirichlet prior over its parameters. It is shown in [24] that one can iterate/integrate over all possible bin boundary configurations efficiently, thus making exact Bayesian inference feasible. Moreover, the marginal likelihood (or model evidence) becomes thus available, which can be used to infer the posterior distribution over all spike counting windows. We make two simplifications to BBCa: 1) z is discrete, because we are counting spikes and 2) we use models with only 1 bin boundary Z_0 in the range r of z, i.e.

$$P(g = l_i | z = z_i) = \begin{cases} p_{l_i} \text{ if } z_i \leq Z_0 \\ q_{l_i} \text{ otherwise} \end{cases} \tag{1}$$

$$\sum_g p_g = 1 \ , \ \sum_g q_g = 1 \tag{2}$$

$$p(p_0, \ldots, p_G) = \frac{\Gamma\left(\sum_g \alpha_g\right)}{\prod_g \Gamma(\alpha_g)} \prod_g p_g^{\alpha_g - 1} \tag{3}$$

$$p(q_0, \ldots, q_G) = \frac{\Gamma\left(\sum_g \beta_g\right)}{\prod_g \Gamma(\beta_g)} \prod_g q_g^{\beta_g - 1} \tag{4}$$

$$p(Z_0) = \frac{1}{|r|}. \tag{5}$$

We have no a priori preferences for any stimulus label, thus we choose $\forall g : \alpha_g = \beta_g = 1$. Since the Dirichlet priors on the p_g and q_g are conjugate to the likelihood of the data (eqn.(1)), the posteriors can be computed in closed form. Further details of the posterior computation after observing a set of stimulus-response pairs (l_i, z_i) are analogous to [24].

The bin membership (higher bin = stimulus has attribute) of a given neural response can then serve as the binary attribute required for FCA, since BBCa weighs bin configurations by their classification (i.e. stimulus label decoding) performance. We proceed in a straight Bayesian fashion: since the bin membership is the only variable we are interested in, all other parameters (counting window size and position, class membership probabilities, bin boundaries) are marginalised. This minimises the risk of spurious results due to "contrived" information (i.e. choices of parameters) made at some stage of the inference process. Afterwards, the probability that the response belongs to the upper bin is thresholded at a probability of 0.5, i.e. if the probability is larger than 0.5, then there will be a cross in the context. Instead of this simple binarisation, other methods of conceptual scaling could be used.

Since BBCa yields exact model evidences, it can also be used for model comparison. Running the algorithm with no bin boundaries in the range of z effectively yields the probability of the data given the "null hypothesis" H_0: z does not contain any information about g. We can then compare it against the alternative hypothesis described above (i.e. the information which bin z is in tells us something about g) to determine whether the cell has responded at all.

4.3 Cell selection

The experimental data consisted of recordings from 26 cells. To minimise the risk that the computed neural responses were a result of random fluctuations, we excluded a cell if 1) H_0 was more probable than 10^{-6} or 2) the posterior standard deviations of the counting window parameters were larger than 20 ms, indicating large uncertainties about the response timing. Cells which did not respond above the threshold included all cells excluded by the above criteria (except one). Furthermore, since not all cells were tested on all stimuli, we also had to select tuples of subsets of cells and stimuli such that all cells in a tuple were tested on all stimuli. Incidentally, this selection can also be accomplished with FCA, by determining the concepts of a context with gIm ="stimulus g was tested on cell m" and selecting those with a large number of stimuli × number of cells. One of these cell and stimulus subset pairs (16 cells, 310 stimuli) was selected for further exemplary analysis, but the lattices computed from the other subset pairs displayed similar features.

5 Results

To analyse the neural code, the thresholded neural responses were used to build stimulus-by-cell-response contexts. We performed FCA on these with ColibriConcepts[1], created stimulus image montages[2] and plotted the lattices[3]. In these graphs, the images represent the formal objects. The top of the frame around each concept image contains the concept number and the list of cells in the intent (which, unfortunately, may be difficult to see in the printed version of the graphs. Moreover, the list is truncated if more than 6 cells are in the intent.).

Fig.2 shows a lattice which has an emphasis on "face" and "head" concepts. The concepts introducing human and cartoon faces (i.e. with extents consisting of general "face" images) tend to be higher up in the lattice and their intents tend to be small. In contrast, the lower concepts introduce mostly single monkey faces (and faces of the monkey's caregivers), with the bottom concepts having intents of ≥ 7 cells. We may interpret this as an indication that the neural code has a higher "resolution" for faces of conspecifics (and other "important" faces) than for faces in general, i.e. other monkeys are represented in greater detail in a monkey's brain than humans or cartoons. This feature can be observed in most lattices we generated. Thus, monkey STSa cells are not just responsive to faces in general, but to specific subclasses, such as monkey faces, in particular.

Fig.3 shows a subgraph from a lattice with full labelling. Full labelling is of interest in these applications because viewing the full extent simultaneously gives an impression of "what this concept is about". The concepts in the left half of the graph are face concepts, whereas the extents of the concepts in the right half also contain a number of non-face stimuli. Most of the latter have something "roundish" about them. The bottom concept, being subordinate to both the "round" and the "face" concepts, contains a stimulus with both characteristics, which points towards a product-of-experts (PoE) encoding [26]. In PoE, each 'expert' can be thought of as an attribute (or attribute combination) of the represented item. These experts are expected to correspond to meaningful aspects of the information items. Several examples of this kind can be found in the other graphs of the complete concept lattices, which cannot be included in this paper.

[1] available at http://code.google.com/p/colibri-concepts/
[2] via ImageMagick, available at http://www.imagemagick.org
[3] with Graphviz, available at http://www.graphviz.org

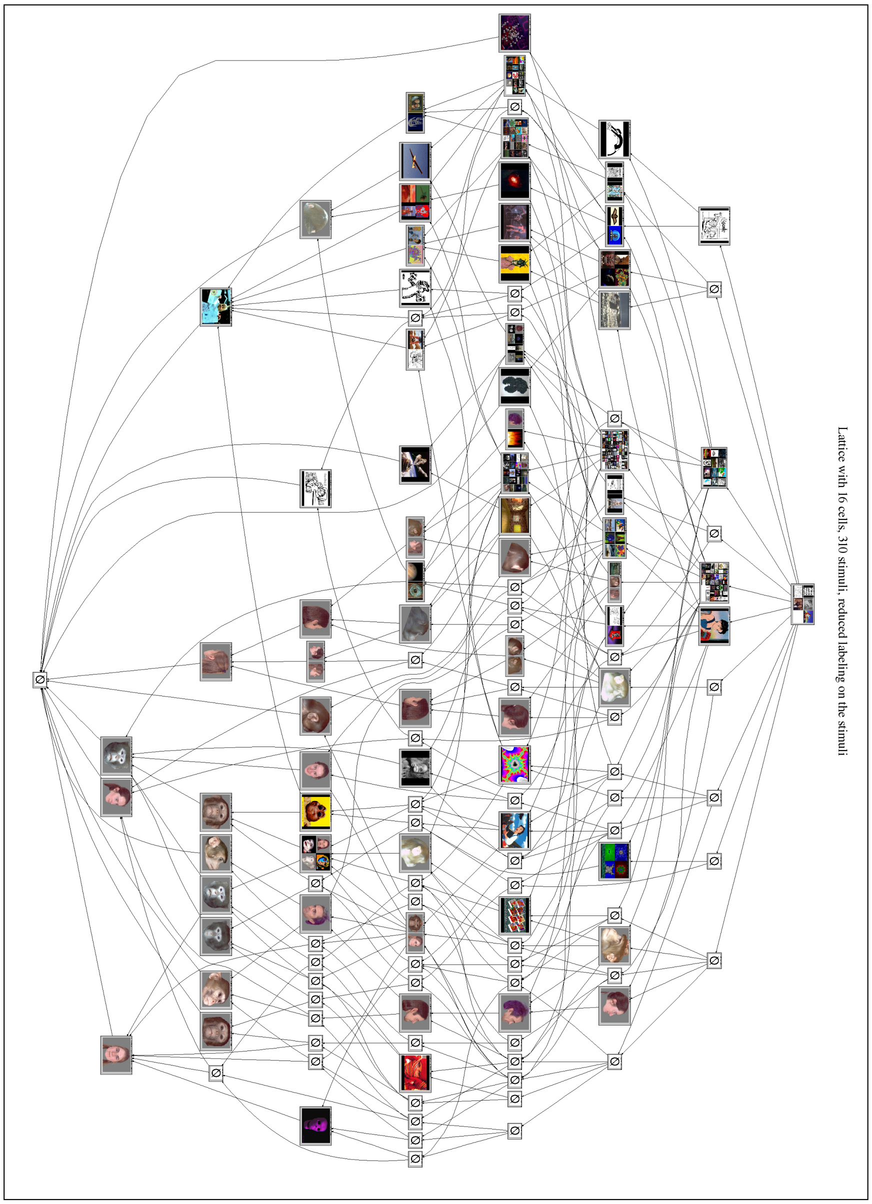


Fig. 2. A lattice with reduced labelling on the stimuli, i.e. stimuli are only shown in their object concepts. The $\emptyset$ indicates that an extent is the intersection of the parent concept extents, i.e. no new stimuli were introduced by this concept.

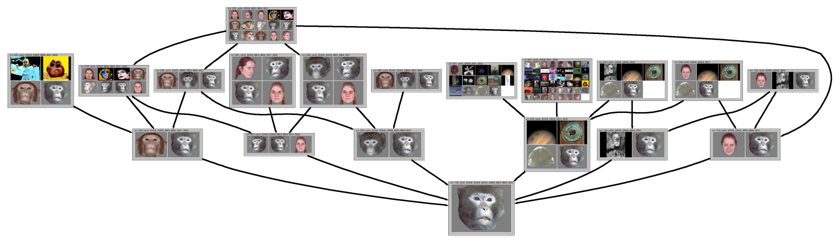


Fig. 3. A subgraph of a lattice with full labelling. The concepts on the right side are not exclusively "face" concepts, but most members of their extents have something "roundish" about them.

6 Conclusion

We demonstrated the potential usefulness of FCA for the exploration and interpretation of neural codes. This technique is feasible even for high-level visual codes, where linear decoding methods [20,21] fail, and it provides qualitative information about the structure of the code which goes beyond stimulus label decoding [1,2,3,4]. The semantic structure of neural data has previously been analysed with tree-based clustering methods [27]. Imposing a tree structure on the data may be inappropriate for neural data that reflects a more general semantic structure, as supported by our results.

Clearly, however, our application of FCA for this analysis is still in its infancy. It would be very interesting to repeat the analysis presented here on data obtained from simultaneous multi-cell recordings, to elucidate whether the conceptual structures derived by FCA are used for decoding by real brains. On a larger scale than single neurons, FCA could also be employed to study the relationships in fMRI data [28].

Acknowledgements D. Endres was supported by MRC fellowship G0501319.

References

1. Georgopoulos, A.P., Schwartz, A.B., Kettner, R.E.: Neuronal population coding of movement direction. Science **233**(4771) (1986) 1416–1419
2. Földiák, P.: The 'Ideal Homunculus': statistical inference from neural population responses. In Eeckmann, F., Bower, J., eds.: Computation and Neural Systems. Kluwer Academic Publishers, Norwell, MA (1993) 55–60
3. Oram, M., Földiák, P., Perrett, D., Sengpiel, F.: The 'Ideal Homunculus': decoding neural population signals. Trends In Neurosciences **21** (June 1998) 259–265
4. Quiroga, R.Q., Reddy, L., Koch, C., Fried, I.: Decoding Visual Inputs From Multiple Neurons in the Human Temporal Lobe. J Neurophysiol **98**(4) (2007) 1997–2007
5. Duda, O., Hart, P., Stork, D.: Pattern classification. John Wiley & Sons, New York, Chichester (2001)

6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York (1991)
7. Földiák, P.: Sparse neural representation for semantic indexing. In: XIII Conference of the European Society of Cognitive Psychology (ESCOP-2003). (2003) http://www.st-andrews.ac.uk/∼pf2/escopill2.pdf.
8. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In Rival, I., ed.: Ordered sets. Reidel, Dordrecht-Boston (1982) 445–470
9. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical foundations. Springer (1999)
10. Ganter, B., Stumme, G., Wille, R., eds.: Formal Concept Analysis, Foundations and Applications. Volume 3626 of Lecture Notes in Computer Science. Springer (2005)
11. Priss, U.: Formal concept analysis in information science. Annual Review of Information Science and Technology **40** (2006) 521–543
12. Földiák, P., Endres, D.: Sparse coding. Scholarpedia **3**(1) (2008) 2984 http://www.scholarpedia.org/article/Sparse_coding.
13. Földiák, P.: Sparse coding in the primate cortex. In Arbib, M.A., ed.: The Handbook of Brain Theory and Neural Networks. second edn. MIT Press (2002) 1064–1068
14. Földiák, P.: Forming sparse representations by local anti-Hebbian learning. Biological Cybernetics **64** (1990) 165–170
15. Olshausen, B.A., Field, D.J., Pelah, A.: Sparse coding with an overcomplete basis set: a strategy employed by V1. Vision Res. **37**(23) (1997) 3311–3325
16. Olshausen, B.: Learning Linear, Sparse, Factorial Codes. Technical Report AIM 1580 (1996)
17. Simoncelli, E.P., Olshausen, B.A.: Natural image statistics and neural representation. Annual Review of Neuroscience **24** (2001) 1193–1216
18. Rolls, E., Treves, A.: The relative advantages of sparse versus distributed encoding for neuronal networks in the brain. Network **1** (1990) 407–421
19. Dayan, P., Abbott, L.: Theoretical Neuroscience. MIT Press, London, Cambridge (2001)
20. Jones, J., Palmer, L.A.: An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology **58**(6) (1987) 1233–1258
21. Ringach, D.L.: Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. Journal of Neurophysiology **88** (2002) 455–463
22. Földiák, P., Xiao, D., Keysers, C., Edwards, R., Perrett, D.: Rapid serial visual presentation for the determination of neural selectivity in area STSa. Progress in Brain Research (2004) 107–116
23. Oram, M.W., Perrett, D.I.: Time course of neural responses discriminating different views of the face and head. Journal of Neurophysiology **68**(1) (1992) 70–84
24. Endres, D., Földiák, P.: Exact Bayesian bin classification: a fast alternative to bayesian classification and its application to neural response analysis. Journal of Computational Neuroscience **24**(1) (2008) 24–35 DOI: 10.1007/s10827-007-0039-5.
25. Endres, D.: Bayesian and Information-Theoretic Tools for Neuroscience. PhD thesis, School of Psychology, University of St. Andrews, U.K. (2006) http://hdl.handle.net/10023/162.
26. Hinton, G.: Products of experts. In: Ninth International Conference on Artificial Neural Networks ICANN 99. Number 470 in ICANN (1999)

27. Kiani, R., Esteky, H., Mirpour, K., Tanaka, K.: Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. Journal of Neurophysiology **97**(6) (April 2007) 4296–4309
28. Kay, K.N., Naselaris, T., Prenger, R.J., Gallant, J.L.: Identifying natural images from human brain activity. Nature **452** (2008) 352–255 http://dx.doi.org/10.1038/nature06713.

Some Links Between Decision Tree and Dichotomic Lattice

S. Guillas, K. Bertet, M. Visani, J-M. Ogier, and N. Girard

Laboratory L3I - University of La Rochelle - France
{sguillas,kbertet,mvisani,jmogier,ngirard02}@univ-lr.fr

Abstract. There are two types of classification methods using a Galois lattice: as most of them rely on selection, recent research work focus on navigation-based approaches. In navigation-oriented methods, classification is performed by navigating through the complete lattice, similar to the decision tree. When defined from binary attributes obtained after a discretization pre-processing step, and more generally when a non-empty set of *complementarity attributes* can be associated to each binary attribute, lattices are denoted as "dichotomic lattices". The *Navigala* approach is a navigation-based classification method that relies on the use of a dichotomic lattice. It was initially proposed for symbol recognition in the field of technical document image analysis. In this paper, we define the structural links between decision trees and dichotomic lattices defined from the same table of data described by binary attributes. Under this condition, we prove both that every decision tree is included in the dichotomic lattice and that the dichotomic lattice is the merger of all the decision trees that can be constructed from the same binary data table.

Key words: Classification, Galois lattice, Concept lattice, Decision tree, Navigation

1 Introduction

Galois lattice (or *concept lattice*) has first been introduced in a formal way in graph and ordered structures theory [1,2,3]. Afterwards it has been developed in the field of Formal Concept Analysis (FCA) [4] for data analysis and classification. The concept lattice structure, based on the notion of *concept*, enables to describe data and to preserve their diversity and complexity. A study realized by Mephu Nguifo and Njiwoua [5] confirms the effectiveness of concept lattices for classification and describes selection-oriented methods. Even though these structures are associated to a high time and space complexity (exponential in the worst case), the technological improvements that have been performed during the last decades enable their use.

Galois lattice gives a representation of all the possible correspondences (denoted as concepts) between a set of *objects* (or examples) O and a set of *attributes* (or features) I. Whereas in decision trees the path from the root to a given leaf is unique, in Galois lattices there are multiple paths from the maximal concept to a given terminal concept. Since a lattice is defined from binary attributes, the continuous-valued primitives have to be discretized (after being normalized) in a pre-processing step.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 193–205,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

There are two types of classification methods using a Galois lattice: as most of them rely on selection, recent research work focus on navigation-based approaches. The selection-oriented methods come from the field of data mining and rely on a selection step where the Galois lattice is used to choose concepts which encode relevant information from the huge amount of available data. The classification step is then performed by some usual classifier (k-nearest neighbours, Bayesian classifier. . .).

On the contrary, in the navigation-oriented methods there is no selection step and classification is performed by navigating through the complete lattice. Similar to the classification tree, we navigate from a node to its successors until a labeled (terminal) concept is reached. Indeed, Galois lattice is a graph whose structure is similar to that of a decision tree. Whereas in decision trees the path from the root to a given leaf is unique, in Galois lattices there are multiple paths from the maximal boundary to a given terminal concept.

This similarity between lattices used by navigation-oriented methods and decision trees has been mentionned and stated in some works [6,7,8]. Similar to the classification tree, we navigate from a node to its successors until a labeled (terminal) concept is reached. It is mentionned in particular for the Navigala method we have developped, dedicated to symbols classification [9,10] for an objective of noisy symbols recognition. In order to reduce the size of the lattice, which is generally more important than the size of the tree, the Navigala method proposes a lattice generation performed on-demand during the classification step.

As a first consequence of the similarity between lattice and decision tree, the navigation-oriented methods shares the advantages of the decision tree in terms of readability and ability to automatically select discriminatory variables among a large number of variables. And, contrary to decision trees where there is a unique navigation path to a given node, lattices propose several paths. This property provides to lattices enhanced robustness towards noise.

In this paper, we precise and extend the links between these two structures of lattice and decision tree in the particular case of dichtomic lattices, i.e lattices defined from binary features where a non-empty set of *complementarity attributes* can be associated to each feature:

- Every decision tree is included in the dichotomic lattice, when both structures are built from the same binary attributes.
- Every dichotomic lattice is the merger of all the decision trees when these structures are built from the same binary attributes.

Galois lattice and the navigation-oriented method Navigala are described in Section 2. Section 3 provides a proper definition for dichotomic lattices and the two main results of this paper concerning structural links between dichotomic lattice and decision tree.

2 Navigala: recognition of symbols by navigation in a Galois lattice

2.1 Galois lattice definition

The *concept lattice* is built from a relation R between objects O and attributes I. This graph is composed of a set of concepts ordered by inclusion. It verifies the properties of a lattice: the relation between concepts is an order relation (transitive, reflexive and antisymmetric), and there are a lower bound and an upper bound for each pair of concepts in the graph. We associate to a set of objects $A \subseteq O$ the set $f(A)$ of attributes in relation R with the objects of A:

$$f(A) = \{x \in I \mid pRx \; \forall \, p \in A\}$$

Dually, for every set of attributes $B \subseteq I$, we define the set $g(B)$ of objects in relation with the attributes of B:

$$g(B) = \{p \in O \mid pRx \; \forall \, x \in B\}$$

The relations between the set of objects and the set of attributes are described by a *formal context*. A formal context C is a triplet $C = (O, I, R)$ represented by a table (see for instance Table 1).

Table 1. Example of formal context

Class	Id	Sunniness			Humidity		Wind	
		Sun	Cloudy	Rain	< 77.5	>= 77.5	Yes	No
Y	1	X			X		X	
N	2	X				X	X	
N	3	X				X		X
N	4	X				X		X
Y	5	X			X			X
Y	6		X			X	X	
Y	7		X			X		X
Y	8		X		X		X	
Y	9		X		X			X
N	10			X		X	X	
N	11			X	X		X	
Y	12			X		X		X
Y	13			X		X		X
Y	14			X		X		X

The two functions f and g defined between objects and attributes form a *Galois connection*. The composition $\varphi = f \circ g$ defined on the attributes set enables to associate to each subset of attributes $X \subseteq I$ the smallest concept containing X: $(g(\varphi(X)), \varphi(X))$. This composition φ verifies the properties of a closure operator: φ is idempotent (*i.e.* $\forall X \subseteq S$, $\varphi^2(X) = \varphi(X)$), extensive (*i.e.* $\forall X \subseteq S$, $X \subseteq \varphi(X)$) and isotone (*i.e.* $\forall X, X' \subseteq S$, $X \subseteq X' \Rightarrow \varphi(X) \subseteq \varphi(X')$).

The *Galois lattice* associated to a formal context C is a graph composed of a set of *formal concepts* equipped with a particular binary relation. Intuitively this graph is a representation of all the possible maximal correspondences between a subset of *objects* (or instances, examples) O and a subset of *attributes* (or primitives, features) I. A *formal concept* is a maximal objects-attributes subset where objects and attributes are in relation. More formally, it is a pair (A, B) with $A \subseteq O$ and $B \subseteq I$, which verifies $f(A) = B$ and $g(B) = A$. Let us introduce the binary relation $\leq$ defined on the set of all the concepts by, for two formal concepts (A_1, B_1) and (A_2, B_2):

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow \left\| \begin{array}{l} A_2 \subseteq A_1 \\ \text{(equivalent to } B_1 \subseteq B_2) \end{array} \right.$$

All the set of formal concepts equipped with the order relation $\leq$ forms a lattice called a *concept lattice* or *Galois lattice*. Thus, for each concepts (A_1, B_1) and (A_2, B_2), it exists a greatest lower bound (resp. a least upper bound) called *meet* (resp. *join*) denoted as $(A_1, B_1) \wedge (A_2, B_2)$ (resp. $(A_1, B_1) \vee (A_2, B_2)$) defined by:

$$(A_1, B_1) \wedge (A_2, B_2) = (g(B_1 \cap B_2), (B_1 \cap B_2)) \quad (1)$$

$$(A_1, B_1) \vee (A_2, B_2) = ((A_1 \cap A_2), f(A_1 \cap A_2)) \quad (2)$$

Therefore, a lattice contains a minimum (resp. maximum) element according to the relation $\leq$ called the *bottom* (resp. *top*) of the lattice, and denoted as $\bot = (O, f(O))$ (resp. $\top = (g(I), I)$). For more information on Galois lattice and closure systems, the reader can refer to [1,4].

Figure 1 shows an example of concept lattice built from the formal context in Table 1. This formal context is composed of a set of 14 objects described by 7 attributes (sun, $cloudy$, $rain$, $hum < 77.5$, $hum \geq 77.5$, $windY$ and $windN$).

2.2 Navigala method description

The navigation-base recognition method named Navigala (NAVIgation into GAlois LAttice) has been introduced in [11]. This method is fitted for recognizing *noisy graphic objects* and especially *symbol images*. Such symbols appear in technical documents such as architectural plans or electrical schemes. Graphic objects may be described by statistical or structural primitives. As statistical features describe the spatial distributions of the pixel values of the symbol, structural primitives describe the spatial or topological relations between some sub-patterns extracted from the symbol images. In the following, the primitives vector of each symbol is called the signature of this symbol.

Navigala is a supervised classification approach, no matter if the discretization preprocessing relies on a supervised or unsupervised criterion. This method relies on the classical steps of recognition: data preparation that mainly consists in discretizing continuous data, learning where the Galois lattice is built, and classification where the samples to recognize are labeled after navigating through the graph until they reach a labeled concept.

Data preparation Firstly, several signatures are extracted from the symbol images: statistical signatures (Fourier-Mellin invariants [12], Radon transform-based Radon transform [13], Zernike moments [14]), and a structural signature named *flexible structural signature* [15]. Data preparation then consists in normalizing the various features. The continuous valued primitives must then be *discretized*. At each step of discretization, a

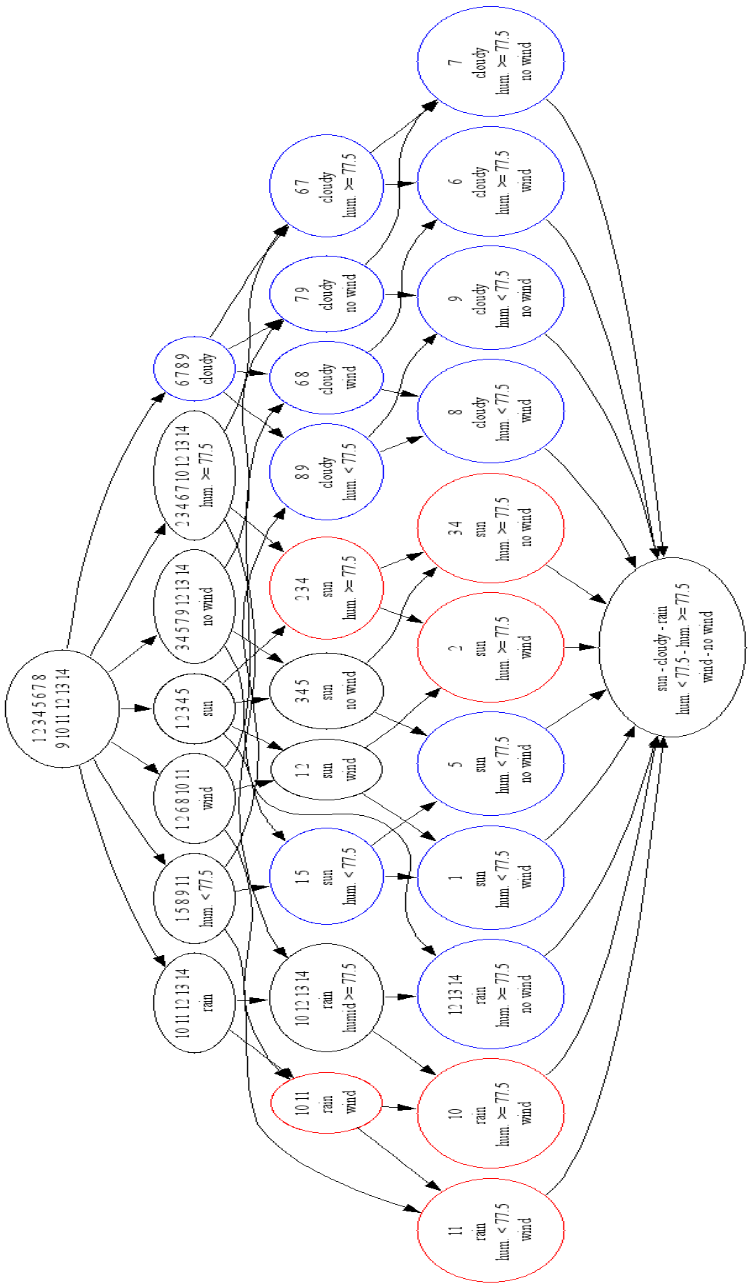


Fig. 1. Example of Galois lattice

criterion selects both the primitive to divide and the optimal cutting point. Let $x \in I$ be a primitive interval composed of values $V_x = (v_1 \dots v_n)$ sorted by ascending order. The interval will be cut between the values v_j and v_{j+1}, where v maximizes a given "cutting" criterion of the primitive values objects. We can define a lot of cutting criteria, supervised or not. Among these criteria, let us mention maximal distance, entropy and Hotelling's coefficient. Discretization is processed until a given *stopping criterion* is met. The stopping criterion used for Navigala is based on class separation, *i.e.* this criterion is met when each class of objects can be represented by its own set of intervals. More precisely, one class can be separated from the others when the objects characterizing this class share at least one interval which enables to distinguish them from the objects of the other classes. At the end of the process, the continuous-valued primitives are converted into intervals of values, called discretized data. Once intervals are computed, they are extended to a fuzzy number.

Learning The discretized data obtained from the data preparation will then be used as a training set, in order to compute the Galois lattice. The generation algorithm [11] is an extension of the Bordat's algorithm [16] since navigation uses the *Hasse diagram* of the lattice (an example of Hasse diagram of a Galois lattice is shown on Figure 1). Once the Hasse diagram is computed, as each concept contains a set of objects, it is possible to label them depending on these objects. Indeed, when all the objects in a given concept correspond to the same class, this concept is named *final concept* and can be labeled. Intuitively, these final concepts correspond to the classes to reach when the Galois lattice is explored for the classification of a new object.

Classification Using the Hasse diagram of the Galois lattice, we can process recognition of the new symbols belonging to the test set. Classification of a new symbol is then processed by navigating through the graph, from the minimal concept (the top $\top$) to a final concept which has been previously labeled by a class. Intuitively, during this progression, we observe a specification of the objects set and a generalization of the attributes set, that is to say that the number of objects is reduced while the number of attributes is increased. Thus, we *refine the description* of the object to recognize, until it corresponds to the description of one of the learning objects whose class will be assigned to the object to recognize. The progression in the graph from a concept to its successor is done according to a *fuzzy distance measure* and a *choice criterion* (for more details please refer to [11]).We estimate the distance between the signature values of the object to recognize and the signatures values of the learning objects, and we choose the successor concept in the Galois lattice whose description best corresponds to the object.

The Galois lattice construction algorithm we use holds several advantages: it is quite easy to implement, and it enables an *on-demand concepts generation* of the Galois lattice. In other words, it enables to generate from a given concept every successor concept in the lattice. This is interesting because it avoids the construction of the whole graph, which can be of exponential complexity in the worst case. Indeed, recognition is performed by exploring only a small region of the lattice. On-demand concepts generation therefore considerably reduces the complexity of the structure generation.

3 Lattice and decision tree

In Navigala, the classification based on a navigation into the graph is quite similar to the one proposed with a decision tree. In this section, we describe both of these structures, and the handled data.

3.1 Decision tree definition

Since years 1960-1970, the decision tree built from a data set has been used in several research works [17,18]. Among the most widely used decision tree generation methods, we can cite CART [19], C4.5 [20] and ID3 [21]. As with a Galois lattice, the data are represented by a table containing a set of objects, set described by a set of attributes. This table can contain discrete, ordinal or continuous data. Decision tree nodes are built from its top, called the root, to its basis where the terminal nodes are called leaves. The construction of a decision tree requires three criteria: a selection criterion, which enables, at each division step, to select one/several feature(s) in the table, a second criterion to discretize the continuous data, and a last criterion to stop the divisions in the tree, which is generally based on a purity measure of the leaves. The root regards all the set of objects in the table; a feature of the table is then selected to separate the objects into two distinct subsets corresponding to two children nodes. This process is likewise iterated on each subset until the stopping criterion is satisfied (see for instance Figure 2).

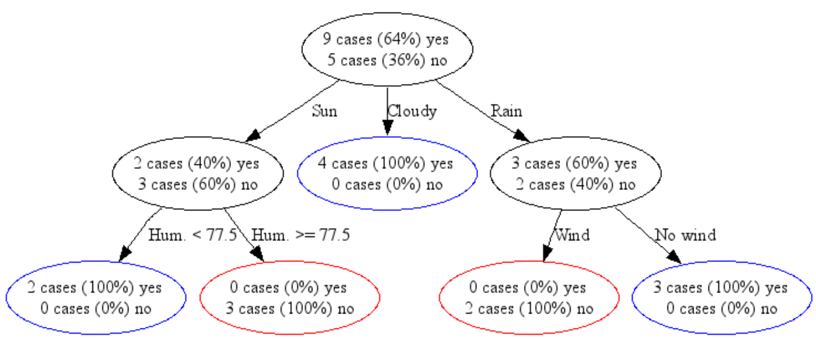


Fig. 2. Example of decision tree

When the features are continuous-valued, as it is the case with the signatures we consider, they need a discretization step which can be processed:

- during the tree construction. Only the selected features will then be discretized.
- before the tree construction, in a pre-processing stage. The data will then be discretized until the classes are separated.

Several heuristics can be used for decision tree construction. For example, a pruning stage can be performed on the decision tree to avoid over-partitioning the data. The pruning principle is to raise into the tree from the leaves by changing nodes in leaves depending on a purity criterion of the nodes. In the structural comparison we describe in the following, the considered decision trees are not pruned.

3.2 Dichotomic lattice

As the decision tree construction infers a discretization of the data, it is possible to consider the binary data table issued from this discretization, and consequently the resulting Galois lattice. We thus find in this binary table (see Table 1), and in the Galois lattice (see Figure 1), the same binary attributes as those proposed by the decision tree (see Figure 2). Thus, when a feature V is proposed, with two children, one for *yes*, and the other for *no*, we had to consider the two binary attributes $V = yes$ and $V = no$. In a more general way, the binary attributes issued from the children of a node are present into the table and separate the set of objects.

In Navigala method, features are continuous data which are discretized in a pre-processing stage in order to obtain classes' separation. The binary attributes in the table are intervals issued from this discretization. A symbol is described by a fixed-size signature before the discretization, and then by a set of binary intervals with the same cardinality after discretization. A symbol is associated to only one interval among the set of intervals issued from a same feature.

Notice that the obtained binary attributes infer an automatic selection of the discriminant features. Indeed, an attribute belonging to all the set of objects will not been proposed in the decision tree, and consequently will not been taken into account in the table. It is the same in Navigala method where a non discretized continuous feature will not appear in the table.

When all objects in a binary table are associated to a same number of binary attributes, the final concepts (*i.e.* the concepts corresponding to a unique class) contain the same number of attributes. The final concepts of a lattice cannot be related the ones to the others (because two concepts in relation $\leq$ can not be composed of a same number of attributes). The final concepts thus have as a unique direct successor the concept $\top$. This property can be found in lattice theory with the notion of *co-atomisticity*. It is the case in our approach Navigala. When discretization is performed (during decision tree construction), the table depends on the proposed attributes in the tree, and two different trees could infer two different binary attributes sets. These two attributes sets can then infer two different lattices. The discretization can also be performed in pre-processing, as in the method Navigala. From this table, several decision trees can be generated but a unique lattice will be associated.

Whatever the case, to each binary attribute x we can associate a non empty set $\overline{X}$ of binary attributes such as the objects having the attribute x, and those having the attributes in $\overline{X}$ are all distinct. The binary attributes are deduced from the decision tree: when x is a feature proposed by a node of the tree, then $\overline{X}$ is a set of all the other features proposed by this same node. Using continuous features discretized in a pre-processing stage, x corresponds to an interval, and $\overline{X}$ contains all remaining intervals corresponding to this same feature. From this property, lattices issued from a tree belong to particular lattices called *dichotomic lattices*. More formally, dichotomic lattices are characterized by the fact to be $\vee$-complementary, that is to say that for each concept (A, B), a *complementary concept* (A', B') always exists such as

$$(A, B) \vee (A', B') = \top = (\emptyset, I) \tag{3}$$

Proposition 1 *Each dichotomic lattice (i.e. lattice issued from a tree) is $\vee$-complementary.*

Proof. Let (A, B) be any concept of a dichotomic lattice. It consists in showing the existence of a complementary concept to (A, B). We consider x any binary attribute of B, and $\overline{x}$ a complementary attribute of x belonging to the set $\overline{X}$. Thus, the objects having x, and those having $\overline{x}$ are distinct. This is formalized by $g(\{x\}) \cap g(\{\overline{x}\}) = \emptyset$. Then we consider the smallest concept containing $\overline{x}$ which, by definition, will be the concept $(g(\varphi(\{\overline{x}\})), \varphi(\{\overline{x}\}))$ where the set of attributes is $\varphi(\{\overline{x}\})$. From the definition of functions f and g, we deduce that $g(\varphi(\{\overline{x}\})) = g(\{\overline{x}\})$, and that $A \subseteq g(\{x\})$. Assuming that $g(\{x\}) \cap g(\{\overline{x}\}) = \emptyset$, we can then deduce that $A \cap g(\varphi(\{\overline{x}\})) = \emptyset$. Consequently, $(A, B) \vee (g(\varphi(\{\overline{x}\})), \varphi(\{\overline{x}\})) = (\emptyset, I)$, and the concept $(g(\varphi(\{\overline{x}\})), \varphi(\{\overline{x}\}))$ is the complementary concept of (A, B). It proves the $\vee$-complementarity of the lattice.

3.3 Structural links between dichotomic lattice and decision tree

A first structural link between decision tree and dichotomic lattice consists in the fact that both structures can be used in classification, and can be defined from a table of binary attributes.

We can notice that the use of navigation-based lattices for classification is similar to the one of decision trees. This similarity is formalized by a structural link between nodes and concepts: indeed, every node in the decision tree may be associated to a unique concept in the lattice. We consider a node n in the tree, and the set of binary attributes X_n proposed from the root to this node. Assuming that these binary attributes belong to the table corresponding to the lattice construction, we then associate to the node n the smallest concept containing the features of X_n:

$$(g(\varphi(X_n)), \varphi(X_n)) \tag{4}$$

Figure 2 presents the decision tree associated to the data of the example. Notice that all the nodes of the decision tree are present into the lattice whatever the construction criterion of the decision tree. Moreover, the structure of the decision tree is also preserved in the lattice as shown in figure 3, where the tree (in bold) is included in the lattice. This property is verified in the general case. Thus, we show that each decision tree is included in the Galois lattice. We also prove that the lattice is the merger of all the decision trees.

Proposition 2 *Each decision tree is included in the dichotomic lattice, when both structures are built from the same binary attributes.*

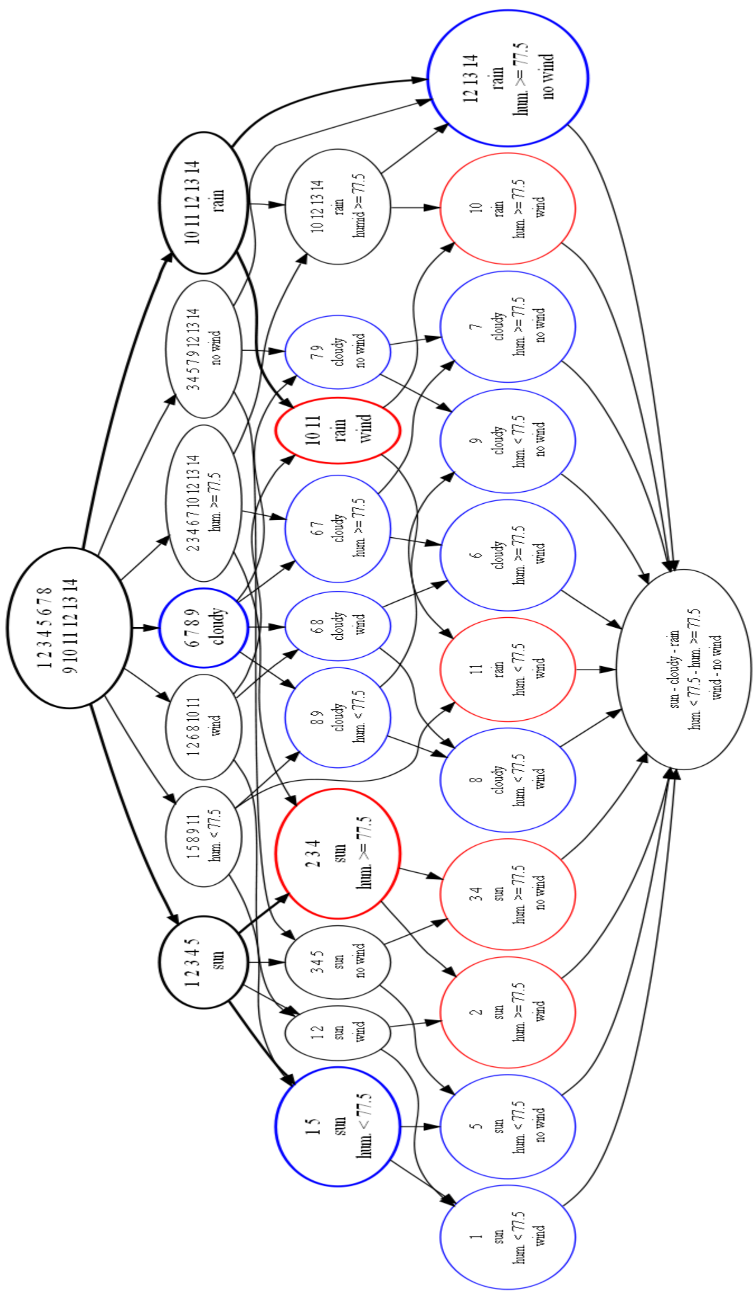


Fig. 3. Inclusion of the decision tree (in bold) in the Galois lattice

Proof. Let us consider a decision tree and a dichotomic lattice issued from the same binary attributes. As mentioned before, these two structures handle the same binary attributes. Moreover, to a node n of the decision tree accessible by validation of the set of attributes X_n we associate the concept $(g(\varphi(X_n)), \varphi(X_n))$. To prove that the decision tree is included into the lattice, it is necessary to prove the three following points:

1. Two different nodes of a decision tree are associated to different concepts:
   By contradiction, when two nodes n_1 and n_2 are associated to a same concept, then $\varphi(X_{n_1}) = \varphi(X_{n_2})$. It means that the same objects share the attributes of X_{n_1} and X_{n_2}, that is in contradiction with the fact that two nodes n_1 and n_2 are two different nodes of the decision tree.
2. When two nodes are ancestors in the decision tree, then their associated concepts are related in the lattice:
   Clearly when a node n_1 is ancestor of a node n_2 in a decision tree, then $X_{n_1} \subseteq X_{n_2}$. The operator φ being isotone, we deduce that $\varphi(X_{n_1}) \subseteq \varphi(X_{n_2})$, and consequently that these two concepts $(g(\varphi(X_{n_1})), \varphi(X_{n_1}))$, $(g(\varphi(X_{n_2})), \varphi(X_{n_2}))$ are related depending on the relation $\leq$.
3. Conversely, when two nodes are not ancestor in the decision tree, then their associated concepts are not related in the lattice:
   When a node n_1 is not ancestor of a node n_2, then we need to consider all the children of the smallest common ancestor to n_1 and n_2, and particularly the child n'_1 ancestor of n_1 and the child n'_2 ancestor of n_2. These two nodes n'_1 and n'_2 exist by construction of the table. Clearly, as n'_1 and n'_2 are brothers, their attributes in the associated concepts, being $\varphi(X_{n'_1})$ and $\varphi(X_{n'_1})$, are not shared by any object. That is formalized by $g(\varphi(X_{n'_1})) \cap g(\varphi(X_{n'_2})) = \emptyset$. Then, n'_1 being ancestor of n_1, we can deduce that $X_{n'_1} \subseteq X_{n_1}$, where $\varphi(X_{n'_1}) \subseteq \varphi(X_{n_1})$ by isotony of the operator φ, and reversely $g(\varphi(X_{n'_1})) \supseteq g(\varphi(X_{n_1}))$ by definition of g. We also have $g(\varphi(X_{n'_2})) \supseteq g(\varphi(X_{n_2}))$ because n'_2 is ancestor of n_2. Thus we deduce that $g(\varphi(X_{n_2})) \cap g(\varphi(X_{n_1})) = \emptyset$, and that proves that the concepts associated to the nodes n_1 and n_2 are not in relation by $\leq$.

Proposition 3 *A dichotomic lattice is the merger of all the decision trees when these structures are built from the same binary attributes.*

Proof. We previously proved that each decision tree is included into the dichotomic lattice built from the same binary attributes. To prove that the dichotomic lattice is the merger of all the decision trees, we must prove that each concept potentially belong to a decision tree. This proof is given by construction.

We consider an any concept (A, B). Then we build the subset of concepts C of the lattice containing: the concept (A, B), a complementary concept (A', B') to (A, B), the minimal concept $\perp$, and all the final successors concepts of (A, B) and (A', B'). The existence of the complementary concept (A', B') is deduced from the $\vee$-complementarity property of the dichotomic lattice. Moreover, it infers that this subset C in addition to the relation $\leq$ forms a tree. Then we add in the set C a maximal number of concepts of the dichotomic lattice such as $(C, \leq)$ preserves the property to be a tree. Thus, by construction, we obtain a sub-tree included into the dichotomic lattice, containing (A, B).

In this sub-tree, the leaves are final concepts and correspond to subsets of objects which can not be separated by any binary attribute, *i.e.* the classes when the data have been discretized until the classes are separated. This tree can thus be considered as a decision tree, what finishes this proof.

4 Conclusion

This paper is about Galois lattice which is used as a classifier in the *Navigala* approach and, more generally, about *dichotomic lattices* defined from a structural way: to every binary feature, a non-empty set of *complementarity features* can be associated.

There is some published work about using Galois lattices as a classifier: as most of the proposed approaches consider the lattice as a concept selection tool, Navigala performs classification by navigating through the lattice from one node to its successors, similar to a classification tree.

As a first consequence, the Navigala approach shares the advantages of the decision tree in terms of readability and ability to automatically select discriminatory variables among a large number of variables. As another consequence, contrary to decision trees where there is a unique navigation path to a given node, lattices propose several paths. This property provides to lattices enhanced robustness towards noise.

The inclusion result (Proposition 2) of this paper implies that navigation paths proposed by decision tree are included in the dichotomic lattice issued from the same binary features. Moreover, Proposition 3 states that the every dichotomic lattice is equal to the merge of navigation paths of all the decision trees.

This work opens the way for the definition of a new method that would combine the advantages of both trees and lattices.

References

1. Barbut, M., Monjardet, B.: Ordre et classification, Algèbre et combinatoire, Paris (1970) 2 tomes.
2. Birkhoff, G.: Lattice theory. 3rd edn. Volume 25. American Mathematical Society (1967)
3. Davey, B., Priestley, H.: Introduction to lattices and orders. 2nd edn. Cambridge University Press (1991)
4. Wille, R.: Restructuring lattice theory: an approach based on hierarchy on contexts. Ordered sets (1982) 445–470
5. Mephu-Nguifo, E., Njiwoua, P.: Treillis des concepts et classification supervisée. Technique et Science Informatiques, RSTI **24** (2005) 449–488 Hermès - Lavoisier, Paris, France.
6. Kuznetsov, S.: Machine learning on the basis of formal concept analysis. **62** (2001) 1543 – 1564
7. Kuznetsov, S.: Machine learning and formal concept analysis. **3029** (2004) 287–312
8. Guillas, S., Bertet, K., Ogier, J.M.: Concept lattice classifier : a first step towards an iterative process of recognition of noised graphic objects. In: Fourth International Conference on Concept Lattices and their Applications (CLA'2006). (2006) 257–263
9. Guillas, S.: Reconnaissance d'objets graphiques détériorés : approche fondée sur un treillis de Galois. PhD thesis, Université de La Rochelle (2007)

10. Guillas, S., Bertet, K., Ogier, J.M.: A generic description of the concept lattices' classifier: Application to symbol recognition. In Wenyin, L., Lladós, J., eds.: Graphics Recognition: Ten Years Review and Future Perspectives - Selected papers from GREC'05. Volume 3926., LNCS (2006) 47–60
11. Bertet, K., Guillas, S., Ogier, J.M.: Extensions of bordat's algorithm for attributes. In: Fifth International Conference on Concept Lattices and their Applications (CLA'2007), Montpellier, France (2007) 38–49
12. Derrode, S., Daoudi, M., Ghorbel, F.: Invariant content-based image retrieval using a complete set of Fourier-Mellin descriptors. Int. Conf. on Multimedia Computing and Systems (ICMCS'99) (1999) 877–881
13. Tabbone, S., Wendling, L.: Recherche d'images par le contenu à l'aide de la transformée de Radon. Technique et Science Informatiques (2003)
14. Teague, M.: Image analysis via the general theory of moments. Journal of Optical Society of America (JOSA) **70** (2003) 920–930
15. Coustaty, M., Guillas, S., Visani, M., Bertet, K., Ogier, J.M.: Flexible structural signature for symbol recognition using a concept lattice classifier. In: Seventh IAPR International Workshop on Graphics Recognition (GREC'07), Curitiba, Brazil (2007)
16. Bordat, J.: Calcul pratique du treillis de Galois d'une correspondance. Math. Sci. Hum. **96** (1986) 31–47
17. Rakotomalala, R.: Graphes d'induction. PhD thesis, Université Claude Bernard, Lyon I (1997)
18. Rakotomalala, R.: Arbres de décision. Revue MODULAD **33** (2005)
19. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth Inc., Belmont, California (1984)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman, Los Altos, California (1993)
21. Quinlan, J.: Induction of decision trees. Machine Learning **1** (1986)

On Generalization of Fuzzy Concept Lattices Based on Change of Underlying Fuzzy Order

Pavel Martinek

Department of Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic
pavel.martinek@upol.cz

Abstract. The paper presents a generalization of the main theorem of fuzzy concept lattices. The theorem is investigated from the point of view of fuzzy logic. There are various fuzzy order types which differ by incorporated relation of antisymmetry. This paper focuses on fuzzy order which uses fuzzy antisymmetry defined by means of multiplication operation and fuzzy equality.

Keywords. Fuzzy order, fuzzy concept lattice

1 Introduction

A notion of fuzzy order has been derived from the classical one by fuzzification the three underlying relations. This led to various versions, at the beginning versions utilizing the classical relation of equality (see e.g. [9]); later versions are more general by introducing fuzzy similarity (or fuzzy equality) instead of the classical equality (see e.g. [4]). Fuzzy similarity is based on idea that relationship between objects A and B should be transformed to a similar relationship between objects A' and B' whenever A', B' are similar to A, B, respectively.

In [3], one of the later definitions of fuzzy order was used to formulate and prove a fuzzy logic extension of the main theorem of concept lattices. The aim of this paper is to enlarge validity of the theorem to more general fuzzy order.

2 Preliminaries

First, we recall some basic notions. It is known that in fuzzy logic an important structure of truth values is represented by a complete residuated lattice (see e.g. [5], [6], [7]).

Definition 1. *A residuated lattice is an algebra* $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$ *such that*

- $\langle L, \wedge, \vee, 0, 1\rangle$ *is a lattice with the least element* 0 *and the greatest element* 1*,*
- $\langle L, \otimes, 1\rangle$ *is a commutative monoid,*

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 207–215,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

– *$\otimes$ and $\rightarrow$ form so-called adjoint pair, i.e. $a \otimes b \leq c$ iff $a \leq b \rightarrow c$ holds for all $a, b, c \in L$.*

Residuated lattice **L** *is called complete if $\langle L, \wedge, \vee \rangle$ is a complete lattice.*

Throughout the paper, **L** will denote a complete residuated lattice. An **L**-*set* (or fuzzy set) A in a universe set X is any mapping $A : X \rightarrow L$, $A(x)$ being interpreted as the truth degree of the fact that "x belongs to A". By L^X we denote the set of all **L**-sets in X. A binary **L**-relation is defined obviously. Operations on L extend pointwise to L^X, e.g. $(A \vee B)(x) = A(x) \vee B(x)$ for any $A, B \in L^X$. As is usual, we write $A \cup B$ instead of $A \vee B$, etc.

L-*equality* (or fuzzy equality) is a binary **L**-relation $\approx \in L^{X \times X}$ such that $(x \approx x) = 1$ (reflexivity), $(x \approx y) = (y \approx x)$ (symmetry), $(x \approx y) \otimes (y \approx z) \leq (x \approx z)$ (transitivity), and $(x \approx y) = 1$ implies $x = y$. We say that a binary **L**-relation $R \in L^{X \times Y}$ is *compatible* with respect to $\approx_X$ and $\approx_Y$ if $R(x, y) \otimes (x \approx_X x') \otimes (y \approx_Y y') \leq R(x', y')$ for any $x, x' \in X,\ y, y' \in Y$. Analogously an **L**-set $A \in L^X$ is *compatible* with respect to $\approx_X$ if $A(x) \otimes (x \approx_X x') \leq A(x')$ for any $x, x' \in X$. Given $A, B \in L^X$, in agreement with [5] we define the subsethood degree $\mathrm{S}(A, B)$ of A in B by $\mathrm{S}(A, B) = \bigwedge_{x \in X} A(x) \rightarrow B(x)$. For $A \in L^X$ and $a \in L$, the set ${}^aA = \{x \in X;\ A(x) \geq a\}$ is called the a-cut of A. Analogously, for $R \in L^{X \times Y}$ and $a \in L$, we denote ${}^aR = \{(x, y) \in X \times Y;\ R(x, y) \geq a\}$. For $x \in X$ and $a \in L$, $\{{}^a/_x\}$ is the **L**-set defined by $\{{}^a/_x\}(x) = a$ and $\{{}^a/_x\}(y) = 0$ for $y \neq x$.

Definition 2. *An* **L**-*order on a set X with an* **L**-*equality $\approx$ is a binary* **L**-*relation $\preceq$ which is compatible with respect to $\approx$ and satisfies the following conditions for all $x, y, z \in X$:*

$$(x \preceq x) = 1 \qquad (\textit{reflexivity}),$$

$$(x \preceq y) \otimes (y \preceq x) \leq (x \approx y) \qquad (\textit{antisymmetry}),$$

$$(x \preceq y) \otimes (y \preceq z) \leq (x \preceq z) \qquad (\textit{transitivity}).$$

(Cf. T-E-ordering from [4].) Since in residuated lattices $x \otimes y \leq x \wedge y$ for every $x, y \in X$, our relation is more general than **L**-order of [3] or [2] where antisymmetry is expressed by condition $(x \preceq y) \wedge (y \preceq x) \leq (x \approx y)$.

Note that

$$(x \preceq y) \otimes (y \preceq x) \leq (x \approx y) \leq (x \preceq y) \wedge (y \preceq x). \tag{1}$$

Indeed, the first inequality represents antisymmetry and the second one follows from compatibility (see proof of Lemma 4 in [3]):
$(x \approx y) = (x \preceq x) \otimes (x \approx x) \otimes (x \approx y) \leq (x \preceq y)$ and analogously, $(x \approx y) \leq (y \preceq x)$.

The inequalities (1) represent the fact that **L**-equality must satisfy the interval confinement as follows:

$$(x \approx y) \in \left[(x \preceq y) \otimes (y \preceq x),\ (x \preceq y) \wedge (y \preceq x)\right].$$

On the other hand, the definition of **L**-order by [3] (which must satisfy the condition $(x \preceq_{[3]} y) \wedge (y \preceq_{[3]} x) \leq (x \approx y)$) leads to firm binding of **L**-equality with the upper bound of previous interval (see Lemma 4 of [3]), i.e.

$$(x \approx y) = (x \preceq_{[3]} y) \wedge (y \preceq_{[3]} x).$$

Now, we can interpret the relationship between **L**-order and **L**-equality defined either in [3] and in this paper as follows. By the definition, **L**-order is dependent on a given **L**-equality. However, if we change point of view and have a look to the inverse "dependence", we can see that

- by [3], an **L**-equality is binded with any corresponding **L**-order firmly,
- in our paper, an **L**-equality has certain freedom (with respect to a corresponding **L**-order).

This will play an important role during generalizing results achieved in [3].

If $\preceq$ is an **L**-order on a set X with an **L**-equality $\approx$, we call the pair $\mathbf{X} = \langle\langle X, \approx\rangle, \preceq\rangle$ an **L**-*ordered set*. In agreement with [3], we say that **L**-ordered sets $\langle\langle X, \approx_X\rangle, \preceq_X\rangle$ and $\langle\langle Y, \approx_Y\rangle, \preceq_Y\rangle$ are *isomorphic* if there is a bijective mapping $h : X \to Y$ such that $(x \approx_X x') = (h(x) \approx_Y h(x'))$ and $(x \preceq_X x') = (h(x) \preceq_Y h(x'))$ hold for any $x, x' \in X$.

Note that in case of firm binding of **L**-equality and **L**-order by [3] (see the note above), preservation of the **L**-order by the bijection h implies also preservation of the **L**-equality. Clearly this is not true for **L**-order defined in this paper.

3 Some properties of fuzzy ordered sets

In this section, we describe some notions and properties related to fuzzy ordered sets which represent appropriate generalizations of notions and facts known from classical (partial) ordered sets. These generalizations were introduced mainly in [2] and [3] (the fact that originally they used less general definition of **L**-order is unimportant).

Definition 3. *For an* **L**-*ordered set* $\langle\langle X, \approx\rangle, \preceq\rangle$ *and* $A \in L^X$ *we define the* **L**-*sets* $\mathcal{L}(A)$ *and* $\mathcal{U}(A)$ *in* X *by*

$$\mathcal{L}(A)(x) = \bigwedge_{x' \in X} (A(x') \to (x \preceq x')) \text{ for all } x \in X,$$

$$\mathcal{U}(A)(x) = \bigwedge_{x'' \in X} (A(x'') \to (x'' \preceq x)) \text{ for all } x \in X.$$

$\mathcal{L}(A)$ *and* $\mathcal{U}(A)$ *are called the lower cone and upper cone of* A, *respectively.*

These **L**-sets can be described as the **L**-sets of elements which are smaller (greater) than all elements of A. We will abbreviate $\mathcal{U}(\mathcal{L}(A))$ by $\mathcal{UL}(A)$, $\mathcal{L}(\mathcal{U}(A))$ by $\mathcal{LU}(A)$ etc.

Definition 4. *For an* $\mathbf{L}$*-ordered set* $\langle\langle X, \approx\rangle, \preceq\rangle$ *and* $A \in L^X$ *we define the* $\mathbf{L}$*-sets* $\inf(A)$ *and* $\sup(A)$ *in* X *by*

$$(\inf(A))(x) = (\mathcal{L}(A))(x) \wedge (\mathcal{UL}(A))(x) \text{ for all } x \in X,$$

$$(\sup(A))(x) = (\mathcal{U}(A))(x) \wedge (\mathcal{LU}(A))(x) \text{ for all } x \in X.$$

$\inf(A)$ *and* $\sup(A)$ *are called the infimum and supremum of* A*, respectively.*

Lemma 1. *Let* $\langle\langle X, \approx\rangle, \preceq\rangle$ *be an* $\mathbf{L}$*-ordered set,* $A \in L^X$*. If* $(\inf(A))(x) = 1$ *and* $(\inf(A))(y) = 1$ *then* $x = y$ *(and similarly for* $\sup(A)$*).*

Proof. The proof is almost verbatim repetition of the proof of Lemma 9 in [3].□

Lemma 2. *For an* $\mathbf{L}$*-ordered set* $\langle\langle X, \approx\rangle, \preceq\rangle$ *and* $A \in L^X$*, the* $\mathbf{L}$*-sets* $\inf(A)$ *and* $\sup(A)$ *are compatible with respect to* $\approx$*.*

Proof. The proof can be found in [2], namely in more general proof of Lemma 5.39 with regard to Remark 5.40. □

Definition 5. *For a set* X *with an* $\mathbf{L}$*-equality* $\approx$*, an* $\mathbf{L}$*-set* $A \in L^X$ *is called an* S*-singleton if it is compatible with respect to* $\approx$ *and there is some* $x_0 \in X$ *such that* $A(x_0) = 1$ *and* $A(x) < 1$ *for* $x \neq x_0$*.*

Remark 1. There are various definitions of fuzzy singletons (see e.g. [8] or [2]). Our definition represents the simplest one, that is why we call it S-singleton. Demanding more conditions than stated would lead to serious troubles in proof of Theorem 2. Note that in case of $\mathbf{L}$ equal to the Boolean algebra $\mathbf{2}$ of classical logic with the support $\{0, 1\}$, S-singletons represent classical one-element sets.

Lemma 3. *For an* $\mathbf{L}$*-ordered set* $\langle\langle X, \approx\rangle, \preceq\rangle$ *and* $A \in L^X$*, if* $(\inf(A))(x_0) = 1$ *for some* $x_0 \in X$ *then* $\inf(A)$ *is an* S*-singleton. The same is true for suprema.*

Proof. The assertion immediately follows from Lemmata 1 and 2. □

Definition 6. *An* $\mathbf{L}$*-ordered set* $\langle\langle X, \approx\rangle, \preceq\rangle$ *is said to be completely lattice* $\mathbf{L}$*-ordered if for any* $A \in L^X$ *both* $\inf(A)$ *and* $\sup(A)$ *are* S*-singletons.*

Theorem 1. *For an* $\mathbf{L}$*-ordered set* $\mathbf{X} = \langle\langle X, \approx\rangle, \preceq\rangle$*, the relation* ${}^1\!\preceq$ *is an order on* X*. Moreover, if* $\mathbf{X}$ *is completely lattice* $\mathbf{L}$*-ordered then* ${}^1\!\preceq$ *is a lattice order on* X*.*

Proof. The proof is analogous to the proof of Theorem 13 in [3]. □

4 Fuzzy concept lattices

We remind some basic facts about concept lattices in fuzzy setting. A formal **L**-*context* is a tripple $\langle X, Y, I\rangle$ where I is an **L**-relation between the sets X and Y (with elements called objects and attributes, respectively). For any **L**-context we can generalize notions introduced in Definition 3 as follows. Let X, Y be sets with **L**-equalities $\approx_X, \approx_Y$, respectively; $I \in L^{X\times Y}$ be an **L**-relation compatible with respect to $\approx_X$ and $\approx_Y$. For any $A \in L^X$, $B \in L^Y$, we define $A^\uparrow \in L^Y$, $B^\downarrow \in L^X$ (see e.g. [1]) by

$$A^\uparrow(y) = \bigwedge_{x\in X} (A(x) \to I(x,y)) \text{ for all } y \in Y,$$

$$B^\downarrow(x) = \bigwedge_{y\in Y} (B(y) \to I(x,y)) \text{ for all } x \in X.$$

Clearly, $A^\uparrow(y)$ describes the truth degree, to which "for each x from A, x and y are in I", and similarly $B^\downarrow(x)$. We will abbreviate $(A^\uparrow)^\downarrow$ by $A^{\uparrow\downarrow}$, $(B^\downarrow)^\uparrow$ by $B^{\downarrow\uparrow}$ etc. The equation $A^\uparrow = A^{\uparrow\downarrow\uparrow}$ holds true for all $A \in L^X$ (see e.g. [1]). Note that if $X = Y$ and $I = \preceq$ is an **L**-order on X, then $A^\uparrow$ coincides with $\mathcal{U}(A)$ and $B^\downarrow$ coincides with $\mathcal{L}(B)$. An **L**-*concept* in a given **L**-context $\langle X, Y, I\rangle$ is any pair $\langle A, B\rangle$ of $A \in L^X$ and $B \in L^Y$ such that $A^\uparrow = B$ and $B^\downarrow = A$ (see [2]).

We denote by $\mathcal{B}(X, Y, I)$ the set of all **L**-concepts given by an **L**-context $\langle X, Y, I\rangle$, i.e.

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

For any $\mathcal{B}(X, Y, I)$, we put (see [3])

$$(\langle A_1, B_1\rangle \preceq_S \langle A_2, B_2\rangle) = \mathrm{S}(A_1, A_2) \text{ for all } \langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathcal{B}(X, Y, I).$$

The **L**-relation $\preceq_S$ obviously satisfies the conditions of reflexivity and transitivity. As to the antisymmetry, we need an **L**-equality. Therefore, consider an arbitrary **L**-equality $\approx$ on the set $\mathcal{B}(X, Y, I)$ such that $\preceq_S$ is compatible with respect to $\approx$ and the inequality

$$(\langle A_1, B_1\rangle \preceq_S \langle A_2, B_2\rangle) \otimes (\langle A_2, B_2\rangle \preceq_S \langle A_1, B_1\rangle) \le (\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle)$$

holds true for every $\langle A_i, B_i\rangle \in \mathcal{B}(X, Y, I),\ i \in \{1, 2\}$. (Existence of such an **L**-equality is demonstrated e.g. by $(\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) = \mathrm{S}(A_1, A_2) \wedge \mathrm{S}(A_2, A_1)$ in [3].) Consequently, $\preceq_S$ is an **L**-order on $\langle \mathcal{B}(X, Y, I), \approx\rangle$ and we get an **L**-ordered set $\langle\langle \mathcal{B}(X, Y, I), \approx\rangle, \preceq_S\rangle$ which will act in further two theorems. Note that the **L**-ordered set is more general than **L**-concept lattice $\langle\langle \mathcal{B}(X, Y, I), \approx\rangle, \preceq_S\rangle$ of [3] because of more general **L**-equality.

The next theorem characterizing **L**-concept lattices needs further denotation. As usual, for an **L**-set A in U and $a \in L$, we denote by $a \otimes A$ the **L**-set such that

$(a \otimes A)(u) = a \otimes A(u)$ for all $u \in U$. If $\mathcal{M}$ is an **L**-set in Y and each $y \in Y$ is an **L**-set in X, we define the **L**-set $\bigcup \mathcal{M}$ in X (see [3]) by

$$(\bigcup \mathcal{M})(x) = \bigvee_{A \in Y} \mathcal{M}(A) \otimes A(x) \quad \text{for all } x \in X.$$

Clearly, $\bigcup \mathcal{M}$ represents a generalization of a union of a system of sets. For an **L**-set $\mathcal{M}$ in $\mathcal{B}(X,Y,I)$, we put $\bigcup_X \mathcal{M} = \bigcup \mathrm{pr}_X(\mathcal{M})$, $\bigcup_Y \mathcal{M} = \bigcup \mathrm{pr}_Y(\mathcal{M})$ where $\mathrm{pr}_X(\mathcal{M})$ is an **L**-set in the set $\{A \in L^X;\ A = A^{\uparrow\downarrow}\}$ of all extents of $\mathcal{B}(X,Y,I)$ defined by $(\mathrm{pr}_X\mathcal{M})(A) = \mathcal{M}(A, A^{\uparrow})$ and, similarly, $\mathrm{pr}_Y(\mathcal{M})$ is an **L**-set in the set $\{B \in L^Y;\ B = B^{\downarrow\uparrow}\}$ of all intents of $\mathcal{B}(X,Y,I)$ defined by $(\mathrm{pr}_Y\mathcal{M})(B) = \mathcal{M}(B^{\downarrow}, B)$. Hence, $\bigcup_X \mathcal{M}$ is the "union of all extents from $\mathcal{M}$" and $\bigcup_Y \mathcal{M}$ is the "union of all intents from $\mathcal{M}$" (see [3]).

Theorem 2. *Let $\langle X,Y,I\rangle$ be an* **L***-context. An* **L***-ordered set $\langle\langle \mathcal{B}(X,Y,I), \approx\rangle, \preceq_S\rangle$ is completely lattice* **L***-ordered set in which infima and suprema can be described as follows: for an* **L***-set $\mathcal{M}$ in $\mathcal{B}(X,Y,I)$ we have*

$$^1\mathrm{inf}(\mathcal{M}) = \left\{ \left\langle (\bigcup_Y \mathcal{M})^{\downarrow}, (\bigcup_Y \mathcal{M})^{\downarrow\uparrow} \right\rangle \right\} \tag{2}$$

$$^1\mathrm{sup}(\mathcal{M}) = \left\{ \left\langle (\bigcup_X \mathcal{M})^{\uparrow\downarrow}, (\bigcup_X \mathcal{M})^{\uparrow} \right\rangle \right\} \tag{3}$$

Proof. The proof of (2) and (3) is analogous to the proof of part (i) of Theorem 14 in [3] where differently defined antisymmetry is not used anywhere. Furthermore by Lemma 3, each **L**-ordered set $\langle\langle \mathcal{B}(X,Y,I), \approx\rangle, \preceq_S\rangle$ is completely lattice **L**-ordered. □

For any completely lattice **L**-ordered set $\mathbf{X} = \langle\langle X, \approx\rangle, \preceq\rangle$, a subset $K \subseteq X$ is called $\{0,1\}$-*infimally dense* ($\{0,1\}$-*supremally dense*) *in* **X** (cf. [3]) if for each $x \in X$ there is some $K' \subseteq K$ such that $x = \bigwedge K'$ ($x = \bigvee K'$). Here $\bigwedge$ ($\bigvee$) means infimum (supremum) with respect to the 1-cut of $\preceq$.

Theorem 3. *Let $\langle X,Y,I\rangle$ be an* **L***-context. A completely lattice* **L***-ordered set $\mathbf{V} = \langle\langle V, \approx_V\rangle, \preceq\rangle$ is isomorphic to an* **L***-ordered set $\langle\langle \mathcal{B}(X,Y,I), \approx\rangle, \preceq_S\rangle$ iff there are mappings $\gamma : X \times L \to V,\ \mu : Y \times L \to V$, such that*

(i) $\gamma(X \times L)$ *is* $\{0,1\}$*-supremally dense in* $\mathbf{V}$,

(ii) $\mu(Y \times L)$ *is* $\{0,1\}$*-infimally dense in* $\mathbf{V}$,

(iii) $((a \otimes b) \to I(x,y)) = (\gamma(x,a) \preceq \mu(y,b))$ *for all* $x \in X,\ y \in Y,\ a,b \in L$.

(iv) $(\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) = \left(\bigvee_{x\in X} \gamma(x, A_1(x)) \approx_V \bigvee_{x \in X} \gamma(x, A_2(x))\right)$ *for all* $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathcal{B}(X,Y,I)$.

Proof. Let γ and μ with the above properties exist. If we define the mapping $\varphi : \mathcal{B}(X,Y,I) \to V$ by $\varphi(A,B) = \bigvee_{x\in X} \gamma(x, A(x))$ for all $\langle A,B\rangle \in \mathcal{B}(X,Y,I)$, then by the proof of Part (ii) of Theorem 14 in [3] (where differently defined antisymmetry is not used anywhere) the mapping φ is bijective and preserves fuzzy order. Thus, we have to prove that it preserves also fuzzy equality. However this is immediate:

$$(\varphi(A_1,B_1) \approx_V \varphi(A_2,B_2)) = (\bigvee_{x\in X} \gamma(x, A_1(x)) \approx_V \bigvee_{x\in X} \gamma(x, A_2(x))) =$$
$$= (\langle A_1,B_1\rangle \approx \langle A_2,B_2\rangle).$$

Conversely, let $\mathbf{V}$ and $\langle\langle \mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$ be isomorphic. Similarly to [3], it suffices to prove existence of mappings γ,μ with desired properties for $\mathbf{V} = \langle\langle \mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$ and for identity on $\langle\langle \mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$ which is obviously an isomorphism. The reason for this simplification lies in the fact that for the general case $\mathbf{V} \cong \langle\langle \mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$ one can take $\gamma\circ\varphi : X\times L \to V,\ \mu\circ\varphi : Y\times L \to V$, where φ is the isomorphism of $\langle\langle \mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$ onto $\mathbf{V}$. If we define $\gamma : X\times L \to \mathcal{B}(X,Y,I),\ \mu : Y\times L \to \mathcal{B}(X,Y,I)$ by

$$\gamma(x,a) = \left\langle \{{}^{a}/_{x}\}^{\uparrow\downarrow}, \{{}^{a}/_{x}\}^{\uparrow} \right\rangle,$$

$$\mu(y,b) = \left\langle \left\{{}^{b}/_{y}\right\}^{\downarrow}, \left\{{}^{b}/_{y}\right\}^{\downarrow\uparrow} \right\rangle$$

for all $x\in X,\ y\in Y,\ a,b\in L$, then by the proof of Part (ii) of Theorem 14 in [3] (where differently defined antisymmetry is not used anywhere) these mappings γ,μ satisfy conditions (i–iii) of our theorem. So, it remains to prove condition (iv), i.e. the equality

$$(\langle A_1,B_1\rangle \approx \langle A_2,B_2\rangle) = \left(\bigvee_{x\in X} \gamma(x, A_1(x)) \approx_V \bigvee_{x\in X} \gamma(x, A_2(x))\right).$$

Since we consider identity on $\langle\langle \mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$, we have $\approx = \approx_V$ and it suffices to prove that $\bigvee_{x\in X} \gamma(x, A(x)) = \langle A,B\rangle$ for all $\langle A,B\rangle \in \mathcal{B}(X,Y,I)$.

We start with proof of the equation $A = \bigcup_{x\in X} \left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow}$ for any $A = A^{\uparrow\downarrow}$. On the one hand we get for each $x'\in X$:

$$\left(\bigcup_{x\in X} \left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow}\right)(x') = \bigvee_{x\in X} \left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow}(x') =$$

$$= \bigvee_{x\in X} \left[\bigwedge_{y\in Y} \left\{{}^{A(x)}/_{x}\right\}^{\uparrow}(y) \to I(x',y)\right] =$$

$$= \bigvee_{x\in X} \left[\bigwedge_{y\in Y} \left(\bigwedge_{\tilde{x}\in X} \left\{{}^{A(x)}/_{x}\right\}(\tilde{x}) \to I(\tilde{x},y)\right) \to I(x',y)\right] =$$

$$= \bigvee_{x\in X}\left[\bigwedge_{y\in Y}(A(x)\to I(x,y))\to I(x',y)\right] \geq$$

$$\geq \bigwedge_{y\in Y}[(A(x')\to I(x',y))\to I(x',y)] \geq$$

$$\geq \bigwedge_{y\in Y} A(x') = A(x').$$

On the other hand we have:

$$\left(\bigcup_{x\in X}\left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow}\right)(x') = \bigvee_{x\in X}\left[\bigwedge_{y\in Y}(A(x)\to I(x,y))\to I(x',y)\right] \leq$$

$$\leq \bigvee_{x\in X}\left[\bigwedge_{y\in Y}\left(\bigwedge_{\tilde{x}\in X} A(\tilde{x})\to I(\tilde{x},y)\right)\to I(x',y)\right] =$$

$$= \bigvee_{x\in X}\left[\bigwedge_{y\in Y} A^{\uparrow}(y)\to I(x',y)\right] =$$

$$= \bigvee_{x\in X} A^{\uparrow\downarrow}(x') = A^{\uparrow\downarrow}(x') = A(x').$$

Using also the definition of γ and Theorem 2, we obtain

$$\bigvee_{x\in X}\gamma(x,A(x)) = \bigvee_{x\in X}\left\langle\left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow},\left\{{}^{A(x)}/_{x}\right\}^{\uparrow}\right\rangle =$$

$$= \left\langle\left(\bigcup_{x\in X}\left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow}\right)^{\uparrow\downarrow},\left(\bigcup_{x\in X}\left\{{}^{A(x)}/_{x}\right\}^{\uparrow\downarrow}\right)^{\uparrow}\right\rangle =$$

$$= \langle A^{\uparrow\downarrow}, A^{\uparrow}\rangle = \langle A, B\rangle.$$

□

Remark 2. Note that the essential difference between Theorem 3 in this paper and Theorem 14, part (ii) in [3] lies in differently defined **L**-ordered sets (see the notes at the end of Section 2). Therefore in comparison to Theorem 14 of [3], Theorem 3 must contain "additional" condition (iv) which is necessary for isomorphism between (more general) **V** and $\langle\langle\mathcal{B}(X,Y,I),\approx\rangle,\preceq_S\rangle$.

5 Work in progress

There is an interesting proposition which deals with a completely lattice **L**-ordered set $\langle\langle\mathcal{B}(V,V,\preceq),\approx_S\rangle,\preceq_S\rangle$ such that

- $\preceq$ is an **L**-order on V,

- $\approx_S$ denotes an **L**-equality defined by $(\langle A_1, B_1\rangle \approx_S \langle A_2, B_2\rangle) = (v_1 \approx v_2)$ where v_i ($i \in \{1,2\}$) is the (unique) element of V such that $(\sup(A_i))(v_i) = 1$. (Thus fuzzy equality between **L**-concepts $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathcal{B}(V, V, \preceq)$ is expressed by fuzzy equality between suprema of their extents.)

Proposition 1. *A completely lattice* **L***-ordered set* $\mathbf{V} = \langle\langle V, \approx_V\rangle, \preceq\rangle$ *is isomorphic to* $\langle\langle \mathcal{B}(V, V, \preceq), \approx_S\rangle, \preceq_S\rangle$.

The proposition represents a corollary of Theorem 3, but an elegant proof of this fact is a matter of further investigations.

6 Acknowledgements

The author wishes to thank the anonymous reviewers for their criticisms and useful comments.

References

1. Bělohlávek, R.: Fuzzy Galois connections, Math. Logic Quarterly 45 (4), 1999, 497–504.
2. Bělohlávek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer, New York, 2002.
3. Bělohlávek, R.: Concept lattices and order in fuzzy logic, Annals of Pure and Appl. Logic 128 (1-3), 2004, 277–298.
4. Bodenhofer, U.: A similarity-based generalization of fuzzy orderings preserving the classical axioms, Internat. J. Uncertain. Fuzziness Knowledge-based Systems 8 (5), 2000, 593–610.
5. Goguen, J. A.: L-fuzzy sets, J. Math. Anal. Appl. 18, 1967, 145–174.
6. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht, 1998.
7. Höhle, U.: On the fundamentals of fuzzy set theory, J. Math. Anal. Appl. 201, 1996, 786–826.
8. Höhle, U.: Many-valued equalities, singletons and fuzzy partitions, Soft Computing 2, 1998, 134–140.
9. Zadeh, L. A.: Similarity relations and fuzzy orderings, Inf. Sci. 3, 1971, 177–200.

On the Isomorphism Problem of Concept Algebras

Léonard Kwuida[1]* and Hajime Machida[2]

[1] Technische Universität Dresden
Institut für Algebra
D-01062 Dresden, Germany

[2] Hitotsubashi University
Department of Mathematics
2-1 Naka, Kunitachi, Tokyo 186-8601, Japan

Abstract. Weakly dicomplemented lattices are bounded lattices equipped with two unary operations to encode a negation on *concepts*. They have been introduced to capture the equational theory of concept algebras [12]. They generalize Boolean algebras. Concept algebras are concept lattices, then complete lattices, with a weak negation and a weak opposition. A special case of the representation problem for weakly dicomplemented lattices, posed in [4] is whether complete weakly dicomplemented lattices are isomorphic to concept algebras. In this contribution we give a negative answer to this question (Theorem 3). We also provide a new proof of a well known result due to M.H. Stone [8], saying that *each Boolean algebra is a field of sets* (Corollary 4).

1 Weak dicomplementation.

Definition 1. *A* **weakly dicomplemented lattice** *is a bounded lattice L equipped with two unary operations $^\triangle$ and $^\triangledown$ called* **weak complementation** *and* **dual weak complementation**, *and satisfying for all $x, y \in L$ the following equations:*

(1) $x^{\triangle\triangle} \leq x$, | *(1')* $x^{\triangledown\triangledown} \geq x$,
(2) $x \leq y \implies x^\triangle \geq y^\triangle$, | *(2')* $x \leq y \implies x^\triangledown \geq y^\triangledown$,
(3) $(x \wedge y) \vee (x \wedge y^\triangle) = x$, | *(3')* $(x \vee y) \wedge (x \vee y^\triangledown) = x$.

We call $x^\triangle$ the **weak complement** *of x and $x^\triangledown$ the* **dual weak complement** *of x. The pair $(x^\triangle, x^\triangledown)$ is called the* **weak dicomplement** *of x and the pair $(^\triangle, ^\triangledown)$ a* **weak dicomplementation** *on L. The structure $(L, \wedge, \vee, ^\triangle, 0, 1)$ is called a* **weakly complemented lattice** *and $(L, \wedge, \vee, ^\triangledown, 0, 1)$ a* **dual weakly complemented lattice**.

* was supported by the grant of Prof. Hajime Machida for a stay in Japan at Hitotsubashi University, Kunitachi, Tokyo, Japan during which this paper was discussed.

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 217–229,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

The following properties are easy to verify: (i) $x \vee x^{\triangle} = 1$, (ii) $x \wedge x^{\triangledown} = 0$, (iii) $0^{\triangle} = 1 = 0^{\triangledown}$, (iv) $1^{\triangle} = 0 = 1^{\triangledown}$, (v) $x^{\triangledown} \leq x^{\triangle}$, (vi) $(x \wedge y)^{\triangle} = x^{\triangle} \vee y^{\triangle}$, (vii) $(x \vee y)^{\triangledown} = x^{\triangledown} \wedge y^{\triangledown}$, (viii) $x^{\triangle\triangle\triangle} = x^{\triangle}$, (ix) $x^{\triangledown\triangledown\triangledown} = x^{\triangledown}$, and (x) $x^{\triangle\triangledown} \leq x^{\triangle\triangle} \leq x \leq x^{\triangledown\triangledown} \leq x^{\triangledown\triangle}$.

Example 1.

(a) The natural examples of weakly dicomplemented lattices are Boolean algebras. For $(B, \wedge, \vee, ^{-}, 0, 1)$ a Boolean algebra, $(B, \wedge, \vee, ^{-}, ^{-}, 0, 1)$ (the complementation is duplicated, i.e. $x^{\triangle} := \bar{x} =: x^{\triangledown}$) is a weakly dicomplemented lattice.
(b) Each bounded lattice can be endowed with a **trivial weak dicomplementation** by defining $(1, 1)$, $(0, 0)$ and $(1, 0)$ as the dicomplement of 0, 1 and of each $x \notin \{0, 1\}$, respectively.

Definition 2. *Let $(P, \leq)$ be a poset and $f : P \to P$ be a map. f is a* **closure operator** *on P if for all $x, y \in P$, $x \leq f(y) \iff f(x) \leq f(y)$. This is equivalent to $x \leq f(x)$, $x \leq y \implies f(x) \leq f(y)$ and $f(f(x)) = f(x)$. Usually we will write a closure operator on a set X to mean a closure operator on the powerset $(\mathcal{P}(X), \subseteq)$ of X. Dually, f is a* **kernel operator** *on P if for all $x, y \in P$, $x \geq f(y) \iff f(x) \geq f(y)$. As above, we will say that f is a kernel operator on X to mean a kernel operator on $(\mathcal{P}(X), \subseteq)$.*

For a weakly dicomplemented lattice $(L, \wedge, \vee, ^{\triangle}, ^{\triangledown}, 0, 1)$, the maps $x \mapsto x^{\triangle\triangle}$ and $x \mapsto x^{\triangledown\triangledown}$ are resp. kernel and closure operators on L. If f is a closure operator (resp. a kernel operator) on a lattice L, then $f(L)$ (with the induced order) is a lattice. Recall that for any closure operator h on L we it holds $h(h(x) \wedge h(y)) = h(x) \wedge h(y)$ as well as $h(h(x) \vee h(y)) = h(x \vee y)$, and for any kernel operator k on L it holds $k(k(x) \wedge k(y)) = k(x \wedge y)$ and $k(k(x) \vee k(y)) = k(x) \vee k(y)$. We denote by P^d the dual poset of $(P, \leq)$, i.e. $P^d := (P, \geq)$. Then f is a kernel operator on P iff f is a closure operator on P^d.

Proposition 1. *Let h be a closure operator and k a kernel operator on a set X. For $A \subseteq X$ define $A^{\triangle_h} := h(X \setminus A)$ and $A^{\triangledown_k} := k(X \setminus A)$.*

(i) $(h\mathcal{P}(X), \cap, \vee^h, ^{\triangle_h}, h\emptyset, X)$*, with $A \vee^h B := h(A \cup B)$, is a weakly complemented lattice.*
(i') $(k\mathcal{P}(X), \wedge_k, \cup, ^{\triangledown_k}, \emptyset, kX)$*, with $A \wedge_k B := k(A \cap B)$, is a dual weakly complemented lattice.*
(ii) If $h\mathcal{P}(X)$ is isomorphic to $k\mathcal{P}(Y)$, then h and k induce weakly dicomplemented lattice structures on $h\mathcal{P}(X)$ and on $k\mathcal{P}(Y)$ that are extensions of those in (i) and (i') above respectively.

Proof. For (i), let h be a closure operator on X; $(h\mathcal{P}(X), \cap, \vee^h, h\emptyset, X)$ is a bounded lattice. So we should only check the equations $(1)-(3)$ in Definition 1. For $x \in h\mathcal{P}(X)$, we have $x^{\triangle\triangle} = h(X \setminus h(X \setminus x)) \subseteq h(X \setminus (X \setminus x)) = h(x) = x$, and (1) is proved. For $x_1 \leq x_2$ in $h\mathcal{P}(X)$, we have $x_1 \subseteq x_2$ and $h(X \setminus x_1) \supseteq h(X \setminus x_2)$, and (2) is proved. Now we consider $x, y \in h\mathcal{P}(X)$. Trivially $(x \cap y) \vee^h (x \cap y^{\triangle_h}) \leq x$. In addition,

$$(x \cap y) \vee^h (x \cap y^{\triangle_h}) = (x \cap y) \vee^h (x \cap h(X \setminus y)) = h((x \cap y) \cup (x \cap h(X \setminus y)))$$
$$\supseteq h((x \cap y) \cup (x \cap (X \setminus y))) = h(x) = x.$$

(i') is proved similarly.

For (ii) we will extend the structures of (i) and (i') to get weakly dicomplemented lattices. By (i), $(h\mathcal{P}(X), \cap, \vee^h, {}^{\triangle_h}, h\emptyset, X)$ is a weakly complemented lattice. Let φ be an isomorphism from $h\mathcal{P}X$ to $k\mathcal{P}X$. We define ${}^{\nabla_\varphi}$ on $h\mathcal{P}(X)$ by: $x^{\nabla_\varphi} := \varphi^{-1}(\varphi(x)^{\nabla_k})$. Then

$$x^{\nabla_\varphi \nabla_\varphi} = \left(\varphi^{-1}\left(\varphi(x)^{\nabla_k}\right)\right)^{\nabla_\varphi} = \varphi^{-1}\left(\varphi\left(\varphi^{-1}\left(\varphi(x)^{\nabla_k}\right)\right)^{\nabla_k}\right) = \varphi^{-1}\left(\varphi(x)^{\nabla_k \nabla_k}\right),$$

and $x^{\nabla_\varphi \nabla_\varphi} \geq \varphi^{-1}(\varphi(x)) = x$. For $x \leq y$ in $h\mathcal{P}X$ we have $\varphi(x) \leq \varphi(y)$ implying $\varphi(x)^{\nabla_k} \geq \varphi(y)^{\nabla_k}$ and $x^{\nabla_\varphi} = \varphi^{-1}(\varphi(x)^{\nabla_k}) \geq \varphi^{-1}(\varphi(y)^{\nabla_k}) = y^{\nabla_\varphi}$. For x, y in $h\mathcal{P}X$, we have

$$\begin{aligned}(x \vee y) \wedge (x \vee y^{\nabla_\varphi}) &= (x \vee y) \wedge (x \vee \varphi^{-1}(\varphi(y)^{\nabla_k})) \\ &= \varphi^{-1}\left((\varphi(x) \vee \varphi(y)) \wedge (\varphi(x) \vee \varphi(y)^{\nabla_k})\right) \\ &= \varphi^{-1}(\varphi(x)) = x.\end{aligned}$$

Therefore $(h\mathcal{P}(X), \cap, \vee^h, {}^{\triangle_h}, {}^{\nabla_\varphi}, h\emptyset, X)$ is a weakly dicomplemented lattice. Similarly $(k\mathcal{P}(X), \wedge^k, \cup, {}^{\triangle_\varphi}, {}^{\nabla_k}, \emptyset, kX)$ with $x^{\triangle_\varphi} := \varphi(\varphi^{-1}(x)^{\triangle_h})$ is a weakly dicomplemented lattice.

Proposition 2. *Let h be a closure operator on X and k a kernel operator on Y such that $h\mathcal{P}(X)$ is isomorphic to $k\mathcal{P}(Y)$. Let φ be an isomorphism from $h\mathcal{P}(X)$ to $k\mathcal{P}(Y)$. We set $L := \{(x, y) \in h\mathcal{P}(X) \times k\mathcal{P}(Y) \mid y = \varphi(x)\}$. L has a weakly dicomplemented lattice structure induced by h and k.*

Proof. By Lemma 1 $(h\mathcal{P}(X), \cap, \vee^h, {}^{\triangle_h}, h\emptyset, X)$ is a weakly complemented lattice and $(k\mathcal{P}(X), \wedge_k, \cup, {}^{\nabla_k}, \emptyset, kX)$ a dual weakly complemented lattice. For every $y \in k\mathcal{P}(Y)$ there is a unique $x \in h\mathcal{P}(X)$ such that $y = \varphi(x)$. For (a, b) and (c, d) in L, we have $a \leq c \iff b \leq d$. We define a relation $\leq$ on L by: $a \leq c \iff: (a, b) \leq (c, d) :\iff b \leq d$. Then $h\mathcal{P}(X) \overset{\pi_1}{\cong} L \overset{\pi_2}{\cong} k\mathcal{P}(Y)$ where π_i is the i^{th} projection. Thus $(L, \leq)$ is a bounded lattice. Moreover $(a, b) \wedge (c, d) = (a \cap c, \varphi(a \cap c))$ and $(a, b) \vee (c, d) = (\varphi^{-1}(b \cup d), b \cup d)$. For $(x, y) \in L$, we define $(x, y)^{\triangle} := (x^{\triangle_h}, \varphi(x^{\triangle_h}))$ and $(x, y)^{\nabla} := (\varphi^{-1}(y^{\nabla_k}), y^{\nabla_k})$. We claim that $(L, \wedge, \vee, {}^{\triangle}, {}^{\nabla}, 0, 1)$ is a weakly dicomplemented lattice. In fact,

$$(x, y)^{\triangle\triangle} = (x^{\triangle_h}, \varphi(x^{\triangle_h}))^{\triangle} = (x^{\triangle_h \triangle_h}, \varphi(x^{\triangle_h \triangle_h})) \leq (x, \varphi(x)) = (x, y).$$

If $(x, y) \leq (z, t)$ in L, we have $x \leq z$ and $y \leq t$, implying $x^{\triangle_h} \geq z^{\triangle_h}$ and $\varphi(x^{\triangle_h}) \geq \varphi(z^{\triangle_h})$; thus $(x, y)^{\triangle} = (x^{\triangle_h}, \varphi(x^{\triangle_h})) \geq (z^{\triangle_h}, \varphi(z^{\triangle_h})) = (z, t)^{\triangle}$.

These prove (1) and (2) of Definition 1. It remains to prove (3). Let (x, y) and (z, t) in L;

$$\begin{aligned}((x,y)\wedge(z,t))\vee((x,y)\wedge(z,t)^{\triangle}) &= (x\cap z,\varphi(x\cap z))\vee((x,y)\wedge(z^{\triangle_h},\varphi(z^{\triangle_h})))\\ &= (x\cap z,\varphi(x\cap z))\vee(x\cap z^{\triangle_h},\varphi(x\cap z^{\triangle_h}))\\ &= (\varphi^{-1}(\varphi(x\cap z)\cup\varphi(x\cap z^{\triangle_h})),\varphi(x\cap z)\cup\varphi(x\cap z^{\triangle_h}))\\ &= ((x\cap z)\vee^h(x\cap z^{\triangle_h}),\varphi((x\cap z)\vee^h(x\cap z^{\triangle_h})))\\ &= (x,\varphi(x)).\end{aligned}$$

and (3) is proved.

The advantage of the weakly dicomplemented lattice L in Lemma 2 is that, in addition to extending the weakly and dual weakly complemented lattice structures induced by h and k, it also keeps track of the closure and kernel systems.

Definition 3. *Let L be a bounded lattice and $x \in L$. The element $x^* \in L$ (resp. $x^+ \in L$) is the pseudocomplement (resp. dual pseudocomplement) of x if*

$$x \wedge y = 0 \iff y \leq x^* \qquad (\textit{resp. } x \vee y = 1 \iff y \geq x^+) \textit{ for all } y \in L.$$

A double p-algebra is a lattice in which every element has a pseudocomplement and a dual pseudocomplement.

Example 2. Boolean algebras are double p-algebras. Finite distributive lattices are double p-algebras. N_5 is a double p-algebra that is not distributive. All distributive double p-algebras are weakly dicomplemented lattices.

The following result give a class of "more concrete" weakly dicomplemented lattices.

Proposition 3. *Let L be a finite lattice. Denote by $J(L)$ the set of join irreducible elements of L and by $M(L)$ the set of meet irreducible elements of L respectively. Define two unary operations $^{\triangle}$ and $^{\triangledown}$ on L by*

$$x^{\triangle} := \bigvee\{a \in J(L) \mid a \not\leq x\} \textit{ and } x^{\triangledown} := \bigwedge\{m \in M(L) \mid m \not\geq x\}.$$

Then $(L, \wedge, \vee, ^{\triangle}, ^{\triangledown}, 0, 1)$ is a weakly dicomplemented lattice. In general, for $G \supseteq J(L)$ and $H \supseteq M(L)$, the operations $^{\triangle_G}$ and $^{\triangledown_H}$ defined by

$$x^{\triangle_G} := \bigvee\{a \in G \mid a \not\leq x\} \quad \textit{and} \quad x^{\triangledown_H} := \bigwedge\{m \in H \mid m \not\geq x\}$$

turn $(L, \wedge, \vee, ^{\triangle_G}, ^{\triangledown_H}, 0, 1)$ into a weakly dicomplemented lattice.

Proof. Let $G \supseteq J(L)$, $b \in G$ and $x \in L$. Then $b \not\leq \bigvee\{a \in G \mid a \not\leq x\}$ implies $b \leq x$; i.e. $b \not\leq x^{\triangle_G} \implies b \leq x$. Thus $x^{\triangle_G\triangle_G} = \bigvee\{b \in G \mid b \not\leq x^{\triangle_G}\} \leq x$ and (1) is proved. For $x \leq y$ we have $\{a \in G \mid a \not\leq x\} \supseteq \{a \in G \mid a \not\leq y\}$ implying $x^{\triangle_G} \geq y^{\triangle_G}$, and (2) is proved. For (3), it is enough to prove that for $a \in J(L)$, $a \leq x \iff a \leq (x \wedge y) \vee (x \wedge y^{\triangle_G})$, since $J(L)$ is $\bigvee$-dense in L. Let $a \leq x$. We have $a \leq y$ or $a \leq y^{\triangle_G}$. Then $a \leq x \wedge y$ or $a \leq x \wedge y^{\triangle_G}$. Thus $a \leq (x \wedge y) \vee (x \wedge y^{\triangle_G})$. The reverse inequality is obvious. $(1') - (3')$ are proved similarly.

Example 3 above is a special case of concept algebras. Before we introduce concept algebras, let us recall some notions from Formal Concept Analysis (FCA). The reader is refered to [5]. Formal Concept Analysis was born in the eighties from the formalization of the notion of *concept* [10]. Traditional philosophers considered a **concept** to be determined by its extent and its intent. The extent consists of all objects belonging to the concept while the intent is the set of all attributes shared by all objects of the concept. In general, it may be difficult to list all objects or attributes of a concept. Therefore a specific *context* should be fixed to enable formalization. A **formal context** is a triple (G, M, I) of sets such that $I \subseteq G \times M$. The members of G are called **objects** and those of M **attributes**. If $(g, m) \in I$, then the object g is said to have m as an attribute. For subsets $A \subseteq G$ and $B \subseteq M$, A' and B' are defined by

$$A' := \{m \in M \mid \ \forall g \in A \ g \, \mathrm{I} \, m\} \quad \text{and} \quad B' := \{g \in G \mid \ \forall m \in B \ g \, \mathrm{I} \, m\}.$$

A **formal concept** of the formal context (G, M, I) is a pair (A, B) with $A \subseteq G$ and $B \subseteq M$ such that $A' = B$ and $B' = A$. The set A is called the **extent** and B the **intent** of the concept (A, B). $\mathfrak{B}(G, M, I)$ denotes the set of all formal concepts of the formal context (G, M, I). For concepts (A, B) and (C, D), (A, B) is called a **subconcept** of (C, D) provided that $A \subseteq C$ (which is equivalent to $D \subseteq B$). In this case, (C, D) is a **superconcept** of (A, B) and we write $(A, B) \leq (C, D)$. The relation **subconcept-superconcept** encodes the **hierarchy on concepts**, namely, that a concept is more general if it contains more objects, and equivalently, if it is determined by less attributes.

Theorem 1 ([10]). *The concept lattice $\mathfrak{B}(G, M, I)$ is a complete lattice in which infimum and supremum are given by:*

$$\bigwedge_{t \in T} (A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)'' \right) \ \textit{and} \ \bigvee_{t \in T} (A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)'', \bigcap_{t \in T} B_t \right).$$

A complete lattice L is isomorphic to $\mathfrak{B}(G, M, I)$ iff there are mappings $\tilde{\gamma} : G \to L$ and $\tilde{\mu} : M \to L$ such that $\tilde{\gamma}(G)$ is supremum-dense, $\tilde{\mu}(M)$ is infimum-dense and $g \, \mathrm{I} \, m \iff \tilde{\gamma} g \leq \tilde{\mu} m$ for all $(g, m) \in G \times M$.

$(\mathfrak{B}(G, M, I); \leq)$ is called the **concept lattice** of the context (G, M, I). All complete lattices are (copies of) concept lattices. We adopt the notations below for $g \in G$ and $m \in M$:

$$g' := \{g\}', \quad m' := \{m\}', \quad \gamma g := (g'', g') \quad \text{and} \quad \mu m := (m', m'').$$

The concept γg is called **object concept** and μm **attribute concept**. The sets γG is supremum-dense and μM infimum-dense in $\mathfrak{B}(G, M, I)$. We usually assume our context clarified, meaning that $x' = y' \implies x = y$ for all $x, y \in G \cup M$. If γg is supremum-irreducible we say that the **object** g is **irreducible**. An **attribute** m is said **irreducible** if the attribute concept μm is infimum-irreducible. A **formal context** is called **reduced** if all its objects and attributes are irreducible. For every finite lattice L there is, up to isomorphism, a unique reduced context

$\mathbb{K}(L) := (J(L), M(L), \leq)$ such that $L \cong \mathfrak{B}(\mathbb{K}(L))$. We call it **standard context** of L. The meet and join operations in the concept lattice can be used to formalize the conjunction and disjunction on concepts [6]. To formalize a negation two operations are introduced as follow:

Definition 4. *Let (G, M, I) be a formal context and (A, B) a formal concept. We define*

$$\text{its } \textbf{weak negation} \text{ by } \quad (A,B)^{\triangle} := \left((G \setminus A)'', (G \setminus A)'\right)$$

$$\text{and its } \textbf{weak opposition} \text{ by } \quad (A,B)^{\triangledown} := \left((M \setminus B)', (M \setminus B)''\right).$$

$\mathfrak{A}(\mathbb{K}) := \left(\mathfrak{B}(\mathbb{K}); \wedge, \vee, ^{\triangle}, ^{\triangledown}, 0, 1\right)$ is called the **concept algebra** *of the formal context $\mathbb{K}$, where $\wedge$ and $\vee$ denote the meet and the join operations of the concept lattice.*

These operations satisfy the equations in Definition 1 (cf. [12]). In fact concept algebras are typical examples of weakly dicomplemented lattices. One of the important and still unsolved problems in this topic is to find out the equational theory of concept algebras; that is the set of all equations valid in all concept algebras. Is it finitely generated? i.e. is there a finite set $\mathcal{E}$ of equations valid in all concept algebras such that each equation valid in all concept algebras follows from $\mathcal{E}$? We start with the set of equations defining a weakly dicomplemented lattice and have to check whether they are enough to represent the equational theory of concept algebras. This representation problem can be split:

strong representation: Describe weakly dicomplemented lattices that are isomorphic to concept algebras.

equational axiomatization: Find a set of equations that generate the equational theory of concept algebras.

concrete embedding: Given a weakly dicomplemented lattice L, is there a context $\mathbb{K}^{\triangle}_{\triangledown}(L)$ such that L can be embedded into the concept algebra of $\mathfrak{A}\left(\mathbb{K}^{\triangle}_{\triangledown}(L)\right)$?

We proved (see [4] or [3]) that finite distributive weakly dicomplemented lattices are isomorphic to concept algebras. However we cannot expect all weakly dicomplemented lattices to be isomorphic to concept algebras, since concept algebras are first of all complete lattices. In Section 3 we will show that being complete is not enough for weakly dicomplemented lattices to be isomorphic to concept algebras. Before that we show in Section 2 that weakly dicomplemented lattices generalize Boolean algebras.

2 Weakly Dicomplemented Lattices with Negation

Example 1 states that duplicating the complementation of a Boolean algebra leads to a weakly dicomplemented lattice. Does the converse hold? The finite case is easily obtained [Corollary 1]. Major parts of this section are taken from [4]. We will also describe weakly dicomplemented lattices whose Boolean part is the intersection of their skeletons (definitions below).

Definition 5. *A weakly dicomplemented lattice is said to be* **with negation** *if the unary operations coincide, i.e., if* $x^{\triangledown} = x^{\triangle}$ *for all* x*. In this case we set* $x^{\triangle} =: \bar{x} := x^{\triangledown}$*.*

Lemma 1. *A weakly dicomplemented lattice with negation is uniquely complemented.*

Proof. $x^{\triangle\triangle} \leq x \leq x^{\triangledown\triangledown}$ implies that $x = \bar{\bar{x}}$. Moreover, $x \wedge \bar{x} = 0$ and $\bar{x}$ is a complement of x. If y is another complement of x then

$$x = (x \wedge y) \vee (x \wedge \bar{y}) = x \wedge \bar{y} \implies x \leq \bar{y}$$

$$x = (x \vee y) \wedge (x \vee \bar{y}) = x \vee \bar{y} \implies x \geq \bar{y}$$

Then $\bar{y} = x$ and $\bar{x} = y$. L is therefore a uniquely complemented lattice.

It can be easily seen that each uniquely complemented atomic lattice is a copy of the power set of the set of its atoms, and therefore distributive. Thus

Corollary 1. *The finite weakly dicomplemented lattices with negation are exactly the finite Boolean algebras.*

Of course, the natural question will be if the converse of Lemma 1 holds. That is, can any uniquely complemented lattice be endowed with a structure of a weakly dicomplemented lattice with negation? The answer is yes for distributive lattices. If the assertion of Corollary 1 can be extended to lattices in general, the answer will unfortunately be no. In fact R. P. Dilworth proved that each lattice can be embedded into a uniquely complemented lattice [**?**]. The immediate consequence is the existence of non-distributive uniquely complemented lattices. They are however infinite. If a uniquely complemented lattice could be endowed with a structure of weakly dicomplemented lattice, it would be distributive. This cannot be true for non distributive uniquely complemented lattices.

Lemma 2. *Each weakly dicomplemented lattice with negation* L *satisfies the de Morgan laws.*

Proof. We want to prove that $\overline{x \wedge y} = \bar{x} \vee \bar{y}$.

$$(\bar{x} \vee \bar{y}) \vee (x \wedge y) \geq \bar{x} \vee (x \wedge \bar{y}) \vee (x \wedge y) = \bar{x} \vee x = 1$$

and

$$(\bar{x} \vee \bar{y}) \wedge (x \wedge y) \leq (\bar{x} \vee \bar{y}) \wedge x \wedge (\bar{x} \vee y) = \bar{x} \wedge x = 0.$$

So $\bar{x} \vee \bar{y}$ is a complement of $x \wedge y$, hence by uniqueness it is equal to $\overline{x \wedge y}$. Dually we have $\overline{x \vee y} = \bar{x} \wedge \bar{y}$.

Now for the distributivity we can show that

Lemma 3. $\overline{x \wedge (y \vee z)}$ *is a complement of* $(x \wedge y) \vee (x \wedge z)$*.*

Proof. Since in every lattice the equation $x \wedge (y \vee z) \geq (x \wedge y) \vee (x \wedge z)$ holds, we have that $\overline{x \wedge (y \vee z)} \leq \overline{(x \wedge y) \vee (x \wedge z)}$; so we have to show only that $\overline{x \wedge (y \vee z)} \vee (x \wedge y) \vee (x \wedge z) = 1$. Using the de Morgan laws and axiom (3) several times we obtain:

$$\begin{aligned}\overline{x \wedge (y \vee z)} \vee (x \wedge y) \vee (x \wedge z) &= \bar{x} \vee (\bar{y} \wedge \bar{z}) \vee (x \wedge y) \vee (x \wedge z)\\ &= \bar{x} \vee (\bar{y} \wedge \bar{z} \wedge x) \vee (\bar{y} \wedge \bar{z} \wedge \bar{x}) \vee (x \wedge y \wedge z)\\ &\quad \vee (x \wedge y \wedge \bar{z}) \vee (x \wedge z \wedge \bar{y})\\ &= \bar{x} \vee (\bar{y} \wedge \bar{z} \wedge \bar{x}) \vee (x \wedge y \wedge z) \vee (x \wedge y \wedge \bar{z})\\ &\quad \vee (x \wedge \bar{y} \wedge z) \vee (x \wedge \bar{y} \wedge \bar{z})\\ &= \bar{x} \vee (\bar{y} \wedge \bar{z} \wedge \bar{x}) \vee (x \wedge y) \vee (x \wedge \bar{y})\\ &= \bar{x} \vee (\bar{y} \wedge \bar{z} \wedge \bar{x}) \vee x\\ &= 1.\end{aligned}$$

Thus $\overline{x \wedge (y \vee z)}$ is a complement of $(x \wedge y) \vee (x \wedge z)$.

Since the complement is unique we get the equality

$$x \wedge (y \vee z) = \overline{\overline{x \wedge (y \vee z)}} = (x \wedge y) \vee (x \wedge z).$$

Thus weakly dicomplemented lattices generalize Boolean algebras in the following sense

Theorem 2. *Boolean algebras with the complementation duplicated*[3] *are weakly dicomplemented lattices. If* $^{\triangle} = ^{\triangledown}$ *in a weakly dicomplemented lattice* L*, then* $(L, \wedge, \vee, ^{-}, 0, 1)$*, with* $\bar{x} := x^{\triangle} = x^{\triangledown}$ *for all* $x \in L$*, is a Boolean algebra.*

As the equality $x^{\triangle} = x^{\triangledown}$ not always holds, we can look for a maximal subset with this property.

Definition 6. *For any weakly dicomplemented lattice* L*, we will call* $B(L) := \{x \in L \mid x^{\triangle} = x^{\triangledown}\}$ *the* **subset of elements with negation**.

As in Definition 5 we denote by $\bar{x}$ the common value of $x^{\triangle}$ and $x^{\triangledown}$. We set $L^{\triangle} := \{a^{\triangle} \mid a \in L\} = \{a \in L \mid a^{\triangle\triangle} = a\}$ and call it the **skeleton** of L, as well as $L^{\triangledown} := \{a^{\triangledown} \mid a \in L\} = \{a \in L \mid a^{\triangledown\triangledown} = a\}$ and call it the **dual skeleton** of L.

Corollary 2. $(B(L), \wedge, \vee, ^{-}, 0, 1)$ *is a Boolean algebra that is a subalgebra of the skeleton and the dual skeleton.*

Proof. From $x^{\triangle} = x^{\triangledown}$ we get $x^{\triangle\triangle} = x^{\triangledown\triangle}$ and $x^{\triangle\triangledown} = x^{\triangledown\triangledown}$. Thus

$$x^{\triangle\triangledown} = x^{\triangle\triangle} = x = x^{\triangledown\triangledown} = x^{\triangledown\triangle}$$

[3] see Example 1

and $B(L)$ is closed under the operations $^\triangle$ and $^\triangledown$. We will prove that $B(L)$ is a subalgebra of L. We consider x and y in $B(L)$. We have

$$(x \wedge y)^\triangle = x^\triangle \vee y^\triangle = x^\triangledown \vee y^\triangledown \leq (x \wedge y)^\triangledown \leq (x \wedge y)^\triangle \text{ and}$$

$$(x \vee y)^\triangledown = x^\triangledown \wedge y^\triangledown = x^\triangle \wedge y^\triangle \geq (x \vee y)^\triangle \geq (x \vee y)^\triangledown.$$

Thus $x \wedge y$ and $x \vee y$ belong to $B(L)$. $B(L)$ is a weakly dicomplemented lattice with negation, and is by Theorem 2, a Boolean algebra.

While proving Corollary 2 we show that $B(L)$ is a subalgebra of L. It is the largest Boolean algebra that is a subalgebra of the skeletons and of L. We call it the **Boolean part** of L. The inclusion $B(L) \subseteq L^\triangle \cap L^\triangledown$ can be strict (see Fig. 1). It would be nice to find under which conditions the Boolean part is the intersection of the skeleton and dual skeleton?

Lemma 4. *If L is a finite distributive lattice with $^\triangledown = *$ (pseudocomplementation) and $^\triangle = +$ (dual pseudocomplementation), then $B(L)$ is the set of complemented elements of L.*

Proof. Let L be a finite distributive lattice with $^\triangledown = *$ and $^\triangle = +$. We denote by $C(L)$ the set of complemented elements of L. Of course $B(L) \subseteq C(L)$. Let $x \in C(L)$. From the distributivity there is a unique elements $z \in L$ such that $x \vee z = 1$ and $x \wedge z = 0$. Then $z \leq x^\triangledown \leq x^\triangle \leq z$, and $x \in B(L)$.

Even in this case, the Boolean part can still be strictly smaller than the intersection of the skeletons (see Fig. 1 below).

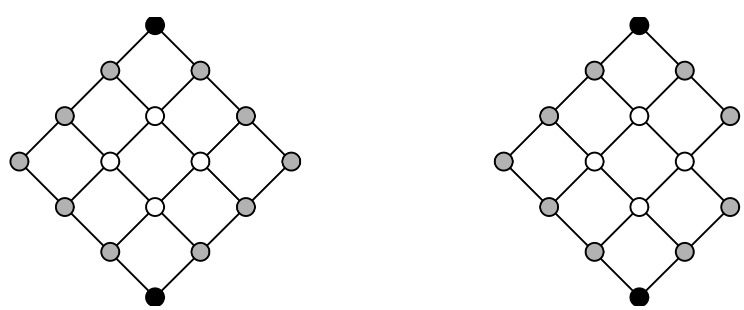


Fig. 1. Examples of dicomplementations. For L_1, the elements c, b and a are each image (of their image). The operation $^\triangle$ is the dual of $^\triangledown$. We have $B(L_1) = \{0, 1\}$, $L_1^\triangle = \{0, 1, c, d, e, c^\triangle, d^\triangle, e^\triangle\}$, $L_1^\triangledown = \{0, 1, c, a, b, c^\triangledown, a^\triangledown, b^\triangledown\}$ and $C(L_1) = \{0, 1, c, a^\triangledown\}$. Thus $B(L_1) \subsetneq C(L_1) = L_1^\triangle \cap L_1^\triangledown$. For L_2, $^\triangle =^+$ and $^\triangledown =^*$. $L_2^\triangle = \{0, 1, c, c^\triangle\}$, $L_2^\triangledown = \{0, 1, c, c^\triangledown\}$, $B(L_2) = \{0, 1\} = C(L_2) \subsetneq \{0, 1, c\} = L_2^\triangle \cap L_2^\triangledown$.

Lemma 5. $B(L) = L^\triangle \cap L^\triangledown$ *iff* $x^{\triangle\triangle} = x^{\triangledown\triangledown} \implies x^{\triangle\triangledown} = x^{\triangledown\triangle}$.

Proof. ($\Rightarrow$). Let $x \in L$ such that $x^{\triangle\triangle} = x^{\bigtriangledown\bigtriangledown}$. Then $x \in L^{\triangle} \cap L^{\bigtriangledown} = B(L)$ and implies $x^{\triangle} = x^{\bigtriangledown}$. Therefore $x^{\triangle\bigtriangledown} = x^{\bigtriangledown\bigtriangledown} = x = x^{\triangle\triangle} = x^{\bigtriangledown\triangle}$.

($\Leftarrow$). Let $x \in L^{\triangle} \cap L^{\bigtriangledown}$. Then $x^{\triangle\triangle} = x = x^{\bigtriangledown\bigtriangledown}$ and implies $x^{\triangle} = x^{\bigtriangledown\triangle\triangle} \leq x^{\bigtriangledown}$. Thus $x^{\triangle} = x^{\bigtriangledown}$, and $x \in B(L)$.

3 Strong representation problem

We start this section by a negative result, namely by showing that *completeness is not enough for weakly dicomplemented lattices to be (copies of) concept algebras.*

Theorem 3. *There is no formal context whose concept algebra is isomorphic to a complete atomfree Boolean algebra.*

Proof. Let B be a complete and atomfree Boolean algebra. By Theorem 1, there is a context (G, M, I) such that $\mathfrak{B}(G, M, I) \cong B$ (lattice isomorphism). Without loss of generality, we can assume that (G, M, I) is a subcontext of $(B, B, \leq)$. We claim that there are $g, h \in G$ with $0 < h < g < 1$. In fact, for an element $g \in G \subseteq B$ with $0 \neq g$ there is $a \in B$ such that $0 < a < g$, since B is atomfree. Moreover G is $\bigvee$-dense in B and then $0 \neq a = \bigvee\{x \in G \mid x \leq a\}$, implying that there $\{x \in G \mid 0 < x \leq a\} \neq \emptyset$. Thus we can choose $h \in G$ with $0 < h \leq a < g$. In the concept algebra of $(G, M, \leq)$ we have $h^{\triangle} = \bigvee\{x \in G \mid x \nleq h\} \geq g > h$. From $h \vee h^{\triangle} = 1$ we get $h^{\triangle} = 1 \neq h'$ (the complement of h in B).

Theorem 3 says that an atomfree Boolean algebra is not isomorphic to a concept algebra. However it can be embedded into a concept algebra. The corresponding context is constructed via ultrafilters. A general construction was presented in [4].

Definition 7. *A* **primary filter** *is a (lattice) filter that contains w or $w^{\triangle}$ for all $w \in L$. Dually, a* **primary ideal** *is an ideal that contains w or $w^{\bigtriangledown}$ for all $w \in L$. $\mathfrak{F}_{\text{pr}}(L)$ denotes the set of all primary filters and $\mathfrak{I}_{\text{pr}}(L)$ the set of primary ideals of L.*

For Boolean algebras, a proper filter F is primary iff it is an ultrafilter, iff it is a prime filter ($x \vee y \in F \implies x \in F$ or $y \in F$). The following result, based on Zorn's lemma provides the sets of $\mathbb{K}^{\triangle}_{\bigtriangledown}$.

Theorem 4 ("Prime ideal theorem"). *For every filter F and every ideal I such that $F \cap I = \emptyset$ there is a primary filter G containing F and disjoint from I. Dually, for every ideal I and every filter F such that $I \cap F = \emptyset$ there is a primary ideal J containing I and disjoint from F.*

Corollary 3. *If $x \nleq y$ in L, then there exists a primary filter F containing x and not y.*

For $x \in L$, we set

$$\mathcal{F}_{\text{x}} := \{F \in \mathfrak{F}_{\text{pr}}(L) \mid x \in F\} \quad \text{and} \quad \mathcal{I}_{\text{x}} := \{I \in \mathfrak{I}_{\text{pr}}(L) \mid x \in I\}.$$

The **canonical context** of a weakly dicomplemented lattice L is the formal context

$$\mathbb{K}_{\triangledown}^{\triangle}(L) := (\mathfrak{F}_{\mathrm{pr}}(L), \mathfrak{I}_{\mathrm{pr}}(L), \square) \quad \text{with } F \,\square\, I :\iff F \cap I \neq \emptyset.$$

The derivation in $\mathbb{K}_{\triangledown}^{\triangle}(L)$ yields, $\mathcal{F}_{\mathrm{x}}' = \mathcal{I}_{\mathrm{x}}$ and $\mathcal{I}_{\mathrm{x}}' = \mathcal{F}_{\mathrm{x}}$ for every $x \in L$. Moreover, the map

$$\begin{aligned} i\colon L &\to \mathfrak{B}\left(\mathbb{K}_{\triangledown}^{\triangle}(L)\right) \\ x &\mapsto (\mathcal{F}_{\mathrm{x}}, \mathcal{I}_{\mathrm{x}}) \end{aligned}$$

is a bounded lattice embedding with $i(x^{\triangledown}) \leq i(x)^{\triangledown} \leq i(x)^{\triangle} \leq i(x^{\triangle})$. If the first and last inequalities above were equalities, we would get a weakly dicomplemented embedding into the concept algebra of $\mathbb{K}_{\triangledown}^{\triangle}(L)$. This would give a solution to the representation problem of weakly dicomplemented lattices.

Theorem 5. *If L is a Boolean algebra, then the concept algebra of $\mathbb{K}_{\triangledown}^{\triangle}(L)$ is a complete and atomic Boolean algebra into which L embeds.*

Proof. If B is a Boolean algebra, then a proper filter F of L is primary iff it is an ultrafilter, and a proper ideal J is primary iff it is maximal. Thus $\mathfrak{F}_{\mathrm{pr}}(L)$ is the set of ultrafilters of L and $\mathfrak{I}_{\mathrm{pr}}(L)$ the set of its maximal ideals. In addition, the complement of an ultrafilter is a maximal ideal and vice-versa. For $F \in \mathfrak{F}_{\mathrm{pr}}(L)$, $L \setminus F$ is the only primary ideal that does not intersect F, and for any $J \in \mathfrak{I}_{\mathrm{pr}}(L)$, $L \setminus J$ is the only primary filter that does not intersect J. Thus the context $\mathbb{K}_{\triangledown}^{\triangle}(L)$ is a copy of $(\mathfrak{F}_{\mathrm{pr}}(L), \mathfrak{F}_{\mathrm{pr}}(L), \neq)$. The concepts of this context are exactly pairs (A, B) such that $A \cup B = \mathfrak{F}_{\mathrm{pr}}(L)$ and $A \cap B = \emptyset$. Thus $\mathfrak{B}(\mathbb{K}_{\triangledown}^{\triangle}(L)) \cong \mathcal{P}(\mathfrak{F}_{\mathrm{pr}}(L))$ and each subset A of $\mathfrak{F}_{\mathrm{pr}}(L)$ is an extent of $\mathbb{K}_{\triangledown}^{\triangle}(L)$. It remains to prove that the lattice embedding

$$\begin{aligned} i\colon L &\to \mathfrak{B}\left(\mathbb{K}_{\triangledown}^{\triangle}(L)\right) \\ x &\mapsto (\mathcal{F}_{\mathrm{x}}, \mathcal{I}_{\mathrm{x}}) \end{aligned}$$

is also a Boolean algebra embedding. If $i(x^{\triangle}) \neq i(x)^{\triangle}$ then there is $F \in \mathcal{F}_{\mathrm{x}^{\triangle}} \setminus (\mathfrak{F}_{\mathrm{pr}}(L) \setminus \mathcal{F}_{\mathrm{x}})'' = \mathcal{F}_{\mathrm{x}^{\triangle}} \setminus (\mathfrak{F}_{\mathrm{pr}}(L) \setminus \mathcal{F}_{\mathrm{x}}) = \emptyset$, which is a contradiction. Similarly $i(x^{\triangledown}) = i(x)^{\triangledown}$. Therefore B embeds into the complete and atomic Boolean algebra $\mathcal{A}\left(\mathbb{K}_{\triangledown}^{\triangle}(L)\right)$ which is a copy of $\mathcal{P}\left(\mathfrak{F}_{\mathrm{pr}}(L)\right)$.

The above result is a new proof to a well-known result (Corollary 4) due to Marshall Stone [8]. The advantage here is that the proof is very simple and does not require any knowledge from topology. Recall that a field of subsets of a set X is a subalgebra of $\mathcal{P}(X)$, .i.e. a family of subsets of X that contains $\emptyset$ and X, and that is closed under union, intersection, and complementation.

Corollary 4 ([8]). *Each Boolean algebra embeds into a field of sets.*

We conclude this section by an example. Consider the Boolean algebra $F\mathbb{N}$ of finite and cofinite subsets of $\mathbb{N}$. It is not complete. But $\mathcal{P}(\mathbb{N})$ is a complete and atomic Boolean algebra containing $F\mathbb{N}$. By Theorem 5 $\mathfrak{A}(\mathbb{K}^{\triangle}_{\triangledown}(B))$ is also a complete and atomic Boolean algebra into which $F\mathbb{N}$ embeds. The atoms of $F\mathbb{N}$ are $\{n\}, n \in \mathbb{N}$. These generate its principal ultrafilters. $F\mathbb{N}$ has exactly one non-principal ultrafilter U (the cofinite subsets). Thus $|F\mathbb{N}| = |\mathbb{N}| + 1 = |\mathbb{N}|$. We can find a bijection let say f between the atoms of $\mathcal{P}(\mathbb{N})$ and the atoms of $\mathfrak{A}(\mathbb{K}^{\triangle}_{\triangledown}(F\mathbb{N}))$. f induces an isomorphism $\hat{f} : \mathcal{P}(\mathbb{N}) \to \mathfrak{A}(\mathbb{K}^{\triangle}_{\triangledown}(F\mathbb{N}))$. Is there a universal property for $\mathfrak{A}(\mathbb{K}^{\triangle}_{\triangledown}(B))$ of Boolean algebras. For example *is* $\mathfrak{A}(\mathbb{K}^{\triangle}_{\triangledown}(B))$ *the smallest complete and atomic Boolean algebra into which* B *can be embedded*?

4 Conclusion

Weakly dicomplemented lattices with negation are exactly Boolean algebras (Thm. 2). Even if they are not always isomorphic to concept algebras (Thm. 3), they embed into concept algebras (Thm. 5). Finite distributive weakly dicomplemented lattices are isomorphic to concept algebras [3]. Extending these results to finite weakly dicomplemented lattices in one sense and to distributive weakly dicomplemented lattices in the other are the next tasks. Finding a kind of universal property to characterize the construction in Thm. 5 is a natural question to be addressed.

References

1. Boole., G.: *An investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities.* Macmillan 1854. Reprinted by Dover Publ. New york (1958).
2. Davey, B. A. Priestley, H. A.: *Introduction to lattices and order.* Second edition Cambridge (2002).
3. Ganter, B. Kwuida, L.: *Finite distributive concept algebras.* Order. (2007).
4. Kwuida, L.: *Dicomplemented lattices. A contextual generalization of Boolean algebras.* Dissertation TU Dresden. Shaker Verlag. (2004).
5. Ganter, B. Wille, R.: *Formal Concept Analysis. Mathematical Foundations.* Springer (1999).
6. Ganter, B. Wille, R.: *Contextual attribute logic.* in W. Tepfenhart, W. Cyre (Eds) *Conceptual Structures: Standards and Practices.* LNAI **1640**. Springer, Heidelberg (1999) 337-338.
7. Priestley, H. A.: *Representation of distributive lattices by means of ordered Stone spaces.* Bull. London Math. Soc. **2** (1970) 186-190.
8. Stone, M. H.: *The theory of representations for Boolean algebras.* Trans. Amer. Math. Soc. **40** (1936) 37-111.
9. Wansing, H. (Ed.): *Negation: a notion in focus. Perspectives in analytical philosophy.* **7** de Gruyter (1996).
10. Wille, R.: *Restructuring lattice theory: an approach based on hierarchies of concepts.* in I. Rival (Ed.) *Ordered Sets.* Reidel (1982) 445-470.

11. Wille, R.: *Restructuring mathematical logic: an approach based on Peirce's pragmatism.* in Ursini, Aldo (Ed.) *Logic and algebra* Marcel Dekker. Lect. Notes Pure Appl. Math. **180** (1996) 267-281.
12. Wille, R.: *Boolean Concept Logic* in B. Ganter & G.W. Mineau (Eds.) ICCS 2000 *Conceptual Structures: Logical, Linguistic, and Computational Issues* Springer LNAI **1867** (2000) 317-331.

Factorization of Concept Lattices with Hedges by Means of Factorization of Residuated Lattices

Michal Krupka

Dept. Computer Science,
Palacký University, Olomouc,
Tomkova 40,
CZ-779 00, Olomouc,
Czech Republic,
michal.krupka@upol.cz

Abstract. In the first part, we extend our results from a previous paper on factorization of residuated lattices to residuated lattices with hedges. In the second part, we show how this result can be applied to the problem of factorization of fuzzy concept lattices with hedges. Our approach is that instead of factorizing the original concept lattice with hedges we construct a new data table with fuzzy values of attributes in a factorized residuated lattice with hedges and show that the induced concept lattice is isomorphic to the factor concept lattice.

1 Introduction

Formal concept analysis (FCA) is a popular method for analysis of object-attribute data [11], [9]. Its aim is to process data in a tabular form (describing objects and their attributes) and extract interesting clusters, called formal concepts, which correspond to maximal rectangles in the processed data table. These formal concepts form a concept lattice, which represents the main output of the method.

In the case of formal concept analysis of data with fuzzy values of attributes the domain for data can consist of more than two elements (representing degrees to which particular objects can have particular attributes). Since the number of formal concepts can be large in this case, several methods of reducing the size of resulting concept lattice have been proposed. In this paper, we consider two of them: factorization and hedges.

The idea behind factorization of fuzzy concept lattices is that instead of considering the original concept lattice, which can be very large, we accept not to distinguish between formal concepts which are sufficiently similar. This can be done by choosing a degree of similarity of formal concepts and factorizing the concept lattice by the tolerance relation induced by this degree. As the result, we obtain a smaller lattice, whose size depends on the prescribed degree. This parametrized size reduction method has been introduced in [1] and further improved in [3], see also [2].

In [8], the notion of fuzzy concept lattice with hedges was introduced (see also [4], [5]). It can be viewed as another tool for reducing size of concept lattices. It introduces two additional parameters, called (truth-stressing) hedges, which are unary functions on the scale of truth degrees and can be seen as truth functions of connectives “very

© Radim Belohlavek, Sergei O. Kuznetsov (Eds.): CLA 2008, pp. 231–241,
ISBN 978–80–244–2111–7, Palacký University, Olomouc, 2008.

true". Hedges can be used as parameters selecting "important attributes" and "important objects". Stronger hedges lead to smaller number of extracted concepts.

In [6], these two approaches (factorization and hedges) were combined and a method of factorizing fuzzy concept lattices with hedges was introduced.

In [17], we dealt with residuated lattices, which are frequently used as structures of truth values in fuzzy logic, and as such are also used in the above papers. We showed (using results of [10] and [18]) that residuated lattices can be factorized by means of a prescribed degree of similarity of truth values. We also stated a general idea of approximate size reduction of fuzzy systems by factorizing the underlying structure of truth values (i.e., a residuated lattice) by a tolerance relation, induced by the user-prescribed degree to which we allow different truth values to be non-distinguishable. We also showed that this general idea is applicable to fuzzy concept lattices: factorized fuzzy concept lattice is in fact isomorphic to another concept lattice, constructed from a data table with values from factor residuated lattice.

In this paper, we first generalize our results from [17] to residuated lattices with hedges. We show that any hedge on a residuated lattice induces a hedge on the factorized residuated lattice. The only limitation is that the prescribed similarity degree must be a fixpoint of the used hedge (similar condition appears also in [6]).

In the next part we show that factor fuzzy concept lattices with hedges can be again described by means of factor residuated lattices with hedges. More precisely, we show that each factor fuzzy concept lattice with hedges is isomorphic to a fuzzy concept lattice with hedges built on a data table with values from the factorized residuated lattice.

This paper is organized as follows. In Section 2 we summarize basic known facts on residuated lattices, fuzzy sets, factorization of residuated lattices and factorization of concept lattices with hedges. In Section 3 we give our two main results on factorization of residuated lattices with hedges and factorization of concept lattices with hedges.

2 Preliminaries

2.1 Residuated lattices and fuzzy sets

We use complete residuated lattices as structures of truth values. We recall only basic facts here, for more detailed review, we refer the reader to [2], [12].

A complete residuated lattice is defined as an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice with the least element 0 and the greatest element 1; $\langle L, \otimes, 1\rangle$ is a commutative monoid (i.e. $\otimes$ is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$); $\otimes$ (product) and $\rightarrow$ (residuum) satisfy so-called adjointness property: $a \otimes b \leq c$ iff $a \leq b \rightarrow c$ for each $a, b, c \in L$. Elements of L are called truth degrees. $\otimes$ and $\rightarrow$ are (truth functions of) "fuzzy conjunction" and "fuzzy implication".

For each complete residuated lattice we consider a derived (truth function of) logical connective $\leftrightarrow$ ("fuzzy equivalence") defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$. $\leftrightarrow$ is called biresiduum and is used for measuring similarity of truth degrees.

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

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

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

$$\text{Gödel:} \quad \begin{aligned} a \otimes b &= \min(a,b), \\ a \to b &= \begin{cases} 1 & \text{if } a \le b, \\ b & \text{otherwise,} \end{cases} \end{aligned} \tag{2}$$

$$\text{Goguen (product):} \quad \begin{aligned} a \otimes b &= a \cdot b, \\ a \to b &= \begin{cases} 1 & \text{if } a \le b, \\ \frac{b}{a} & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

Complete residuated lattices on $[0,1]$ given by (1), (2), and (3) are called standard Łukasiewicz, Gödel, Goguen (product) algebras, respectively.

The class of complete residuated lattices include finite structures as well. For instance, we can put $L_{n+1} = \{a_0 = 0, a_1, \ldots, a_n = 1\} \subseteq [0,1]$, where $a_0 < \cdots < a_n$ are equidistant and $\otimes$ and $\to$ are restrictions of the operations from (1). In this case, the residuated lattice $\mathbf{L_{n+1}} = \langle L_{n+1}, \min, \max, \otimes, \to, 0, 1\rangle$ is called an equidistant Łukasiewicz chain.

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

A hedge (or truth stresser) on residuated lattice $\mathbf{L}$ is a unary operation $*$ satisfying (i) $1^* = 1$, (ii) $a^* \le a$, (iii) $(a \to b)^* \le a^* \to b^*$, (iv) $a^{**} = a^*$, for $a, b \in L$. A hedge $*$ is a (truth function of) logical connective "very true" [13].

Among all hedges on any residuated lattice, the greatest one is given by $a^* = a$ and is called (obviously) identity. The smallest hedge is called globalization and is given by $1^* = 1$ and $a^* = 0$ for $a < 1$. In Fig. 1 there are depicted all possible hedges on $\mathbf{L_5}$.

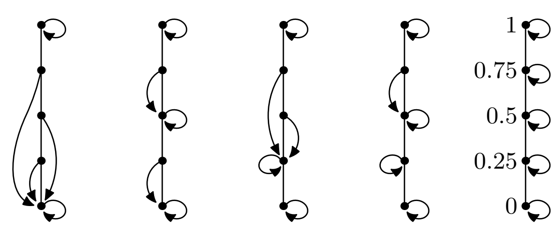


Fig. 1. All hedges on $\mathbf{L_5}$

Element $a \in L$ is said to be a fixpoint of hedge $*$ if $a^* = a$. For two fixpoints a_1, a_2 of $*$, the product $a \otimes b$ is also a fixpoint of $*$.

Recall that an $\mathbf{L}$-set (or fuzzy set) A in universe X is a mapping $A: X \to L$. For any $x \in X$, $A(x)$ is interpreted as the degree to which x belongs to A. For two such $\mathbf{L}$-sets

A_1, A_2, the degree of their similarity $A_1 \approx^X A_2 \in L$ is defined by

$$A_1 \approx^X A_2 = \bigwedge_{x \in X} A_1(x) \leftrightarrow A_2(x). \tag{4}$$

2.2 Factorization of residuated lattices

We use factorization of residuated lattices by compatible tolerances as the main tool in this paper. Regarding factorization of (complete) ordinary lattices we use results of Czédli [10] and Wille [18].

Recall that tolerance on a set X is a relation $\sim$ which is reflexive and symmetric. Each tolerance induces a covering of its underlying set, called the factor set. This set consists of all maximal blocks of the tolerance, i.e., the maximal subsets whose any two elements are in $\sim$. In the case of tolerance $\sim$ on the set X, the factor set is denoted $X/\sim$.

Compatible tolerance relation on a complete lattice $\mathbf{L}$ is a tolerance which preserves suprema and infima, i.e., a tolerance $\sim$ on $\mathbf{L}$ is compatible if from $a_j \sim b_j$ for any $j \in J$ follows $\bigvee_{j\in J} a_j \sim \bigvee_{j\in J} b_j$ and $\bigwedge_{j\in J} a_j \sim \bigwedge_{j\in J} b_j$.

For $a \in L$ we denote

$$a^{\sim} = \bigvee\{b \in L \mid a \sim b\}, \quad a_{\sim} = \bigwedge\{b \in L \mid a \sim b\}, \tag{5}$$

$$[a]_{\sim} = [a_{\sim}, (a_{\sim})^{\sim}], \quad [a]^{\sim} = [(a^{\sim})_{\sim}, a^{\sim}] \tag{6}$$

($[a_1, a_2]$ denotes the interval $\{b \in L \mid a_1 \leq b \leq a_2\}$).

Maximal blocks of $\sim$ are exactly sets $[a]_{\sim}$ and $[a]^{\sim}$, i.e., it holds $L/\sim = \{[a]_{\sim} \mid a \in L\} = \{[a]^{\sim} \mid a \in L\}$.

Ordering on the set $L/\sim$ is introduced using suprema of maximal blocks and can be equivalently introduced using infima. For blocks $B_1, B_2 \in L/\sim$ we set

$$B_1 \leq B_2 \quad \text{iff} \quad \bigvee B_1 \leq \bigvee B_2. \tag{7}$$

The set $L/\sim$ together with this ordering is a complete lattice, which is denoted by $\mathbf{L}/\sim$.

Now suppose that $\mathbf{L}$ is a residuated lattice. The following results can be found in [2], [3], where a more general approach is presented, namely sets of fixpoints of $\mathbf{L}$-closure operators are considered in place of residuated lattice $\mathbf{L}$.

For $e \in L$ we denote the e-cut of biresiduum in $\mathbf{L}$ by $\sim_e^{\mathbf{L}}$ or simply $\sim_e$. By definition of e-cuts of fuzzy relations, for any $a_1, a_2 \in L$, $a_1 \sim_e a_2$ if and only if $a_1 \leftrightarrow a_2 \geq e$. $\sim_e$ is a compatible tolerance on $\mathbf{L}$.

We introduce the following simplified notations: $a_e = a_{\sim_e}$, $a^e = a^{\sim_e}$, $[a]_e = [a]_{\sim_e}$, $[a]^e = [a]^{\sim_e}$. The factor lattice $\mathbf{L}/\sim_e$ will be denoted by $\mathbf{L}/e$.

It holds for any $a \in L$, $a_e = e \otimes a$, $a^e = e \to a$. As a consequence, we obtain the following equalities, which hold for any maximal block $B \in L/\sim_e$: $\bigvee B = e \to \bigwedge B$, $\bigwedge B = e \otimes \bigvee B$.

In [17] we introduced a structure of residuated lattice on the factor set L/e as follows. For $B_1, B_2 \in L/e$ we set

$$B_1 \otimes B_2 = \left[\bigvee B_1 \otimes \bigvee B_2\right]_e, \tag{8}$$

$$B_1 \to B_2 = \left[\bigvee B_1 \to \bigvee B_2\right]_e. \tag{9}$$

Now the set L/e together with elements $0,1 \in L/e$ and operations $\wedge, \vee$ given by the factor lattice structure and together with operations $\otimes, \rightarrow$ introduced in (8) and (9) is a complete residuated lattice, which is denoted by $\mathbf{L}/e$. More formally, $\mathbf{L}/e$ is equal to the tuple $\langle L/e, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$.

In the following lemma, we introduce some basic properties of factor residuated lattices which will be needed later. For more systematic approach, the reader can refer to [17].

Lemma 1. *For any $a_1, a_2 \in L$, $B_1, B_2 \in L/e$ it holds*

$$[a_1 \rightarrow a_2]_e \leq [a_1]_e \rightarrow [a_2]_e, \tag{10}$$

$$[a_1 \rightarrow (e \rightarrow a_2)]_e = [a_1]_e \rightarrow [e \rightarrow a_2]_e, \tag{11}$$

$$\bigvee (B_1 \rightarrow B_2) = \bigvee B_1 \rightarrow \bigvee B_2. \tag{12}$$

2.3 Fuzzy concept lattices with hedges

In this section, we recall some basic notions and notations and state some basic results on fuzzy concept lattices with hedges and their factorization. We refer the reader to [2], [6], [8] for details.

Let X, Y be nonempty sets, $I\colon X \times Y \rightarrow L$ an $\mathbf{L}$-relation between X and Y. The triple $\langle X, Y, I\rangle$ is called a formal $\mathbf{L}$-context, elements of X and Y are called objects and attributes, respectively. $\langle X, Y, I\rangle$ represents a data table which assigns to each $x \in X$ and $y \in Y$ a truth degree $I(x,y) \in L$ to which object x has the attribute y.

For a hedge $*_X$ on $\mathbf{L}$ and $\mathbf{L}$-set $A \in L^X$ of objects we define an $\mathbf{L}$-set $A^{\uparrow} \in L^Y$ of attributes by

$$A^{\uparrow}(y) = \bigwedge_{x \in X} \left(A(x)^{*_X} \rightarrow I(x,y)\right). \tag{13}$$

Similarly, for any hedge $*_Y$ and $\mathbf{L}$-set B of attributes we define an $\mathbf{L}$-set $B^{\downarrow}$ of objects by

$$B^{\downarrow}(x) = \bigwedge_{y \in Y} \left(B(y)^{*_Y} \rightarrow I(x,y)\right). \tag{14}$$

The following lemma summarizes basic properties of mappings $\uparrow$ and $\downarrow$ [4]:

Lemma 2. *Mappings $\uparrow$ and $\downarrow$ defined by* (13) *and* (14) *satisfy the following properties:*

1. $A^{*_X} \leq A^{\uparrow\downarrow}$ *and* $B^{*_Y} \leq B^{\downarrow\uparrow}$*;*
2. $A_1 \leq A_2$ *implies* $A_2^{\uparrow} \leq A_1^{\uparrow}$, *and* $B_1 \leq B_2$ *implies* $B_2^{\downarrow} \leq B_1^{\downarrow}$ *(antitony);*
3. $A^{\uparrow} = A^{*_X\uparrow}$ *and* $B^{\downarrow} = B^{*_Y\downarrow}$*;*
4. $A^{\uparrow *_Y} \leq A^{\uparrow\downarrow\uparrow} \leq A^{*_X\uparrow}$ *and* $B^{\downarrow *_X} \leq B^{\downarrow\uparrow\downarrow} \leq B^{*_Y\downarrow}$*;*
5. $A^{\uparrow\downarrow} = A^{\uparrow\downarrow\uparrow\downarrow}$ *and* $B^{\downarrow\uparrow} = B^{\downarrow\uparrow\downarrow\uparrow}$.

Next we set

$$\mathscr{B}(X^{*_X}, Y^{*_Y}, I) = \{\langle A, B\rangle \in L^X \times L^Y \mid A^{\uparrow} = B, B^{\downarrow} = A\}. \tag{15}$$

We define a partial ordering on $\mathscr{B}(X, Y, I)$ by

$$\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle \quad \text{iff} \quad A_1 \leq A_2 \tag{16}$$

(or, equivalently, $B_2 \leq B_1$). $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$ with this ordering is a complete lattice, called an **L**-concept lattice induced by $\langle X, Y, I \rangle$ and hedges $*_X$, $*_Y$.

Elements $\langle A, B \rangle$ of $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$ are called formal concepts, for each formal concept $\langle A, B \rangle$, A is called its extent, B intent. Formal concepts are interpreted as concepts/clusters hidden in the data table. Namely, the conditions $A^{\uparrow} = B$ and $B^{\downarrow} = A$ say that B is the collection of all attributes shared by all objects (for which it is very true that they are) from A, and A is the collection of all objects sharing all attributes (for which it is very true that they are) from B.

The main idea of adding hedges to fuzzy concept lattices is that using hedges, one can affect the size of concept lattices. Namely, if we choose both $*_X$, $*_Y$ to be identities, we obtain an ordinary fuzzy concept lattice. Other choices lead to smaller concept lattices. For example, if both $*_X$, $*_Y$ are globalizations then the generated concept lattice consists of so called crisply generated formal concepts [7]. If $*_X$ and $*_Y$ are globalization and identity (respectively) then $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$ is isomorphic to so-called one-sided concept lattice [15].

Now we recall the parametrized concept lattice factorization method, as introduced in [1], and then mention its generalization to fuzzy concept lattices with hedges.

As we mentioned in Introduction, factorization represents another attempt to reduce the size of fuzzy concept lattice. In this method, user choses a degree $e \in L$ to which he/she considers two different concepts to be similar. Factorizing-out similar concepts by a tolerance relation induced by e a smaller lattice is obtained. This lattice do not preserve information on differences between similar concepts. Reader can refer [6], [8] for details on factorization of concept lattices and its generalization to concept lattices with hedges.

We introduce a similarity relation $\approx$ on the set $\mathscr{B}(X, Y, I)$ of all formal concepts of $\langle X, Y, I \rangle$ by

$$\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle = A_1 \approx^X A_2 \tag{17}$$

(see (4)).

$\langle A_1, B_1 \rangle \approx \langle A_2, B_2 \rangle$ is called the degree of similarity of formal concepts $\langle A_1, B_1 \rangle$ and $\langle A_2, B_2 \rangle$. $\approx$ is known to be a fuzzy equivalence on $\mathscr{B}(X, Y, I)$.

Since $\approx$ is a fuzzy equivalence on $\mathscr{B}(X, Y, I)$ then, for any user-chosen threshold $e \in \mathbf{L}$, the e-cut ${}^e\!\approx$ is a (crisp) tolerance relation ("being similar to degree at least e") on $\mathscr{B}(X, Y, I)$. This tolerance is compatible with the lattice structure on $\mathscr{B}(X, Y, I)$.

Maximal blocks of ${}^e\!\approx$ are exactly intervals $[\langle A, B \rangle]_{^e\approx}$ (or, equivalently, intervals $[\langle A, B \rangle]^{^e\approx}$, see (6)), and the factor set $\mathscr{B}(X, Y, I)/{}^e\!\approx$ together with the ordering given by (7) is a complete lattice.

This result can also be generalized to fuzzy concept lattices with hedges. First we show some properties of the fuzzy equivalence $\approx^X$ (resp. $\approx^Y$) on L^X (resp. L^Y) with connection to functions $\uparrow$ and $\downarrow$ [6]:

Lemma 3. *For $A_1, A_2 \in L^X$ and $B_1, B_2 \in L^Y$ we have $(A_1 \approx^X A_2)^{*_X} \leq A_1^{\uparrow} \approx^Y A_2^{\uparrow}$ and $(B_1 \approx^Y B_2)^{*_Y} \leq B_1^{\downarrow} \approx^X B_2^{\downarrow}$.*

For a concept lattice $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$, similarity of concepts is defined as above, as well as its e-cut, used for factorization. The factor set $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)/{}^e\!\approx$ together with the ordering given by (7) is again a complete lattice. The structure of maximal blocks of ${}^{e}\!\approx$ on $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$ is given by the following lemma.

Lemma 4. *For $\langle A,B\rangle \in \mathscr{B}(X,Y,I)$ we have*

1. $\langle A,B\rangle^{e\approx} = \langle (e \to A)^{\uparrow\downarrow}, (e \otimes B)^{\downarrow\uparrow}\rangle$,
2. $\langle A,B\rangle_{e\approx} = \langle (e \otimes A)^{\uparrow\downarrow}, (e \to B)^{\downarrow\uparrow}\rangle$,
3. $\langle A,B\rangle^{e\approx} = ((\langle A,B\rangle^{e\approx})_{e\approx})^{e\approx}$,
4. $\langle A,B\rangle_{e\approx} = ((\langle A,B\rangle_{e\approx})^{e\approx})_{e\approx}$.

3 Results

3.1 Factorization of residuated lattices with hedges

The first main result of this paper concerns introducing a hedge on the factor residuated lattice $\mathbf{L}/e$ induced by a hedge on the original residuated lattice $\mathbf{L}$.

Suppose that $*$ is a hedge on residuated lattice $\mathbf{L}$ and $e \in L$ is its fixpoint, i.e., $e^* = e$. We define a new unary operation $*^e$ (or, simply, $*$ if e and underlying residuated lattice are obvious) on $\mathbf{L}/e$ by setting for any $B \in L/e$,

$$B^{*^e} = \left[\left(\bigvee B\right)^*\right]_e. \tag{18}$$

We have the following result for the new operation $*^e$:

Theorem 1. *If $e \in L$ is a fixpoint of the hedge $*$ then the operation $*^e$ on $\mathbf{L}/e$ is a hedge.*

Proof. Let $1 \in L$ and $\mathbf{1} \in L/e$ be unite elements. We have $\mathbf{1} = [1]_e$ and

$$\mathbf{1}^{*^e} = ([1]_e)^{*^e} = [1^*]_e = \mathbf{1},$$

which proves condition (i) for hedges.

Now let $B \in L/e$. Then

$$B^{*^e} = \left[\left(\bigvee B\right)^*\right]_e \leq \left[\bigvee B\right]_e = B,$$

which proves condition (ii).

To prove condition (iii) we use Lemma 1 and obtain for $B_1, B_2 \in L/e$,

$$\begin{aligned}(B_1 \to B_2)^{*^e} &= \left[\left(\bigvee (B_1 \to B_2)\right)^*\right]_e = \left[\left(\bigvee B_1 \to \bigvee B_2\right)^*\right]_e \leq \\ &\leq \left[\left(\bigvee B_1\right)^* \to \left(\bigvee B_2\right)^*\right]_e \leq \left[\left(\bigvee B_1\right)^*\right]_e \to \left[\left(\bigvee B_2\right)^*\right]_e = \\ &= B_1^{*^e} \to B_2^{*^e}.\end{aligned}$$

Let $B \in L/e$. To prove the equality $B^{*^e} = B^{*^e *^e}$ we show that infima of both sides are equal. Denote $\bigvee B = a$. We have $\bigwedge B^{*^e} = e \otimes a^*$ and $\bigwedge B^{*^e *^e} = e \otimes (e \to e \otimes a^*)^*$. Now, from condition (iii) for hedges and from the fact that $e \otimes a^*$ is a fixpoint of $*$ (both e and a^* are fixpoints) we obtain

$$\bigwedge B^{*^e *^e} \leq e \otimes (e^* \to (e \otimes a^*)^*) = e \otimes (e \to e \otimes a^*) = \bigwedge B^{*^e}.$$

The opposite inequality $\bigwedge B^{*^e} \leq \bigwedge B^{*^e *^e}$ follows from $(e \to e \otimes a^*)^* \leq e \to e \otimes a^*$ by multiplying both sides by e. This proves the remaining condition (iv) for hedges.

3.2 Factorization of fuzzy concept lattices with hedges

In this section, we present our second main result: the factorized **L**-concept lattice $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)/^e\approx$ is isomorphic to an $\mathbf{L}/e$-concept lattice, constructed from a formal $\mathbf{L}/e$-context, which is easily computable from the original formal **L**-context $\langle X, Y, I\rangle$.

For any **L**-set $A \in \mathbf{L}^X$ we shall use the symbols A^e, A_e, $[A]^e$, $[A]_e$ as before, where e is identified with the constant mapping $x \mapsto e$. We have $A^e, A_e \in \mathbf{L}^X$, $[A]^e, [A]_e \in (\mathbf{L}^X)/e$.

In what follows, we shall not distinguish between sets L^X/e and $(L/e)^X$ and their elements. For example, we can consider $[A]_e$ as an element of $(L/e)^X$, having $[A(x)]_e = [A]_e(x) \in L/e$, for any $x \in X$ (see [17] for details).

For a formal context $\langle X, Y, I\rangle$, the **L**-relation I is a mapping $I\colon X \times Y \to L$. Using results from [17], we define an $\mathbf{L}/e$-relation $[I]^e\colon X \times Y \to L/e$ by

$$[I]^e(x,y) = [I(x,y)]^e \tag{19}$$

(like before, we do not distinguish between elements of $(L/e)^{X\times Y}$ and $L^{X\times Y}/e$).

Let $\langle X, Y, I\rangle$ be a formal context, $*_X$, $*_Y$ hedges, $e \in L$ a fixed threshold. We consider a new formal $\mathbf{L}/e$-context $\langle X, Y, [I]^e\rangle$. Using results of previous section, we introduce two thresholds $*^e_X$, $*^e_Y$ on the factor residuated lattice $\mathbf{L}/e$ such that e is their common fixpoint. Then we construct the concept lattice $\mathscr{B}(X^{*^e_X}, Y^{*^e_Y}, [I]^e)$.

When the underlying residuated lattice and e are obvious, we also denote the thresholds $*^e_X$, $*^e_Y$ simply by $*_X$, $*_Y$. Since there will be no possibility of confusion, we also denote the formal-context-defining operators with respect to the formal context $\langle X, Y, [I]^e\rangle$ and hedges $*^e_X$, $*^e_Y$ again by $^\uparrow$, and $^\downarrow$.

Lemma 5. *For any $\bar{A} \in L^X/e$ with $A = \bigvee \bar{A}$ it holds $\bar{A}^\uparrow = [A^\uparrow]^e$. For any $\bar{B} \in L^Y/e$ with $B = \bigvee \bar{B}$ it holds $\bar{B}^\downarrow = [B^\downarrow]^e$.*

Proof. From basic properties of blocks of compatible tolerances in residuated lattices and from (11) we obtain

$$\begin{aligned}
\bar{A}^\uparrow(y) &= \bigwedge_{x\in X} \bar{A}^{*^e_X}(x) \to [I]^e(x,y) = \\
&= \bigwedge_{x\in X} \bar{A}^{*^e_X}(x) \to [e \to I(x,y)]_e = \\
&= \bigwedge_{x\in X} [A^{*_X}(x)]_e \to [e \to I(x,y)]_e = \\
&= \bigwedge_{x\in X} [A^{*_X}(x) \to (e \to I(x,y))]_e = \\
&= \bigwedge_{x\in X} [e \to (A^{*_X}(x) \to I(x,y))]_e = \\
&= \bigwedge_{x\in X} [A^{*_X}(x) \to I(x,y)]^e = \\
&= \left[\bigwedge_{x\in X} (A^{*_X}(x) \to I(x,y))\right]^e = \\
&= [A^\uparrow(y)]^e.
\end{aligned}$$

The second statement follows by duality.

Lemma 6. *For any $\bar{A} \in L^X/e$, if $A \in \bar{A}$ then $A^{\uparrow} \in \bar{A}^{\uparrow}$. For any $\bar{B} \in L^Y/e$, if $B \in \bar{B}$ then $B^{\downarrow} \in \bar{B}^{\downarrow}$.*

Proof. This is a simple consequence of Lemma 5. If $A \in \bar{A}$ then $A \leq \bigvee \bar{A}$ and $A \approx^X \bigvee \bar{A} \geq e$. Hence $A^{\uparrow} \geq (\bigvee \bar{A})^{\uparrow}$ (Lemma 2, part 2) and $A^{\uparrow} \approx^Y (\bigvee \bar{A})^{\uparrow} \geq e^{*_X} = e$ (Lemma 3). Thus, $A^{\uparrow} \in [(\bigvee \bar{A})^{\uparrow}]^e = \bar{A}^{\uparrow}$ (Lemma 5). The second statement can be proved similarly.

Lemma 7. *For $\langle \bar{A}, \bar{B} \rangle \in \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e)$, $(\bigvee \bar{B})^{\downarrow}$ is the least fixpoint of $\uparrow\downarrow$ in $\bar{A}$.*

Proof. Denote $B_0 = \bigvee \bar{B}$, $A_0 = B_0^{\downarrow}$. First we show that A_0 is a fixpoint of $\uparrow\downarrow$. The element $A_0^{\uparrow}$ is a fixpoint of $\downarrow\uparrow$ (Lemma 2, part 5). We have $B_0^{*_Y} \leq A_0^{\uparrow}$ (Lemma 2, part 1) and $A_0^{\uparrow} \leq B_0$ (Lemma 6, applied twice). Hence for fixpoint $A_0^{\uparrow\downarrow}$ of $\uparrow\downarrow$ we obtain (using Lemma 2, part 2), $B_0^{\downarrow} \leq A_0^{\uparrow\downarrow} \leq B_0^{*_Y\downarrow}$. But from Lemma 2, part 3, we have $B_0^{\downarrow} = B_0^{*_Y\downarrow}$, which shows that A_0 is a fixpoint of $\uparrow\downarrow$.

Now from antitony of $\uparrow$ and $\downarrow$ (Lemma 2, part 2) we have for any fixpoint $A \in \bar{A}$: $A \geq \bigwedge \bar{A}$, $A^{\uparrow} \leq (\bigwedge \bar{A})^{\uparrow} \leq B_0$ (Lemma 6), which leads to $A_0 \leq A^{\uparrow\downarrow} = A$.

Lemma 8. *For every $\langle \bar{A}, \bar{B} \rangle \in \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e)$, the set $F(\langle \bar{A}, \bar{B} \rangle)$ of all $\langle A, B \rangle$ from $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$ such that $A \in \bar{A}$, is a maximal block of ${}^e\!\approx$ (i.e., $F(\langle \bar{A}, \bar{B} \rangle)$ belongs to $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)/{}^e\!\approx$).*

Proof. According to Lemma 7, $A_0 = (\bigvee \bar{B})^{\downarrow}$ is the least fixpoint of $\uparrow\downarrow$ in $\bar{A}$. From Lemma 5 we have $e \to A_0 = \bigvee \bar{A}$ and $(e \to A_0)^{\uparrow\downarrow} = A_1$, where A_1 is the greatest fixpoint of $\uparrow\downarrow$ in $\bar{A}$. According to Lemma 6, $A_1 \in \bar{A}$.

It remains to be shown (Lemma 4) that $A_0 = (e \otimes A_1)^{\uparrow\downarrow} \in \bar{A}$. We have $(\bigvee \bar{A})^{*_X} \leq A_1 \leq \bigvee \bar{A}$ (Lemma 2, part 1) and from Lemma 2, parts 2, 3, the intent $B_1 = A_1^{\uparrow}$ is equal to $(\bigvee \bar{A})^{\uparrow}$. Hence, $\bigvee \bar{B} = e \to B_1$ (Lemma 5) and $(e \to B_1)^{\downarrow\uparrow}$ is the greatest intent of $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$ from $\bar{B}$. According to Lemma 4, the corresponding extent is equal to A_0. Applying Lemma 6 now completes the proof.

Lemma 9. *For any maximal block $K = [\langle A_0, B_0 \rangle, \langle A_1, B_1 \rangle] \in \mathscr{B}(X^{*_X}, Y^{*_Y}, I)/{}^e\!\approx$ there is exactly one formal concept $G(K) = \langle \bar{A}, \bar{B} \rangle \in \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e)$ such that $\bigwedge \bar{A} \leq A_0$, $A_1 \leq \bigvee \bar{A}$. It holds $\bar{A} = [A_0]^e$.*

Proof. Since $A_0 \;{}^e\!\approx A_1$ then there exists a maximal block $A' \in L^X/e$ such that $A_0 \in A'$, $A_1 \in A'$. From Lemma 6 we have $A_0 \in A'^{\uparrow\downarrow}$, $A_1 \in A'^{\uparrow\downarrow}$. This gives existence of at least one $\langle \bar{A}, \bar{B} \rangle$ with desired properties.

Now suppose that $\langle \bar{A}, \bar{B} \rangle \in \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e)$ is such that $\bigwedge \bar{A} \leq A_0$, $A_1 \leq \bigvee \bar{A}$. The element $(\bigvee \bar{B})^{\downarrow}$ is the least fixpoint of $\uparrow\downarrow$ in $\bar{A}$ (Lemma 7). Hence, $(\bigvee \bar{B})^{\downarrow} = A_0$ (K is a maximal block). From Lemma 5 we have $\bar{A} = [A_0]^e$ which proves the uniqueness of $\bar{A}$ as well as the desired equality.

Lemmas 8 and 9 give us mapping $F\colon \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e) \to \mathscr{B}(X^{*_X}, Y^{*_Y}, I)/{}^e\!\approx$ and mapping $G\colon \mathscr{B}(X^{*_X}, Y^{*_Y}, I)/{}^e\!\approx \to \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e)$ which are obviously mutually inverse. Using mapping F, we state our main result:

Theorem 2. *Mapping F is an isomorphism of lattices.*

Proof. It remains to be shown that F and G are morphisms of ordered sets. For two elements $\langle \bar{A}, \bar{B} \rangle, \langle \bar{C}, \bar{D} \rangle \in \mathscr{B}(X^{*_X}, Y^{*_Y}, [I]^e)$, denote $F(\langle \bar{A}, \bar{B} \rangle) = [\langle A_0, B_0 \rangle, \langle A_1, B_1 \rangle]$ and, similarly, $F(\langle \bar{C}, \bar{D} \rangle) = [\langle C_0, D_0 \rangle, \langle C_1, D_1 \rangle]$ (intervals taken in $\mathscr{B}(X^{*_X}, Y^{*_Y}, I)$).

If $\langle \bar{A}, \bar{B} \rangle \leq \langle \bar{C}, \bar{D} \rangle$ then $\bigvee \bar{A} \leq \bigvee \bar{C}$, from which and from Lemma 7 it follows $B_1 = (\bigvee \bar{A})^{\uparrow} \geq (\bigvee \bar{C})^{\uparrow} = D_1$. This means $[\langle A_0, B_0 \rangle, \langle A_1, B_1 \rangle] \leq [\langle C_0, D_0 \rangle, \langle C_1, D_1 \rangle]$.

To prove the opposite we start with $A_0 \leq C_0$. This and Lemma 5 give $\bigvee \bar{A} = e \to A_0 \leq e \to C_0 = \bigvee \bar{C}$, which finishes the proof.

4 Conclusion

The two main results of this paper can be interpreted as follows. If we are trying to reduce the complexity of some concept lattice with hedges by factorization, then we are, in fact, constructing another concept lattice with hedges, which is built over a data table with values in some factorized residuated lattice. Thus, the problem of factorization of concept lattice by similarity is replaced with the problem of factorization of the used set of truth degrees (residuated lattice) which indicate the similarity levels.

This paper extends our previous results from [17], where we considered residuated lattices and fuzzy concept lattices without hedges.

There is even more general approach ("Generalized concept lattice", [16]), which contains the notion of fuzzy concept lattice with hedges as a special case [14]. There arises a question whether the method of factorization of concept lattices can be generalized to this case. This question is open; the main obstacle seems to be that in this general framework there is no known natural notion of similarity of concepts.

References

1. Belohlavek, R.: Similarity relations in concept lattices. J. Logic Comput., 10 (6), 823–845 (2000)
2. Belohlavek, R.: Fuzzy Relational Systems, Foundations and Principles. Kluwer, New York (2002)
3. Belohlavek, R. et al.: Fast factorization by similarity in formal concept analysis of data with fuzzy attributes. J. Comput. Syst. Sci., 73 (6), 1012–1022 (2007)
4. Belohlavek, R., Funiokova, T., Vychodil, V.: Galois connections with hedges. In: Yingming Liu, Guoqing Chen, Mingsheng Ying (Eds.): Fuzzy Logic, Soft Computing & Computational Intelligence: Eleventh International Fuzzy Systems Association World Congress (Vol. II), 2005, pp. 1250–1255. Tsinghua University Press and Springer, ISBN 7–302–11377–7.
5. Belohlavek, R., Outrata, J. and Vychodil, V.: Thresholds and shifted attributes in formal concept analysis of data with fuzzy attributes. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) Lecture Notes in Artificial Intelligence, 4068, pp. 117–130. Springer, Berlin/Heidelberg (2006)
6. Belohlavek, R., Outrata, J., Vychodil, V.: Fast Factorization by Similarity of Fuzzy Concept Lattices with Hedges. Int. J. Found. Comput. Sci. 19(2): 255-269 (2008)
7. Belohlavek, R., Sklenář, V., Zacpal, J.: Crisply generated fuzzy concepts. In: B. Ganter and R. Godin (Eds.): ICFCA 2005, Lecture Notes in Computer Science 3403, pp. 268–283, Springer-Verlag, Berlin/Heidelberg, 2005.

8. Belohlavek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: FUZZ-IEEE 2005, The IEEE International Conference on Fuzzy Systems, May 22–25, 2005, Reno (Nevada, USA), pp. 663–668.
9. Carpineto, C., Romano, G.: Concept Data Analysis. Theory and Applications. J. Wiley (2004).
10. Czédli, G.: Factor lattices by tolerances. Acta Sci. Math., 44, 35–42 (1982)
11. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Founda- tions. Springer-Verlag, Berlin (1999).
12. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
13. Hájek, P.: On very true. Fuzzy sets and systems 124(2001), 329–333.
14. Krajči, S.: Every concept lattice with hedges is isomorphic to some generalized concept lattice. In: R. Belohlavek, V. Snasel (Eds.), Proceedings of CLA 2005: The 3rd international conference on Concept Lattices and Their Applications, Olomouc, Czech Republic, September 2005, pp. 1–9
15. Krajči, S.: Cluster based efficient generation of fuzzy concepts. Neural Network World 5(2003), 521–530.
16. Krajči, S.: The basic theorem on generalized concept lattice. In: R. Belohlavek, V. Snasel (eds.), CLA 2004, Ostrava, Proceedings of the 2nd international workshop, pp. 25–33
17. Krupka, M.: Factorization of residuated lattices with application to concept lattices. In: R.Trappl (ed.) Cybernetics and Systems 2008, pp. 21–25. Austrian Society for Cybernetic Studies, Vienna (2008)
18. Wille, R.: Complete tolerance relations of concept lattices. In: G. Eigenthaler et al. Contributions to General Algebra, Vol. 3, pp. 397–415, Hölder-Pichler-Tempsky, Wien (1985)

Author Index

Editors: Radim Belohlavek, Sergei O. Kuznetsov

Publisher & Print: Palacký University, Olomouc
Křížkovského 8, 771 47 Olomouc
Czech Republic

Title: CLA 2008, Proceedings of the Sixth International Conference on Concept Lattices and Their Applications

Place, year, edition: Olomouc, 2008, 1st

Page count: xii+243

Impression: 70

Not for sale

ISBN 978–80–244–2111–7