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

LNAI 7278

Formal Concept Analysis

10th International Conference, ICFCA 2012
Leuven, Belgium, May 2012
Proceedings

Springer

Lecture Notes in Artificial Intelligence 7278

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

LNAI Founding Series Editor

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

Formal Concept Analysis

10th International Conference, ICFCA 2012
Leuven, Belgium, May 7-10, 2012
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Florent Domenach
University of Nicosia, Computer Science Department
46 Makedonitissas Ave., 1700 Nicosia, Cyprus
E-mail: domenach.f@unic.ac.cy

Dmitry I. Ignatov
National Research University Higher School of Economics
Pokrovsky Boulevard, 109028 Moscow, Russia
E-mail: dignatov@hse.ru

Jonas Poelmans
Katholieke Universiteit Leuven, Faculty of Business and Economics
Naamsestraat 69, 3000 Leuven, Belgium
E-mail: jonas.poelmans@econ.kuleuven.be

ISSN 0302-9743
e-ISSN 1611-3349
ISBN 978-3-642-29891-2
e-ISBN 978-3-642-29892-9
DOI 10.1007/978-3-642-29892-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012936071

CR Subject Classification (1998): I.2, G.2.1-2, F.4.1-2, D.2.4, H.3

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers presented at the 10th International Conference on Formal Concept Analysis (ICFCA 2012) held during May 7–10, at the Katholieke Universiteit Leuven, Belgium.

There were 68 submissions by authors from 27 countries. Each submission was reviewed by at least three Program Committee members, and 20 regular papers (29%) were accepted. The program also included six invited talks on topical issues: Recent Advances in Machine Learning and Data Mining, Mining Terrorist Networks and Revealing Criminals, Concept-Based Process Mining, and Scalability Issues in FCA and Rough Sets. The corresponding abstracts are gathered in the first section of this volume. Another 14 papers were assessed as valuable for discussion at the conference and were therefore collected in the supplementary proceedings.

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

The conference aims to bring together researchers and practitioners working on theoretical or applied aspects of formal concept analysis within major related areas such as mathematics, computer and information sciences and their diverse applications to fields such as software engineering, linguistics, life and social sciences.

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

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

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

March 2012

Florent Domenach
Dmitry I. Ignatov
Jonas Poelmans

Organization

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

Conference Chair

Jonas Poelmans	Katholieke Universiteit Leuven, Belgium

Conference Organizing Committee

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

Program Chairs

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

Editorial Board

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

Program Committee

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

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

External Reviewers

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

Sponsoring Institutions

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

Table of Contents

Invited Talks

Dark Web: Exploring and Mining the Dark Side of the Web 1
Hsinchun Chen

Declarative Modeling for Machine Learning and Data Mining 2
Luc De Raedt

Can Concepts Reveal Criminals? 3
Paul Elzinga

Cartification: From Similarities to Itemset Frequencies 4
Bart Goethals

Processes Are Concepts, Aren't They? 5
Ir. Edward Peters

Rough Sets and FCA – Scalability Challenges 6
Dominik Ślęzak

Regular Papers

Approximating Concept Stability 7
Mikhail A. Babin and Sergei O. Kuznetsov

Logical Analysis of Concept Lattices by Factorization 16
Eduard Bartl and Michal Krupka

Basic Level of Concepts in Formal Concept Analysis 28
Radim Belohlavek and Martin Trnecka

A Peep through the Looking Glass: Articulation Points in Lattices 45
Anne Berry and Alain Sigayret

Practical Use of Formal Concept Analysis in Service-Oriented Computing 61
Stéphanie Chollet, Vincent Lestideau, Yoann Maurel, Etienne Gandrille, Philippe Lalanda, and Olivier Raynaud

Publication Analysis of the Formal Concept Analysis Community 77
Stephan Doerfel, Robert Jäschke, and Gerd Stumme

Understanding the Semantic Structure of Human fMRI Brain Recordings with Formal Concept Analysis 96
Dominik Endres, Ruth Adam, Martin A. Giese, and Uta Noppeney

Cubes of Concepts: Multi-dimensional Exploration of Multi-valued Contexts 112
Sébastien Ferré, Pierre Allard, and Olivier Ridoux

Ordinal Factor Analysis 128
Bernhard Ganter and Cynthia Vera Glodeanu

A Macroscopic Approach to FCA and Its Various Fuzzifications 140
Tim B. Kaiser and Stefan E. Schmidt

A Connection between Clone Theory and FCA Provided by Duality Theory 148
Sebastian Kerkhoff

Formal Concept Discovery in Semantic Web Data 164
Markus Kirchberg, Erwin Leonardi, Yu Shyang Tan, Sebastian Link, Ryan K.L. Ko, and Bu Sung Lee

Concept Lattices of Incomplete Data 180
Michal Krupka and Jan Lastovicka

Formal Concept Analysis as a Framework for Business Intelligence Technologies 195
Juraj Macko

Good Classification Tests as Formal Concepts 211
Xenia A. Naidenova

Modeling Preferences over Attribute Sets in Formal Concept Analysis 227
Sergei Obiedkov

Finding Top-N Colossal Patterns Based on Clique Search with Dynamic Update of Graph 244
Yoshiaki Okubo and Makoto Haraguchi

Quantitative Concept Analysis 260
Dusko Pavlovic

Some Notes on Managing Closure Operators 278
Sebastian Rudolph

Distributed Formal Concept Analysis Algorithms Based on an Iterative MapReduce Framework 292
Biao Xu, Ruairí de Fréin, Eric Robson, and Mícheál Ó Foghlú

Author Index 309

Dark Web: Exploring and Mining the Dark Side of the Web

Hsinchun Chen

Artificial Intelligence Lab, MIS Dept.
P.O. Box 210108, Tucson, AZ 85721-0108, USA
hchen@eller.arizona.edu

Abstract. This talk will review the emerging research in Terrorism Informatics based on a web mining perspective. Recent progress in the internationally renowned Dark Web project will be reviewed, including: deep/dark web spidering (web sites, forums, Youtube, virtual worlds), web metrics analysis, dark network analysis, web-based authorship analysis, and sentiment and affect analysis for terrorism tracking. In collaboration with selected international terrorism research centers and intelligence agencies, the Dark Web project has generated one of the largest databases in the world about extremist/terrorist-generated Internet contents (web sites, forums, blogs, and multimedia documents). Dark Web research has received significant international press coverage, including: Associated Press, USA Today, The Economist, NSF Press, Washington Post, Fox News, BBC, PBS, Business Week, Discover magazine, WIRED magazine, Government Computing Week, Second German TV (ZDF), Toronto Star, and Arizona Daily Star, among others. For more Dark Web project information, please see: http://ai.eller.arizona.edu/research/terror/.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, p. 1, 2012.

Declarative Modeling for Machine Learning and Data Mining

Luc De Raedt

Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200A, 3001 Heverlee, Belgium
luc.deraedt@cs.kuleuven.be

Abstract. Despite the popularity of machine learning and data mining today, it remains challenging to develop applications and software that incorporates machine learning or data mining techniques. This is because machine learning and data mining have focussed on developing high-performance algorithms for solving particular tasks rather than on developing general principles and techniques.

I propose to alleviate these problems by applying the constraint programming methodology to machine learning and data mining and to specify machine learning and data mining problems as constraint satisfaction and optimization problems. What is essential is that the user be provided with a way to declaratively specify what the machine learning or data mining problem is rather than having to outline how that solution needs to be computed. This corresponds to a model + solver-based approach to machine learning and data mining, in which the user specifies the problem in a high level modeling language and the system automatically transforms such models into a format that can be used by a solver to efficiently generate a solution. This should be much easier for the user than having to implement or adapt an algorithm that computes a particular solution to a specific problem.

I shall illustrate this using our results on constraint programming for itemset mining [1] and probabilistic programming. Some further ideas along these lines are contained in [2].

References

1. Guns, T., Nijssen, S., De Raedt, L.: Itemset mining: A constraint programming perspective. Artificial Intelligence 175(12-13), 1951–1983 (2011)
2. De Raedt, L., Nijssen, S.: Towards Programming Languages for Machine Learning and Data Mining (Extended Abstract). In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Raś, Z.W. (eds.) ISMIS 2011. LNCS, vol. 6804, pp. 25–32. Springer, Heidelberg (2011)

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, p. 2, 2012.

Can Concepts Reveal Criminals?

Paul Elzinga

Amsterdam-Amstelland Police, James Wattstraat 84,
1000 CG Amsterdam, The Netherlands
paul.elzinga@amsterdam.politie.nl

Abstract. In 2005 the Amsterdam-Amstelland police introduced Intelligence-led Policing as a management paradigm. The goal of ILP is to optimally use the information which becomes available after police patrols, motor vehicle inspections, video camera recordings, etc. to prevent crimes where possible and optimally allocate available resources. This policy has resulted in an increasing number of textual reports, video materials, etc. every year. Until now, this vast amount of information was not easily accessible because good analysis methods were missing and as a consequence hardly used by the criminal intelligence departments. In the first part of this talk I will give a short overview of traditional statistical methods such as hot spot analysis which have been used to make this information accessible and steer police actions. In the second part of this talk I will present using some real life cases how FCA was used to identify criminals involved in human trafficking, terrorism, robberies, etc. In the third part of this talk I would like to evoke a lively discussion on the potential of FCA related algorithms and methods for analyzing textual reports, internet data such as twitter feeds, browsing behavior of visitors of radical Islamic websites, etc.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, p. 3, 2012.

Cartification: From Similarities to Itemset Frequencies

Bart Goethals

University of Antwerp, Dept. of Mathematics and Computer Science
Middelheimlaan 1, B-2020 Antwerpen, Belgium
bart.goethals@ua.ac.be

Abstract. Suppose we are given a multi-dimensional dataset. For every point in the dataset, we create a transaction, or cart, in which we store the k-nearest neighbors of that point for one of the given dimensions. The resulting collection of carts can then be used to mine frequent itemsets; that is, sets of points that are frequently seen together in some dimensions. Experimentation shows that finding clusters, outliers, cluster centers, or even subspace clustering becomes easy on the cartified dataset using state-of-the-art techniques in mining interesting itemsets.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, p. 4, 2012.

Processes Are Concepts, Aren't They?

Ir. Edward Peters[1,2]

[1] K.U.Leuven, Faculty of Business and Economics,
Naamsestraat 69, 3000 Leuven, Belgium
[2] OpenConnect Systems, 2711 LBJ Freeway Suite 700,
Dallas, TX 75234, United States of America
epeters@oc.com

Abstract. Discovery is an information / technical approach to the important managerial problem of decision making under not only uncertainty, but also actually, "unknown unknowns". Formal Concept Analysis (FCA) is an elegant mathematically grounded theory that complements Data Discovery particularly well, especially for data with no predefined structure. It allows for discovering emergent, a-priori unknown concepts in these data. Process Discovery deals with emergent behaviour in business workflows. In its current state-of-the-art, Process Discovery combines machine-learning techniques utilizing Hidden Markov Model (HMM) representations of Processes. In one of our research lines, we investigate how FCA can improve and complement Process Discovery. Although the inclusion of temporal data and events in FCA allows for the creation of "early warning" and trend-models, HMM's are needed for a deep understanding of the processes and their intrinsic complexities. However, FCA can assist significantly in the post-processing and understanding of HMM's, in particular in the presence of process variations and exceptions. But FCA allows also for the detection of recurrent, coherent Process steps, which can be considered "service steps" in business processes. Ultimately, an appropriate mathematical representation of HMM's should allow for the application of algebraic extensions of FCA for discovering Processes and their variations as mathematical concepts as well. Some initial work on Process patterns gives inspiring research directions. Real-life case materials from Healthcare Administration, Customer Contact-Center's and Financial Services illustrate this keynote lecture.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, p. 5, 2012.

Rough Sets and FCA – Scalability Challenges*

Dominik Ślęzak

Institute of Mathematics, University of Warsaw
ul. Banacha 2, 02-097 Warsaw, Poland
Infobright Inc.
ul. Krzywickiego 34, lok. 219, 02-078 Warsaw, Poland

Rough Sets (RS) [1,2,3] and Formal Concept Analysis (FCA) [4,5] provide foundations for a number of methods useful in data mining and knowledge discovery at different stages of data preprocessing, classification and representation. RS and FCA are often applied together with other techniques in order to cope with real-world challenges. It is therefore important to investigate various ways of extending RS/FCA notions and algorithms in order to facilitate dealing with truly large and complex data. This talk attempts to categorize some ideas of how to scale RS and FCA methods with respect to a number of objects and attributes, as well as types and cardinalities of attribute values. We discuss a usage of analytical database engines [6] and randomized heuristics [7] to compute approximate, yet meaningful results. We also discuss differences and similarities in algorithmic bottlenecks related to RS and FCA, illustrating that these approaches should be regarded as complementary rather than competing methodologies. As a case study, we consider the tasks of data analysis and knowledge representation arising within a research project aiming at enhancing semantic search of diverse types of content in a large repository of scientific articles [8].

References

1. Pawlak, Z., Skowron, A.: Rudiments of Rough Sets. Inf. Sci. 177(1), 3–27 (2007)
2. Pawlak, Z., Skowron, A.: Rough Sets and Boolean Reasoning. Inf. Sci. 177(1), 41–73 (2007)
3. Pawlak, Z., Skowron, A.: Rough Sets: Some Extensions. Inf. Sci. 177(1), 28–40 (2007)
4. Poelmans, J., Kuznetsov, S., Ignatov, D., Dedene, G., Elzinga, P., Viaene, S.: Formal Concept Analysis in Knowledge Processing: A Survey on Models and Techniques. Inf. Sci. (2012)
5. Poelmans, J., Ignatov, D., Kuznetsov, S., Dedene, G., Elzinga, P., Viaene, S.: Formal Concept Analysis in Knowledge Processing: A Survey on Applications. Inf. Sci. (2012)
6. Ślęzak, D., Synak, P., Toppin, G., Wróblewski, J., Borkowski, J.: Rough SQL - Semantics and Execution. In: Proc. of IPMU (to appear, 2012)
7. Ślęzak, D., Janusz, A.: Ensembles of Bireducts: Towards Robust Classification and Simple Representation. In: Kim, T.-H., Adeli, H., Ślęzak, D., Sandnes, F.E., Song, X., Chung, K.-I., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 64–77. Springer, Heidelberg (2011)
8. Ślęzak, D., Janusz, A., Świeboda, W., Nguyen, H.S., Bazan, J.G., Skowron, A.: Semantic Analytics of PubMed Content. In: Holzinger, A., Simonic, K.-M. (eds.) USAB 2011. LNCS, vol. 7058, pp. 63–74. Springer, Heidelberg (2011)

* Partly supported by grant 2011/01/B/ST6/03867 from Ministry of Science and Higher Education of Republic of Poland, and National Centre for Research and Development (NCBiR) under grant SP/I/1/77065/10 by strategic scientific research and experimental development program: "Interdisciplinary System for Interactive Scientific and Scientific-Technical Information".

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, p. 6, 2012.

Approximating Concept Stability

Mikhail A. Babin and Sergei O. Kuznetsov

National Research University Higher School of Economics,
Pokrovskii bd. 11, 109028 Moscow, Russia
mikleb@yandex.ru, skuznetsov@hse.ru

Abstract. Concept stability was used in numerous applications for selecting concepts as biclusters of similar objects. However, scalability remains a challenge for computing stability. The best algorithms known so far have algorithmic complexity quadratic in the size of the lattice. In this paper the problem of approximate stability computation is analyzed. An approximate algorithm for computing stability is proposed. Its computational complexity and results of computer experiments in comparing stability index and its approximations are discussed.

Keywords: concept stability, approximate counting, computational complexity.

1 Introduction

The approaches to data analysis and data mining using concept lattices for clustering and ontology engineering often encounter the problem of the large number of concepts of a formal context. There may be exponentially many formal concepts wrt. the size of the underlying context, the problem of computing the number of formal concepts given a context being #P-complete [5]. Several indices were proposed for measuring concept quality, such as concept stability [1,6,8,9], probability and separation [13]. Stability was used in numerous applications for selecting concepts as biclusters of similar objects, e.g., in technical diagnostics [1], in detecting scientific subcommunities [9,11,10], in planing medical treatment [12,17], or in grouping French verbs [14,16,15]. In [13] the authors compared filtration based on various indices and their linear combinations for data recovery. Linear index combinations that showed the best performance in computer experiments on concept filtration use stability with large weights. However, a potential constraint for applying stability for large data is the complexity of its computation, shown to be #P-complete in [1,8]. Sergei Obiedkov et al. proposed [11] an algorithm for computing stability index for all concepts using the concept lattice. This algorithm was quite good in practical applications so far, but in the worst case its complexity is quadratic in the size of the lattice (which itself can be exponential in the input context size). In this paper we consider the problem of approximate stability computation. We propose an approach to approximation, consider its computational complexity and discuss results of computer experiments in comparing stability index and its approximations. The rest of the paper is organized as follows. In the next section we recall

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 7–15, 2012.

main definitions related to FCA and concept stability, in Section 3 we discuss the complexity of approximations of the number of all closed and nonclosed sets, in Section 4 we consider computation of stability and in Section 5 we discuss results of computer experiments.

2 Main Definitions

2.1 FCA

Here we briefly recall the FCA terminology [3]. Let G and M be sets, called the set of objects and attributes, respectively. Let I be a relation $I \subseteq G \times M$ between objects and attributes: for $g \in G, m \in M, gIm$ holds iff the object g has the attribute m. The triple $\mathbb{K} = (G, M, I)$ is called a *(formal) context.* If $A \subseteq G, B \subseteq M$ are arbitrary subsets, then a *Galois connection* is given by the following *derivation operators*:

$$A' = \{m \in M \mid gIm \ \forall g \in A\}$$

$$B' = \{g \in G \mid gIm \ \forall m \in B\}$$

The pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *(formal) concept (of the context* $\mathbb{K}$*)* with *extent* A and *intent* B (in this case we have also $A'' = A$ and $B'' = B$).

The operation $(\cdot)''$ is a closure operator [3], i.e. it is idempotent ($X'''' = X''$), extensive ($X \subseteq X''$), and monotone ($X \subseteq Y \Rightarrow X'' \subseteq Y''$). Sets $A \subseteq G$, $B \subseteq M$ are called *closed* if $A'' = A$ and $B'' = B$. Obviously, extents and intents are closed sets. The set of attributes B is *implied by the set of attributes* A, or the implication $A \to B$ holds, if all objects from G that have all attributes from the set A also have all attributes from the set B, i.e. $A' \subseteq B'$. Implications obey the Armstrong rules:

$$\frac{}{A \to A}, \quad \frac{A \to B}{A \cup C \to B}, \quad \frac{A \to B, B \cup C \to D}{A \cup C \to D}.$$

A subset $X \subseteq M$ *respects* an implication $A \to B$ if $A \subseteq X$ implies $B \subseteq X$. Every set of implications $\mathfrak{J}$ on the set M defines a closure operator $(\cdot)^{\mathfrak{J}}$ on M, where a subset of M is closed iff it respects all implications from $\mathfrak{J}$

2.2 Stability

The notion of stability of a formal concept was first introduced in [1,8] and now is used in a slightly revised form from [9,11].

Definition 1. *Let $\mathbb{K} = (G, M, I)$ be a formal context and (A, B) be a formal concept of $\mathbb{K}$. The* (intensional) stability *$\sigma_{in}(A, B)$, or $\sigma_{in}(A)$, is defined as follows:*

$$\sigma_{in}(A, B) = \frac{|C \subseteq A \mid C' = B|}{2^{|A|}}$$

The *extentional stability* is defined in the dual way:

$$\sigma_{ex}(A, B) = \sigma_{ex}(B) = \frac{|C \subseteq B \mid C'' = B|}{2^{|B|}}.$$

Usually, when it does not lead to misunderstanding, subscripts $_{in}$ and $_{ex}$ are omitted.

The numerator of intensional stability $\gamma(A, B) = |C \subseteq A \mid C' = B|$ is the number of all generators of the concept (A, B), so

$$2^{|A|} = \sum_{(C,D) \leq (A,B)} \gamma(C, D)$$

and

$$\gamma(A, B) = \sum_{(C,D) \leq (A,B)} 2^{|C|} \mu((C, D), (A, B)),$$

where $\mu(x, y)$ is the Möbius function of the concept lattice. Thus, stability nominator is dual to powersets of extents of the concept lattice wrt. the Möbius function of the concept lattice. This is reflected in the algorithm from [11] for computing stability, which is implicitly based on inclusion-exclusion principle, like standard algorithms for computing the Möbius function of a lattice.

3 Approximation of the Number of Closed and Nonclosed Sets

Many counting problems in FCA are known to be #P-complete but it does not imply that they cannot be solved approximately in polynomial time. For example, the problem of counting satisfying assignments for a DNF (unlike the dual problem for CNF) can be solved approximately using so-called FPRAS [2]. A *randomized approximation scheme* for a counting problem $f \colon \Sigma^* \to \mathbb{N}$ (e.g., the number of formal concepts of a context) is a randomized algorithm that takes as input an instance $x \in \Sigma^*$ (e.g. a formal context $\mathbb{K} = (G, M, I)$) and an error tolerance $\varepsilon > 0$, and outputs a number $N \in \mathbb{N}$ such that, for every input instance x,

$$Pr[(1 - \varepsilon) f(x) \leq N \leq (1 + \varepsilon) f(x)] \geq \frac{3}{4}$$

If the time of randomized approximation scheme is polynomial in $|x|$ and ε^{-1}, then this algorithm is called *fully polynomial randomized approximation scheme*, or FPRAS.

Below, for the problem *Problem* we will denote the number of solutions of *Problem* on corresponding input (which will be clear from the context) by $|\#Problem|$.

Given a hypergraph $G = (V, \mathcal{E}),\ \mathcal{E} = \{E_1, \ldots, E_m\}$, a subset $U \subseteq V$ is called *independent set* if $E_i \nsubseteq U$ for any $1 \le i \le m$ and is called *coindependent set* if $U \nsubseteq E_i$ for any $1 \le i \le m$.

Problem 1. Counting independent set (#IS)
INPUT: A hypergraph G.
OUTPUT: The number of independent sets of all sizes of G

It is known that there is no FPRAS for #IS unless $RP = NP$ (see [4]) when the hypergraph is a simple graph. So we can see this problem is hard even when V and $\{\emptyset\}$ are not edges of the hypergraph.

We also need the formulation of the following problem.

Problem 2. Counting coindependent sets (#CIS)
INPUT: A hypergraph $\mathcal{G} = (V, \mathcal{E}),\ \mathcal{E} = \{E_1, \ldots, E_m\},\ E_i \subseteq V$.
OUTPUT: The number of coindependent sets of G.

Note that set $U \subset V$ is an independent set of a hypergraph $G = (V, \mathcal{E}),\ \mathcal{E} = \{E_1, \ldots, E_m\}$ iff $V \backslash U$ is a coindependent set of the hypergraph $G' = (V, \mathcal{E}'),\ \mathcal{E}' = \{V \setminus E_1, \ldots, V \setminus E_m\}$. Thus there is no FPRAS for #CIS, unless $RP = NP$.

Now we are ready to discuss complexity of the counting problems for nonclosed sets of a formal context, closed sets of the closure system given by implication base, and nonclosed sets of the closure system given by implication base.

Problem 3. Counting nonclosed sets (#NC)
INPUT: A formal context $\mathbb{K} = (G, M, I)$
OUTPUT: The number of sets $A \subset M$ that $A'' \neq A$

Proposition 2. *There is no FPRAS for #NC, unless* $RP = NP$

Proof. Consider any input instance $(V, \mathcal{E}),\ V = \{v_1, \ldots, v_n\}\ \mathcal{E} = \{E_1, \ldots, E_m\}$ of #CIS. From this instance we construct the formal context $\mathbb{K} = (G, V, I)$ with the set of object intents $\bigcup_{1 \le i \le m} E_i \cup \{E_i \setminus \{v_1\}\} \cup \{E_i \setminus \{v_2\}\} \cup \ldots \cup \{E_i \setminus \{v_n\}\}$. Obviously, the set $A \subseteq V$ is a coindependent set of hypergraph $(V, \mathcal{E})$ iff $A'' \neq A$ or $A = V$ for the context $\mathbb{K}$. Hence $|\#CIS| = |\#NC| + 1$. □

Problem 4. Counting closed sets of implication base ($\#C_{\mathfrak{J}}$)
INPUT: An inplication base $\mathfrak{J} = \{A_1 \to B_1, \ldots, A_m \to B_m\},\ A_i, B_i \subseteq M$
OUTPUT: The number of closed sets of $\mathfrak{J}$.

Proposition 3. *There is no FPRAS for* $\#C_{\mathfrak{J}}$*, unless* $RP = NP$

Proof. Consider any input instance $(V, \mathcal{E})$, $\mathcal{E} = \{E_1, \ldots, E_m\}$ of #IS. From this instance let us construct the implication base $\mathfrak{J} = \{E_1 \to V, \ldots, E_m \to V\}$ (implications are defined on the set V). Obviously, a set U is an independent set of hypergraph $(V, \mathcal{E})$ iff U is closed set of $\mathfrak{J}$ and $U \neq V$. Hence $|\#IS| = |\#C_{\mathfrak{J}}|-1$. □

Since a closed set wrt. an implication base can be represented as a satisfying assignment of a Horn CNF, we immediately get

Corollary 1. There is no FPRAS for the counting problem of Horn CNF satisfiability ($\#Horn\ SAT$), unless $NP = RP$.

Problem 5. Counting nonclosed set of implication base ($\#NC_{\mathfrak{J}}$)
INPUT: An inplication base $\mathfrak{J} = \{A_1 \to B_1, \ldots, A_m \to B_m\},\ A_i, B_i \subseteq M$
OUTPUT: The number of nonclosed sets of $\mathfrak{J}$.

Proposition 4. *There is FPRAS for* $\#NC_{\mathfrak{J}}$

Proof. Consider an instance $\mathfrak{J} = \{A_1 \to B_1, \ldots, A_m \to B_m\},\ A_i, B_i \subseteq X$ of $\#NC_{\mathfrak{J}}$. Closed sets of implication base $\mathfrak{J}$ are in one-to-one correspondence with the satisfying truth assignments of the corresponding Horn CNF $f_{\mathfrak{J}}$. Thus nonclosed sets of $\mathfrak{J}$ are in one-to-one correspondence with satisfying truth assignments of DNF $\neg f_{\mathfrak{J}}$. There is a known FPRAS for the counting problem of satisfying assignments of a DNF [2]. □

It is worth to note that exact complexity of approximate counting of a closed set of a formal context is open, but it is known that this problem is complete in class $\#RH\Pi_1$ [4]. All of the above results of this section can be summarized in the following table.

Table 1. Complexity of closed/nonclosed sets counting

	#closed sets	#nonclosed sets
$cs(\mathbb{K})$	$\#RH\Pi_1$-complete	no FPRAS, unless $RP = NP$
$cs(\mathfrak{J})$	no FPRAS, unless $RP = NP$	FPRAS

$cs(\mathbb{K})$ denotes the case where a closure system is given by context $\mathbb{K}$.
$cs(\mathfrak{J})$ denotes the case where a closure system is given by implication base $\mathfrak{J}$.

4 Computation of Stability

Recall that exact computing of concept stability is an #P-complete problem[1,6,8]. Moreover, there is no FPRAS for computing stability, unless $RP = NP$. In order to show this fact consider the context from the proof of proposition 2. Clearly $\sigma(M) = (|\#NC|+1)/2^{|M|}$. Here we discuss how to approximate stability with bounded absolute error. By definition of stability, stability of an intent A of a formal context $\mathbb{K} = (G, M, I)$ equals to the probability that a closure of a random subset of A is equal to A, i.e. $\sigma(A) = Pr(X'' = A)$, when X is chosen uniformly and random from 2^A. Thus to estimate $\sigma(A)$ we can use a Monte Carlo method.

```
GetStability(A, N)
answer ← 0
for i ← 1 to N
      do pick random subset X of A
         if X'' = A
            then answer ← answer + 1
answer ← answer / N
return answer
```

Recall Chernoff-Hoeffding theorem with simplified bounds [2].

Theorem 5 (Chernoff-Hoeffding). *Let $X_1, X_2, \ldots, X_N$ be independent and identically distributed random variables with $p = E(X_i)$. Then*

$$Pr(\frac{1}{N}\sum X_i \leq p - \varepsilon) \leq \exp(-2\varepsilon^2 N)$$

It is easy to get the following proposition which states that for sufficiently large $N = N(\varepsilon, \delta)$, the probability of $|\,answer - \sigma(A)| \geq \varepsilon$ is not greater than δ.

Proposition 6. *The Monte Carlo method yields an approximation to $\sigma(A)$ with probability at least $1 - \delta$ and absolute error ε provided*

$$N > \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

Proof. If we take random variables to be $1 - X_i$, and substitute them in the inequality of the Chernoff-Hoeffding theorem, then we have $p = E(1 - X_i)$ and we get

$$Pr(\frac{1}{N}\sum X_i \geq p + \varepsilon) \leq \exp(-2\varepsilon^2 N).$$

Hence

$$Pr(|\frac{1}{N}\sum X_i - p| \geq \varepsilon) \leq$$
$$\leq Pr(\frac{1}{N}\sum X_i \leq p - \varepsilon) + Pr(\frac{1}{N}\sum X_i \geq p + \varepsilon) \leq$$
$$\leq 2\exp(-2\varepsilon^2 N).$$

Consider random variables X_i such that $X_i = 1$ iff $X'' = A$ in the i-th iteration of GetStability and $X_i = 0$ otherwise. Thus $\frac{1}{N}\sum X_i = answer$, where *answer* is returned by GetStability(A,N) at the ith iteration. Absolute error probability is $Pr(|\,answer - p| \geq \varepsilon) \leq 2\exp(-2\varepsilon^2 N) \leq \delta$. Hence $2\varepsilon^2 N \geq \ln \frac{2}{\delta}$. □

We can use results of this algorithm to select top approximate stable concepts using the following straightforward algorithm.

TopStableConcepts($\mathbb{K}, \gamma_0$)

```
answer ← ∅
for every concept C = (A, A′) of 𝕂
    do if approxStability(A) > σ_θ
        then answer ← answer ∪ {(A, A′)}
return answer
```

5 Experimental Results

In this section we discuss experimental results in computing stability approximations for random contexts of various sizes and density. The results of the approximate stability computation on random contexts are presented in Figure 1 and Figure 2. The Y-axis (labeled as *Error*) gives the relative error

$$|S(\mathbb{K}, \tilde{\sigma}, \sigma_\theta) \Delta S(\mathbb{K}, \sigma, \sigma_\theta)| / |S(\mathbb{K}, \sigma, \sigma_\theta)|.$$

Here $S(\mathbb{K}, \sigma, \sigma_\theta)$ denotes the set of all concepts with stability $\sigma \geq \sigma_\theta$; $S(\mathbb{K}, \tilde{\sigma}, \sigma_\theta)$ denotes the set of all concepts with approximate stability $\tilde{\sigma} \geq \sigma_\theta$, where σ_θ is a parameter (*stability threshold*). For every pair $g \in G$, $m \in M$ of a random context $\mathbb{K} = (G, M, I)$ one has $(g, m) \in I$ with probability d called *context density*.

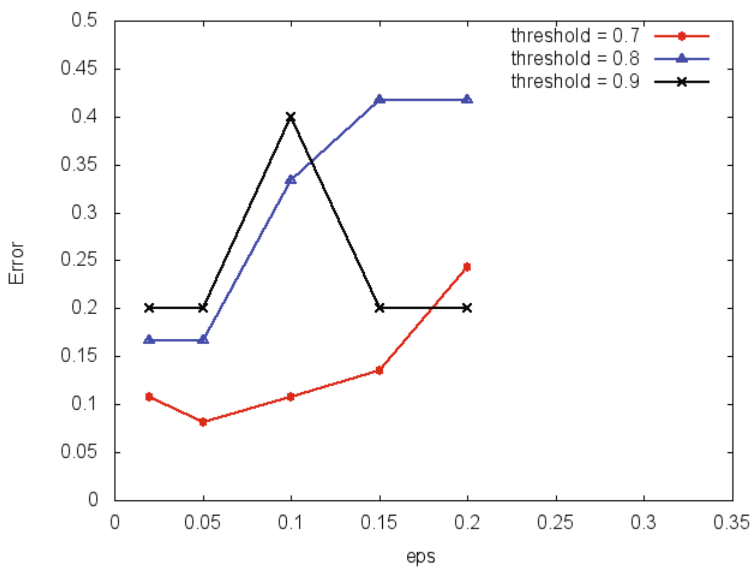


Fig. 1. Approximation quality for random contexts 100×30 with density 0.3

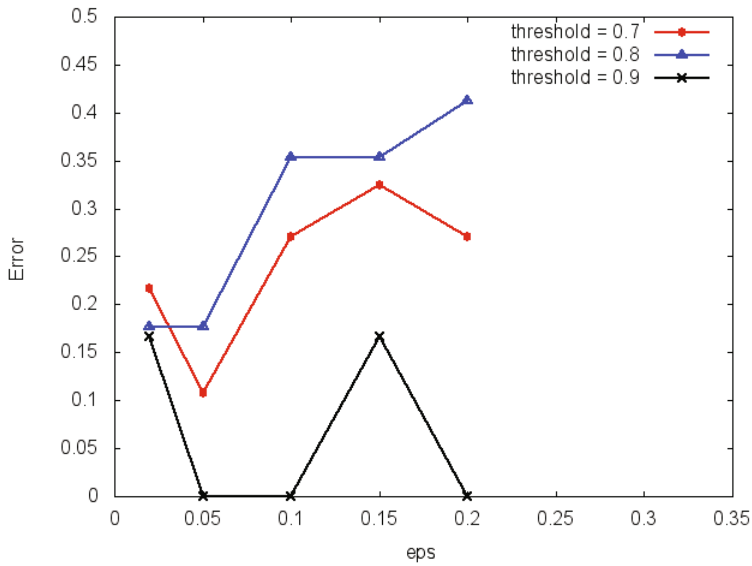


Fig. 2. Approximation quality for random contexts 150×30 with density 0.2

The results of computer experiments show that the algorithm for computing approximated stability algorithm has better precision when stability threshold is lower. This behaviour is consistent with theory, since when the stability threshold is high the number of stable concept is small and a small deviation of this threshold can result in significant change of the relative number of "stable" concepts (i.e. concepts with approximate stability larger than threshold).

6 Conclusion

The problem of approximate stability computation was analyzed. Approximate solution of the problem was shown to be hard: the existence of FPRAS solving this problem would imply NP = RP. An approximate algorithm for computing stability, which can run in reasonable time for approximations with bounded absolute error was proposed. Its computational complexity and results of computer experiments in comparing stability index and its approximations were discussed. The results show that the approximations are better when stability threshold is low. Further study will be related to comparing approximate stability to other concept interestingness measures, such as independence, probability, wrt. computation time and selectiveness. Another challenging task would be the generation of interesting concepts without generating the set of all concepts.

Acknowledgments. This work was partially supported by Russian Foundation for Basic Research, project no. 08-07-92497-NTsNILa.

References

1. Kuznetsov, S.O.: Stability as an estimate of the degree of substantiation of hypotheses derived on the basis of operational similarity. Autom. Document. Math. Ling. (Nauchn. Tekh. Inf., Ser. 2) (12), 21–29 (1990)
2. Motwani, R., Raghavan, P.: Randomized Algorithms. Cambridge University Press (1995)
3. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
4. Dyer, M., Goldberg, L.A., Greenhill, C., Jerrum, M.: On the Relative Complexity of Approximate Counting Problems. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 108–119. Springer, Heidelberg (2000)
5. Kuznetsov, S.O.: On Computing the Size of a Lattice and Related Decision Problems. Order 18(4), 313–321 (2001)
6. Kuznetsov, S.O.: Stability of a Formal Concept. In: San-Juan, E. (ed.) Proc. 4th Journee d'Informatique Messine (JIM 2003), Metz (2003)
7. Roth, C., Obiedkov, S., Kourie, D.: Towards Concise Representation for Taxonomies of Epistemic Communities. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) CLA 2006. LNCS (LNAI), vol. 4923, pp. 240–255. Springer, Heidelberg (2008)
8. Kuznetsov, S.O.: On Stability of a Formal Concept. Annals of Mathematics and Artificial Intelligence 49, 101–115 (2007)
9. Kuznetsov, S.O., Obiedkov, S., Roth, C.: Reducing the Representation Complexity of Lattice-Based Taxonomies. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 241–254. Springer, Heidelberg (2007)
10. Roth, C., Obiedkov, S., Kourie, D.: Towards Concise Representation for Taxonomies of Epistemic Communities. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) CLA 2006. LNCS (LNAI), vol. 4923, pp. 240–255. Springer, Heidelberg (2008)
11. Obiedkov, S.A., Roth, C., Kourie, D.G.: On Succinct Representation of Knowledge Community Taxonomies with Formal Concept Analysis. Int. J. Found. Comput. Sci. 19(2), 383–404 (2008)
12. Jay, N., Kohler, F., Napoli, A.: Analysis of Social Communities with Iceberg and Stability-Based Concept Lattices. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 258–272. Springer, Heidelberg (2008)
13. Klimushkin, M., Obiedkov, S., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 255–266. Springer, Heidelberg (2010)
14. Falk, I., Gardent, C., Lorenzo, A.: Using Formal Concept Analysis to Acquire Knowledge about Verbs. In: Proc. 7th of the International Conference on Concept Lattices and Their Applications (CLA 2010), pp. 151–162 (2010)
15. Falk, I., Gardent, C.: Bootstrapping a Classification of French Verbs Using Formal Concept Analysis. In: Interdisciplinary Workshop on Verbs, Pisa, Italy (2011)
16. Falk, I., Gardent, C.: Combining Formal Concept Analysis and Translation to Assign Frames and Thematic Role Sets to French Verbs. In: Proc. International Conference Concept Lattices and Their Applications (CLA 2011), Nancy, France (2011)
17. Egho, E., Jay, N., Raissi, C., Napoli, A.: A FCA-based Analysis of Sequential Care Trajectories. In: Proc. 8th of the International Conference on Concept Lattices and Their Applications (CLA 2011), Nancy, France, pp. 362–376 (2011)

Logical Analysis of Concept Lattices by Factorization

Eduard Bartl and Michal Krupka

Palacky University
{eduard.bartl,michal.krupka}@upol.cz
http://www.inf.upol.cz/

Abstract. Reducing the size of concept lattices is a well-known problem in Formal Concept Analysis. A particular instance of this problem is the size reduction of concept lattices using factorization by complete tolerances. We show that all complete tolerances on a complete lattice (i.e., all possible ways of factorizing the lattice) with a naturally-defined operation of multiplication form a residuated lattice. This allows looking at the set of all complete tolerances as a scale of truth degrees using which we can evaluate formulas of predicate logic specifying the desired parameters of the factorization. We present illustrative example to clarify our approach.

Keywords: block relation, complete residuated lattice, complete tolerance, formal concept analysis.

1 Introduction

There is a well-known fact that even small formal contexts generate concept lattices consisting of a large number of formal concepts. In some particular cases, providing such a complex lattice does not bring any significant added value to a user. This issue can be addressed by applying some kind of reduction method.

In this paper we use factorization as a tool for simplifying examined data. The price for this simplification are small (acceptable) inaccuracies introduced into the data. We give to a user the possibility to formulate an additional relationship in the data which was not originally present. Such a relationship represents the above mentioned acceptable inaccuracy and therefore reduces the concept lattice.

In our approach to the size reduction of concept lattices we consider some formal concepts as similar according to particular criteria. Similarity of the formal concepts can be naturally modeled by complete tolerances (completeness property is required since we need to preserve structure of a concept lattice). The reduction of concept lattices based on complete tolerances was introduced in [3] and further developed in [6]. The problem is how to intuitively specify a complete tolerance on the complete lattice we want to factorize. In this paper we provide an answer to this question.

First, we propose a theorem that a set of all complete tolerances on a complete lattice along with a naturally-defined operation of multiplication form a residuated lattice. Such a structure is called a tolerance residuated lattice. Obviously

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 16–27, 2012.

it is convenient to think of the complete lattice on which we define complete tolerances as a concept lattice. It is then possible to work with the objects, attributes and with an incidence relation between them. Second, we establish a logical calculus where a tolerance residuated lattice serves as the scale of truth values. Formulas of this logical calculus specify so-called block relations on the given concept lattice which are in a one-to-one correspondence with the complete tolerances on the concept lattice.

One can describe a block relation in natural language and then translate it into a formula of proposed logical calculus. This formula is eventually evaluated to an element of the tolerance residuated lattice. In other words, the formula is evaluated to a truth value specifying the way how a given concept lattice is going to be factorized.

In the paper we also present an illustrative example explaining the proposed method.

2 Preliminaries

2.1 Formal Concept Analysis

Formal Concept Analysis has been introduced in [8], our basic reference is [3]. A *formal context* is a triple $\langle X, Y, I\rangle$ where X is a set of objects, Y a set of attributes and $I \subseteq X \times Y$ a binary relation between X and Y. For $\langle x, y\rangle \in I$ it is said "The object x has the attribute y".

For subsets $A \subseteq X$ and $B \subseteq Y$ we set

$$A^{\uparrow_I} = \{y \in Y \mid \text{for each } x \in A \text{ it holds } \langle x, y\rangle \in I\},$$
$$B^{\downarrow_I} = \{x \in X \mid \text{for each } y \in B \text{ it holds } \langle x, y\rangle \in I\}.$$

If $A^{\uparrow_I} = B$ and $B^{\downarrow_I} = A$, then the pair $\langle A, B\rangle$ is called a *formal concept* of $\langle X, Y, I\rangle$. The set A is called the *extent of* $\langle A, B\rangle$, the set B the *intent of* $\langle A, B\rangle$.

A partial order $\leq$ on the set $\mathcal{B}(X, Y, I)$ of all formal concepts of $\langle X, Y, I\rangle$ is defined by $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ iff $A_1 \subseteq A_2$ (iff $B_2 \subseteq B_1$). $\mathcal{B}(X, Y, I)$ along with $\leq$ is a complete lattice and is called the *concept lattice* of $\langle X, Y, I\rangle$.

2.2 Complete Tolerances and Block Relations

The results of this subsection are taken from [9], see also [3].

A *tolerance on a set* X is a binary relation t which is reflexive and symmetric. Each tolerance on X induces a covering of X, called the *factor* (*quotient*) *set*. This covering consists of all maximal blocks of the tolerance, i.e., maximal (with respect to set inclusion) subsets $B \subseteq X$ such that for any $a, b \in B$ it holds $\langle a, b\rangle \in t$. The factor set of X induced by a tolerance t is denoted X/t. Note that X/t is a covering of X, but need not be a partition. X/t is a partition of X if and only if t is transitive (thus an equivalence relation).

A *complete tolerance* on a complete lattice $\mathbf{V} = \langle V, \wedge, \vee, 0, 1\rangle$ is a tolerance which preserves suprema and infima. More precisely, a tolerance t on $\mathbf{V}$ is complete if from $\langle a_j, b_j\rangle \in t$ for all $j \in J$ it follows $\langle \bigvee_{j\in J} a_j, \bigvee_{j\in J} b_j\rangle \in t$ and $\langle \bigwedge_{j\in J} a_j, \bigwedge_{j\in J} b_j\rangle \in t$ (J is an arbitrary index set).

For a complete tolerance t on $\mathbf{V}$ and $v \in V$ we denote

$$v^t = \bigvee\{w \in V \mid \langle v, w\rangle \in t\}, \quad v_t = \bigwedge\{w \in V \mid \langle v, w\rangle \in t\}, \tag{1}$$

$$[v]_t = [v_t, (v_t)^t], \quad [v]^t = [(v^t)_t, v^t], \tag{2}$$

where $[v_1, v_2]$ denotes the interval $\{w \in V \mid v_1 \leq w \leq v_2\}$.

The mapping $v \mapsto v^t$ (resp. $v \mapsto v_t$) is a $\bigwedge$-morphism (resp. $\bigvee$-morphism) of $\mathbf{V}$. The equations (2) describe all maximal blocks of t. It holds ([2,9]):

$$L/t = \{[v]_t \mid v \in V\} = \{[v]^t \mid v \in V\}.$$

An ordering on the set V/t is introduced using suprema of maximal blocks and can be equivalently introduced using infima. For blocks $B_1, B_2 \in V/t$ we set

$$B_1 \leq B_2 \quad \text{iff} \quad \bigvee B_1 \leq \bigvee B_2 \quad \left(\text{iff} \quad \bigwedge B_1 \leq \bigwedge B_2\right). \tag{3}$$

The set V/t together with this ordering is a complete lattice, which is denoted by $\mathbf{V}/t$.

The following result ([3,9]) characterizes all systems of maximal blocks of complete tolerances on complete lattices.

Lemma 1. *Let $\mathcal{V}$ be a set of intervals in a complete lattice $\mathbf{V}$ such that $\mathcal{V}$ is a covering of $\mathbf{V}$. Then $\mathcal{V}$ is the set of all maximal blocks of some complete tolerance on $\mathbf{V}$ if and only if for each $[u_1, v_1], [u_2, v_2] \in \mathcal{V}$ it holds $u_1 \leq u_2$ iff $v_1 \leq v_2$ and for each $[u_k, v_k] \in \mathcal{V}$, $k \in K$, the element $\bigwedge_{k\in K} v_k$ (resp. $\bigvee_{k\in K} u_k$) is the upper (resp. lower) bound of some interval from $\mathcal{V}$.*

We denote by $\mathrm{CTol}\,\mathbf{V}$ the set of all complete tolerances on a complete lattice $\mathbf{V}$. $\mathrm{CTol}\,\mathbf{V}$ forms a closure system on V. Therefore, $\mathrm{CTol}\,\mathbf{V}$ along with ordering $\leq$ defined as $s \leq t$ iff $t \subseteq s$ forms a complete lattice. Infima and suprema in $\mathrm{CTol}\,\mathbf{V}$ are then given by

$$\bigwedge_{k\in K} t_k = C_{\mathbf{V}}\left(\bigcup_{k\in K} t_k\right), \qquad \bigvee_{k\in K} t_k = \bigcap_{k\in K} t_k, \tag{4}$$

where $C_{\mathbf{V}}\colon 2^V \to 2^V$ is the closure operator corresponding to the closure system $\mathrm{CTol}\,\mathbf{V}$. The least element $0_{\mathrm{CTol}\,\mathbf{V}}$ in $\mathrm{CTol}\,\mathbf{V}$ is the total relation $0_{\mathrm{CTol}\,\mathbf{V}} = V \times V$, the greatest element $1_{\mathrm{CTol}\,\mathbf{V}}$ is the equality (diagonal) on V, i.e. $1_{\mathrm{CTol}\,\mathbf{V}} = \{\langle v, v\rangle \mid v \in V\}$.

The following equalities describe the mappings (1) for the infimum and the supremum of complete tolerances $t_k \in \mathrm{CTol}\,\mathbf{V}$, $k \in K$:

$$v^{\bigcap_{k\in K} t_k} = \bigwedge_{k\in K} v^{t_k}, \qquad v_{\bigcap_{k\in K} t_k} = \bigvee_{k\in K} v_{t_k}. \tag{5}$$

Let $\langle X, Y, I\rangle$ be a formal context. A relation $J \supseteq I$ is called a *block relation on* $\langle X, Y, I\rangle$ if for each $x \in X$ the set $\{x\}^{\uparrow_J}$ is an intent of I and for each $y \in Y$ the set $\{y\}^{\downarrow_J}$ is an extent of I.

The set $BR(X, Y, I)$ of all block relations on a formal context $\langle X, Y, I\rangle$ is a closure system. We consider $BR(X, Y, I)$ together with the dual structure of complete lattice:

$$\bigwedge_{k\in K} J_k = C_I\Big(\bigcup_{k\in K} J_k\Big), \qquad \bigvee_{k\in K} J_k = \bigcap_{k\in K} J_k, \tag{6}$$

where C_I is the closure operator induced by the closure system $BR(X, Y, I)$.

By [9], the complete lattices $\mathrm{CTol}\,\mathcal{B}(X, Y, I)$ and $BR(X, Y, I)$ are isomorphic, for corresponding $t \in \mathrm{CTol}\,\mathcal{B}(X, Y, I)$ and $J \in BR(X, Y, I)$ it holds

$$\langle x, y\rangle \in J \quad \text{iff} \quad \langle \gamma(x), \gamma(x) \wedge \mu(y)\rangle \in t, \tag{7}$$

$$\text{iff} \quad \langle \gamma(x) \vee \mu(y), \mu(y)\rangle \in t, \tag{8}$$

$$\langle\langle A_1, B_1\rangle, \langle A_2, B_2\rangle\rangle \in t \quad \text{iff} \quad (A_1 \times B_2) \cup (A_2 \times B_1) \subseteq J, \tag{9}$$

where γ and μ are the mappings $x \mapsto \langle\{x\}^{\uparrow\downarrow}, \{x\}^{\uparrow}\rangle$ and $y \mapsto \langle\{y\}^{\downarrow}, \{y\}^{\downarrow\uparrow}\rangle$, respectively.

2.3 Residuated Lattices

In fuzzy logic, complete residuated lattices are frequently used as basic structures of truth degrees. A *complete residuated lattice* [1,4,7] is a structure $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that: (i) $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice, i.e., a partially ordered set in which arbitrary infima and suprema exist; (ii) $\langle L, \otimes, 1\rangle$ is a commutative monoid, i.e., $\otimes$ is a binary operation which is commutative, associative, and $a \otimes 1 = a$ for each $a \in L$; (iii) $\otimes$ and $\rightarrow$ satisfy adjointness, i.e., $a \otimes b \leq c$ iff $a \leq b \rightarrow c$. 0 and 1 denote the least and greatest elements. The partial order of $\mathbf{L}$ is denoted by $\leq$.

Elements a of L are called *truth degrees.* $\otimes$ and $\rightarrow$ (truth functions of) "fuzzy conjunction" and "fuzzy implication".

Common examples of complete residuated lattices include those defined on a unit interval, (i.e., $L = [0, 1]$) or on a finite chain in the unit interval, e.g. $L = \{0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1\}$, $\wedge$ and $\vee$ being minimum and maximum, $\otimes$ being a left-continuous t-norm with the corresponding $\rightarrow$.

Lemma 2. *Let* $\mathbf{L} = \langle L, \wedge, \vee, \otimes, 0, 1\rangle$ *be a structure satisfying conditions (i) and (ii) of the definition of residuated lattices, such that* $\langle L, \wedge, \vee, 0, 1\rangle$ *is a complete lattice. Then the following two conditions are equivalent:*

1. *There exists a binary operation* $\rightarrow$ *on* L*, satisfying the condition (iii).*
2. $\mathbf{L}$ *satisfies*

$$a \otimes \bigvee_{j\in J} b_j = \bigvee_{j\in J} a \otimes b_j. \tag{10}$$

For $\rightarrow$ *it holds*

$$a \rightarrow b = \bigvee\{c \in L \mid a \otimes c \leq b\}. \tag{11}$$

3 Main Results

3.1 Tolerance Residuated Lattice

Let $\mathbf{V}$ be a complete lattice. We set for each $s, t \in \mathrm{CTol}\,\mathbf{V}$,

$$s \otimes t = st \cap ts, \tag{12}$$

$$s \rightarrow t = \bigvee\{r \in \mathrm{CTol}\,\mathbf{V} \mid s \otimes r \leq t\}, \tag{13}$$

where st denotes the composition of the complete tolerances s and t.

Lemma 3. *For each* $s, t \in \mathrm{CTol}\,\mathbf{V}$ *the relations* $s \otimes t$ *and* $s \rightarrow t$ *are complete tolerances. For each* $u \in V$ *it holds*

$$u^{s\otimes t} = (u^s)^t \wedge (u^t)^s, \qquad u_{s\otimes t} = (u_s)_t \vee (u_t)_s, \tag{14}$$

$$u^{s\rightarrow t} = \bigwedge_{s\otimes r\leq t} u^r, \qquad u_{s\rightarrow t} = \bigvee_{s\otimes r\leq t} u_r. \tag{15}$$

Proof. $s \otimes t$ is evidently a tolerance. We have for each $u, v \in V$, $\langle u, v\rangle \in s \otimes t$ iff $\langle u, v\rangle \in st$ and $\langle u, v\rangle \in ts$. Now, $\langle u, v\rangle \in st$ iff there is $w \in V$ such that $\langle u, w\rangle \in s$ and $\langle w, v\rangle \in t$. This follows $w \leq u^s$ and $v \leq w^t$, which gives $v \leq (u^s)^t$. Similarly we obtain $v \leq (u^t)^s$, and, together, $v \leq (u^s)^t \wedge (u^t)^s$. Since $\langle u, u^s\rangle \in s$ and $\langle u^s, (u^s)^t\rangle \in t$, then $\langle u, (u^s)^t\rangle \in st$. Similarly, $\langle u, (u^t)^s\rangle \in ts$ and put together, $\langle u, (u^s)^t \wedge (u^t)^s\rangle \in s \otimes t$.

Thus, $(u^s)^t \wedge (u^t)^s$ is the greatest element which is in the relation $s \otimes t$ with u. Similarly it can be proved that $(u_s)_t \wedge (u_t)_s$ is the least element which is in the relation $s \otimes t$ with u.

Moreover, from the beginning of the proof it follows that $\langle u, v\rangle \in s \otimes t$ iff $v \leq (u^s)^t \wedge (u^t)^s$ and, similarly $\langle v, u\rangle \in s \otimes t$ iff $v \geq (u_s)_t \wedge (u_t)_s$. Therefore, if we set $u^{s\otimes t} = (u^s)^t \wedge (u^t)^s$ and $u_{s\otimes t} = (u_s)_t \vee (u_t)_s$ then intervals $[u, v]$, where $v = u^{s\otimes t}$ and $u = v_{s\otimes t}$, form the system of all maximal blocks of the tolerance $s \otimes t$. It can be easily verified that this system satisfies all the assumptions of the system $\mathcal{V}$ from Lemma 1. Thus, $s \otimes t$ is a complete tolerance satisfying (14).

The relation $s \rightarrow t$ is an intersection of complete tolerances, thus a complete tolerance, (15) follows from (5).

Theorem 1. *The tuple* $\langle \mathrm{CTol}\,\mathbf{V}, \wedge, \vee, \otimes, \rightarrow, 0_{\mathrm{CTol}\,\mathbf{V}}, 1_{\mathrm{CTol}\,\mathbf{V}}\rangle$ *is a complete residuated lattice.*

Proof. The operation $\otimes$ is evidently commutative and $1_{\mathrm{CTol}\,\mathbf{V}}$ is its unit element. Thus, by Lemma 2 it suffices to verify the distributivity law

$$t \otimes \bigvee_{k\in K} s_k = \bigvee_{k\in K} t \otimes s_k.$$

We have by Lemma 3, Lemma 1, (4), and (5) (all big suprema, infima, and intersections are taken over all $k \in K$),

$$\begin{aligned} u^{t\otimes\bigvee s_k} &= (u^t)^{\bigvee s_k} \wedge (u^{\bigvee s_k})^t = (u^t)^{\bigcap s_k} \wedge (u^{\bigcap s_k})^t \\ &= \Big(\bigwedge (u^t)^{s_k}\Big) \wedge \Big(\bigwedge u^{s_k}\Big)^t = \bigwedge (u^t)^{s_k} \wedge (u^{s_k})^t, \\ u^{\bigvee t\otimes s_k} &= u^{\bigcap t\otimes s_k} = \bigwedge u^{t\otimes s_k} = \bigwedge (u^t)^{s_k} \wedge (u^{s_k})^t. \end{aligned}$$

Thus, $u^{t\otimes\bigvee s_k} = u^{\bigvee t\otimes s_k}$, and, similarly, $u_{t\otimes\bigvee s_k} = u_{\bigvee t\otimes s_k}$.

Definition 1. The residuated lattice from Theorem 1 is denoted $\mathrm{CTol}\,\mathbf{V}$ and called *the tolerance residuated lattice of* $\mathbf{V}$.

3.2 First-Order Fuzzy Logic for Factorizing Concept Lattices

In the previous section it was shown how to build a complete residuated lattice using complete tolerances. The class of all complete residuated lattices $\mathcal{L}$ represents a suitable and sufficiently general class of structures of truth values. Using a tolerance residuated lattice $\mathbf{L} \in \mathcal{L}$ we can then assign a truth values to the formulas of a certain fuzzy logic, particularly of a many-sorted first-order fuzzy logic. In other words, a many-sorted first-order fuzzy logic may be used as a tool for specifying the desired parameters of the size-factorization of concept lattices.

We start with definition of a language FCL of our many-sorted first-order fuzzy logic.

Definition 2. A *language FCL* is of type $\langle S, R, F, C, \sigma\rangle$, where $S = \{\mathbb{X}, \mathbb{Y}\}$ is a set of sorts, $R = \{r_\sim, r_A, r_{A_1}, \ldots, r_B, r_{B_1}, \ldots\}$ is a set of relation symbols with arities $\sigma(r_\sim) = \mathbb{X} \times \mathbb{Y}$, $\sigma(r_A) = \sigma(r_{A_1}) = \cdots = \mathbb{X}$, $\sigma(r_B) = \sigma(r_{B_1}) = \cdots = \mathbb{Y}$, and $F = \emptyset$ is an empty set of function symbols. By $\mathbbm{x}, \mathbbm{x}_1, \ldots$ we denote variables of the sort $\mathbb{X}$, and by $\mathbbm{y}, \mathbbm{y}_1, \ldots$ variables of the sort $\mathbb{Y}$. We use logical connectives $\curlyvee, \curlywedge, \otimes, \Rightarrow$, quantifiers $\exists, \forall$, and truth constant $\mathbf{0}$.

Atomic formulas of the language FCL are for example $r_\sim(\mathbbm{x}, \mathbbm{y})$, $r_A(\mathbbm{x})$, $r_B(\mathbbm{y})$, or $\mathbf{0}$. Formulas are defined inductively using logical connectives and quantifiers. Moreover, we use the abbreviations: $\neg\, \varphi$, $(\varphi \Leftrightarrow \psi)$, $\mathbf{1}$ is the abbreviation for $(\varphi \Rightarrow \mathbf{0})$, $((\varphi \Rightarrow \psi) \curlywedge (\psi \Rightarrow \varphi))$, and $(\mathbf{0} \Rightarrow \mathbf{0})$, respectively.

In the next definition we suppose a tolerance residuated lattice $\mathbf{L} = \langle BR(X, Y, I), \vee, \wedge, \otimes, \rightarrow, 0, 1\rangle$.

Definition 3. An $\mathbf{L}$-*structure* for the language FCL is $\mathbf{M} = \langle M, R^M, F^M\rangle$, where $M = \{X, Y\}$, $R^M = \{r^M_\sim, r^M_A, r^M_{A_1}, \ldots, r^M_B, r^M_{B_1}, \ldots\}$, $F^M = \emptyset$. The fuzzy relation $r^M_\sim : X \times Y \rightarrow BR(X, Y, I)$ is defined as:

$$r^M_\sim(x, y) = \bigvee\{J \in BR(X, Y, I) \mid \langle x, y\rangle \in J\}$$

for all $x \in X$, $y \in Y$. The fuzzy sets $r^M_A, r^M_{A_1}, \dots : X \to BR(X,Y,I)$, where $A, A_1, \dots \subseteq X$, are given by:

$$r^M_A(x) = \bigvee\{J \in BR(X,Y,I) \mid x \in A^{\uparrow_J\downarrow_J}\},$$
$$r^M_{A_1}(x) = \bigvee\{J \in BR(X,Y,I) \mid x \in A_1^{\uparrow_J\downarrow_J}\},$$
$$\dots$$

for all $x \in X$. Similarly, the fuzzy sets $r^M_B, r^M_{B_1}, \dots : Y \to BR(X,Y,I)$, where $B, B_1, \dots \subseteq Y$, are defined in the way:

$$r^M_B(y) = \bigvee\{J \in BR(X,Y,I) \mid y \in B^{\downarrow_J\uparrow_J}\},$$
$$r^M_{B_1}(y) = \bigvee\{J \in BR(X,Y,I) \mid y \in B_1^{\downarrow_J\uparrow_J}\},$$
$$\dots$$

for all $y \in Y$.

For the sake of brevity, we will write just $\sim$ (using infix notation), $A, A_1, \dots,$ $B, B_1, \dots$ instead of $r^M_\sim$, $r^M_A, r^M_{A_1}, \dots$, $r^M_B, r^M_{B_1}, \dots$, respectively. Moreover, in what follows we will use instead of all relation symbols $r_A, r_{A_1}, \dots, r_B, r_{B_1}, \dots$ just symbols r_A and r_B.

For **M**-valuation v assigning to every variable $\mathbb{x}$ a value $v(\mathbb{x}) \in X$ and to every variable $\mathbb{y}$ a value $v(\mathbb{y}) \in Y$ we can define a value of a variable $\mathbb{x}$ as $||\mathbb{x}||_{\mathbf{M},v} = v(\mathbb{x})$ and a value of a variable $\mathbb{y}$ as $||\mathbb{y}||_{\mathbf{M},v} = v(\mathbb{y})$. The atomic formulas are then evaluated as follows:

$$\begin{aligned}
||r_\sim(\mathbb{x},\mathbb{y})||_{\mathbf{M},v} &= v(\mathbb{x}) \sim v(\mathbb{y}),\\
||r_A(\mathbb{x})||_{\mathbf{M},v} &= A(v(\mathbb{x})),\\
||r_B(\mathbb{y})||_{\mathbf{M},v} &= B(v(\mathbb{y})),\\
||\mathbf{0}||_{\mathbf{M},v} &= 0 = X \times Y,\\
||\mathbf{1}||_{\mathbf{M},v} &= 1 = I.
\end{aligned}$$

The formulas get the value inductively in a usual way:

$$\begin{aligned}
||\neg\,\varphi||_{\mathbf{M},v} &= ||\varphi||_{\mathbf{M},v} \to 0,\\
||(\varphi \curlyvee \psi)||_{\mathbf{M},v} &= ||\varphi||_{\mathbf{M},v} \vee ||\psi||_{\mathbf{M},v},\\
||(\varphi \curlywedge \psi)||_{\mathbf{M},v} &= ||\varphi||_{\mathbf{M},v} \wedge ||\psi||_{\mathbf{M},v},\\
||(\varphi \otimes \psi)||_{\mathbf{M},v} &= ||\varphi||_{\mathbf{M},v} \otimes ||\psi||_{\mathbf{M},v},\\
||(\varphi \Rightarrow \psi)||_{\mathbf{M},v} &= ||\varphi||_{\mathbf{M},v} \to ||\psi||_{\mathbf{M},v},\\
||(\varphi \Leftrightarrow \psi)||_{\mathbf{M},v} &= ||\varphi||_{\mathbf{M},v} \leftrightarrow ||\psi||_{\mathbf{M},v},
\end{aligned}$$

$$||(\exists \mathtt{x})\varphi||_{\mathbf{M},v} = \bigvee\{||\varphi||_{\mathbf{M},v'} \mid v' =_{\mathtt{x}} v\},$$
$$||(\exists \mathtt{y})\varphi||_{\mathbf{M},v} = \bigvee\{||\varphi||_{\mathbf{M},v'} \mid v' =_{\mathtt{y}} v\},$$
$$||(\forall \mathtt{x})\varphi||_{\mathbf{M},v} = \bigwedge\{||\varphi||_{\mathbf{M},v'} \mid v' =_{\mathtt{x}} v\},$$
$$||(\forall \mathtt{y})\varphi||_{\mathbf{M},v} = \bigwedge\{||\varphi||_{\mathbf{M},v'} \mid v' =_{\mathtt{y}} v\}$$

for every formulas φ and ψ.

For instance, proposition "there exists an object x from set a A such that x has attribute y_1 and does not have attribute y_2" can be expressed using FCL by the formula:

$$(\exists \mathtt{x})(r_A(\mathtt{x}) \Rightarrow (r_\sim(\mathtt{x}, \mathtt{y}_1) \otimes \neg\, r_\sim(\mathtt{x}, \mathtt{y}_2))).$$

4 Illustrative Example

As an illustrative example we consider size reduction of concept lattice $\mathcal{B}(X, Y, I)$ depicted in Fig. 1 (right). This concept lattice is constructed from the simple formal context $\langle X, Y, I\rangle$, where I is a relation between sets $X = \{\mathtt{M}, \mathtt{F}, \mathtt{J}\}$ and $Y = \{\mathtt{ef}, \mathtt{rm}, \mathtt{of}\}$, see Fig. 1 (left). The symbol `M` denotes Mercedes Smart, `F` Ford Focus, and `J` Jeep Cherokee, while `ef` denotes fuel efficient, `rm` roomy, and `of` off-road ability.

I	ef	rm	of
M	×		
F	×	×	
J		×	×

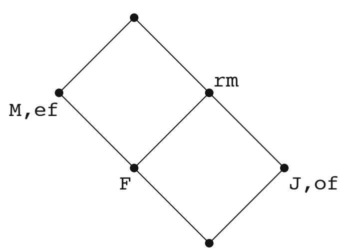


Fig. 1. Formal context (left) and corresponding concept lattice (right)

There exist ten block relations on the concept lattice $\mathcal{B}(X, Y, I)$. Their ordering is shown in Fig. 2, by crosses we denote the pairs of the relation I, by dots there are denoted the pairs of the set-difference $J \setminus I$, where $J \in BR(X, Y, I)$. The corresponding complete tolerances are ordered in the same way as we can see in Fig. 3.

Truth value of the proposition "Ford Focus is a fuel efficient car" is 1 since $\langle \mathtt{F}, \mathtt{ef}\rangle \in I$, while truth value of the proposition "Mercedes Smart is a roomy car" is obviously lesser than 1, particulary equal to h. The value h is the smallest (w.r.t. set inclusion) block relation containing $\langle \mathtt{M}, \mathtt{rm}\rangle$, see Fig. 4 (left). By saying "Mercedes Smart is a roomy car" we actually do not make any difference between

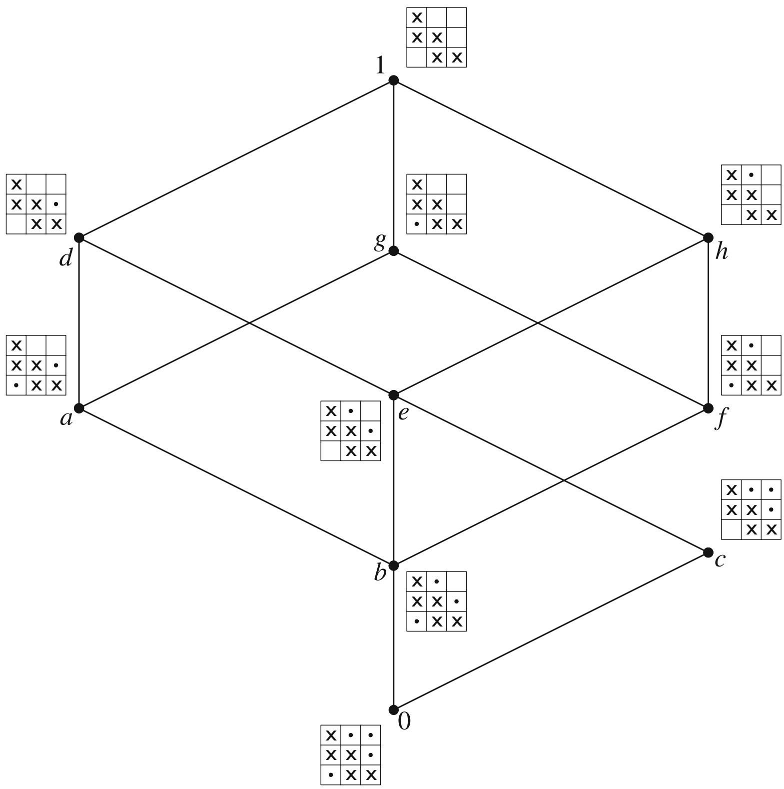


Fig. 2. Tolerance residuated lattice consisting of truth values $0, a, b, c, d, e, f, g, h, 1$ which are expressed using block relations

Mercedes Smart and Ford Focus, i.e. we merge the formal concepts M, ef, and F. As a result the concept lattice $\mathcal{B}(X, Y, I)$ is reduced to the four-element lattice depicted in Fig. 4 (right).

The proposition "Mercedes Smart is a roomy car" becomes fully true when we commit the inaccuracy that Mercedes Smart and Ford Focus are of the same size. This fact corresponds to the evaluation of the proposition "Mercedes Smart is a roomy car" using **L**-structure, where **L** is the tolerance residuated lattice defined on $BR(X, Y, h)$, i.e. it is defined on the set of all block relations on the reduced concept lattice in Fig. 4 (right).

The proposition "every fuel efficient vehicle has an off-road ability" is rather untrue, its truth value is equal to c. Indeed, we evaluate the FCL formula $(\forall \mathbb{x})(r_A(\mathbb{x}) \Rightarrow (r_{\sim}(\mathbb{x}, \mathbb{y}))$, where A is the set of all fuel efficient cars, i.e. $A = \{\mathtt{M}, \mathtt{F}\}$, and **M**-valuation v assigns to the variable $\mathbb{y}$ the attribute of (off-road ability). Therefore, we get:

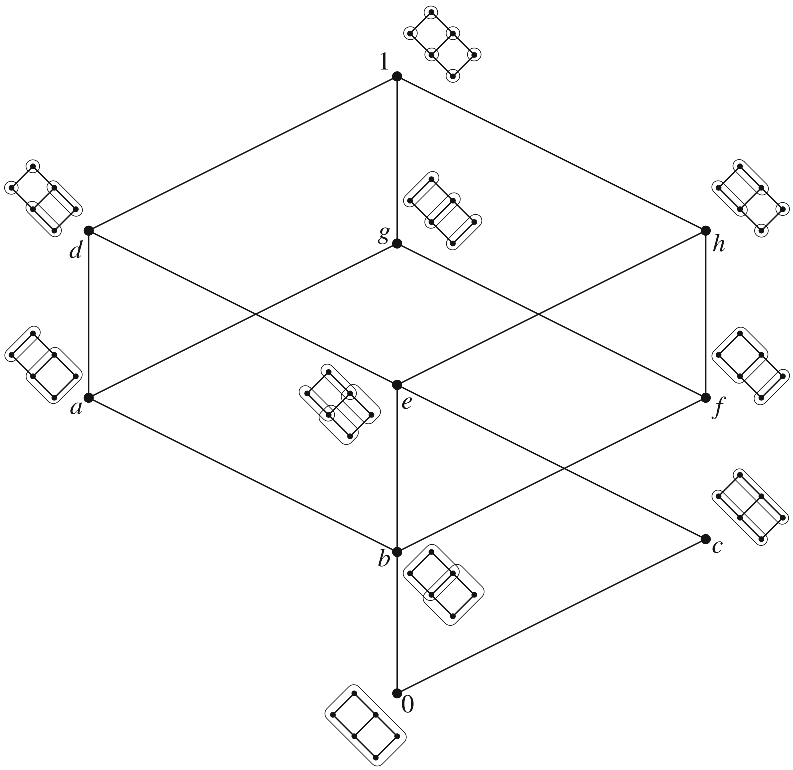


Fig. 3. Tolerance residuated lattice consisting of truth values $0, a, b, c, d, e, f, g, h, 1$ which are expressed using complete tolerances

I	ef	rm	of
M	×	•	
F	×	×	
J		×	×

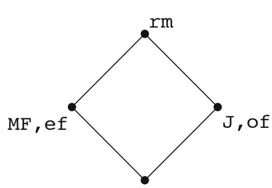


Fig. 4. Block relation h (left) and corresponding reduced concept lattice (right)

$$||(\forall \mathbb{x})(r_A(\mathbb{x}) \Rightarrow (r_\sim(\mathbb{x}, \mathbb{y}))||_{\mathbf{M},v} = \bigwedge_{x \in X} A(x) \rightarrow (x \sim \texttt{of})$$
$$= (g \rightarrow 1) \wedge (1 \rightarrow d) \wedge (1 \rightarrow c) = c.$$

The proposition "every fuel efficient vehicle has an off-road ability" therefore reduces the given concept lattice to the two-element concept lattice, see Fig. 5 (right).

I	ef	rm	of
M	×	•	•
F	×	×	•
J		×	×

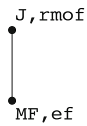


Fig. 5. Block relation c (left) and corresponding reduced concept lattice (right)

5 Conclusion

The main result of the paper consists in showing that a set of all block relations on a concept lattice (or generally, on an arbitrary complete lattice) with an operation of multiplication form a residuated lattice.

Although complete residuated lattices can be defined as non-linear structures (i.e. some elements might be incomparable), in practical applications entirely linear ones are used. The reason is that one may consider incomparable truth values as something artificial. This phenomenon causes the complete residuated lattices to lose their expressing power and theoretical beauty. Using our approach we can easily model the situations where non-linear residuated lattices are naturally employed.

The above mentioned result allows us to consider a logical calculus where such a residuated lattice serves as the scale of truth values. A formula of this calculus, being interpreted by a block relation on the concept lattice, relates certain objects and attributes which where not originally related. Additional information imposed by a formula merges certain formal concepts and therefore, creates a concept lattice consisting of a smaller number of formal concepts.

6 Future Work

In the future we are going to focus on the operations on a tolerance residuated lattice. We are also going to implement the algorithms based on the proposed method and use them on real data sets.

Moreover, it will be definitely useful to generalize the presented results for the fuzzy concept lattices since the problem with a far too big number of formal concepts becomes even more serious in the fuzzy case.

Acknowledgment. Support by project reg. no. CZ.1.07/2.3.00/20.0059 of the European Social Fund in the Czech Republic is gratefully acknowledged.

References

1. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic Publishers, Norwell (2002)
2. Czédli, G.: Factor lattices by tolerances. Acta Sci. Math. 44, 35–42 (1982)

3. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer (1999)
4. Hájek, P.: Metamathematics of Fuzzy Logic (Trends in Logic). Springer (November 2001)
5. Konecny, J., Krupka, M.: Block relations in fuzzy setting. In: Napoli, A., Vychodil, V. (eds.) CLA 2011: Proceedings of the 8th International Conference on Concept Lattices and Their Applications, pp. 115–130 (2011)
6. Krupka, M.: On Factorization of Concept Lattices by Incompatible Tolerances. In: Valtchev, P., Jaschke, R. (eds.) ICFCA 2011. LNCS, vol. 6628, pp. 167–182. Springer, Heidelberg (2011)
7. Ward, M., Dilworth, R.P.: Residuated lattices. Transactions of the American Mathematical Society 45, 335–354 (1939)
8. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, Boston, pp. 445–470 (1982); seminal publication on Formal Concept Analysis
9. Wille, R.: Complete tolerance relations of concept lattices. In: Eigenthaler, G., et al. (eds.) Contributions to General Algebra, vol. 3, pp. 397–415. Hölder-Pichler-Tempsky, Wien (1985)

Basic Level of Concepts in Formal Concept Analysis

Radim Belohlavek and Martin Trnecka

Data Analysis and Modeling Lab (DAMOL)
Department of Computer Science, Palacky University, Olomouc
17. listopadu 12, CZ-77146 Olomouc, Czech Republic
radim.belohlavek@acm.org, martin.trnecka@gmail.com

Abstract. The paper presents a preliminary study on basic level of concepts in the framework of formal concept analysis (FCA). The basic level of concepts is an important phenomenon studied in the psychology of concepts. We argue that this phenomenon may be utilized in FCA for selecting important concepts. In the other direction, we argue that FCA provides a simple framework for studying the phenomenon itself. In this preliminary study, we attempt to draw the attention of researchers in FCA to the basic level of concepts. Furthermore, we propose a formalization of one of the several existing psychological views of the basic level, present experiments, and outline future research in this direction.

Keywords: formal concept analysis, psychology, concepts, basic level.

1 Introduction

1.1 Motivation

It is a well-known fact that, usually, a concept lattice contains a considerable number of formal concepts. When assessed by domain experts, some of the concepts are found more important (or natural) than others. This observation may be utilized for selecting only some formal concepts—the important ones—and filtering out the others. In the literature, one may find several approaches that implement this general idea of selecting important concepts. In this paper, we propose and approach based on a phenomenon well-known in the psychology of concepts, namely the basic level of concepts. In addition to the aim of utilizing the basic level of concepts in FCA, we would like to draw the attention of researchers in FCA to this phenomenon as well as to argue that FCA may be seen as a simple formal framework for studying this phenomenon itself and thus be of interest for the psychologists of concepts (see also [1]).

1.2 Paper Overview

In Sections 1.3 and 1.4, we provide an overview of related work and preliminaries from FCA. In Section 2, we describe the phenomenon of the basic level of

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 28–44, 2012.

concepts and provide main references to the work in the psychology of concepts. An approach to formalize one psychological view of the basic level of concepts is presented in Section 3. Examples and experiments which demonstrate this approach are provided in Section 4. In Section 5 we conclude the paper and outline future research.

1.3 Related Work

The most relevant to this paper is the work on the stability of a formal concept and other indices that seek to assign indices to formal concepts representing the importance of formal concepts [14,15]. In the approaches mentioned in the foregoing paragraph, no other information is used to compute the indices than the one contained in the formal context (or the concept lattice, which is uniquely determined by the formal context). Another idea, initiated in [2] and later developed in [3,4], proposes to utilize the background knowledge to select important concepts. Other approaches to selection of only certain formal concepts have been proposed in [7,8,16,23].

1.4 Preliminaries and Notation

We assume that the reader is familiar with basic notions of formal concept analysis [11]. A formal context is denoted by $\langle X, Y, I\rangle$. Formal concepts of $\langle X, Y, I\rangle$ are denoted by $\langle A, B\rangle$. A pair $\langle A, B\rangle$ consisting of $A \subseteq X$ and $B \subseteq Y$ is called a formal concept if and only if $A^{\uparrow} = B$ and $B^{\downarrow} = A$ where

$$A^{\uparrow} = \{y \in Y \mid \text{ for each } x \in X : \langle x, y\rangle \in I\},$$
$$B^{\downarrow} = \{x \in X \mid \text{ for each } y \in Y : \langle x, y\rangle \in I\}$$

are the set of all attributes common to all objects from A and the set of all objects having all the attributes from B, respectively. The set of all formal concepts of $\langle X, Y, I\rangle$ is denoted by $\mathcal{B}(X, Y, I)$. $\mathcal{B}(X, Y, I)$ equipped with a subconcept-superconcept partial order $\leq$ is the concept lattice of $\langle X, Y, I\rangle$.

2 Basic Level of Concepts in the Psychology of Concepts

The psychology of concepts is a field in cognitive psychology studying human concepts and their cognitive role. The psychology of concepts has been developed systematically since 1950s, but one may find related work much before the 1950s, with [10] being probably the first work using what has become the most common approach to experimental studies of concepts [18]. A comprehensive overview of the main issues involved in the psychology of concepts is given in [19] which is our main source in this paper.

An important place in the studies in the psychology of concepts is occupied by concept hierarchy, i.e. a particular way of organizing concepts using the superordinate-subordinate (superconcept-subconcept) relationship. An extensive experimental work has been done on various issues surrounding concept hierarchies. One of them, central to this paper, is the basic level of concepts.

The basic level of concepts is known from ordinary life. When we see a particular dog, say a German Shepherd named Alex, we say “This is a dog,” rather than “This is a German Shepherd” or “This is a mammal.” That is, to name the object, we use a particular concept that we somehow prefer to other concepts. In this case, we prefer the concept of a dog to the concept of a German Shepherd and to the concept of a mammal. The preferred concepts are called the concepts of the basic level.[1]

In the psychology of concepts, the basic level of concepts is intuitively understood as

> “. . . the most natural, preferred level at which to conceptually carve up the world. The basic level can be seen as a compromise between the accuracy of classification at a maximally general level and the predictive power of a maximally specific level.” [19, p. 210].

[5] is the first work suggesting that people consistently use a kind of a middle level concepts in speach. Since then, several studies in basic level have been conducted. For example, it has been observed [19, Chap. 7] that people across cultures tend to use the same level of concepts when naming plants and animals (the level corresponds to genus). In addition, it has been observed experimentally that people with less knowledge about a domain tend to use more general concepts while people with extensive domain knowledge tend to use more specific concepts as the basic level concepts.

For our purpose, it is important to note that there exist several informal definitions, of the basic level. Some of them have been developed by Eleanor Rosch [20,21]. The one we attempt to use and formalize within FCA in this paper is based on experimental work according to which the basic level is distinguished by the number of common attributes. According to these experiments, when people are asked to list attributes of a concept from the basic level and its superordinate and subordinate concepts, the following pattern may be observed. Only a few attributes are listed for the concepts of a superordinate level, while many more attributes are listed for the concepts of the basic level and the subordinate concepts. Moreover, the number of attributes listed for the subordinate concepts is only slightly larger than the one for the basic level. In addition, the experiments

[1] Related is our experience from explaining FCA. It is quite common that people not familiar with FCA ask questions like: “Given a particular object, what is the (appropriate) formal concept to which the object belongs?” To people familiar with FCA, such question may suggest that the person asking does not understand FCA well yet, because there are usually several formal concepts that cover a particular object, the most specific of them being the object concept generated by the given object. Nevertheless, such question may probably be seen as a manifestation of the phenomenon of the basic level of concepts which is inherently present in human reasoning with concepts: The person is asking for a concept from the basic level.

show that the nature of the attributes plays a role as well. Namely, for superordinate concepts, mostly functional attributes were listed, such as "keeps you warm"; for basic level, people listed nouns and adjectives as properties, such as buttons, belt loops; for subordinate concepts, additional adjectives were listed, such as those referring to color. In [21], the authors concluded what may be considered a generalized view of the above-described observation: The objects of the basic level concepts are similar to each other, the objects of the superordinate concepts are significantly less similar, while the objects of the subordinate concepts are only slightly more similar. This way of characterization of the basic level is utilized in Section 3.

Due to the lack of space in this paper, we do not comment in detail on the other psychological views of the basic level. Let us only note that the literature contains studies of several interesting aspects of the basic level, including the role of the basic level in cognitive processes such as the speed of classification or predictive capability. Let us also note that many studies were performed within specific domains and that the basic level was described using domain-specific criteria such as types of movements or visual characteristics of the objects.

3 An Approach to Basic Level in FCA

In this section, we propose an approach to identifying concepts of a basic level in a concept lattice inspired by Rosch's definition of basic level concepts described in Section 2. Informally, we call a cohesion of a concept a measure of whether the objects to which that concept applies are pairwise similar. According to Rosch's definition, a formal concept $\langle A, B\rangle$ belongs to the basic level if it satisfies the following properties:

(BL1) $\langle A, B\rangle$ has a high cohesion.
(BL2) $\langle A, B\rangle$ has a significantly larger cohesion than its upper neighbors.
(BL3) $\langle A, B\rangle$ has only a slightly smaller cohesion than its lower neighbors.

Note that the upper neighbors of $\langle A, B\rangle$ are the concepts that are more general than $\langle A, B\rangle$ and are directly above $\langle A, B\rangle$ in the hierarchy of concepts. The lower neighbors are defined analogously. The sets of all upper and lower neighbors of c (i.e. elements covering c and covered by c) is denoted by $\mathcal{UN}(c)$ and $\mathcal{LN}(c)$, respectively. That is,

- $\mathcal{UN}(c) = \{d \in \mathcal{B}(X, Y, I) \mid c < d$ and there is no d' for which $c < d' < d\}$,
- $\mathcal{LN}(c) = \{d \in \mathcal{B}(X, Y, I) \mid c > d$ and there is no d' for which $c > d' > d\}$.

Furthermore, we use the following notation and its variants:

- $sim(x_1, x_2)$ denotes the degree (or index) of similarity of objects x_1 and x_2.
- $coh(c)$ denotes the degree (or index) of cohesion of formal concept c.

Similarity of objects x_1 and x_2 can naturally be assessed by similarity of their corresponding intents, i.e. by similarity of $\{x_1\}^\uparrow$ and $\{x_2\}^\uparrow$. That is, given an appropriate similarity measure sim_Y with $sim_Y(B_1, B_2)$ being a similarity degree of sets $B_1, B_2 \subseteq Y$ of attributes, we may put

$$sim(x_1, x_2) = sim_Y(\{x_1\}^\uparrow, \{x_2\}^\uparrow). \tag{1}$$

In our experiments, we used the following well-known functions for sim_Y:

$$sim_{\mathrm{SMC}}(B_1, B_2) = \frac{|B_1 \cap B_2| + |Y - (B_1 \cup B_2)|}{|Y|}, \tag{2}$$

$$sim_{\mathrm{J}}(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}. \tag{3}$$

Note that (2) is the simple matching coefficient and that (3) is the Jaccard index [22]. sim_{SMC} is the number of attributes on which B_1 and B_2 agree (either $y \in B_1$ and $y \in B_2$, or $y \notin B_1$ and $y \notin B_2$) divided by the number of all attributes. sim_{J} is the number of attributes that belong to both B_1 and B_2 divided by the number of all attributes that belong to B_1 or B_2. That is, while sim_{SMC} treats both presence and non-presence of attributes symmetrically, sim_{J} disregards non-presence. This is the main conceptual difference between sim_{SMC} and sim_{J}.

A simple approach to measure the cohesion $coh(A, B)$ for a formal concept $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ is the following:

$$coh^{\varnothing}(A, B) = \frac{\sum_{\{x_1, x_2\} \subseteq A, x_1 \neq x_2} sim(x_1, x_2)}{|A| \cdot (|A| - 1)/2}. \tag{4}$$

That is, $coh^{\varnothing}(A, B)$ is the average similarity of two objects covered by the formal concept $\langle A, B\rangle$. Alternatively, we might put

$$coh^{\mathrm{m}}(A, B) = \min_{x_1, x_2 \in A} sim(x_1, x_2), \tag{5}$$

in which case the cohesion is the least similarity degree of any two objects covered by $\langle A, B\rangle$.

We are now going to assign to every formal concept $\langle A, B\rangle$ of $\langle X, Y, I\rangle$ a degree $BL(A, B)$ to which $\langle A, B\rangle$ is a concept from the basic level. Given that the concepts from the basic level need to satisfy conditions (BL1), (BL2), and (BL3), it seems natural to construe $BL(A, B)$ as the degree to which a conjunction of the three propositions, (BL1), (BL2), and (BL3), is true. That is, to put

$$BL(A, B) = \mathcal{C}(\alpha_1(A, B), \alpha_2(A, B), \alpha_3(A, B)), \tag{6}$$

where

- $\alpha_i(A, B)$ is the degree to which condition (BLi) is satisfied, $i = 1, 2, 3$,
- $\mathcal{C}$ is a "conjunctive" aggregation function; that is, if propositions φ_1, φ_2, and φ_3 have truth degrees α_1, α_2, and α_3, respectively, then the conjunction φ_1 *and* φ_2 *and* φ_3 has the truth degree $\mathcal{C}(\alpha_1, \alpha_2, \alpha_3)$.

A simple form of $\mathcal{C}$ is obtained by taking a t-norm [13] $\otimes$ and to put

$$\mathcal{C}(\alpha_1, \alpha_2, \alpha_3) = \alpha_1 \otimes \alpha_2 \otimes \alpha_3.$$

We assume that the truth degrees are numbers in $[0, 1]$ and use the product (Goguen) t-norm, given by $a \otimes b = a \cdot b$, in the experiments.

For $\alpha_1(A, B)$, $\alpha_2(A, B)$, and $\alpha_3(A, B)$, the following definitions seem natural choices (coh^* denotes $coh^{\varnothing}$ or coh^{m}, see (4) and (5)):

$$\alpha_1^*(A, B) = coh^*(A, B), \tag{7}$$

$$\alpha_2^{\varnothing *}(A, B) = 1 - \frac{\sum_{c \in \mathcal{UN}(A,B)} coh^*(c)/coh^*(A, B)}{|\mathcal{UN}(A, B)|}, \tag{8}$$

$$\alpha_2^{\mathrm{m}*}(A, B) = 1 - \max_{c \in \mathcal{UN}(A,B)} coh^*(c)/coh^*(A, B), \tag{9}$$

$$\alpha_3^{\varnothing *}(A, B) = \frac{\sum_{c \in \mathcal{LN}(A,B)} coh^*(A, B)/coh^*(c)}{|\mathcal{LN}(A, B)|}, \tag{10}$$

$$\alpha_3^{\mathrm{m}*}(A, B) = \min_{c \in \mathcal{LN}(A,B)} coh^*(A, B)/coh^*(c). \tag{11}$$

Remark 1. Let us explain the meaning of formulas (7)–(11). The formulas are designed so that the values of $\alpha_1(A, B)$, $\alpha_2(A, B)$, and $\alpha_3(A, B)$ (and their variants given by the superscripts) may naturally be interpreted as the truth degrees to which the propositions in (BL1), (BL2), and (BL3) are true.

Ad (7): Clearly, (7) may be interpreted as the truth degree of (BL1).

Before discussing the other formulas, let us observe that if $coh^*(c_1) \leq coh^*(c_2)$, then $\frac{coh^*(c_1)}{coh^*(c_2)} \in [0, 1]$ may be interpreted as the truth degree of "$coh^*(c_1)$ is only slightly smaller than $coh^*(c_2)$", and hence $1 - \frac{coh^*(c_1)}{coh^*(c_2)} \in [0, 1]$ may be interpreted as the truth degree of proposition "$coh^*(c_1)$ is significantly smaller than $coh^*(c_2)$". Assume therefore that in the factions $\frac{coh^*(c_1)}{coh^*(c_2)}$ in (7)–(11), we always have $coh^*(c_1) \leq coh^*(c_2)$.

Ad (8): Since

$$\alpha_2^{\varnothing *}(A, B) = \frac{\sum_{c \in \mathcal{UN}(A,B)} 1 - coh^*(c)/coh^*(A, B)}{|\mathcal{UN}(A, B)|},$$

it follows that $\alpha_2^{\varnothing *}(A, B)$ may be interpreted as the truth degree of "on average, the upper neighbors of $\langle A, B \rangle$ have a significantly smaller cohesion than $\langle A, B \rangle$", which is one possible meaning of (BL2).

Ad (9): Since

$$\alpha_2^{\mathrm{m}*}(A, B) = \min_{c \in \mathcal{UN}(A,B)} 1 - coh^*(c)/coh^*(A, B),$$

$\alpha_2^{\mathrm{m}*}(A, B)$ may be interpreted as the truth degree of "each upper neighbor of $\langle A, B \rangle$ has a significantly smaller cohesion than $\langle A, B \rangle$", which is another possible reading of (BL2).

Ad (10): For the same reasons as in the case of (8), $\alpha_3^{\varnothing *}(A, B)$ may be interpreted as the truth degree of "on average, $\langle A, B\rangle$ has only lightly smaller cohesion than its lower neighbors", which is one possible meaning of (BL3).

Ad (11): For the same reasons as in the case of (10), $\alpha_3^{\mathrm{m}*}(A, B)$ may be interpreted as the truth degree of "$\langle A, B\rangle$ has only a slightly small than cohesion than each of its lower neighbors", which is another possible reading of (BL3).

The interpretation described in (7)–(11) is correct if, as was mentioned in the above remark, in the factions $\frac{coh^*(c_1)}{coh^*(c_2)}$ in (7)–(11), we have $coh^*(c_1) \leq coh^*(c_2)$.

This is the case of coh^{m}, as the following lemma shows.

Lemma 1. *If* $\langle A_1, B_1\rangle \leq \langle A_2, B_2\rangle$ *then* $coh^{\mathrm{m}}(A_2, B_2) \leq coh^{\mathrm{m}}(A_1, B_1)$.

Proof. Immediate to verify. □

However, for $coh^{\varnothing}$ such property no longer holds. Namely, the cohesion of an upper neighbor of a concept may be greater than that of the concept itself as the following example shows.

Example 1. Consider the formal context in Table 1. One may check that for the formal concepts $\langle A_1, B_1\rangle = \{\{x_1, x_2\}, \{y_1, y_2\}\}$ and $\langle A_2, B_2\rangle = \{\{x_1, x_2, x_3\}, \{y_1\}\}$ we have $\langle A_2, B_2\rangle \in \mathcal{UN}(A_1, B_1)$ and yet:

Table 1. Formal context from Example 1

	y_1	y_2	y_3	y_4	y_5
x_1	×	×	×		×
x_2	×	×		×	
x_3	×		×	×	×

$$coh^{\varnothing}(A_1, B_1) = \frac{sim(x_1, x_2)}{1} = \frac{2}{5} < \frac{\frac{2}{5} + \frac{3}{5} + \frac{2}{5}}{3} = \frac{sim(x_1, x_2) + sim(x_1, x_3) + sim(x_2, x_3)}{3} = coh^{\varnothing}(A_2, B_2),$$

for both $sim = sim_{\mathrm{SMC}}$ and $sim = sim_{\mathrm{J}}$.

We propose the following solution to this problem. Instead of considering $\mathcal{UN}(A, B)$, i.e. all the upper neighbors of $\langle A_1, B_1\rangle$, we consider only

$$\mathcal{UN}^{\leq}(A, B) = \{c \in \mathcal{UN}(A, B) \mid coh^{\varnothing}(c) \leq coh^{\varnothing}(A, B)\},$$

i.e. only the upper neighbors with a smaller cohesion in (8) and (9). In addition, it seems natural to disregard $\langle A, B\rangle$ as a candidate for a basic level concept if the number of "wrong upper neighbors" is relatively large, i.e. if $\frac{|\mathcal{UN}^{\leq}(A,B)|}{|\mathcal{UN}(A,B)|} < \theta$

for some parameter θ. θ itself may be subject to experiments ($\theta = 1$ means that we require all the upper neighbors to have cohesion $\leq$ the cohesion of $\langle A, B\rangle$). Likewise, instead of considering $\mathcal{LN}(A, B)$, we consider only

$$\mathcal{LN}^{\geq}(A, B) = \{c \in \mathcal{LN}(A, B) \mid coh^{\varnothing}(c) \geq coh^{\varnothing}(A, B)\}$$

in (10) and (11), and a similar condition for the number of "wrong lower neighbors" given by θ. Therefore, we get the following formulas ($||\varphi||$ denoted the truth degree of condition φ):

$$\alpha_1^*(A, B) = coh^*(A, B),$$

$$\alpha_2^{\varnothing *}(A, B) = [1 - \frac{\sum_{c \in \mathcal{UN}^{\leq}(A,B)} coh^*(c)/coh^*(A, B)}{|\mathcal{UN}^{\leq}(A, B)|}] \cdot || \frac{|\mathcal{UN}^{\leq}(A, B)|}{|\mathcal{UN}(A, B)|} \geq \theta||,$$

$$\alpha_2^{\mathrm{m}*}(A, B) = [1 - \max_{c \in \mathcal{UN}^{\leq}(A,B)} coh^*(c)/coh^*(A, B)] \cdot || \frac{|\mathcal{UN}^{\leq}(A, B)|}{|\mathcal{UN}(A, B)|} \geq \theta||,$$

$$\alpha_3^{\varnothing *}(A, B) = [\frac{\sum_{c \in \mathcal{LN}^{\geq}(A,B)} coh^*(A, B)/coh^*(c)}{|\mathcal{LN}^{\geq}(A, B)|}] \cdot || \frac{|\mathcal{LN}^{\geq}(A, B)|}{|\mathcal{LN}(A, B)|} \geq \theta||,$$

$$\alpha_3^{\mathrm{m}*}(A, B) = [\min_{c \in \mathcal{LN}^{\geq}(A,B)} coh^*(A, B)/coh^*(c)] \cdot || \frac{|\mathcal{LN}^{\geq}(A, B)|}{|\mathcal{LN}(A, B)|} \geq \theta||.$$

Lemma 2. *$\alpha_2^{\varnothing *}(A, B) \geq \alpha_2^{\mathrm{m}*}(A, B)$ and $\alpha_3^{\varnothing *}(A, B) \geq \alpha_3^{\mathrm{m}*}(A, B)$.*

Proof. Immediate to verify.

Now, according to (6) every concept $\langle A, B\rangle$ gets assigned a degree $BL(A, B)$ to which $\langle A, B\rangle$ may be considered as a concept of a basic level. The concepts with high degrees are therefore considered as important ones. The degrees may be used to rank the concepts accordingly, i.e. to sort them from those with the highest basic level degrees to the lowest.

4 Experiments

We performed several experiments with the above method to identify basic level concepts. Our main aim was to see if the method is able to identify concepts that humans would naturally use as basic level concepts. Clearly, the subjectivity factor plays a significant role. We therefore selected datasets describing commonly known objects, for which most people would probably agree on the basic level concepts, or at least agree on whether a given concept can be regarded as a basic level concept.

We were not checking the results of our method for a given dataset against a psychological experiment involving a group of respondents telling their basic level concepts for the dataset. This important step, particularly from the psychology point of view, is left for future. Instead, we tried to see whether our method identifies basic level concepts in a reasonable way based on our intuition.

For every dataset $\langle X, Y, I\rangle$, we observed the basic level degrees of all concepts of the concept lattice $\mathcal{B}(X, Y, I)$. We report the results for the following combinations $BL_{\rm s}^{\rm c,a}(A, B)$: s is SMC or J and indicates whether $sim_{\rm SMC}$ or $sim_{\rm J}$ was used; c is $\varnothing$ or m and indicates whether $coh^{\varnothing}$ or $coh^{\rm m}$ was used; a is $\varnothing$ or m and indicates whether $\alpha_2^{\varnothing *}$ and $\alpha_3^{\varnothing *}$, or $\alpha_2^{\rm m*}$ and $\alpha_3^{\rm m*}$ was used. We report the results for $\theta = 1$. Note that we have:

Lemma 3. $BL_{\rm s}^{\rm c,\varnothing}(A, B) \geq BL_{\rm s}^{\rm c,m}(A, B)$ *for any* s *and* c*, provided* $\mathcal{C}$ *is isotone.*

Proof. Directly from Lemma 2 and the fact that $\mathcal{C}$ in (6) is isotone.

4.1 Experiment 1

The dataset in Table 2 contains selected sports and their attributes. There are the following formal concepts in this dataset (for convenience, we list them in the form $c_i = \langle A, B\rangle$ where A and B are the extent and intent of c_i):
$c_1 = \langle\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}, \{\}\rangle$,
$c_2 = \langle\{1, 2, 4, 11, 13, 14, 15, 20\}, \{10\}\rangle$,
$c_3 = \langle\{3, 5, 6, 7, 8, 9, 10, 12, 16, 17, 18, 19\}, \{9\}\rangle$, $c_4 = \langle\{3, 4, 14, 16\}, \{8\}\rangle$,
$c_5 = \langle\{3, 16\}, \{8, 9\}\rangle$, $c_6 = \langle\{1, 2, 3, 4, 11, 13, 14, 15, 17, 19\}, \{5\}\rangle$,
$c_7 = \langle\{1, 2, 4, 11, 13, 14, 15\}, \{5, 10\}\rangle$, $c_8 = \langle\{3, 17, 19\}, \{5, 9\}\rangle$,
$c_9 = \langle\{3, 4, 14\}, \{5, 8\}\rangle$, $c_{10} = \langle\{4, 14\}, \{5, 8, 10\}\rangle$,
$c_{11} = \langle\{5, 6, 7, 8, 9, 10, 12, 16, 18, 20\}, \{4\}\rangle$,
$c_{12} = \langle\{5, 6, 7, 8, 9, 10, 12, 16, 18\}, \{4, 9\}\rangle$, $c_{13} = \langle\{5, 6, 7, 8, 9, 18\}, \{4, 7, 9\}\rangle$,
$c_{14} = \langle\{5, 6, 7, 8, 18\}, \{4, 6, 7, 9\}\rangle$, $c_{15} = \langle\{4, 16, 17, 18, 19, 20\}, \{3\}\rangle$,
$c_{16} = \langle\{4, 20\}, \{3, 10\}\rangle$, $c_{17} = \langle\{16, 17, 18, 19\}, \{3, 9\}\rangle$, $c_{18} = \langle\{4, 16\}, \{3, 8\}\rangle$,
$c_{19} = \langle\{4, 17, 19\}, \{3, 5\}\rangle$, $c_{20} = \langle\{17, 19\}, \{3, 5, 9\}\rangle$, $c_{21} = \langle\{16, 18, 20\}, \{3, 4\}\rangle$,
$c_{22} = \langle\{20\}, \{3, 4, 10\}\rangle$, $c_{23} = \langle\{16, 18\}, \{3, 4, 9\}\rangle$, $c_{24} = \langle\{16\}, \{3, 4, 8, 9\}\rangle$,
$c_{25} = \langle\{18\}, \{3, 4, 6, 7, 9\}\rangle$, $c_{26} = \langle\{9, 10, 11, 12, 13, 14, 15\}, \{2\}\rangle$,
$c_{27} = \langle\{11, 13, 14, 15\}, \{2, 5, 10\}\rangle$, $c_{28} = \langle\{14\}, \{2, 5, 8, 10\}\rangle$,
$c_{29} = \langle\{9, 10, 12\}, \{2, 4, 9\}\rangle$, $c_{30} = \langle\{9\}, \{2, 4, 7, 9\}\rangle$,
$c_{31} = \langle\{1, 2, 3, 4, 5, 6, 7, 8\}, \{1\}\rangle$, $c_{32} = \langle\{3, 5, 6, 7, 8\}, \{1, 9\}\rangle$,
$c_{33} = \langle\{1, 2, 3, 4\}, \{1, 5\}\rangle$, $c_{34} = \langle\{1, 2, 4\}, \{1, 5, 10\}\rangle$, $c_{35} = \langle\{3, 4\}, \{1, 5, 8\}\rangle$,
$c_{36} = \langle\{3\}, \{1, 5, 8, 9\}\rangle$, $c_{37} = \langle\{5, 6, 7, 8\}, \{1, 4, 6, 7, 9\}\rangle$,
$c_{38} = \langle\{4\}, \{1, 3, 5, 8, 10\}\rangle$, $c_{39} = \langle\{\}, \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\rangle$
Table 3 contains concepts c_1–c_{39} and their basic level degrees. The corresponding concept lattice is depicted using a reduced labeling in Fig. 1.

Note first that in accordance with Lemma 3, the basic level degrees in column $BL_{\rm SMC}^{\varnothing\varnothing}$ are greater than or equal to those in column $BL_{\rm SMC}^{\varnothing \rm m}$; and that the same is true when comparing columns $BL_{\rm SMC}^{\rm m\varnothing}$ to $BL_{\rm SMC}^{\rm mm}$, $BL_{\rm J}^{\varnothing\varnothing}$ to $BL_{\rm J}^{\varnothing \rm m}$, and $BL_{\rm J}^{\rm m\varnothing}$ to $BL_{\rm J}^{\rm mm}$.

Second, note that it seems not to matter very much whether $\alpha_2^{\varnothing *}$ and $\alpha_3^{\varnothing *}$, or $\alpha_2^{\rm m*}$ and $\alpha_3^{\rm m*}$ is used. This observation needs to be explored further in future. On the other hand, it matters significantly whether $coh^{\varnothing}$ or $coh^{\rm m}$ is used. According to our intuition and the results of this and other experiments we performed, we

Table 2. Sports and their attributes

		on land	on ice	in water	collective sport	individual sport	using ball	needs opponent	multiple disciplines	points	time
		1	2	3	4	5	6	7	8	9	10
Run	1	×				×					×
Orienteering	2	×				×					×
Gymnastics	3	×				×			×	×	
Triathlon	4	×		×		×			×		×
Football	5	×			×		×	×		×	
Inline Hockey	6	×			×		×	×		×	
Tennis	7	×			×		×	×		×	
Baseball	8	×			×		×	×		×	
Ice Hockey	9		×		×			×		×	
Curling	10		×		×					×	
Cross-country Skiing	11		×			×					×
Synchronized Skating	12		×		×					×	
Alpine Skiing	13		×			×					×
Biathlon	14		×			×			×		×
Speed Skating	15		×			×					×
Synchronized Swimming	16			×	×				×	×	
Diving	17			×		×				×	
Water Polo	18			×	×		×	×		×	
Underwater Diving	19			×		×				×	
Rowing	20			×	×						×

hypothesize that $coh^{\varnothing}$ is better to use than coh^{m}. Again, a more detailed study is needed to support this claim.

Third, and most importantly for the purpose of our paper, let us consider the concepts that have been indicated as basic level concepts by the method. Due to lack of space, we consider the selection by $BL^{\varnothing\varnothing}_{\mathrm{SMC}}$ only. The concepts with a non-zero degree, depicted by square nodes in Fig. 1, sorted in a descending way are c_{29} (which can be verbally described as "winter collective sports"; encompassing Ice Hockey, Curling, Synchronized Skating), c_{27} ("individual winter sports"; Cross-country Skiing, Alpine Skiing, Biathlon, Speed Skating), c_{32} ("land sports evaluated by points"), c_{26} ("winter sports"), c_{13} ("collective sports with opponent"), c_8 ("individual sports"), c_{14} ("ball games"), c_{34} ("land sports evaluated by time"), c_{19} ("individual water sports"), c_{31} ("land sports"). Arguably, all of them are likely to be considered natural, basic level concepts. On the other hand,

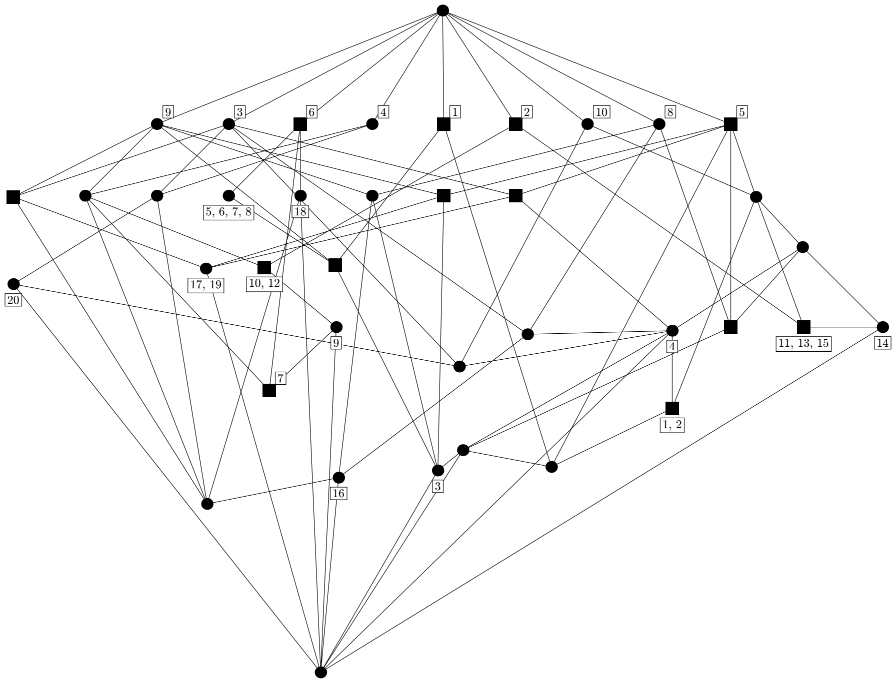


Fig. 1. Concept lattice of Table 2

among the concepts not selected for basic level are, for example, c_{35} ("individual land sports with multiple disciplines", encompassing Gymnastics and Triathlon), c_{30} ("collective winter sports with opponent evaluated by points" consisting of Ice Hockey), c_{28} ("individual winter sports with multiple disciplines evaluated by time" consisting of Biathlon), or c_{18} ("sports performed in water with multiple disciplines" encompassing Triathlon and Synchronized Swimming). These concepts are not likely to be regarded as basic level concepts. From this and other experiments we conclude that the method we present in this paper tends to select natural concepts likely to be considered basic level concepts and not to select clear non-basic level concepts. For the "borderline cases", however, it happens that seemingly natural concepts are not selected and, vice versa, that the selected concepts do not seem likely to be considered basic level concepts. It has to be noted that this study presents an initial study in the presented problem. As such, we consider the method promising and giving good results already at this stage.

An important observation, that we made in several experiments, worth noting here is that which concepts are considered as basic level concepts very much depends on the dataset and the selected attributes in particular. Typically, a

Table 3. Basic level degrees of the concepts of Table 2

intent of concept $\langle A, B\rangle$										basic level degree of $\langle A, B\rangle$							
on land	on ice	in water	collective sport	individual sport	using ball	needs opponent	multiple disciplines	points	time	$BL^{\emptyset\emptyset}_{\mathrm{SMC}}$	$BL^{\emptyset\mathrm{m}}_{\mathrm{SMC}}$	$BL^{\mathrm{m}\emptyset}_{\mathrm{SMC}}$	$BL^{\mathrm{mm}}_{\mathrm{SMC}}$	$BL^{\emptyset\emptyset}_{\mathrm{J}}$	$BL^{\emptyset\mathrm{m}}_{\mathrm{J}}$	$BL^{\mathrm{m}\emptyset}_{\mathrm{J}}$	$BL^{\mathrm{mm}}_{\mathrm{J}}$
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	1	0	0	0.33	0.33	0	0	0.08	0.08
0	0	0	0	0	0	0	0	1	0	0	0	0.11	0.09	0	0	0.05	0.04
0	0	0	0	0	0	0	1	0	0	0	0	0.21	0.20	0	0	0.06	0.06
0	0	0	0	0	0	0	1	1	0	0	0	0.15	0.12	0	0	0.07	0.06
0	0	0	0	1	0	0	0	0	0	0	0	0.31	0.29	0.12	0.11	0.08	0.07
0	0	0	0	1	0	0	0	0	1	0	0	0.08	0.07	0	0	0.09	0.07
0	0	0	0	1	0	0	0	1	0	0.10	0.07	0.21	0.14	0.10	0.10	0.10	0.09
0	0	0	0	1	0	0	1	0	0	0	0	0.13	0.09	0.05	0.01	0.12	0.11
0	0	0	0	1	0	0	1	0	1	0	0	0.07	0.07	0	0	0.08	0.08
0	0	0	1	0	0	0	0	0	0	0	0	0.22	0.20	0	0	0.07	0.06
0	0	0	1	0	0	0	0	1	0	0	0	0.10	0.06	0	0	0.08	0.05
0	0	0	1	0	0	1	0	1	0	0.11	0.10	0.16	0.14	0.16	0.15	0.13	0.11
0	0	0	1	0	1	1	0	1	0	0.07	0.07	0.08	0.08	0.11	0.11	0.11	0.11
0	0	1	0	0	0	0	0	0	0	0	0	0.03	0.03	0	0	0.04	0.04
0	0	1	0	0	0	0	0	0	1	0	0	0.15	0.06	0	0	0.06	0.06
0	0	1	0	0	0	0	0	1	0	0	0	0.26	0.18	0.08	0.04	0.11	0.07
0	0	1	0	0	0	0	1	0	0	0	0	0.10	0.05	0	0	0.05	0.04
0	0	1	0	1	0	0	0	0	0	0.05	0.02	0.15	0.06	0.09	0.07	0.06	0.06
0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0.22	0.12	0	0	0.10	0.06
0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	0	0	0	0.09	0.07	0	0	0.09	0.08
0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0.14	0.14	0.13	0.13	0.11	0.10	0.03	0.03
0	1	0	0	1	0	0	0	0	1	0.18	0.13	0.36	0.27	0.29	0.23	0.38	0.31
0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0.20	0.19	0.41	0.36	0.27	0.24	0.40	0.35
0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0.05	0.05	0.03	0.03	0.11	0.10	0.04	0.03
1	0	0	0	0	0	0	0	1	0	0.15	0.13	0.13	0.10	0.21	0.20	0.05	0.05
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0.19	0.16
1	0	0	0	1	0	0	0	0	1	0.06	0.05	0.12	0.08	0.10	0.09	0.14	0.12
1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0.07	0.05
1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Table 4. Animals and their attributes

	gives milk	gives meat	gives fur	gives eggs	warmblooded	coldblooded	walks on 2 legs	walks on 4 legs	lives in water	lives on land	flies	usually has name	omnivore	carnivore	herbivore	used to pull
German Shepherd					×			×		×		×		×		
Labrador Retrieval					×			×		×		×		×		
European Shorthair Cat					×			×		×		×		×		
Persian Cat					×			×		×		×		×		
Sheep	×	×	×		×			×		×					×	
Ouessan Sheep	×	×	×		×			×		×					×	
Domestic Goat	×	×			×			×		×					×	
Mountain Goat		×			×			×		×					×	
Chicken		×		×	×		×			×	×		×			
Czech Gold Chicken		×		×	×		×			×	×		×			
Pig		×			×			×		×			×			
Indochinese Warty Pig		×			×			×		×			×			
Domestic Horse		×			×			×		×		×			×	×
Pony		×			×			×		×		×			×	×
Donkey	×	×			×			×		×		×			×	×
Rabbit		×			×			×		×					×	
Coypu		×			×			×	×	×					×	
Kosovo Rooster		×		×	×		×			×	×		×			
Domestic Goose		×	×	×	×		×		×	×	×				×	
Lesser White-fronted Goose	×	×	×	×		×		×	×	×				×		
Wild Mallard Duck		×	×		×		×		×	×	×		×			
Domestic Duck		×	×	×	×		×		×	×	×		×			
Domestic Pigeon		×			×		×			×	×				×	
Common Ostrich		×	×	×	×		×			×			×			
Domestic Turkey		×		×	×		×			×					×	
Budgerigar					×		×			×	×	×			×	
Carp		×				×			×				×			
Trout		×				×			×					×		

Table 5. Basic level degrees of the concepts of Table 4

intent of concept $\langle A, B\rangle$																basic level degree of $\langle A, B\rangle$							
gives milk	gives meat	gives fur	gives eggs	warmblooded	coldblooded	walks on 2 legs	walks on 4 legs	lives in water	lives on land	flies	usually has name	omnivore	carnivore	herbivore	used to pull	$BL^{\emptyset\emptyset}_{\mathrm{SMC}}$	$BL^{\emptyset\mathrm{m}}_{\mathrm{SMC}}$	$BL^{\mathrm{m}\emptyset}_{\mathrm{SMC}}$	$BL^{\mathrm{mm}}_{\mathrm{SMC}}$	$BL^{\emptyset\emptyset}_{\mathrm{J}}$	$BL^{\emptyset\mathrm{m}}_{\mathrm{J}}$	$BL^{\mathrm{m}\emptyset}_{\mathrm{J}}$	$BL^{\mathrm{mm}}_{\mathrm{J}}$
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	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0.03	0.03	0.05	0.05	0.06	0.06	0.07	0.07
1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0.04
0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0.02	0.02	0.14	0.13	0.07	0.07	0.08	0.07
0	1	1	1	1	0	1	0	0	1	0	0	0	0	0	0	0.01	0	0	0	0.03	0.01	0	0
0	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0.05	0.03	0.09	0.05	0.07	0.04	0.13	0.08
0	1	1	1	1	0	1	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0.08	0.02
0	1	1	0	1	0	1	0	0	1	0	0	0	0	0	0	0.07	0.03	0	0	0.11	0.06	0	0
0	1	1	0	1	0	1	0	1	1	1	0	0	0	0	0	0.06	0.03	0.13	0.05	0.10	0.06	0.18	0.08
0	1	1	0	1	0	1	0	1	1	1	0	1	0	0	0	0.09	0.04	0.18	0.12	0.14	0.06	0.27	0.17
0	1	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0.10	0.08	0.03	0.02	0.06	0.06
0	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0.02	0.01	0.05	0.05	0.03	0.03	0.08	0.07
0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0.02	0.02	0.05	0.05	0.05	0.04	0.08	0.08
0	1	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0.04	0.02	0	0	0.07	0.05	0.08	0.02
0	1	0	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0.16	0.08
0	1	0	1	1	0	1	0	0	1	0	0	0	0	1	0	0.03	0.02	0	0	0.04	0.04	0	0
0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0.03	0.02	0	0	0.04	0.04	0.05	0.02
0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0.07	0.02	0.16	0.05	0.10	0.04	0.10	0.06
0	1	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0.02	0.01	0.08	0.05	0.04	0.03	0.09	0.07
0	1	0	0	1	0	1	0	0	1	1	0	1	0	0	0	0.04	0.01	0	0	0.06	0.03	0	0
0	1	0	0	1	0	1	0	0	1	1	0	0	0	1	0	0.04	0.02	0	0	0.05	0.04	0	0
0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0.05	0.03	0	0	0.08	0.04	0	0
0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0.08	0.04	0.18	0.04	0.10	0.06	0.11	0.05
0	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0.11	0.09	0.23	0.18	0.20	0.18	0.36	0.28
0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0.06	0.02	0.15	0.05	0.07	0.04	0.12	0.06
0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0.10	0.07	0.14	0.09	0.17	0.12	0.10	0.08
0	1	0	0	1	0	0	0	1	1	0	0	0	0	1	0	0.01	0	0.13	0.09	0.03	0	0.08	0.07
0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0.07	0.04	0.13	0.04	0.09	0.06	0.09	0.07
0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0.04	0.02	0.03	0
0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0.19	0.19	0.38	0.38	0.09	0.09	0.25	0.25
0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.09	0.06	0	0	0.04	0.03
0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0.03	0.02
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.09	0.07
0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0.02	0.02	0	0	0.04	0.03	0.09	0.07
0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0.11	0.11	0	0	0.18	0.17	0.10	0.08
0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0.01	0.01	0	0	0.02	0.02	0.05	0.05
0	0	0	0	1	0	1	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0.01	0.01	0	0
0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0.05	0.01	0	0
0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0.11	0.11	0.28	0.28	0.11	0.10	0.13	0.13
0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	0.04	0.03	0.04	0.04	0.07	0.05	0.04	0.03
0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0.12	0.11	0	0	0.14	0.14	0.19	0.18
0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0.03	0.01	0	0	0.06	0.04	0.03	0.01
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0.07	0.07	0	0	0.10	0.09	0.10	0.08
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0.14	0.14	0.14	0.14	0.15	0.15	0.02	0.02

human expert tends to take into account other information (not only the attributes present in the dataset) to assess which concepts are the basic level concepts. This makes it difficult to objectively assess the quality of a particular basic level function. Rather than instructing the human expert "You must use only the attributes and objects in the dataset to tell which concepts are basic level concepts for our data", which the expert is likely to inadvertently disobey anyway, we learned that the dataset must be balanced in that it must contain the main relevant attributes people would naturally take into account when telling the basic level concepts. From this point of view, it is certainly desirable to do more experiments with the sport dataset and its variants, such as the one that would distinguish attribute "points" as to whether points are assigned by jury or whether this means that points are scored such as in Ice Hockey, which indeed impacts the basic level.

4.2 Experiment 2

The dataset in Table 4 contains selected animals and their attributes. Table 5 contains the concepts of this dataset and their basic level degrees. Due to lack of space, we leave the interpretation of these results to the reader.

5 Conclusions and Future Research

We proposed a method that utilizes a psychological phenomenon of basic level of concepts to select possibly important, natural concepts from a concept lattice and presented first results and experience obtained from experiments. The future research will focus on the following issues:

- Psychological experiments. First, to help assess the quality of the functions for the basic level degrees, aiming to benefit the process of selecting important concepts from the concept lattice. Second, to help better understand the phenomenon of the basic level, thus aiming to benefit the psychology of concepts itself. Our experimental work opens several questions for psychological research such as whether the basic level may contain comparable concepts or whether and in what sense the collection of basic level concepts needs to be exhaustive.
- Comparison with other techniques to select important formal concepts, in particular with the stability index [14,15]. A more detailed study of the the mutual relationship of the various basic level degree functions, utilizing statistical analyses.
- Utilizing further results of the studies of the basic level in the psychology of concepts, in particular utilizing those from which quantitative criteria for the basic level can be obtained[19, p. 213].
- Comparing the idea of basic level concepts with the heuristics known from cluster analysis [9].
- A theoretical study of the issues pertaining to basic level, involving existing work on similarity in concept lattices such as [17].
- Design of efficient algorithms to compute basic level concepts.

Acknowledgment. Supported by Grant No. P103/10/1056 of the Czech Science Foundation.

References

1. Belohlavek, R., Klir, G.J. (eds.): Concepts and Fuzzy Logic. MIT Press, Cambridge (2011)
2. Bělohlávek, R., Sklenář, V.: Formal Concept Analysis Constrained by Attribute-Dependency Formulas. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 176–191. Springer, Heidelberg (2005)
3. Bělohlávek, R., Vychodil, V.: Formal Concept Analysis with Constraints by Closure Operators. In: Schärfe, H., Hitzler, P., hrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 131–143. Springer, Heidelberg (2006)
4. Belohlavek, R., Vychodil, V.: Formal concept analysis with background knowledge: attribute priorities. IEEE Trans. Systems, Man, and Cybernetics, Part C 39(4), 399–409 (2009)
5. Brown, R.: How shall a thing be called? Psychological Review 65, 14–21 (1958)
6. Carpineto, C., Romano, G.: Concept Data Analysis. Theory and Applications. J. Wiley (2004)
7. Cellier, P., Ferré, S., Ridoux, O., Ducassé, M.: A parameterized algorithm to explore formal contexts with a taxonomy. Int. J. Found. Comput. Sci. 2, 319–343 (2008)
8. Dias, S.M., Vieira, N.J.: Reducing the size of concept lattices: The JBOS Approach. In: Proc. CLA 2010. CEUR WS, vol. 672, pp. 80–91 (2010) ISBN 978–84–614–4027–6
9. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis, 4th edn. Arnold, London (2001)
10. Fisher, S.C.: The process of generalizing abstraction; and its product, the general concept. Psychological Monographs XXI(2.90), 1–209 (1916)
11. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer, Berlin (1999)
12. Jaccard, P.: Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 547–579 (1901)
13. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Dordrecht, Kluwer (2000)
14. Klimushkin, M., Obiedkov, S., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 255–266. Springer, Heidelberg (2010)
15. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence 49, 101–115 (2007)
16. Kwuida, L., Missaoui, R., Ben Amor, B., Boumedjout, L., Vaillancourt, J.: Restrictions on concept lattices for pattern management. In: Proc. CLA 2010. CEUR WS, Vol. 672, pp. 235–246 (2010) ISBN 978–84–614–4027–6
17. Lengnink, K.: Formalisierungen von Ähnlichkeit aus Sicht der Formalen Begriffsanalyse. PhD Thesis, TH Darmstadt, Shaker Verlag, Aachen (1996)
18. Machery, E.: 100 years of psychology of concepts: the theoretical notion of concept and its operationalization. Stud. Hist. Phil. Biol. & Biomed. Sci. 38, 63–84 (2007)

19. Murphy, G.L.: The Big Book of Concepts. MIT Press, Cambridge (2002)
20. Rosch, E.: Principles of categorization. In: Rosch, E., Lloyd, B.B. (eds.) Cognition and Categorization, pp. 27–48. Erlbaum, Hillsdale (1978)
21. Rosch, E., et al.: Basic objects in natural categories. Cognitive Psychology 8, 382–439 (1976)
22. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley (2006)
23. Xie, Z.: Φ-generalized concept lattice models: From power concepts to formal concepts, and then to robust concepts. In: CLA 2006, pp. 219–230 (2006)

A Peep through the Looking Glass: Articulation Points in Lattices

Anne Berry and Alain Sigayret

LIMOS UMR CNRS 6158*
Ensemble Scientifique des Cézeaux, Université Blaise Pascal,
63 173 Aubière, France
{berry,sigayret}@isima.fr

Abstract. We define as an 'articulation point' in a lattice an element which is comparable to all the other elements, but is not extremum.

We investigate a property which holds for both the lattice of a binary relation and for the lattice of the complement relation (which we call the mirror relation): one has an articulation point if and only if the other has one also.

We give efficient algorithms to generate all the articulation points. We discuss artificially creating such an articulation point by adding or removing crosses of the relation, and also creating a chain lattice.

We establish the strong relationships with bipartite and co-bipartite graphs; in particular, we derive efficient algorithms to compute a minimal triangulation and a maximal sub-triangulation of a co-bipartite graph, as well as to find the clique minimal separators and the corresponding decomposition.

Keywords: articulation point, chain lattice, complement relation, co-bipartite graph, minimal triangulation, clique separator decomposition.

1 Introduction

In previous papers we showed and exploited the strong relationship between the lattice built on the maximal bicliques of a bipartite graph and the minimal separators of the co-bipartite graph, which is the complement of the bipartite graph [4].

A question which we have often been asked is: "But what can you say about the lattice of the bipartite complement ?"

In this paper, we begin our investigation of this question with a simple property which is common to both lattices: a lattice has an articulation point if and only if the lattice of the bipartite complement has an articulation point. What we call an 'articulation point' in a lattice is an element which is comparable to all the other elements, but is not an extremum. The removal of such an 'articulation point' disconnects the lattice diagram.

* Research partially supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 45–60, 2012.

In order to avoid sentences such as "The co-bipartite graph which is the complement of the bipartite complement ...", we have chosen to refer to the complement relation as the 'mirror relation'. We will investigate properties of the relations, lattices and graphs seen with this 'looking glass'.

We characterize in terms of binary relation the cases where the lattice has an articulation point. We give an efficient algorithm to find all the articulation points of a lattice. We also examine the case where there is no articulation point: we can create one by either adding or removing an inclusion-minimal set of elements of the relation.

We then go on to discuss the case where all the elements of the lattice are articulation points. Such a lattice is a chain, as well as its mirror lattice. We give a linear-time algorithm to recognize a chain lattice, and then use it to embed a lattice into a chain by adding or removing an inclusion-minimal set of crosses of the relation.

In all cases, we explore the relationship with the bipartite and co-bipartite graphs involved. Our approach uses lattice theory to propose an alternate process for several graph algorithms. We also derive a new and more efficient algorithm to compute a minimal triangulation and a maximal subtriangulation of a co-bipartite graph.

The paper is organized as follows: Section 2 gives preliminary notations and results. Sections 3 deals with lattices endowed with an articulation point. Section 4 discusses the algorithmic issues of finding these articulation points efficiently. Section 5 investigates lattices which are chains. Section 5.3 addresses the issues related to artificially creating a lattice which is a chain; in particular we improve the triangulation of a co-bipartite graph. We conclude in Section 6.

2 Preliminaries

As our results are pertaining to both lattices and graphs, we give the necessary notions for both fields.

2.1 Relations, Concepts and Lattices

Given a finite set $\mathcal{O}$ of *objects* (which we will denote by numbers in our examples) and a finite set $\mathcal{A}$ of *attributes*, (which we will denote by lowercase letters), we will consider a binary **relation** $\mathcal{R}$ as a subset of the Cartesian product $\mathcal{O} \times \mathcal{A}$. We will refer to elements of $\mathcal{R}$ as **crosses**. For $x \in \mathcal{A}$, we will denote $\mathcal{R}(x) = \{y \in \mathcal{A} \,|\, (x, y) \in \mathcal{R}\}$, and for $y \in \mathcal{O}, \mathcal{R}(y) = \{x \in \mathcal{O} \,|\, (x, y) \in \mathcal{R}\}$. For $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{A}$, subrelation $\mathcal{R}(X, Y)$ denotes the restriction of $\mathcal{R}$ to X and Y: $(x, y) \in \mathcal{R}(X, Y)$ *iff* $(x, y) \in \mathcal{R}$ and $x \in X$ and $y \in Y$. The **mirror relation** of $\mathcal{R}$ is the relation $\overline{\mathcal{R}} \subseteq \mathcal{O} \times \mathcal{A}$ such that $(x, y) \in \overline{\mathcal{R}}$ *iff* $(x, y) \notin \mathcal{R}$.

The triple $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ is called a **context** [12]; a **concept** of this context is a maximal sub-product $X \times Y \subset \mathcal{R}$, denoted (X, Y): $\forall x \in X, \ \forall y \in Y, \ (x, y) \in \mathcal{R}$, and $\forall x \in \mathcal{O} - X \ \exists y' \in Y \,|\, (x, y') \notin R$, and $\forall y \in \mathcal{A} - Y \ \exists x' \in X \,|\, (x', y) \notin R$.

X is called the **extent** of concept (X, Y), and Y its **intent**. In our examples, we will shorten the notations using for instance $(12, abcde)$ instead of $(\{1, 2\}, \{a, b, c, d, e\})$.

A **lattice** is a partially ordered set in which every pair $\{e, e'\}$ of elements has both a lowest upper bound and a greatest lower bound. A finite lattice has two **extremal** elements: a lowest element, called the **bottom** element, and a greatest element, called the **top** element. A lattice is graphically represented by its **Hasse diagram**: transitivity and reflexivity arcs are omitted, and the orientation from bottom to top is implicit. In the Hasse diagrams, only the objects or attributes which appear for the first time are represented, as detailed in the example given in Subsection 2.6. Our lattices are drawn with the program 'Concept Explorer' [1]. A **maximal chain** of a lattice is a path (all the elements are comparable) from bottom to top in the Hasse diagram.

The concepts of a context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ are ordered by inclusion on their intents: $(X, Y) < (X', Y')$ *iff* $X \subset X'$ *iff* $Y' \subset Y$. This defines a finite lattice called a **concept lattice** (or Galois lattice [11]) denoted $\mathcal{L}(\mathcal{R})$. Two concepts (X, Y) and (X', Y') are comparable if $X \subset X'$ or $X' \subset X$. A concept (X', Y') is a descendant of (X, Y) if $X \subset X'$. Concept (X', Y') is said to cover concept (X, Y) if $X \subset X'$ and there is no concept (X'', Y'') such that $X \subset X'' \subset X'$.

For any concept (X, Y), the descendants of (X, Y) form a sub-lattice, which is isomorphic to the lattice formed on Bordat's subrelation [7] $\mathcal{R}(\mathcal{O} - X, Y)$: any concept (W, Z) of this relation corresponds to concept $(W + X, Z)$ of the original relation.

The reader is referred to [12] and [11] for details on lattices and ordered sets.

2.2 Graphs

An undirected finite graph is denoted $G = (V, E)$, where V is the vertex set, $|V| = n$, and $E \subset V^2$ is the edge set, $|E| = m$. The **neighborhood** $N_G(x)$ of vertex x in graph G is the set of vertices $y \neq x$ such that xy is an edge of E (we then say that x and y **see** each other). The neighborhood $N_G(X)$ of a set X of vertices is $N(X) = (\bigcup_{x \in X} N(x)) - X$. $G(X)$ denotes the **subgraph induced** by X in G, *i.e.* the subgraph of G with vertex set X and edges set $\{xy \in E \,|\, x, y \in X\}$.

A **clique** is a set X of vertices with all possible edges (i.e. $\forall x, y \in X, x \neq y, xy \in E$). A **maximal clique module** is a clique X such that $\forall x, y \in X, N(x) \cup \{x\} = N(y) \cup \{y\}$, and which is maximal for this property. A **stable set** (or independent set) is a set X of vertices with no edge (i.e. $\forall x, y \in X, xy \notin E$). A **path** in a graph is a sequence $(x_0, ..., x_k)$ of vertex such that, for any $i \in [0, k[$, $x_i x_{i+1}$ is an edge of the graph. A **cycle** of length k is a path $(x_0, ..., x_k)$ with $x_0 = x_k$ and $k > 2$. A **chord** in a cycle is an edge between two non-consecutive vertices of the cycle. A C_4 is an induced chordless cycle on 4 vertices, a $2K_2$ is the complement of a C_4. A **connected component** of a graph is an inclusion-maximal set of vertices in which there is a path between any pair of distinct vertices. A graph is said to be connected if it has only one connected component, and disconnected otherwise.

The **complement** of graph $G = (V, E)$ is graph $\overline{G} = (V, \overline{E})$ with $\overline{E} = \{xy \,|\, x \neq y \text{ and } xy \notin E\}$.

Minimal Separators

A **separator** S of a connected graph $G = (V, E)$ is a subset of vertices the removal of which disconnects the graph; a separator S is called a **minimal separator** if there are at least two connected components X and Y of $G(V - S)$ such that $N(X) = N(Y) = S$.

A separator S is called a **clique separator** if it is a separator and a clique; we will say that we **saturate** a non-clique separator S if we add all missing edges necessary to make S a clique. Clique minimal separator **decomposition** is a graph decomposition which repeatedly uses a clique minimal separator S to replace the current graph $G = (V, E)$ with subgraphs $G(C_i \cup N(C_i))$, where C_i is a connected component of $G(V - S)$; the final set of subgraphs obtained are called '**atoms**' (see [3] for full details on this decomposition). A minimal separator S is said to **cross** another minimal separator S' if S' has at least on vertex in each connected component of $G(V - S)$ [17]. Saturating a minimal separator S causes all the minimal separators which cross S to disappear. Thus a minimal separator is a clique if and only if it crosses no other minimal separator [17], [5].

Chordal Graphs and Triangulation

A graph is said to be **chordal** (or **triangulated**) if it contains no chordless induced cycle of length strictly greater than three. **Minimal triangulation** is the process of embedding a graph $G = (V, E)$ into a chordal graph $H = (V, E+F)$ by the *addition* of an inclusion-minimal set F of edges: H is chordal but fails to remain chordal if any proper subset of edges $F' \subset F$ is removed. A graph is chordal if and only if all its minimal separators are cliques. Repeatedly saturating a minimal separator is a process which yields a minimal triangulation [5].

A **maximal subtriangulation** $H' = (V, E - F')$ is a chordal graph obtained from graph $G = (V, E)$ by *removing* an inclusion-minimal set of edges.

2.3 Bipartite Graphs

A **bipartite graph** $G = (V_1 + V_2, E)$ is a graph whose vertex set can be bipartitioned into two disjoint sets V_1 and V_2, each inducing a stable set. A **biclique** $(X + Y)$ in a bipartite graph, with $X \subseteq V_1$ and $Y \subseteq V_2$, is defined as having all possible edges: $\forall x \in X, \forall y \in Y,\ xy \in E$. We will say that vertex $x \in X$ (resp. $y \in Y$) is **universal** if x sees all the vertices of Y (resp. X).

We will call **mirror** (or bipartite complement) of bipartite graph $G = (V_1 + V_2, E)$ the bipartite graph $mir(G) = (V_1 + V_2, F)$ such that $\forall x \in V_1, y \in V_2$, $xy \in F$ *iff* $xy \notin E$.

Any context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ is associated with bipartite graph $bip(\mathcal{R}) = (\mathcal{O}+\mathcal{A}, E)$, where $xy \in E$ if $(x, y) \in \mathcal{R}$. There is a one-to-one correspondence between the maximal bicliques of $G = bip(\mathcal{R})$ and the concepts of $\mathcal{L}(\mathcal{R})$.

2.4 Co-bipartite Graphs

A **co-bipartite** graph is a graph which is the complement of a bipartite graph. The vertex set of a co-bipartite graph can thus be partitioned into two disjoint sets V_1 and V_2, each inducing a clique. Any minimal separator S of a co-bipartite graph defines exactly two connected components, X and Y, with $X \subset V_1$ and $Y \subset V_2$ and $S = N(X) = N(Y)$ [4].

We will call **mirror** of co-bipartite graph $G = (V_1 + V_2, E)$ the co-bipartite graph $mir(G) = (V_1 + V_2, F)$ with the same cliques sets X and Y, and where for $x \in V_1$ and $y \in V_2$, $xy \in F$ *iff* $xy \notin E$.

The reader is referred to [18] and [9] for details on graphs.

2.5 Lattices and Co-bipartite Graphs

Any context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ is associated with a concept lattice $\mathcal{L}(\mathcal{R})$, a bipartite graph $bip(\mathcal{R})$ built on stable sets $\mathcal{O}$ and $\mathcal{A}$, and a co-bipartite graph $cobip(\mathcal{R})$ built on cliques $\mathcal{O}$ and $\mathcal{A}$, where xy is an external edge of $cobip(\mathcal{R})$ *iff* $xy \notin \mathcal{R}$.

Theorem 1. *[4] Let $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ be a context, let $cobip(\mathcal{R})$ be the corresponding co-bipartite graph. Then (X, Y) is a concept of $\mathcal{R}$ if and only if $S = V - (X \cup Y)$ is a minimal separator of $cobip(\mathcal{R})$, minimally separating $X \subset \mathcal{O}$ from $Y \subset \mathcal{A}$.*

Characterization 1. *[4] Given a context $(\mathcal{O},\mathcal{A},\mathcal{R})$, concepts (X,Y) and (X',Y') are comparable elements of $\mathcal{L}(\mathcal{R})$ if and only if their respective associated minimal separators $S = (\mathcal{O} - X) \cup (\mathcal{A} - Y)$ and $S' = (\mathcal{O} - X') \cup (\mathcal{A} - Y')$ are non-crossing minimal separators of $cobip(\mathcal{R})$.*

2.6 Example

Figure 1 shows a relation $\mathcal{R}$ with its associated bipartite graph $bip(\mathcal{R})$, the corresponding co-bipartite graph $cobip(\mathcal{R})$, and the associated concept lattice $\mathcal{L}(\mathcal{R})$, as well as the mirror objects associated with $\mathcal{R}$: the complement relation $\overline{\mathcal{R}}$ with its associated graph $bip(\overline{\mathcal{R}})$, the corresponding co-bipartite graph $cobip(\overline{\mathcal{R}})$, and the associated concept lattice $\mathcal{L}(\overline{\mathcal{R}})$.

3 Lattices with an Articulation Point

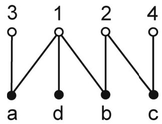
We will first characterize the relations whose lattices are endowed with an articulation point, and then examine how this is translated in the mirror relation.

Definition 1. *Let $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ be a context. A concept (X, Y) which is not the top or bottom element is called an **articulation point** of $\mathcal{L}(\mathcal{R})$ if it is comparable with all the other elements of $\mathcal{L}(\mathcal{R})$.*

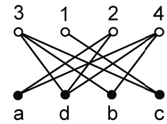
$\mathcal{R}$	a	b	c	d
1	×	×		×
2		×	×	
3	×			
4			×	

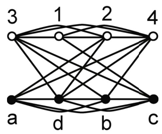
$\mathcal{R}$ and its mirror relation $\overline{\mathcal{R}}$

$\overline{\mathcal{R}}$	a	b	c	d
1			×	
2	×			×
3		×	×	×
4	×	×		×

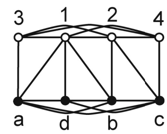


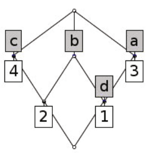
bip($\mathcal{R}$) and its mirror *bip*($\overline{\mathcal{R}}$)





cobip($\mathcal{R}$) and its mirror *cobip*($\overline{\mathcal{R}}$)





$\mathcal{L}(\mathcal{R})$ and its mirror lattice $\mathcal{L}(\overline{\mathcal{R}})$

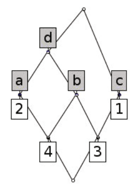


Fig. 1. A relation $\mathcal{R}$, its complement $\overline{\mathcal{R}}$, the associated graphs and lattices

3.1 Cases Where the Lattice Has an Articulation Point

Characterization 2. *Let* (X, Y) *be a concept of* $\mathcal{L}(\mathcal{R})$*.* (X, Y) *is an articulation point of* $\mathcal{L}(\mathcal{R})$ *if and only if in* $bip(\mathcal{R})$, $(\mathcal{O}-X)\cup(\mathcal{A}-Y)$ *is a stable set containing at least one vertex of* $\mathcal{O}$ *and at least one vertex of* $\mathcal{A}$*.*

Proof: Let $G = bip(\mathcal{R})$. (X, Y) is an articulation point: by definition, (X, Y) is comparable to all the other concepts, and by Characterization 1, $S = (\mathcal{O} - X) \cup (\mathcal{A} - Y)$ is a clique in co-bipartite graph *cobip*($\mathcal{R}$), and therefore a stable set in *bip*($\mathcal{R}$). There must be at least two concepts (W, Z) with $W \subset X$ and (W', Z') with $Z' \subset Y$, else (X, Y) is extremum. If there is a concept (X, Y) such that $G(V-(X+Y))$ is a stable set containing at least one vertex of $\mathcal{O}$ and at least one vertex of $\mathcal{A}$, then there is at least one element of $\mathcal{L}(G)$ above (X, Y) and at least on element below, so (X, Y) is not extremum; suppose there is a concept (X', Y') which is not comparable with (X, Y): let x be a vertex of $X - X'$, y a vertex of $Y - Y'$; xy is an edge of $G(V-(X+Y))$, which then fails to be a stable set. □

We are now ready to present our main theorem:

Theorem 2. *A concept lattice* $\mathcal{L}(\mathcal{R})$ *has an articulation point if and only if its mirror concept lattice* $\mathcal{L}(\overline{\mathcal{R}})$ *has an articulation point.*

Proof: Let $G = (V_1 + V_2, E)$ be a bipartite graph, let $X + Y$ be a maximal biclique of G such that $(V_1 - X) \cup (V_2 - Y)$ induces a stable set. The mirror of G is a bipartite graph in which, since $X \neq \emptyset$ and $Y \neq \emptyset$, $(V_1 - X) + (V_2 - Y)$ is a maximal biclique; $X \cup Y$ induces a stable set, so Theorem 2 follows from Characterization 2. □

Figure 2 illutrates Theorem 2 and Characterization 2.

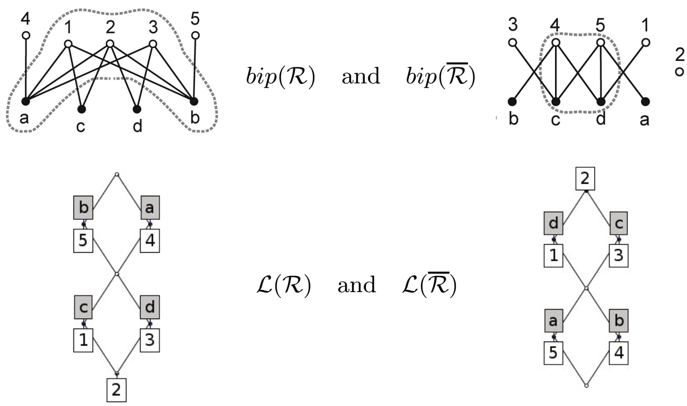


Fig. 2. Bipartite graph $bip(\mathcal{R})$ has a partition into maximal biclique $\{1, 2, 3, a, b\}$ and stable set $\{4, 5, c, d\}$; the corresponding lattice $\mathcal{L}(\mathcal{R})$ has an articulation point: $(123, ab)$. The mirror bipartite graph $bip(\overline{\mathcal{R}}) = mir(bip(\mathcal{R}))$ has also a partition (maximal biclique $\{4, 5, c, d\}$ and stable set $\{1, 2, 3, a, b\}$); the corresponding mirror lattice $\mathcal{L}(\overline{\mathcal{R}})$ has an articulation point: $(45, cd)$.

Let us remark that a similar class of bipartite graphs, called 'K+S' graphs was studied in [13], [14]. 'K+S' graphs are the bipartite graphs which can be partitioned into a maximal biclique and a stable set, and is thus a superclass of the bipartite graphs whose associated lattice has an articulation point: the reduced relation of a 'K+S' graph may correspond to a disconnected bipartite graph, so the corresponding lattice cannot have an articulation point. This is the case for instance for relation $\mathcal{R} = \{(1, a), (2, b), (3, a), (3, b)\}$.

3.2 Expressing the Mirror Articulation Point

When concept (X, Y) is an articulation point of $\mathcal{L}(\mathcal{R})$, we could expect that $(\mathcal{O}-X, \mathcal{A}-Y)$ is an articulation point of $\mathcal{L}(\overline{\mathcal{R}})$. However, when an object x of X and/or a attribute y of Y fails to see vertices of the stable set $S = V-(X+Y)$, this is not exactly the expression of the mirror articulation point. The following theorem details the possible cases.

Theorem 3. *Let (X, Y) be an articulation point of $\mathcal{L}(G)$; then*

1. *If all the vertices of* X *and all the vertices of* Y *see* $S{=}V{-}(X{+}Y)$*, then* $(\mathcal{O}{-}X, \mathcal{A}{-}Y)$ *is an articulation point of* $\mathcal{L}(\overline{\mathcal{R}})$*.*
2. *If a set* X' *of objects of* X *(resp.* $Y' \subset Y$*) fail to see vertices of the stable set* $S{=}V{-}(X{+}Y)$ *but every attribute in* Y *(resp. object in* X*) sees* S*, then* $((\mathcal{O}{-}X){+}X', \mathcal{A}{-}Y)$ *[resp.* $(\mathcal{O}{-}X, (\mathcal{A}{-}Y){+}Y')$*] is an articulation point of* $\mathcal{L}(\overline{\mathcal{R}})$*.*
3. *If a set* X' *of objects of* X *and a set* Y' *of properties of* Y *fail to see vertices of the stable set* $S = V{-}(X{+}Y)$*, then* $((\mathcal{O}{-}X){+}X', \mathcal{A}{-}Y)$ *and* $((\mathcal{O}{-}X), (\mathcal{A}{-}Y){+}Y')$ *are two articulation points of* $\mathcal{L}(\overline{\mathcal{R}})$*.*

Proof:
Case 1: all the vertices of X and of Y see vertices of the stable set $S{=}V{-}(X{+}Y)$. $(\mathcal{O}{-}X, \mathcal{A}{-}Y)$ is an articulation point of $\mathcal{L}(\overline{\mathcal{R}})$: $\mathcal{O}{-}X \cup \mathcal{A}{-}Y$ is a stable set of $bip(\mathcal{R})$ by Characterization 2, so $(\mathcal{O}{-}X, \mathcal{A}{-}Y)$ is a biclique of $\mathcal{L}(\overline{\mathcal{R}})$. This biclique is maximal, since in $bip(\mathcal{R})$ no vertex of $(X{+}Y)$ fails to see $(\mathcal{O}{-}X, \mathcal{A}{-}Y)$.
Case 2: a set X' of objects of X fail to see vertices of the stable set $S{=}V{-}(X{+}Y)$ but every attribute in Y sees S. (Note that all the vertices of X' are equivalent, so if the relation is reduced, there is only one such object x'.) In this case, X' sees all the vertices of S in $mir(G)$, so $mir(G)((\mathcal{O}{-}X){+}(\mathcal{A}{-}Y))$ cannot be a maximal biclique; the corresponding maximal biclique will include X', so $((\mathcal{O}{-}X){+}X', \mathcal{A}{-}Y)$ is an articulation point of $\mathcal{L}(\overline{\mathcal{R}})$.
Naturally, the dual situation where a set of properties fails to see S is similar.
Case 3: Using the previous case, the existence of non-empty X' and Y' insure that $((\mathcal{O}{-}X){+}X', \mathcal{A}{-}Y)$ and $((\mathcal{O}{-}X), ((\mathcal{A}{-}Y){+}Y')$ are two distinct articulation points of $\mathcal{L}(\overline{\mathcal{R}})$. □

Remark 1. Conversely, there may be two consecutive articulation points of $\mathcal{L}(\mathcal{R})$ which correspond to a single one in the mirror lattice. In this case, both are irreducible elements of $\mathcal{L}(\mathcal{R})$.

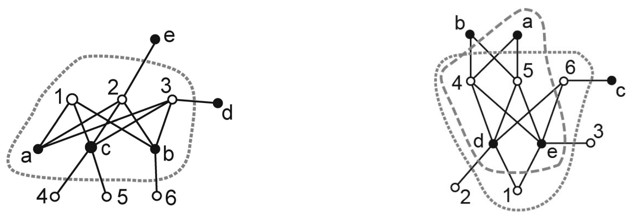
Figure 3 gives an example of a relation corresponding to Case 3 of Theorem 3.

3.3 Impact on the Co-bipartite Graph

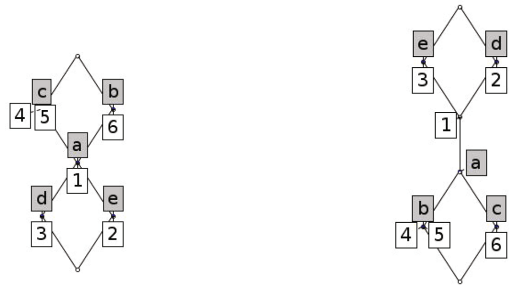
[4] showed that an articulation point of the lattice corresponds to a clique minimal separator of the co-bipartite graph. Let us remark that the converse does not hold: when there is a set X of universal vertices in bipartite graph $bip(\mathcal{R})$, for instance $X \subset \mathcal{O}$, then the neighborhood $N(X)$ of X in co-bipartite graph $cobip(\mathcal{R})$ is a clique separator, separating X from $\mathcal{A}$. The mirror co-bipartite has the same clique separator.

Property 1. Let $\mathcal{R}$ be a relation and $G{=}cobip(\mathcal{R})$ be its associated co-bipartite graph. Then $\mathcal{L}(\mathcal{R})$ has an articulation point (X, Y) if and only if G has a clique minimal separator $S = (\mathcal{O}{-}X){+}(\mathcal{A}{-}Y)$, minimally separating X from Y.

Corollary 1. *A co-bipartite graph has a clique minimal separator if and only if its mirror co-bipartite graph has a clique minimal separator.*



a) G=$bip(\mathcal{R})$ **b)** $bip(\overline{\mathcal{R}})$



c) $\mathcal{L}(\mathcal{R})$ **d)** $\mathcal{L}(\overline{\mathcal{R}})$

Fig. 3. a) A bipartite graph with partition into maximal biclique $\{1, 2, 3, a, b, c\}$ and stable set $\{4, 5, 6, d, e\}$ where 1 and a see only the biclique. **b)** The mirror bipartite graph. **c)**, **d)** The corresponding lattices, where $\mathcal{L}(\mathcal{R})$ has 1 articulation point but $\mathcal{L}(\overline{\mathcal{R}})$ has 2 corresponding articulation points.

3.4 Artificially Creating an Articulation Point of the Lattice

As described in [4], given a relation $\mathcal{R}$, an articulation point in lattice $\mathcal{L}(\mathcal{R})$ can be created by choosing a concept (X, Y), and saturating the corresponding minimal separator $S = (\mathcal{O}{-}X) \cup (\mathcal{A}{-}Y)$ of *cobip*$(\mathcal{R})$. This means that we modify relation $\mathcal{R}$ by *removing* any crosses from $\mathcal{R}(\mathcal{O} - X, \mathcal{A} - Y)$, obtaining relation $\mathcal{R}'$. This causes articulation point (X, Y) to appear in $\mathcal{L}(\mathcal{R}')$. A concept has disappeared from $\mathcal{L}(\mathcal{R})$ if and only if it is incomparable with concept (X, Y) in $\mathcal{L}(\mathcal{R})$. Similarly, a minimal separator S' disappears from the set of minimal separators of *cobip*$(\mathcal{R})$ if and only if S' crosses minimal separator S in *cobip*$(\mathcal{R})$.

Note that in the mirror relation $\overline{\mathcal{R}}$, crosses are *added* to create an articulation point of $\mathcal{L}(\mathcal{R}')$; $\mathcal{L}(\overline{\mathcal{R}})$ is also reorganized, but in a less straightforward fashion.

Figure 4 illustrates what happens when a concept is forced into an articulation point.

4 Finding the Articulation Points of a Lattice

If a concept (X, Y) is an articulation point of $\mathcal{L}(G)$, then (X, Y) appears on any maximal chain of $\mathcal{L}(G)$. Thus we will first compute a maximal chain, and then use it to determine efficiently which concepts are articulation points.

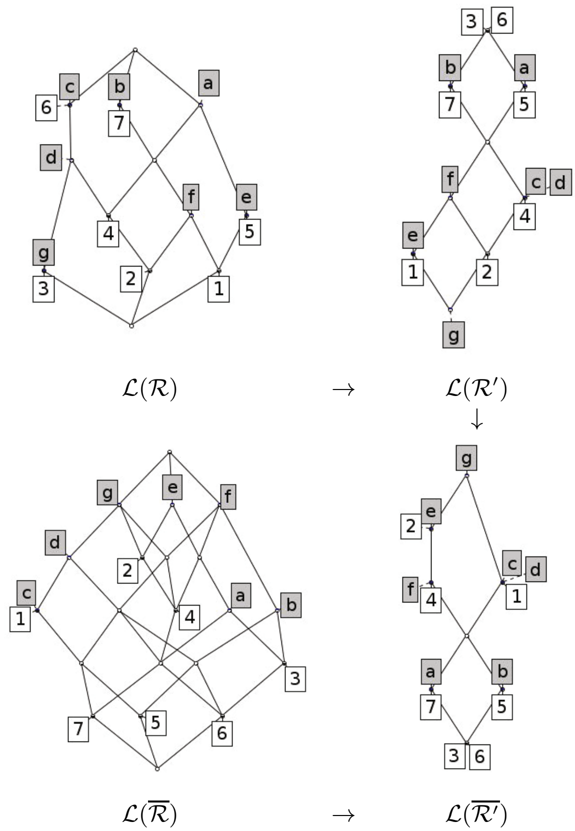


Fig. 4. A lattice on relation $\mathcal{R}$, the sublattice on relation $\mathcal{R}'$ obtained by forcing concept $(124, ab)$ of $\mathcal{L}(\mathcal{R})$ into an articulation point by removing crosses of $\mathcal{R}$, the corresponding mirror lattices

4.1 Computing a Maximal Chain of the Lattice

A maximal chain of the lattice can be computed using the sequence of degrees in a binary relation. We give an algorithm which repeatedly: finds an object x of maximum degree, whose intent $Y = \mathcal{R}(x)$ will belong to a concept (X, Y) covering the bottom element; removes $\mathcal{A} - Y$; uses the universal objects of the obtained subrelation to define the extent X of Y; and then removes X to compute the next concept of the maximal chain in the new relation, which is Bordat's subrelation for concept (X, Y). This corresponds to the process outlined in [4] to compute a maximal chain in $O(|\mathcal{O} + \mathcal{A}|.|\mathcal{R}|)$ time, for which we present a more efficient algorithm MAX-CHAIN.

Theorem 4. *Algorithm* MAX-CHAIN *computes a maximal chain of a lattice $\mathcal{L}(\mathcal{R})$ in $O(min(|\mathcal{R}|, |\overline{\mathcal{R}}|))$ time.*

ALGORITHM MAX-CHAIN

Input : A context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$
Output: A maximal chain $\mathscr{C}$ of $\mathcal{L}(\mathcal{R})$
prefix $\leftarrow \emptyset$; $\mathscr{C} \leftarrow \emptyset$;
repeat
  Choose an object x of maximum degree; $X \leftarrow \{x\}$;
  $Y \leftarrow \mathcal{R}(x)$;
  remove x and $\mathcal{A} - Y$ from $\mathcal{R}$;
  $U \leftarrow$ set of universal vertices of $\mathcal{R}$;
  $X \leftarrow X + U$;
  remove all vertices of U from $\mathcal{R}$;
  add concept $(prefix + X, Y)$ to $\mathscr{C}$;
  prefix $\leftarrow prefix + X$;
until $\mathcal{R}$ *is empty*;

Proof: Let x be an object of maximum degree in $\mathcal{R}$, then $\forall y \in \mathcal{O}$, $\mathcal{R}(x) \not\subset \mathcal{R}(y)$ [4]; x, with its equivalent objects forming set X, yields the extent of concept (X, Y) covering the bottom element, with $Y = \mathcal{R}(X)$ [4]. Bordat's subrelation, in which (X, Y) will correspond to the bottom concept, is then computed, by removing x and $\mathcal{A}-Y$ and finding the universal vertices, which will be the vertices which were in the same maximal clique module as x. The next computed element will be an atom of the new lattice.

Computing and ordering the degrees requires $O(|\mathcal{R}|)$ time. If a correct data structure is used (an adjacency list of $bip(\mathcal{R})$ linked to an ordered list of the degrees), the sequence of degrees can be updated in $O(1)$ time for each removal of crosses to form the new subrelation, so the overall cost for these updates costs $O(|\mathcal{R}|)$ time.

All these steps can be done in $\overline{\mathcal{R}}$ equivalently, so the overall time required is $O(min(|\mathcal{R}|, |\overline{\mathcal{R}}|))$. □

Example 1. Let us consider the lattice from Figure 5. Let us choose 1, which is of maximum degree; $\mathcal{R}(1) = \{a, b, c, d, e, f\}$; concept $(1, abcdef)$ is tentatively created. 1 and g are removed from the relation: there is no universal vertex, so $(1, abcdef)$ is a concept.

In the new relation, 2 is of maximum degree (5); $X \leftarrow 2$, $Y \leftarrow abcde$; we remove 2 and f from the relation; there is no universal vertex, so the next concept on our maximal chain will be $(12, abcde)$.

3 is now of maximum degree; $X \leftarrow 3$, $Y \leftarrow abce$; 3 and d are removed from the relation; 4 is now universal, so $X \leftarrow 34$; 4 is removed from the relation; concept $(1234, abce)$ is created.

5 is now of maximum degree; $X \leftarrow 5$, $Y \leftarrow abc$; 5 and e are removed from the relation; the new relation has no universal vertex, so the next concept is $(12345, abc)$.

6 is now of maximum degree; $X \leftarrow 6$, $Y \leftarrow bc$; 6 and a are removed and the relation becomes empty; the last concept is $(123456, bc)$.

We have generated maximal chain: $((1, abcdef), (12, abcde), (1234, abce),$ $(12345, abc), (123456, bc))$.

	a	b	c	d	e	f	g	degree
1	×	×	×	×	×	×		6
2	×	×	×	×	×		×	6
3	×	×	×		×			4
4	×	×	×		×			4
5	×	×	×	×				4
6		×	×					2
7	×							1

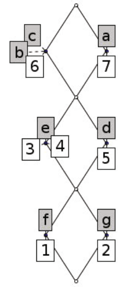


Fig. 5. A lattice with several articulation points. Example 1 computes maximal chain $((1, abcdef), (12, abcde), (1234, abce), (12345, abc), (123456, bc))$.

4.2 Computing the Articulation Points from a Maximal Chain of the Lattice

Once we have obtained a maximal chain $((X_1, Y_1), (X_2, Y_2), \ldots, (X_k, Y_k))$, we want to test each concept (X_i, Y_i) as to whether it is an articulation point. If it is, then by Theorem 2, $S = (\mathcal{O} - X) \cup (\mathcal{A} - Y)$ will be empty, as S is a clique separator of *cobip*$(\mathcal{R})$. Thus we will need to test for emptiness $\mathcal{R}(\mathcal{O}{-}X_1, \mathcal{A}{-}Y_1)$, $\mathcal{R}(\mathcal{O}{-}X_2, \mathcal{A}{-}Y_2)$, $\ldots$, $\mathcal{R}(\mathcal{O}{-}X_k, \mathcal{A}{-}Y_k)$. Since $Y_k \subset \ldots Y_2 \subset Y_1$, we need only to test $\mathcal{R}(\mathcal{O}{-}X_1, \mathcal{A}{-}Y_1)$, $\mathcal{R}(\mathcal{O}{-}X_2, Y_1{-}Y_2)$, $\ldots$, $\mathcal{R}(\mathcal{O}{-}X_k, Y_{k-1}{-}Y_k)$.

Algorithm Articulations computes the set of articulation points of a lattice.

ALGORITHM ARTICULATIONS
Input : A context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$, a maximal chain $((X_1, Y_1), \ldots, (X_k, Y_k))$ of $\mathcal{L}(\mathcal{R})$.
Output: Set $\mathscr{M}$ of articulation points of $\mathcal{L}(\mathcal{R})$
$\mathscr{M} \leftarrow \emptyset$;
for i=*1* **to** k **do**
  $X \leftarrow \mathcal{O} - X_i$;
  if i=*1* **then**
    $Y \leftarrow \mathcal{A} - Y_1$;
  else
    $Y \leftarrow (Y_{i-1} - Y_i)$;
  if $X \times Y = \emptyset$ **then**
    $\mathscr{M} \leftarrow \mathscr{M} + (X_i, Y_i)$;

In Example 1, for maximal chain: $(\ (1, abcdef),\ (12, abcde),\ (1234, abce),\ (12345, abc),\ (123456, bc)\ )$, we will test: $\mathcal{R}(\{2,3,4,5,6,7\}, \{g\})$, $\mathcal{R}(\{3,4,5,6,7\}, \{f\})$, $\mathcal{R}(\{5,6,7\}, \{d\})$, $\mathcal{R}(\{6,7\}, \{e\})$, $\mathcal{R}(\{7\}, \{a\})$. We will find $(2, g) \in \mathcal{R}$, so $(1, abcdef)$ is not an articulation point, $\mathcal{R}(\{34567\}, \{f\}) = \emptyset$, so $(12, abcde)$ is an articulation point, $(5, d) \in \mathcal{R}$, so $(1234, abce)$ is not an articulation point, $\mathcal{R}(\{67\}, \{e\}) = \emptyset$, so $(12345, abc)$ is an articulation point, and finally $(7, a) \in \mathcal{R}$, so $(123456, bc)$ is not an articulation point.

4.3 Finding the Clique Minimal Separator Decomposition of a Co-bipartite Graph

Finding the articulation points of a lattice is equivalent to finding the clique minimal separators of the corresponding co-bipartite graph. Thus we can use Algorithm ARTICULATIONS to compute the clique minimal separators of a co-bipartite graph efficiently, and also easily extract the 'atoms' of the decomposition by clique minimal separators, and the mirror 'atoms'. This requires $O(min(|\mathcal{R}|, |\overline{\mathcal{R}}|))$ time to compute, *i.e.* less than the number of edges of the co-bipartite graph, since only the external edges are traversed.

Theorem 5. *Let $G = (\mathcal{O} + \mathcal{A}, E)$ be a co-bipartite graph on cliques $\mathcal{O}$ and $\mathcal{A}$, let $\mathcal{R}$ and $bip(\mathcal{R})$ be the corresponding bipartite graph and relation on $\mathcal{O} + \mathcal{A}$ (where $xy \in E$ if and only if $xy \notin \mathcal{R}$). Algorithm* ARTICULATIONS *returns an ordered set of articulation points of $\mathcal{L}(\mathcal{R})$, call it $((X_1, Y_1), \ldots, (X_k, Y_k))$. The clique minimal separators of G are $S_1 = (\mathcal{O} - X_1) \cup (\mathcal{A} - Y_1), \ldots, S_k = (\mathcal{O} - X_k) \cup (\mathcal{A} - Y_k)$. The corresponding atoms by clique minimal separator decomposition are: $T_1 = \mathcal{O} \cup (\mathcal{A} - Y_1), T_2 = (\mathcal{O} - Y_1) \cup (\mathcal{A} - Y_2), \ldots, T_{k+1} = (\mathcal{O} - Y_1) \cup \mathcal{A}$.*

In Example 1, with maximal chain: ($(1, abcdef)$, $(12, abcde)$, $(1234, abce)$, $(12345, abc)$, $(123456, bc)$), the articulation points are: $(12, abcde)$ and $(12345, abc)$. The clique minimal separators of $cobip(\mathcal{R})$ are: $\{3, 4, 5, 6, 7, f, g\}$ and $\{6, 7, d, e, f, g\}$ the atoms by clique minimal separator decomposition of $cobip(\mathcal{R})$ will be: $\mathcal{O} \cup \mathcal{A} - \{a, b, c, d, e\}$, $\mathcal{O} - \{1, 2\} \cup \mathcal{A} - \{a, b, c\}$ and $\mathcal{O} - \{1, 2, 3, 4, 5\} \cup \mathcal{A}$, so the atoms obtained by clique minimal separator decomposition are: $\{1, 2, 3, 4, 5, 6, 7, f, g\}$, $\{3, 4, 5, 6, 7, d, e, f, g\}$ and $\{6, 7, a, b, c, d, e, f, g\}$.

5 Lattices Where Every Concept Is an Articulation Point

When every concept is an articulation point, the lattice is just one maximal chain, which we call a chain lattice. In this case, the relation is a 'Guttmann scale': by ordering the elements by decreasing degree, a full triangular matrix is obtained.

5.1 Chain Lattices and the Corresponding Graphs

By Theorem 2, every articulation point of $\mathcal{L}(\mathcal{R})$ corresponds to (at least one) articulation point of $\mathcal{L}(\overline{\mathcal{R}})$, so the following holds:

Property 2. $\mathcal{L}(\mathcal{R})$ is a chain lattice if and only if $\mathcal{L}(\overline{\mathcal{R}})$ is a chain lattice.

In view of the discussion from Subsection 3.2, the chain and the mirror chain do not necessarily have the same number of elements, although one can not be more than twice the length of the other. When $\mathcal{L}(\mathcal{R})$ is a chain lattice, in $cobip(\mathcal{R})$ all the minimal separators are clique separators, so the co-bipartite graph is chordal.

Theorem 6. *A co-bipartite graph is chordal if and only if its mirror co-bipartite graph is chordal.*

Since $cobip(\mathcal{R})$ is chordal, it has no C_4, so $bip(\mathcal{R})$ has no $2K_2$; such a bipartite graph is called a 'chain graph'. Our results give an alternate proof of the result from [18]:

Property 3. A bipartite graph G is a chain graph if and only if $mir(G)$ is a chain graph.

5.2 Recognizing Chain Lattices and the Corresponding Graphs

We will now see that we can test, in the same $O(min\{|\mathcal{R}|, |\overline{\mathcal{R}}|\})$ time as Algorithm Max-Chain, the three equivalent properties:

- whether $\mathcal{L}(\mathcal{R})$ is a chain lattice;
- whether $bip(\mathcal{R})$ is a chain graph;
- whether $cobip(\mathcal{R})$ is chordal.

Given a context $(\mathcal{O}, \mathcal{A}, \mathcal{R})$, we can efficiently recognize whether $\mathcal{L}(\mathcal{R})$ is a chain, using the results from Section 4: while computing a maximal chain $((X_1, Y_1), \ldots, (X_k, Y_k))$ of the lattice, add a counter which keeps track of the number of crosses of $\mathcal{R}$ involved; in the end, the lattice is a chain if and only if the counter's value is exactly $|\mathcal{R}|$. This is the same as testing whether $|X_1|.|Y_1| + \Sigma_{i=2}^{k} (|X_i| - |X_{i-1}|) \,.\, |Y_i| = |\mathcal{R}|$.

Recently, many graph recognition algorithms endeavour to add a 'certificate' to the answer; a certificate provides the user with a structure which is easy to verify and which enables to quickly check that the answer is indeed correct. In the case of chain graphs, for example, a recent result gives a certifying algorithm [15].

For chain graphs, we can provide a negative certificate in the form of an extraneous element $(x, z) \in \mathcal{R}$ found in $\mathcal{R}(\mathcal{O} - X_i, Y_{i-1} - Y_i)$ which prevents $\mathcal{L}(\mathcal{R})$ from being a chain. In this case, (x, z) corresponds to the lowest concept (X_i, Y_i) which is not an articulation point of $\mathcal{L}(\mathcal{R})$, with $x \in X_i$ and $z \notin Y_i$. In the lattice, there will be at least one concept which is not comparable with (X_i, Y_i), for instance the concept whose intent is $\mathcal{R}(x)$. In a similar fashion, $S = (\mathcal{O} - X_i) \cup (\mathcal{A} - Y_i)$ will be a non-clique minimal separator of $cobip(\mathcal{R})$, as edge xz is missing, certifying that $cobip(\mathcal{R})$ fails to be chordal. Finally, $\forall y' \in (Y_i - \mathcal{R}(x)), \forall x' \in (X_i - X_{i-1})$, $\{x, x', y, y'\}$ induces a $2K_2$ in $bip(\mathcal{R})$, a certificate that $bip(\mathcal{R})$ fails to be a chain graph.

5.3 Creating a Chain Lattice and Corresponding Graph Embeddings

We will now examine what happens when we restrict lattice $\mathcal{L}(\mathcal{R})$ to one of its maximal chains. To do this, we will compute a maximal chain, $((X_1, Y_1), (X_2, Y_2), \ldots, (X_k, Y_k))$, as discussed in Subsection 4.1. We will then remove all crosses which do not correspond to this chain, i.e. we will need to empty $\mathcal{R}(\mathcal{O}-X_1, \mathcal{A}-Y_1), \mathcal{R}(\mathcal{O}-X_2, \mathcal{A}-Y_2), \ldots, \mathcal{R}(\mathcal{O}-X_k, \mathcal{A}-Y_k)$, which as discussed before is equivalent to emptying $\mathcal{R}(\mathcal{O}-X_1, \mathcal{A}-Y_1), \mathcal{R}(\mathcal{O}-X_2, Y_1-Y_2), \ldots, \mathcal{R}(\mathcal{O}-X_i, Y_{i-1}-Y_i), \ldots, \mathcal{R}(\mathcal{O}-X_k, Y_{k-1}-Y_k)$.

As before, this can be done in $O(min(|R|, |\overline{R}|))$ time.
In Example 1 and the lattice from Figure 5, with maximal chain $((1, abcdef)$, $(12, abcde)$, $(1234, abce)$, $(12345, abc)$, $(123456, bc))$, relation $\mathcal{R}$ will be restricted to $\mathcal{R}'$, which can be re-organized into a triangular matrix:

$\mathcal{R}'$	b	c	a	e	d	f	g
1	×	×	×	×	×	×	
2	×	×	×	×	×		
3	×	×	×	×			
4	×	×	×	×			
5	×	×	×				
6	×	×					
7							

Since all the minimal separators of $cobip(\mathcal{R})$ have been saturated by restricting $\mathcal{R}$ to chain $\mathcal{R}'$, a minimal triangulation of $cobip(\mathcal{R})$ is thereby computed, by adding to $cobip(\mathcal{R})$ any missing edge from the Cartesian product $(\mathcal{O}-X_1) \times (\mathcal{A}-Y_1), (\mathcal{O}-X_2) \times (\mathcal{A}-Y_2), \ldots, (\mathcal{O}-X_k) \times (\mathcal{A}-Y_k)$. An existing algorithm [16] computes a minimal triangulation of a claw-free AT-free graph in linear time; co-bipartite graphs are claw-free AT-free graphs [4]; however, the above process can be considered as an improvement on the linear-time, since only the external edges of the co-bipartite are counted in the complexity analysis, but not the edges which lie inside cliques on $\mathcal{O}$ and $\mathcal{A}$.

Since the computed triangulation of $cobip(\mathcal{R})$ is minimal, we can ensure that we have removed an inclusion-minimal set of crosses from $\mathcal{R}$ to obtain a chain lattice; we also have removed an inclusion-minimal set of edges from $bip(\mathcal{R})$ to reduce it to a chain graph.

When examining what happens in the mirror relation $\overline{\mathcal{R}}$, we see that we have computed in $O(min(|R|, |\overline{R}|)$ time:

– a maximal sub-triangulation of the mirror co-bipartite graph $cobip(\overline{\mathcal{R}})$, for which the best known algorithm was the general one in $O(nm)$ time [2].
– a minimal embedding of $bip(\overline{\mathcal{R}})$ into a chain graph.

6 Conclusion and Perspectives

In this paper, we investigate a property which is true in the relation and in the complement relation (which we call the mirror relation). This leads us to present linear time algorithms for both lattice and graph problems, such as computing a maximal chain of the lattice and computing a minimal triangulation of a co-bipartite graph.

When an articulation point is artificially created in a lattice $\mathcal{L}(\mathcal{R})$ by removing crosses from $\mathcal{R}$, we do not know exactly what happens to the mirror lattice $\mathcal{L}(\overline{\mathcal{R}})$, which is a strangely distorted image of $\mathcal{L}(\mathcal{R})$. We conjecture that the number of concepts decreases. The set of concepts may be contracted in a fashion which is exploitable, yielding more information than $\mathcal{L}(\mathcal{R})$, where a set of concepts is simply removed.

We also leave open the question of how the Galois subhierarchy is impacted by these transformations of the relation.

References

1. Yevtushenko, S.A.: System of data analysis "Concept Explorer". In: Proc. 7th Nat. Conf. on Artif. Intell. KII 2000, Ru., pp.127–134 (2000) (in Russian), http://conexp.sourceforge.net/download.html, release 1.3. © Yevtushenko, S.A., et al. (2000-2006)
2. Berry, A., Heggernes, P., Villanger, Y.: A vertex incremental approach for maintaining chordality. Discrete Mathematics 306, 318–336 (2006)
3. Berry, A., Pogorelcnik, R., Simonet, G.: An introduction to clique minimal separator decomposition. Algorithms 3(2), 197–215 (2010)
4. Berry, A., Sigayret, A.: Representing a concept lattice by a graph. Discrete Applied Mathematics 144(1-2), 27–42 (2004)
5. Berry, A., Bordat, J.-P., Heggernes, P., Simonet, G., Villanger, Y.: A wide-range algorithm for minimal triangulation from an arbitrary ordering. Journal of Algorithms 58(1), 33–66 (2006)
6. Birkhoff, G.: Lattice Theory, 3rd edn. American Mathematical Society (1967)
7. Bordat, J.-P.: Calcul pratique du treillis de Galois d'une correspondance. Mathématiques, Informatique et Sciences Humaines 96, 31–47 (1986)
8. Brandstädt, A., Hammer, P.L., Le, V.B., Lozin, V.V.: Bisplit graphs. Discrete Mathematics 299(1-3), 11–32 (2005)
9. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. SIAM Monographs on Discrete Math. Appl., Philadelphia, Vol. 3, (1999)
10. Carroll, L.: Through the Looking-Glass, and What Alice Found There (1871)
11. Caspard, N., Leclerc, B., Monjardet, B.: Ensembles ordonnés finis: concepts, résultats et usages. Mathémathiques et Applications 60 (2007)
12. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
13. Fouquet, J.-L., Giakoumakis, V., Vanherpe, J.-M.: Linear Time Recognition and Optimizations for Weak-Bisplit Graphs, Bi-Cographs and Bipartite P6-Free Graphs. International Journal of Foundations of Computer Science 14(1), 107–136 (2003)
14. Fouquet, J.-L., Giakoumakis, V., Vanherpe, J.-M.: Bipartite Graphs Totally Decomposable by Canonical Decomposition. International Journal of Foundations of Computer Science 10(4), 513–534 (1999)
15. Heggernes, P., Kratsch, D.: Linear-time certifying recognition algorithms and forbidden induced subgraphs. Nordic Journal of Computing 14, 87–108 (2007)
16. Meister, D.: Recognition and computation of minimal triangulations for AT-free claw-free and co-comparability graphs. Discr. Appl. Math. 146(3), 193–218 (2005)
17. Parra, A., Scheffler, P.: How to Use the Minimal Separators of a Graph for Its Chordal Triangulation. In: Fülöp, Z. (ed.) ICALP 1995. LNCS, vol. 944, pp. 123–134. Springer, Heidelberg (1995)
18. Spinrad, J.P.: Efficient Graph Representations. American Mathematical Society (2003)

Practical Use of Formal Concept Analysis in Service-Oriented Computing

Stéphanie Chollet[1], Vincent Lestideau[2], Yoann Maurel[2], Etienne Gandrille[2], Philippe Lalanda[2], and Olivier Raynaud[3]

[1] Laboratoire de Conception et d'Intégration des Systèmes
F-26902, Valence cedex 9, France
stephanie.chollet@lcis.grenoble-inp.fr

[2] Laboratoire d'Informatique de Grenoble
F-38041, Grenoble cedex 9, France
{vincent.lestideau,yoann.maurel,etienne.gandrille, philippe.lalanda}@imag.fr

[3] Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
F-63173, Aubière cedex, France
raynaud@isima.fr

Abstract. Pervasive applications are encountered in a number of settings, including smart houses, intelligent buildings or connected plants. Service-Oriented Computing is today the technology of choice for implementing and exposing resources in such environments. The selection of appropriate services at the right moment in order to compose meaningful applications is however a real issue. In this paper, we propose a FCA-based solution to this problem. We have integrated FCA algorithms in our pervasive gateways and adapted them in order to allow efficient runtime selection of heterogeneous and dynamic services. This work has been applied to realistic use cases in the scope of a European project.

Keywords: Service-Oriented Computing, Pervasive environment, Service classification.

1 Introduction

Service-Oriented Computing (SOC) [13] brings software qualities of major importance. As with any planned reuse approach, it supports rapid, high quality development of software applications. Using existing, already tested, software elements is likely to reduce the time needed to build up an application and improve its overall quality. The key concept of the service-oriented approach is the notion of service. A service is a software entity that provides a set of functionalities described in a service description. The service description contains information on the service's functional part, but also on its non-functional aspects. Based on such specification, a service consumer can search for services that meet its requirements, select a compliant service and invoke it [6].

Web Services are the most popular and well-known technology for implementing Service-Oriented Architectures, both in the industry and in the academia.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 61–76, 2012.

A provider of Web Services can describe their service's functional and non-functional characteristics in a WSDL[1] file and then registers the service description in an UDDI[2] service registry. A client, or consumer, can search the UDDI registry for services that meet their requirements. Consumers use the SOAP[3] protocol to communicate with Web Services.

However, Web Services are not the only technology that implement the Service-Oriented approach. Web Services technology is dominant to integrate IT applications. However, in many other domains such as pervasive environments, the Service-Oriented approach is a solution of choice. Consequently, many technologies have been implemented and adapted to these domains. For instance, UPnP[4] or DPWS[5] are preferred in small area networks for devices whereas OSGi[6] and iPOJO[8] are often used in centralized and embedded equipments.

Service-Oriented Computing has thus evolved to support dynamic service discovery and lazy inter-service binding. Weak coupling between consumers and providers reduces dependencies among composition units, letting each element evolve separately. Late-binding and substitutability improve adaptability: a service chosen or replaced at runtime, based on its current availability and properties, is likely to better fulfill the consumer expectations. Such characteristics are essential when building pervasive applications with strong adaptability requirements. A key point is the ability to select at anytime the relevant service available in the registry to realize an application. The selection of services is a well-known complex problem. This problem has been particularly studied in the domain of Web Services [1,3].

However, we think that this problem can not be restricted to a unique technology. Today, applications are composed of heterogeneous and dynamic services. Applications frequently need to integrate UPnP-based and DPWS-based filed devices and Web Services for remote applications. In this paper, we propose to tackle the problem of heterogeneous and dynamic service selection in the context of pervasive applications. In realistic use cases studied in the European OSAMI[7] project, one salient problem is the number of services to be considered. We investigate a solution based on Formal Concept Analysis (FCA) [10] to select pervasive services in a reactive and efficient fashion. Our proposition, that has been implemented and tested with the industrial partners of OSAMI, brings significant results in terms of efficiency and adaptability.

The paper is organized as follows: in Section 2, we detail the challenges of service selection in pervasive applications. In Section 3, some background about

[1] *Web Services Description Language*, http://www.w3.org/TR/wsdl
[2] *Universal Description Discovery and Integration*, http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm
[3] *Simple Object Access Protocol*, http://www.w3.org/TR/soap/
[4] *Universal Plug and Play*, http://www.upnp.org
[5] *Device Profile for Web Services*, http://specs.xmlsoap.org/ws/2006/02/devprof/devicesprofile.pdf
[6] *Open Service Gateway initiative*, http://www.osgi.org/download/r4v41/r4.core.pdf
[7] *Open Source Ambient Intelligence Commons for an Open and Sustainable Internet*, http://www.osami-commons.org/

FCA is provided. Section 4 outlines the general idea of our FCA-based approach, detailed in Section 5: the service registry, the decision structure the selection algorithms. In Section 7, we present the implementation and experimental results. Before conclusion, Section 8 lists and discusses the related work.

2 Challenges of Service Selection

In Service-Oriented approach, a consumer must select the relevant service before invocation. The selection depends on the consumer requirements and service registry. The information shared between the different actors is the service description. For instance, from the consumer perspective, a Web Service is a black box that provides no technical details on its implementation. The only available information in the WSDL file includes the Web Service functionalities, certain characteristics such as non-functional properties, location and invocation instructions. Figure 1 recapitulates the different standards used by a set of service technologies for service description and service registry.

	Web Services	UPnP	DPWS	OSGi	iPOJO
Service Description	WSDL	UPnP Device Description	WSDL	Java Interface + Properties	Java Interface + Properties
Service Registry	UDDI registry	No registry	No registry	OSGi registry	OSGi registry
Service Discovery	Active	Active and passive	Active and passive	Active and passive	Active and passive
Service Notification and Service Withdrawal		Multicast	Multicast	Event	Event
Communication	SOAP	SOAP	SOAP	Java	Java

Fig. 1. Variety of standards implementing SOC

Service discovery uses network protocols which allow automatic detection of services and/or devices. There are two kinds of service discovery: active - the consumer uses the service registry to discover the service - or passive - the service announces its arrival/departure on the network. Consequently, service technologies supporting passive discovery use multicast or event protocols. For instance, in pervasive environment, services are dynamic: smart devices join and leave the network at unpredictable times; back office applications are regularly updated. Services related to these devices are very volatile. In fact, devices connections and disconnections can be caused by many factors as diverse as users moves, battery problems, users demands, updates [11]. Then, two primitives (notification and withdrawal) can be added to support the dynamicity of services.

To conclude, there is a large variety of service technologies and they use heterogeneous technologies to describe, to discover and to communicate. Note that the service selection is complex due to this multitude of technologies and to the dynamicity of services.

In addition, today applications are more and more composed of multiple heterogeneous services. For exemple, an application for acquisition chain requires devices (UPnP and/or DPWS) to acquire data and services for the business (Web Services) such as to analyze or to store data. We have extended the basic SOC pattern in order to support heterogeneity (Figure 2).

A key issue in such context lies in the runtime selection of relevant services in environments filled with devices and applications. Service selection has become a challenge in pervasive environments with the increasing number of dynamic devices, often providing close functionalities but with different technologies and different descriptions.

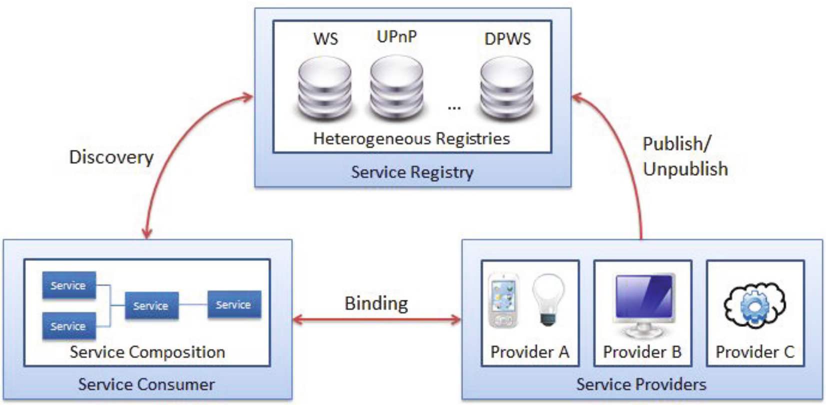


Fig. 2. SOC adapted to the service heterogeneity

3 Theoretical Fundations: Formal Concept Analysis

Formal Concept Analysis (FCA) [10] is a theoretical and mathematical framework used to classify. We propose to use the Formal Concept Analysis method to classify available services at runtime. We very shortly define the main concepts of FCA. The purpose of FCA is to build a partially ordered structure, called concept lattice, from a formal context.

Definition 1. *A **formal context** $\mathbb{K}$ is a set of relations between objects and attributes. It is denoted by $\mathbb{K} = (O, A, R)$ where O and A are respectively sets of **Objects** and **Attributes**, and R is a **Relation** between O and A.*

Definition 2. *A **formal concept** C is a pair (E, I) where E is a set of objects called **Extent**, I is a set of attributes called **Intent**, and all the objects in E are in relation R with all the attributes in I.*

Thus, the Extent of a concept is the set of **all** objects sharing a set of common attributes, and the Intent is the set of **all** attributes shared by the objects of the Extent. Formally:

- $E = \{o \in O, \forall i \in I, (o, i) \in R\}$,
- $I = \{a \in A, \forall e \in E, (e, a) \in R\}$.

Consequently, a formal concept $C = (E, I)$ is made of the objects in E which are exactly the set of objects sharing the attributes in I.

Let X a set of attributes. We define the function $Closure_{\mathbb{K}}(X)$ which associates to X the concept made of the set of objects sharing X and the other attributes shared by this set of objects. Note that the computation of a formal concept from a set of attributes X of size n has a complexity of $\mathcal{O}(n \times m)$ where m is the number of objects.

The set $\mathcal{C}(\mathbb{K})$ of all concepts induced by a context can be ordered using the following partial order relation: $(E_1, I_1) <_{\mathcal{C}} (E_2, I_2)$ if $E_2 \subset E_1$ and $I_1 \subset I_2$.

Definition 3. *A **concept lattice** is defined as the pair $(\mathcal{C}(\mathbb{K}), \leq_{\mathcal{C}})$. Let two concepts (E_1, I_1) and (E_2, I_2) we say that (E_2, I_2) is a successor of (E_1, I_1) if $(E_1, I_1) <_{\mathcal{C}} (E_2, I_2)$. Given I_1 a subset of A, we note by **successors**(I_1) the set of successors of the concept (E_1, I_1). The concept lattice can be represented by a particular graph called Hasse Diagram.*

Note that the computation of a concept lattice from a formal context has a complexity of $\mathcal{O}((n+m) \times m \times |\mathcal{C}(\mathbb{K})|)$ where n is the number of attributes and m is the number of objects ([12]). Most of the time we have $n << m$ and the complexity becomes $\mathcal{O}(m^2 \times |\mathcal{C}(\mathbb{K})|)$.

4 Global Approach

It is assumed that we have an application which is a composition of abstract services defined at design time. During this step, a set of architectural constraints are defined. These constraints are the expected functionnalities and/or non-functional properties; they are considered as mandatory features. From design time to runtime we must consider as much on these architectural constraints as runtime environment constraints (*i.e.* arrival and departure of services).

It is required to select a concrete service for each abstract service of the composition. Current approaches only select services from architectural constraints. In addition, these approaches realize service selection just before runtime. Consequently, they do not completely handle the environment dynamicity.

Our objective is to ensure at runtime the more adapted configuration according to the architectural constraints, the current environment and user preferences. Figure 3 illustrates our approach which is divided into three parts:

- **Setting forth user requirements.** The user requirements are expressed by a request which contains the architectural constraints. These constraints are composed of mandatory and optional features defined by the architect at design time.

- **Storage of available services in the environment.** The service registry supports the dynamicity and the heterogeneity of services. It is global and it contains the service descriptions annotated with user properties (QoS).
- **Service selection.** The selection process is divided into two parts. First, a set of services is selected and classified according to the defined user requirements. A decision structure is built at that time. Second, one approriate service must be chosen from this structure. We define adapted algorithms to search in this structure the most appropriate service according to the user preferences.

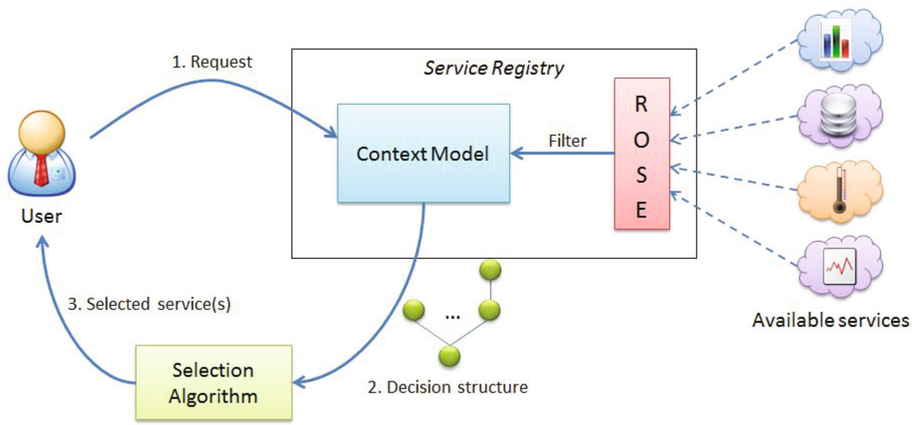


Fig. 3. Global approach

The following section is divided into three parts in which we detail the service registry, the computation of the decision structure and the selection algorithms.

5 Application to Services

5.1 Service Registry

The service registry is the central element of our approach. Its goal is to maintain a global view of all the available services at runtime. It is composed of two parts:

- ROSE [4]: an integration platform monitoring the runtime environment, *i.e.* it traces services availability and provides information about them;
- a context model: it stores and maintains the information recovered by ROSE.

ROSE is an OSGi-based open source middleware[8] It detects all the services being in the environment. This tool is able to support active and passive service

[8] http://wiki.chameleon.ow2.org/xwiki/bin/view/Main/Rose

discovery. Concretly, it supports multicast and event mechanisms used to service notification and service withdrawal. These capabilities are essential because services related to devices are very volatile.

Since ROSE detects all the services being in the environment, we have defined a filter to compute an application-specific context model. The filter allows to specify the services of interest. In fact, only the services of interest for the application are selected. For instance, a multimedia entertainment application requires multimedia services such as movies library, TV... Humidity sensors will be ignored.

In our project, we have adapted the context model to the FCA formal context. The context model can be seen as a relation between the filtered services and the possible service features of the application domain. We can categorize the service features into three main groups (Table 1):

- The service technologies (Web Service, UPnP, DPWS...),
- The service functionalities,
- The non-functional properties required and/or provided by the service.

Table 1. Context model as a formal context

	$t_1, ..., t_i$	$f_1, ..., f_j$	$nf_1, ..., nf_k$
s_1			
...			
s_n			

We have tested our work with security properties (authentication, confidentiality, integrity...), a particular non-functional property that can be expressed as a boolean value (*i.e.* services are secured or not). However, all non-functional properties are not boolean. But, in [15,9,2], authors have investigated the link between FCA with numerical data.

5.2 Decision Structure

From the context model expressed as a formal context, we extract a set of ordered formal concepts, *i.e.* an extract of the concept lattice. Given that the computation of a lattice has a complexity in $\mathcal{O}(m^2 \times |\mathcal{C}(\mathbb{K})|)$ and that the space complexity is in $\mathcal{O}(2^n)$ since the number of concepts is potentially 2^n, it is not realistic to dynamically compute the entire concept lattice for each user request. We propose a way to compute only the interesting concepts.

In the concept lattice, we can distinguish two exclusive groups of concepts, as illustrated in Figure 4:

- **concepts with no real meaning.** These concepts contain in their intent a set of properties which is not usable. For example, all the concepts with an intent composed of only non-functional properties do not make sense. The *top* and the *bottom* of the lattice are also meaningless. The *top* contains in its intent all the attributes, *i.e.* all the functional and non-functional properties, and the extent is empty because no service can provide all the properties. Similarly, the *bottom* contains in its extent all the services and the intent is empty because it is not possible to have a common property for all the services. For example, the type of service is an exclusive property.

– **concepts with sense.** Contrary to the previous group, the intent of the concepts makes sense, *i.e.* the intent contains coherent information. For example, at least one functionality is in the intent.

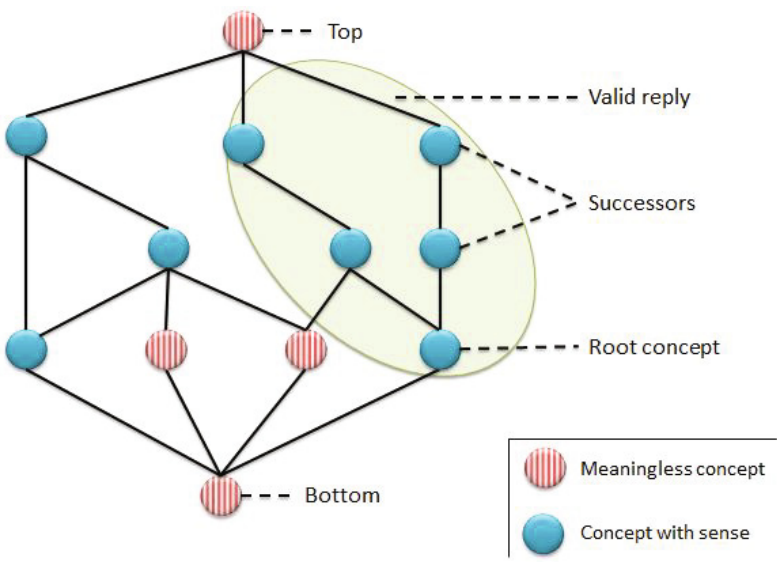


Fig. 4. Computation of the decision structure

This classification into concepts with or without applicative meaning is the key of our approach. According to concept semantics, we can compute only the interesting concepts and not the entire lattice. The interesting concepts are a subset of meaningful concepts extracted from the lattice. The subset is a structure where the root element is a formal concept and the nodes are the successors of the formal concept.

The decision structure is computed in response of a user request. In the following, a user request (denoted in bold) is defined by a set of mandatory features (denoted by MF). The result of the request is set of formal concepts in which the extent (denoted by S) contains all services sharing a set of common features, the mandatory features and possibly a new set of found features (denoted by FF).

In the following, we present two kinds of user requests based on the selection of a service for a workflow activity. First, the selection can be only based on mandatory features. Second, the selection can be based on mandatory and optional features.

The solution for the selection of services from a set of mandatory features is the computation a formal concept called root concept in which the intent contains the mandatory features: $(S; \mathbf{MF} \cup FF)$. The sets S and FF can be empty. If the extent S is empty, there is no service available providing the mandatory features.

The selection based on mandatory and optional features is an extension of the previous selection. To take into account the optional features, we propose to compute the successors of the root concept $(S; \mathbf{MF} \cup FF)$. The computation of the succesors is then an extract of the concept lattice that can be viewed as the decision structure: $(S; \mathbf{MF} \cup FF) \cup \texttt{successors}(\mathbf{MF} \cup FF)$. This decision structure is limited to the set of successors in the concept lattice.

Table 2 is an illustration of a context model. The context model is a simplification of a real context model because we have only few service characteristics (attributes) and eight available services. For clarification purposes, the extract of context model only contains the functional attributes *Temperature* and *Humidity* and only the services providing the *Temperature* functionality.

Table 2. Extract of the context model

	WS	UPnP	DPWS	Temperature (T)	Humidity (H)	Authentication (A)	Confidentiality (C)	Integrity (I)
S_1		X		X				X
S_2		X		X			X	X
S_3			X	X	X			X
S_4			X		X		X	X
S_5		X		X	X			X
S_6		X		X		X		
S_7		X		X				
S_8			X	X				X

For example, the administrator can select all the services providing the temperature functionality. The selection result for *Temperature* activity is the formal concept $(\{S_1, S_2, S_3, S_5, S_6, S_7, S_8\}; \{\mathbf{Temperature}\})$. All the services except S_4 provide the *Temperature* functionality. The administrator can refine his request: the classification of the *Temperature* services can be computed from the concept $(\{S_1, S_2, S_3, S_5, S_6, S_7, S_8\}; \{\mathbf{Temperature}\})$ previously obtained. The successors of this concept constitutes the decision structure (Figure 5).

At the bottom of the figure, we find the concept $(\{S_1, S_2, S_3, S_5, S_6, S_7, S_8\};$ $\{\mathbf{Temperature}\})$. Services are classified according to their characteristics. In [7], we have extended our approach to the composition of decision structures in response to complex user requests.

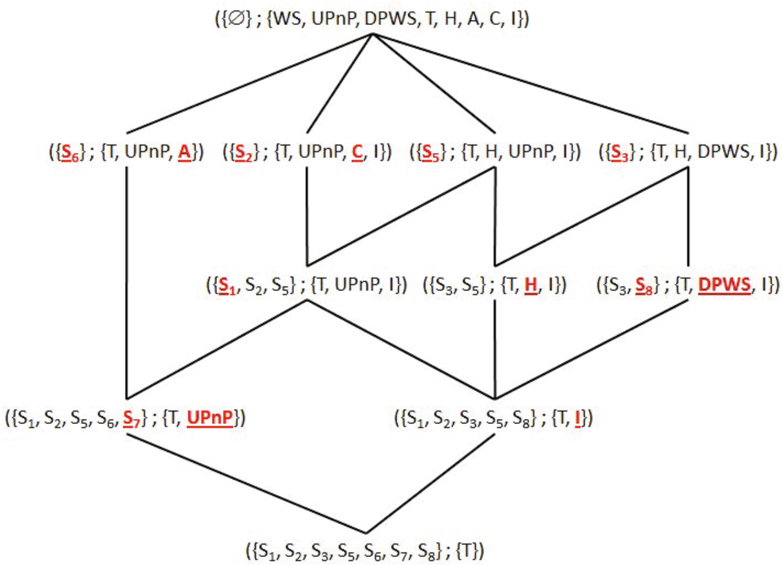


Fig. 5. Example of an extract lattice for *Temperature* activity

5.3 Algorithms for Selection

Depending on the expected response of the request (one or more services), it may be necessary to choose among all possible services. For this, it is possible to use different selection algorithms:

A selection based only on mandatory features: computation of a formal concept $(S; \mathbf{MF} \cup FF)$**.** Some naive but efficient algorithms with a complexity ($\mathcal{O}(1)$) will return one service among the extent S (*e.g.* "first", random). If the selected service is not available at runtime (realistic use case in pervasive computing), it is possible to search the extent S to choose another available service (algorithms with a complexity ($\mathcal{O}(n)$). The ability to use the extent without recomputing the formal concept is a major advantage of our solution.

A selection based on mandatory and optional features: computation of a decision structure $(S; \mathbf{MF} \cup FF) \;\cup\; \texttt{successors}(\mathbf{MF} \cup FF)$**.** In a decision structure, services are classified according to a set of optional features defined by the user at specification time. Being guided by selection policies (*i.e.* QoS-based, target environment description-based) when exploring the decision structure allows fine-grained service selection. In this case, the time complexity of the selection is linear in the number of successors of the root concept in the concept lattice.

For instance, with the decision structure previously built in Figure 5, an optional criterion for the user request *Temperature* can be the services implemented in the UPnP technology. Then, the right side of the tree can be pruned. Services S_1, S_2, S_5, S_6 and S_7 provide the functionality *Temperature* with an UPnP implementation.

In another example, let us consider that the user wants at least two *Temperature* services with, if possible, confidentiality and integrity properties for the data exchange. Only service S_2 provides the confidentiality (C) and integrity (I) properties and it is also implemented with UPnP technology ($\{S_2\}; \{T, UPnP, C, I\}$). However, thanks to the decision tree, the user can relax the constraints. Services S_1 and S_5 have the same features than S_2 but the confidentiality property ($\{S_1, S_2, S_5\}; \{T, UPnP, I\}$).

The ability to use the decision structure at runtime (without rebuilding) is also a major advantage of our solution.

6 Reacting to Service Availability at Runtime

In our approach, the decision structure is stored in memory in order to be reactive to events related to services (departure and re-arrival). In addition, we keep a pointer to the formal concept from which the service was selected. For instance, the service S_1 is a selected service extracted from the formal concept ($\{S_1, S_2, S_5\}; \{T, UPnP, I\}$) (Figure 5).

At runtime several scenarios can occur. For example, in the case of service departure, several options are possible:

- First, we can choose a replacement among the other services of the extent of the pointed concept (subject to the service availability). In the example, we can consider that S_1, S_2 and S_3 are owned by a same equivalent class; they have the same features.
- Second, if there is no available service in the extent of the pointed concept, we can use the decision tree for backtracking at runtime. More specifically by selecting a service among the extents in one (or more) of the predecessors (*i.e.* $\{S_1, S_2, S_5, S_6, S_7\}$ and ($\{S_1, S_2, S_3, S_5, S_8\}$). It is also possible to restart a partial tree search by relaxing a few constraints.

In this way, we can quickly and easily (at runtime) find a replacement service while ensuring the most appropriate configuration. In addition, our approach can also take into account the re-arrival of a more appropriate service.

7 Implementation and Validation

In this section, we present the implementation of our approach and the experimental results. This work has been validated in the European OSAMI project.

7.1 Implementation

To validate our approach, we made a Java implementation. Our implementation has been cut into three modules (Figure 6):

– A data acquisition module;
– A processing module, containing an algorithm for the formal concept computation;
– A renderer module, to display the results.

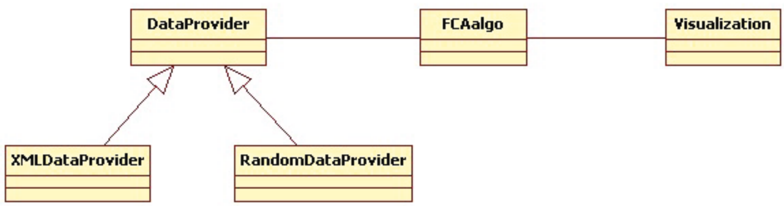


Fig. 6. Application architecture

For the needs of our experiment, two implementations have been realized. The first one loads static data from an XML file; the second uses a random number generator to generate data. Loading data from a static XML file allowed us to test our algorithm on a test bench; using the random number generator, we were able to test the robustness of our algorithm for big data sets. Data generation is performed in several stages. The first one generates the communication protocol (UPnP, DPWS), and the following ones the associated functional and non-functional properties.

To obtain valid data using random generation, this one was made in several stages. In every stage, attributes laws of generation are function of the the previous stages results. Then specificities associated with every technology are taken into account.

In the processing module, to obtain short processing times and low memory footprint with big data sets, we used bit fields, and took care of allocating only the strictly necessary memory.

The last part which is the renderer module, is implanted using JGraph[9], a graph drawing open source software component written in Java. This graphic feedback allowed a fast interpretation of the obtained lattices.

7.2 Experimental Results

In this section, we present the result of our experimentations. The goal of this work is to prove the feasability of our approach, *i.e.* introducing FCA in the

[9] http://www.jgraph.com/

service selection. Then, we have evaluated our work with a performance study. Experimentations have been made on a 2.20 GHz Intel Core Duo, 4GB of memory, Windows 7, 32 bits. We have fixed to 24 the number of attributes for the context model (3 technologies, 11 functionalities and 10 security properties).

Number of Computed Concepts. To test the feasability of our approach, we have evaluated the number of computed concepts according to the number of available services and the size of the request. The request contains:

– No constraint, *i.e.* equivalent to compute the entire lattice,
– One functional constraint, *i.e.* the minimal use case because the user knows at least the expected functionality,
– One functional constraint and the technology used to implement the service.

For this experimentation, we count the number of computed concepts (Figure 7) for these requests according to the available services in the context model.

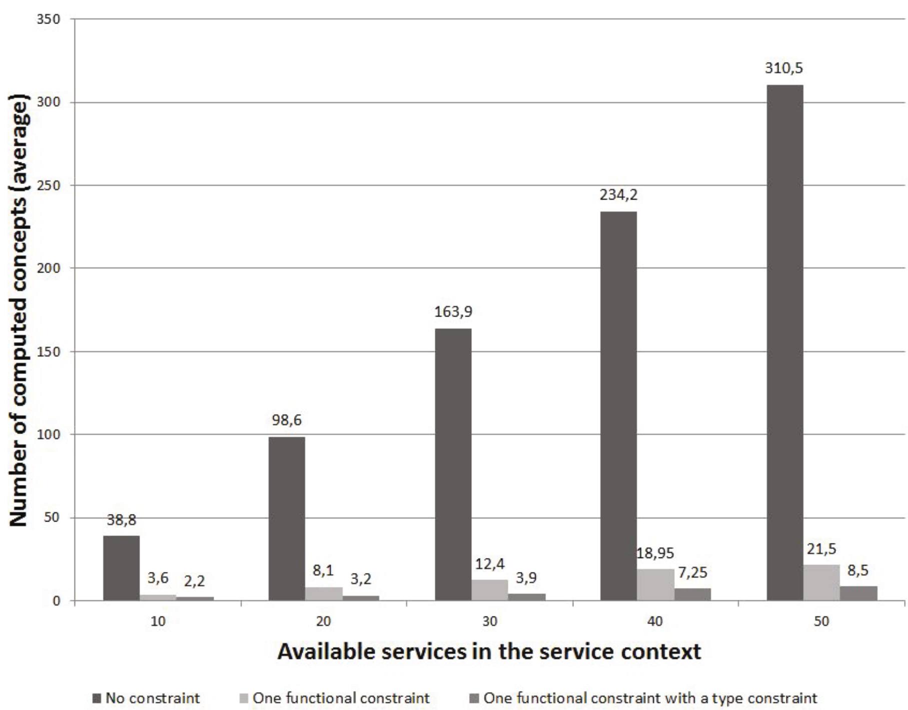


Fig. 7. Number of computed concepts in function of the available services

We note that the computation of only interesting concepts largely decreases the number of computed concepts. For a request based on one functionality, the decrease is 92% in average; for the second type of request, the decrease is 96%. Consequently, we have studied the computation time for evaluating the performance of our approach.

Computation Time. The major inconvenience of using FCA is the complexity of algorithm to compute a lattice: $\mathcal{O}((n+m) \times m \times |\mathcal{C}(\mathbb{K})|)$ where n is the number of attributes (properties) and m is the number of objects (available services). Even if we have decreased the number of computed concepts, we compute an ordered set of concepts. In a pervasive environment, we must propose this set in a "reasonable" time, *i.e.* due to the dynamicity of the application, and support a registry containing numerous services. For instance, a large building or plant can be approximately composed of one thousand devices (services). To test the reactivity of our approach, we have studied the computation time.

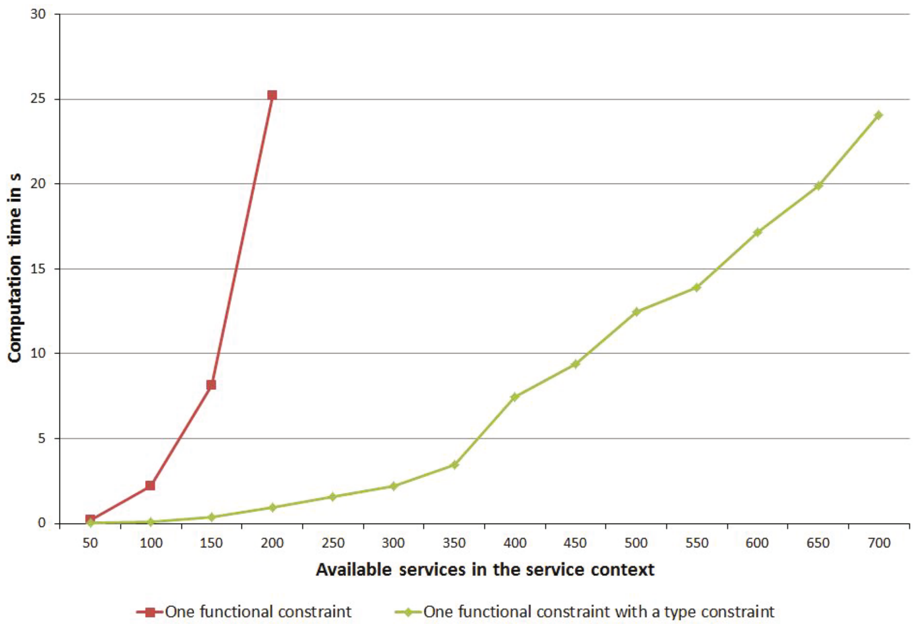


Fig. 8. Computation time as a function of the available services

In Figure 8, we have represented the computation time for two types of requests. We have not represented the computation time for a request with no constraint but it is approximately 160s for 50 available services and 2700s for 100 available services. This is clearly not realistic in a pervasive environment. However, for the selection, the request contains at least the service functionality and the computation time stays "reasonable" (25s) with 200 available services in the context. The request with one functionality constrained by a technology - more realistic request - can be executed also in 25s but for 700 available services.

8 Related Work

The problem of service selection, depending on service classification and FCA, has been studied by a few authors. Bruno et al. [5] propose an approach based on machine-learning techniques to support service classification and annotation. Peng et al. [14] classify Web Services in a concept lattice. Services are classified according to their functional operations regardless of non-functional aspects. Azmeh et al. [3] classify Web Services by their calculated QoS levels and composability modes. This classification is made with a Relational Concept Analysis approach, an extension of FCA. In these approaches, the inconvenient of using FCA is the large size of the concept lattice even if optimizations such as minimizing the number of attributes are proposed. However, these approaches are interesting in the assumption that the concept lattice is computed only once. But in the pervasive domain, services regularly appear and disappear, which means recalculating the lattice. Moreover, these approaches can not manage simultaneously different technologies (UPnP, DPWS...).

The key element of our approach is the service registry definition. We have adapted the service registry to the FCA context model. Ait Ameur [1] proposes to adapt the registry to a semantic registry in which semantic Web Services are stored. The introduction of ontologies allows to define a subsumption relationship between services that expresses a substituability relationship between these services. In our approach, ontologies can be added in the filter of the service registry in order to minimize the number of attributes in the context model.

9 Conclusion

Pervasive applications are made of heterogenous and dynamic services requiring a runtime adaptability and context-sensitive selections. In this paper, we have proposed an FCA-based solution for this selection problem. The integration and adaptation of FCA in our pervasive platform allows efficient runtime selection of heterogeneous and dynamic services. The characteristics of formal concept and decision structure avoid reiterate each time the selection algorithm which significantly improves performance at runtime. Our work has been applied on realistic use cases of the OSAMI European project and the results of experimentation show that it is possible to integrate FCA-based approach in dynamic context.

In the future, we will be interested in automating the selection in the decision structure of the appropiated service. We propose to replace the user by an autonomic manager to be more reactive to the context.

References

1. Ait-Ameur, Y.: A semantic repository for adaptive services. In: IEEE Congress on Services, pp. 211–218. IEEE Computer Society, Los Alamitos (2009)
2. Assaghir, Z., Kaytoue, M., Meira, W., Villerd, J.: Extracting decision trees from interval pattern concept lattices. In: Concept Lattices and their Applications (2011)

3. Azmeh, Z., Driss, M., Hamoui, F., Huchard, M., Moha, N., Tibermacine, C.: Selection of composable web services driven by user requirements. In: IEEE International Conference on Web Services, ICWS 2011, pp. 395–402. IEEE Computer Society, Los Alamitos (2011)
4. Bardin, J., Lalanda, P., Escoffier, C.: Towards an Automatic Integration of Heterogeneous Services and Devices. In: Proceedings of IEEE Asia-Pacific Services Computing Conference, pp. 171–178. IEEE Computer Society, Los Alamitos (2010)
5. Bruno, M., Canfora, G., Penta, M.D., Scognamiglio, R.: An approach to support web service classification and annotation. In: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 2005), pp. 138–143. IEEE Computer Society, Los Alamitos (2005)
6. Chollet, S., Lalanda, P., Bardin, J.: Service-Oriented Computing: from Web Services to Service-Oriented Components. In: Lee, J., Ma, S-P., Liu, A. (eds.) Service Life Cycle Tools and Technologies: Methods, Trends and Advances, pp. 1–20. IGI Global (2011)
7. Chollet, S., Lestideau, V., Lalanda, P., Maurel, Y., Colomb, P., Raynaud, O.: Building FCA-based Decision Trees for the Selection of Heterogeneous Services. In: SCC 2011: Proceedings of the 2011 IEEE International Conference on Services Computing, pp. 616–623. IEEE Computer Society, Washington, DC (2011)
8. Escoffier, C., Hall, R.S., Lalanda, P.: iPOJO: an Extensible Service-Oriented Component Framework. In: IEEE International Conference on Services Computing (SCC), pp. 474–481. IEEE Computer Society, Los Alamitos (2007)
9. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
10. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations. Springer, Heidelberg (1999)
11. Lalanda, P., Bourcier, J., Bardin, J., Chollet, S.: Development of service-oriented pervasive home applications. In: Al-Qutayri, M.A. (ed.) Smart Home Systems, pp. 1–16. I. Book (February 2010)
12. Nourine, L., Raynaud, O.: A fast incremental algorithm for building lattices. Journal of Experimental & Theoretical Artificial Intelligence 14(2-3), 217–227 (2002)
13. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering, Los Alamitos, CA, USA, pp. 3–12 (December 2003)
14. Peng, D., Huang, S., Wang, X., Zhou, A.: Management and retrieval of web services based on formal concept analysis. In: Proceedings of the The Fifth International Conference on Computer and Information Technology, pp. 269–275. IEEE Computer Society, Los Alamitos (2005)
15. Polaillon, G.: Interpretation and reduction of galois lattices of complex data. In: Rizzi, A., Vichi, M., Bock, H.-H. (eds.) Advances in Data Science and Classification, pp. 433–440. Springer (1998)

Publication Analysis of the Formal Concept Analysis Community

Stephan Doerfel, Robert Jäschke, and Gerd Stumme

Knowledge & Data Engineering Group,
University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany
http://www.kde.cs.uni-kassel.de/

Abstract. We present an analysis of the publication and citation networks of all previous editions of the three conferences most relevant to the FCA community: ICFCA, ICCS and CLA. Using data mining methods from FCA and graph analysis, we investigate patterns and communities among authors, we identify and visualize influential publications and authors, and we give a statistical summary of the conferences' history.

Keywords: bibliometrics, citation analysis, community, data mining, influence.

1 Introduction

On the occasion of the 10th anniversary of the *International Conference on Formal Concept Analysis (ICFCA)* we are presenting a quantitative and qualitative analysis of all papers published at the previous editions of ICFCA. Additionally, we included the two related conference series *International Conference on Conceptual Structures (ICCS)* and *Concept Lattices and their Applications (CLA)* to extend the range of analyzed publications relevant to Formal Concept Analysis.

Being active members of the FCA community, our intention for this analysis was to gain more insights into the structure of our community and its relationship to closely related disciplines. We will address questions that every researcher is asking himself from time to time, such as

– Which are the most influential authors, papers, and conferences?
– Who is cooperating with whom on which topics?
– Who is citing whom?

We will target these and other questions on three different levels: on the conference level, the author level, and the paper level.

This paper will allow long-term participants of one or more of these conference series to gauge their perception about their community. It may also allow newcomers a faster access to the community by being pointed to the must-read papers and to the different schools of thought that are attending these conferences. Last but not least, we intend to spark further research about our community's structure. To this end, we publicly provide the dataset which is underlying this paper's analysis at http://www.kde.cs.uni-kassel.de/datasets/

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 77–95, 2012.

The structure of this paper is as follows: In the next section, we discuss related work. Section 3 describes the dataset of publications in detail. In Section 4, we briefly introduce the various analysis methods that we used. Section 5 provides the results of the analysis – this is the main contribution of this paper. Finally, in Section 6, we briefly address future work.

2 Related Work

The field of research we are dealing with in this paper is *bibliometrics*, the science of analyzing (scientific) literature. Subjects of analysis are, among others, the statistical and structural properties of citation or collaboration networks and measures of influence and impact of publications, authors, journals or conferences. Given the multitude of bibliometric publications it is difficult to provide the most relevant pointers. A good starting point are dedicated journals, e.g., the *Scientometrics* journal.

Some recent analyses with a focus on (parts of) computer science include [8] and [1]. In the latter the authors discuss graph properties like connectivity and degree distributions in the citation graph of a publication corpus. An analysis of collaboration networks including the discussion of community structure and the small-world phenomenon is given in [8]. Tilley and Eklund use FCA for a qualitative analysis of 47 publications from software engineering in [15]. They relate publications to software-related activities and classify them by the lines of code of a particular programming language, applied in the publications.

Poelmans et al. combine text mining and FCA to provide a survey on the FCA literature related to knowledge discovery [10] (140 publications) and information retrieval [11] (103 publications). Using a thesaurus of relevant terms, the retrieved papers are classified and visualized using a concept lattice. In the sequel the focus of both papers is a detailed survey of some of the publications under study. An early practical application of FCA to the management of literature is presented in [12], where meta data of publications is used to search and visualize a given publication corpus.

In contrast to these previous papers we neither focus on a detailed analysis of a small publication corpus, nor on a rough statistical analysis of a large scale corpus. The medium size of our corpus (954 publications with 17121 citations) still allows us to look at specific authors or publications. We provide the first analysis of the three conference series, in particular the first analysis with a focus on FCA that is applied next to such diverse methods as graph partitioning and ranking.

3 Dataset

We first describe how we collected the publication corpus and then define the data structures upon which our analysis is based.

Table 1. Venues of the three conference series

ICCS	1993: Quebec City (CA), 1994: College Park (US), 1995: Santa Cruz (US), 1996: Sydney (AU), 1997: Seattle (US), 1998: Montpellier (FR), 1999: Blacksburg (US), 2000: Darmstadt (DE), 2001: Stanford (US), 2002: Borovets (BG), 2003: Dresden (DE), 2004: Huntsville (US), 2005: Kassel (DE), 2006: Aalborg (DK), 2007: Sheffield (UK), 2008: Toulouse (FR), 2009: Moscow (RU), 2010: Kuching (MY), 2011: Derby (UK)
ICFCA	2003: Darmstadt (DE), 2004: Sydney (AU), 2005: Lens (FR), 2006: Dresden (DE), 2007: Clermont-Ferrand (FR), 2008: Montreal (CA), 2009: Darmstadt (DE), 2010: Agadir (MA), 2011: Nicosia (CY)
CLA	2004: Ostrava (CZ), 2005: Olomouc (CZ), 2006: Hammamet (TN), 2007: Montpellier (FR), 2008: Olomouc (CZ), 2010: Sevilla (ES), 2011: Nancy (FR)

3.1 Gathering and Preprocessing

For our analysis we gathered meta data for all papers published at any of the past editions (up to 2011) of the three conference series ICCS, ICFCA, and CLA, i.e., 19 editions of ICCS, 9 editions of ICFCA, and 7 editions of CLA,[1] see Table 1. ICCS began as a conference on Conceptual Graphs (CG), with first FCA papers in 1995, and a balanced contribution of CG and FCA papers a few years later; while both ICFCA and CLA focus on FCA topics.

We collected data like paper titles, authors and their cited references from the publisher website SpringerLink[2] (ICCS and ICFCA) or extracted them from the paper's PDFs of CLA's website.[3] In our dataset, invited talks, regular and short papers are treated the same; poster sessions, satellite workshops as well as separate 'contributions' proceedings were not considered.

To gain knowledge about publications citing any of the conference papers, we retrieved citations from Microsoft Academic Search.[4] Note that these citations only roughly reflect the real number of citations a publication received, since this search engine relies on citation data that is available on the web and can only to a certain extent remove errors and correctly match different citation variants.

Our preprocessing included the extraction of authors, titles, years, and references from HTML and PDF files using regular expressions and manual work. Further, we implemented several normalization and completion steps for the titles and author names to allow matching and duplicate detection and an extensive manual error correction. Therefore, we employed the normalization steps described in [16] with an additional removal of diacritics (e.g., 'ä' and 'á' were

[1] The first edition of the CLA 2002 in Horní Bečva was a small seminar with four talks and hence no published proceedings exist.

[2] http://www.springerlink.com/

[3] http://cla.inf.upol.cz/papers.html

[4] http://academic.research.microsoft.com/

replaced by 'a'). We used different heuristics, e.g., the Levenshtein distance, to find errors in author names and titles. All references without authors (often encountered for cited web pages) were removed from the dataset.

Since many publications were cited as different editions or prior to their publication ('to appear'), we normalized the publication year by dating back different editions to the earliest mentioned date of publication. For example, the collected papers of Charles S. Peirce [47] were cited with different publication years (1931, 1935, 1953, 1958, 1966) which we normalized to 1931.

For the first ICFCA 2003 in Darmstadt no proceedings were published. Thus, we used the book from 2005 [33] which contains contributions from the participants of the first ICFCA on the state of the art on FCA and its applications.

Finally, we would like to point out that – since the focus of our analysis is on the three conference series – many publications related to FCA (in particular journal articles) have not been included in the dataset. The results presented in this paper should be interpreted with this fact in mind.

3.2 Notations and Derived Data Structures

From the collected data we derived several structures (graphs and formal contexts) that are described in detail in the following. All structures that use the references were created after removing self-citations (cited publications where one of the authors is also an author of the citing paper).

We denote the set of all authors that published at any of the three conferences by A and the set of all papers published at any of the conferences by P.

Authorship. The formal context $\mathbb{K}_{\mathrm{pa}} = (P, A, I_{\mathrm{pa}})$, with $(p, a) \in I_{\mathrm{pa}}$ iff a is an author of paper p, describes who authored which publication.

The graph of co-authorship $\mathfrak{G}_{\mathrm{coa}}$ is an undirected, weighted graph with A as node set. Two authors are connected, iff they published together and their edge's weight is the number of co-authored publications at the conferences.

In Section 5.2, we cluster (partition) $\mathfrak{G}_{\mathrm{coa}}$ and use these clusters as attributes of formal contexts. We denote by $C_n(\mathfrak{G}_{\mathrm{coa}})$ the set containing the n clusters with the highest cardinality.

Citations. The directed, weighted graph $\mathfrak{G}_{\mathrm{cit}}$ again has the authors in A as nodes. An edge (a, b) with weight w indicates that in all considered publications, w times, some publication of b was referenced by a.

Conferences. To analyze the distribution of all authors over the three conference series, we use $\mathbb{K}_{\mathrm{conf}} = (A, \{ICCS, ICFCA, CLA\}, \mathbb{N}, I_{\mathrm{conf}})$, a many-valued context where $(a, c, n) \in I_{\mathrm{conf}}$, iff a published exactly n papers at conference c.

4 Definitions and Methodology

In this section, we give a brief overview of the different algorithms and methods we use in our analysis. Most of the FCA notions are explained in great detail in the textbook [5]. In Section 5.2, we discuss the extents of an *iceberg lattice* of a context, i.e., an ordered subset of the concept lattice containing only concepts with extents larger (w.r.t cardinality) than a given threshold (minimum support). Iceberg lattices and a construction algorithm are explained in [13].

In the same section, we analyze *communities* of co-authorship. Intuitively, communities are certain subsets of some larger set of entities, such that the members of a subset are somewhat more related or similar to each other than they are to others. There is, however, no generally accepted formal definition of the notion of a "community". Here, by communities we mean the classes of a partitioning on the node set of a given graph. To create such a partitioning and its visualization for the co-authorship graph $\mathfrak{G}_{\text{coa}}$, we laid out the graph using the force directed graph visualization provided by Graphviz [4]. Then the *GMap* algorithm (again Graphviz) based on [9] was applied to discover communities of collaborators. GMap optimizes its output clustering w.r.t. *modularity*, which is a community quality measure that compares the number of co-author edges within each community to the expected value for this number in an equivalent random graph. Finally, *Voronoi diagrams* are used to draw the 'borders' between the different 'countries'.

In Section 5.2, we also apply different node centrality measures which indicate the importance of nodes within the citation graph $\mathfrak{G}_{\text{cit}}$. Next to the simple measures *in-degree* (number of edges pointing towards a node) and *in-strength* (sum of the weights of all edges pointing towards a node), we use *PageRank* [2] to rank authors of the conferences. PageRank is an eigenvector-based measure that was originally developed to measure the importance of web pages according to the link structure of the World Wide Web. To assign a score to each node in a graph, a linear equation system is solved which integrates the adjacency matrix of the graph and a probabilistic component. The main idea of the ranking is that important nodes are pointed to by other (important) nodes. In our scenario of citations, an author is considered important (i.e., has a high PageRank), if he or she is cited by many other important authors.

Based on a similar idea, the (also eigenvector-based) *HITS algorithm* [6] determines *hubs* and *authorities* in a graph. Roughly speaking, hubs are nodes that point to many good authorities in the graph. Authorities are those nodes that are referenced by many good hubs. In the citation graph, an author is a good hub, if he or she references many authors that have high values as authorities (e.g., authors of survey papers). Of interest for us, however, are the authorities, i.e., authors that have been cited by authors with high hub values.

5 Results

Now, we present the results of our analysis along the three dimensions of conferences (Section 5.1), authors (Section 5.2), and publications (Section 5.3).

5.1 Conferences

We start the section on conferences by some basic statistics (cf. Tables 2 and 3) that give an overview of the conference history. The two lower blocks of Table 2

Table 2. The history of the three conference series in numbers

	ICCS	ICFCA	CLA	total
editions	19	9	7	35
publications	567	208	179	954
avg. publications per edition	29.84	23.11	25.57	27.26
authors	542	218	269	872
avg. publications per author	2.04	1.94	1.62	2.25
‘outgoing’ citations (publications that have been cited by the conferences’ papers)				
citations	10131	4328	2662	17121
cited authors	5871	2655	2027	8513
cited publications	6079	2406	1668	8813
self-citations	2255 (≈22 %)	965 (≈22 %)	529 (≈20 %)	3749 (≈21 %)
‘incoming’ citations (conference papers that have been cited)				
citations	3202	1322	153	4677
citing publications	1776	985	134	2522
cited publications	404 (≈71 %)	128 (≈62 %)	47 (≈26 %)	579 (≈61 %)

show statistics for two types of citations: ‘outgoing’, i.e., citations we extracted from the conference papers, and ‘incoming’, i.e., publications that cite one of the papers published at one of the conferences. The fraction of 20–22 % self-citations is comparable to or lower than prior results (e.g., [14] reports 38 % for mathematical publications). The lower fraction of publications at ICFCA and CLA that have been cited (last row) can partly be explained by the young age of these two conferences.

Table 3. The top five contributing authors of each conference. In case of a tie all authors with the same number of publications are listed.

ICCS	ICFCA	CLA	total
R. Wille (24)	R. Wille (14)	S. Ben Yahia (13)	R. Wille (42)
G.W. Mineau (19)	P. Eklund (11)	R. Bělohlávek (11)	S.O. Kuznetsov (27)
J.F. Sowa (14)	P. Valtchev (10)	A. Napoli (10)	P. Eklund (26)
S.O. Kuznetsov (13)	B. Ganter (10)	E. Mephu Nguifo (8)	B. Ganter (24)
M. Keeler (13)	S.O. Kuznetsov (8)	V. Vychodil (7)	P. Valtchev (20)
	S. Ferré (8)	M. Huchard (7)	G.W. Mineau (20)
	L. Nourine (8)	J. Outrata (7)	

Publication Habits. To gain insights into the publication habits we consider the many-valued context $\mathbb{K}_{\text{conf}}$. Through conceptual scaling this context is transformed into the single-valued context

$$\mathbb{K}_{\text{freq}} = (A, \{CLA, ICCS, ICFCA, 3 \times CLA, 3 \times ICCS, 3 \times ICFCA\}, I_{\text{freq}})$$

where each author coincides with a conference if he or she published there at least once. An author incides with one of the other three attributes if he or she published at the corresponding conference at least three times. The threshold of three was selected since publishing three times at the same conference series indicates already a certain commitment to it. On the other hand, we did not set a higher value, since especially CLA and ICFCA are young conferences (seven and nine editions, resp.). The line diagram of the context's concept lattice is depicted in Figure 1, where the values below each concept count the number of authors in the concept extent (support values). Exemplarily, the top contributing authors from Table 3 are annotated at their object concepts. To interpret the lattice, one has to keep in mind that ICCS runs more than twice as long as the other two conference series, naturally resulting in higher author participation: 542 authors vs. 218 (ICFCA) and 269 (CLA). Of the 872 authors, 127 (14.6 %) published at least at two and 30 (3.4 %) of them at all three conference series.

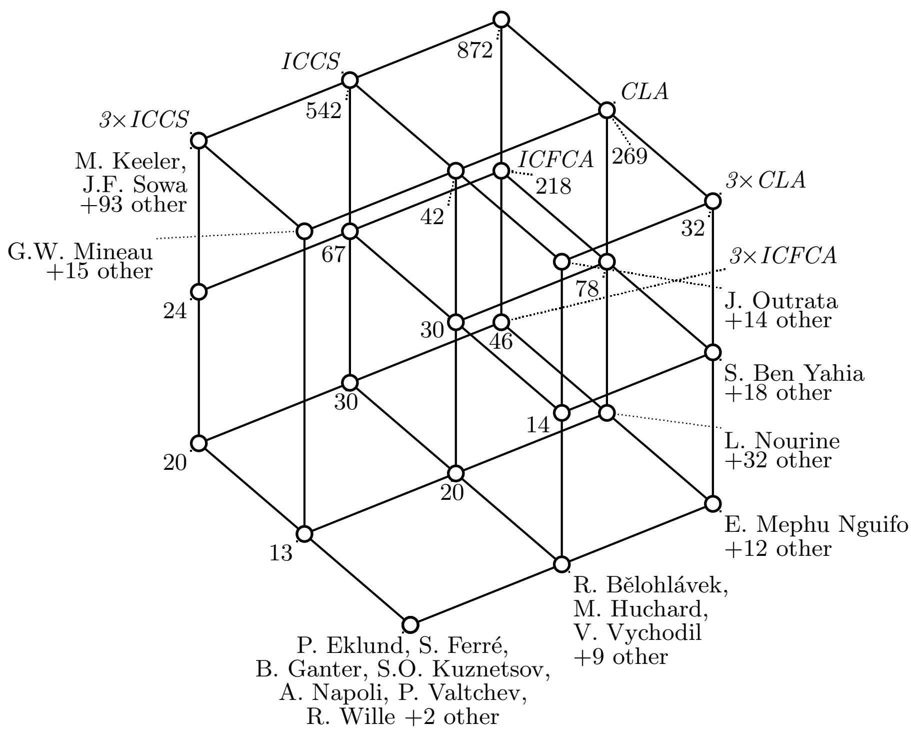


Fig. 1. The concept lattice for the author-conference context $\mathbb{K}_{\text{freq}}$, annotated with support values and the top contributing authors mentioned in Table 3

The Duquenne-Guigues base of implications contains – aside from the trivial rules resulting from the choice of scales – only two rules:

1. 3×ICCS and 3×CLA $\Longrightarrow$ 3×ICFCA
2. 3×ICCS and ICFCA and CLA $\Longrightarrow$ 3×ICFCA.

The first rule states that any author who frequently published at both *ICCS* and *CLA* also frequently published at ICFCA. Similar rules do not hold for the other combinations of conferences. However, several association rules with high confidence further confirm the bonds between the three conferences. The following list contains those rules with a confidence greater or equal to 80% (each given with its absolute support and confidence):

1. 3×CLA and ICCS $\Longrightarrow$ ICFCA (15/93 %)
2. 3×CLA and 3×ICFCA $\Longrightarrow$ ICCS (13/92 %)
3. 3×CLA and ICCS and ICFCA $\Longrightarrow$ 3×ICFCA (14/86 %)
4. 3×ICCS and ICFCA $\Longrightarrow$ 3×ICFCA (24/83 %)
5. 3×ICCS and CLA $\Longrightarrow$ 3×ICFCA (16/81 %).

Roughly speaking, these rules express the fact that many authors who frequently published a paper at ICCS or CLA also (frequently) published a paper at ICFCA.

Author Fluctuation. Now, we want to answer the question, *How many new authors can the conferences attract each year?* Therefore, we investigate for each year which fraction of authors of all accepted publications is 'new', i.e., has never before published a paper at the corresponding conference. As can be seen in Figure 2, for the first edition of each conference this fraction naturally is equal to 1 and has a decreasing trend for the immediately following years. On the contrary, the fraction of authors that appeared at a conference for the 'last' time (negative bars) naturally increases to -1 for last year's conferences. Therefore, we omitted the first (last) two editions of each conference for the calculation of the mean first (last) fractions. For all three conferences, on average, over half of the authors never published before at the conference. We conclude that the conferences are able to attract new authors each year. Similarly, on average, half of the authors did not publish again. Thus, there is a considerable exchange of authors and possibly ideas. For CLA, both values are considerably higher, meaning that this young conference still has a high fluctuation rate. Another observation is the steady increase of newcomers in the years from 2003 to 2007 for ICCS, followed by a sharp drop in 2008. This is also reflected by the absolute counts (not shown here) that drop from 58 'newcomers' in 2007 to only 15 in 2008 and the similar behaviour for those years with the 'last' authors. One explanation is given by the absolute numbers of authors for these years: 90 (2007) and 47 (2008), i.e., a decrease by a factor of two. Nevertheless, this might not be the only explanation, since in the following year 2009 only 40 authors published at ICCS but both the fraction of 'newcomers' and 'lasttimers' increases. We could not find a convincing explanation for this phenomenon, but plan to specifically compare the collaboration graphs of these years.

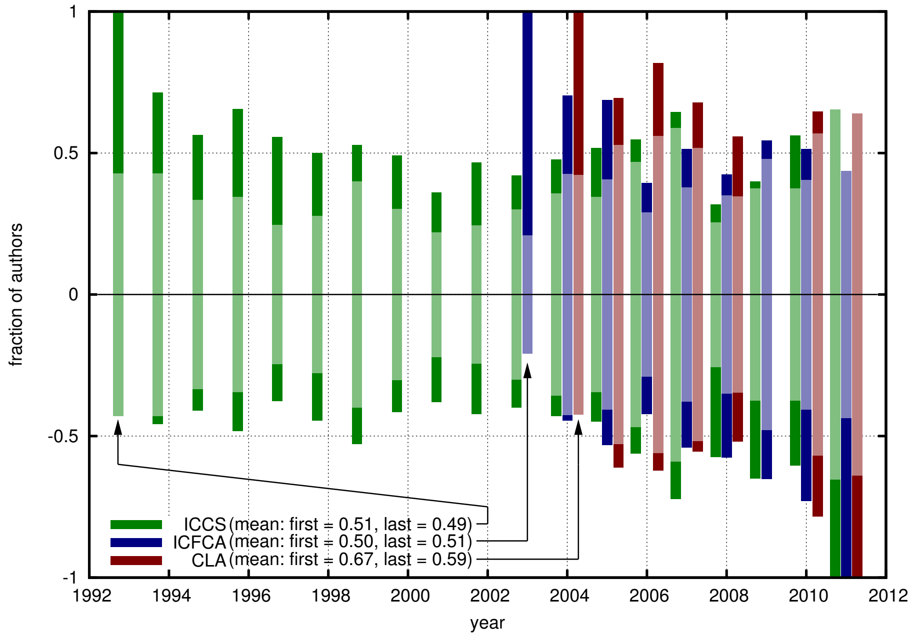


Fig. 2. Fluctuation of authors for each conference. The dark positive (negative) bars depict the fraction of authors that submitted a paper to the corresponding conference for the first (last) time in that year. The light bars in front of them depict the fraction of authors for which that year was also the only year (up to now) they submitted a paper (note that this measure is symmetric with respect to 'first' and 'last'.) For the calculation of the mean values for first (last), the first (last) two editions of each conference were omitted.

5.2 Authors

We analyze collaboration and influence between the authors of the conferences.

The Structure of the Community. First, we take a look at the co-authorship structure of the conferences. The most frequent collaborators can be read off from an iceberg lattice (frequent closed itemsets) of the publication-author-context $\mathbb{K}_{pa}$. Setting for instance the minimum support (minimum number of publications) to six, the following ten pairs[5] constitute the only (non singleton) intents of the iceberg lattice (given with their absolute support):[6] R. Bělohlávek/V. Vychodil (10), S. Ferré/O. Ridoux (9), J. Ducrou/P. Eklund (8), M.R. Hacene/P. Valtchev (8), P. Øhrstrøm/H. Schärfe (8), R. Godin/P. Valtchev (7), E. Mephu

[5] The fact that only pairs show up indicates that there were no teams of three or more authors who published more than six papers together.

[6] We do not show the iceberg lattice, due to space restrictions, and to the fact that it is structurally just an anti-chain.

Nguifo/S. Ben Yahia (7), M. Ducassé/S. Ferré (6), B. Ganter/S.O. Kuznetsov (6) and T. Hamrouni/S. Ben Yahia (6). Using a lower minimum support threshold of 4 yields another 12 concepts with 5 publications and 8 concepts with 4 publications in the extent. Among them are three concepts with intents containing more than just two authors: P. Cellier/M. Ducassé/S. Ferré (5), T. Hamrouni/E. Mephu Nguifo/S. Ben Yahia (5) and M.R. Hacene/M. Huchard/P. Valtchev (4).

The co-author graph $\mathfrak{G}_{\text{coa}}$ reveals interesting patterns of collaboration within and between the FCA and CG (Conceptual Graphs) communities. The map in Figure 3 shows a clustering created by GMap [3]. Connected components that contain less than four authors or that are based on less than four papers have been omitted for the sake of legibility. The width of the edges between two co-authors reflects the number of publications they have written together at any of the three conferences; similarly, the size of the author names depicts the number of published papers.

The giant connected component (GCC) of the graph is divided into 13 clusters (1–13) and contains 314 of the 482 authors shown on the map. The second largest component (clusters 14 and 15) contains the second largest cluster (14) with 52 members mostly belonging to the Conceptual Graph (CG) sub-community that is based in France. The remaining five large clusters (with more than ten members) are not connected. Based on our knowledge of the community they can roughly be classified to belong to the CG community (clusters 17–19) and to the FCA community (clusters 16 and 20). Adepts of the conferences can discover many further interesting aspects in this collaboration graph. Due to space restrictions we only want to outline that the CG community forms more separate clusters than the FCA community. Besides the five mentioned separate clusters, we consider only three of the 13 clusters of the GCC to be part of the core CG community (clusters 4, 5, and 9). Except for cluster 10 (the Description Logics community) all remaining clusters of the GCC belong to the FCA community. Finally, we would like to point out the remarkable role of G. Mineau (cluster 5) as a bridge between two CG clusters and the FCA community.

Topics of the Clusters. To get an idea about the topics that the authors of single clusters deal with, we visualize their citations of the most often cited publications and authors in two concept lattices (Figure 4). For legibility, we restrict this analysis to the set C_8 of the eight largest clusters (each containing more then 24 authors, while the others contain at most 14 authors), i.e., the clusters 1–7 and 14. Many different ways of choosing attribute sets and incidence relations are conceivable and it would be interesting to observe the influence of these choices. In this paper, we choose the following two examples for visualization: We construct the contexts $\mathbb{K}_{\text{cp}} = (C_8, P_{20}, I_{\text{cp}})$ and $\mathbb{K}_{\text{ca}} = (C_8, A_{C5}, I_{\text{ca}})$. Hereby, the set P_{20} contains the 20 most often cited publications of the corpus. In contrast to that, the set A_{C5} contains for each of the eight clusters the top five authors w.r.t to the number of papers – with at least one author from the cluster – that reference them. A cluster c is set in relation with a publication p (an author a), if p (a) is cited by at least three (five) papers from c. Figures 4(a) and 4(b) show the resulting lattice diagrams.

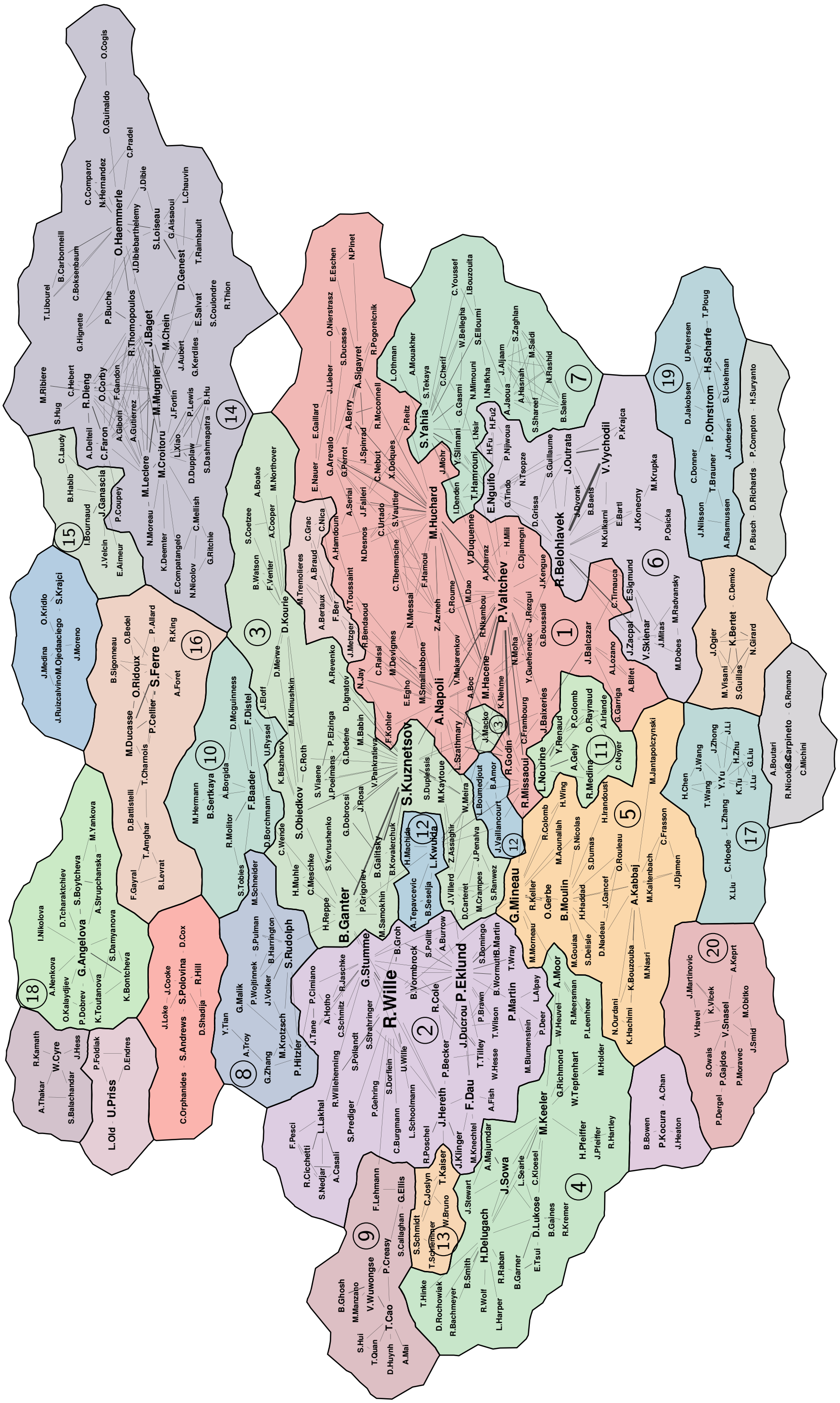


Fig. 3. A map of the co-author graph. Isolated 'islands' with less than four publications or less than four authors have been removed.

Both lattices seem to reflect the two main schools of the considered conferences: FCA and CG. Each cluster cites one of their cornerstone-publications ([60] and [54]) and their creators (R. Wille and J.F. Sowa). Clearly, clusters 1, 6 and 7 belong to the FCA community and clusters 4 and 14 to the CG community, while 2, 3 and 5 cite publications and authors from both. The philosophical foundations of C.S. Peirce are important for clusters 2 and 4. In the FCA community, we can see the high impact of the foundations book [5] by B. Ganter and R. Wille and of papers on implications and association rules. The topics of the papers further suggest that clusters 2 and 4 might be more interested in mathematical and philosophical foundations while clusters 1, 6 and 7 often cite important algorithmic publications.

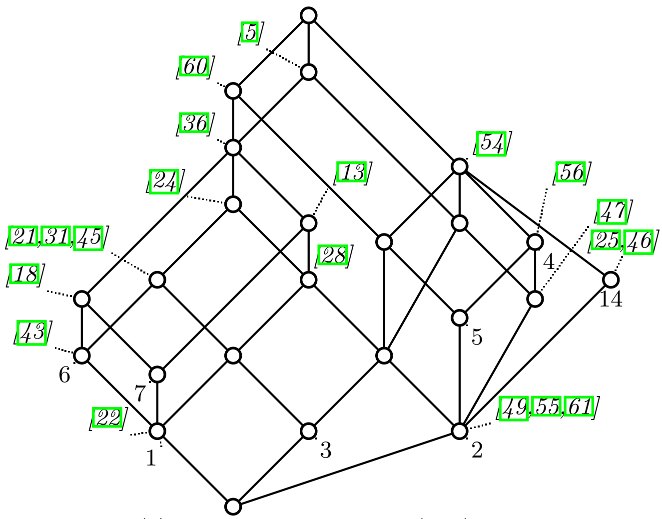
Table 4. Top ten rankings for the network analysis measures in-degree, in-strength, PageRank and authority (HITS, cf. Section 3.2) in $\mathfrak{G}_{\text{cit}}$

	in-degree		in-strength		PageRank		authority	
1	R. Wille	443	R. Wille	1877	J.F. Sowa	.101	R. Wille	.161
2	B. Ganter	424	B. Ganter	1322	R. Wille	.068	B. Ganter	.087
3	J.F. Sowa	307	J.F. Sowa	1033	B. Ganter	.043	G. Stumme	.042
4	G. Stumme	211	G. Stumme	570	M.-L. Mugnier	.021	L. Lakhal	.031
5	R. Godin	156	M.-L. Mugnier	427	M. Chein	.020	J.F. Sowa	.030
6	S.O. Kuznetsov	151	L. Lakhal	412	G. Ellis	.017	S. Prediger	.023
7	R. Missaoui	134	R. Godin	374	G. Stumme	.014	M.J. Zaki	.019
8	G.W. Mineau	128	M. Chein	360	O. Gerbé	.014	R. Godin	.019
9	L. Lakhal	127	S.O. Kuznetsov	349	S. Prediger	.013	S.O. Kuznetsov	.018
10	P. Eklund	124	C. Carpineto	264	G.W. Mineau	.011	C. Carpineto	.017

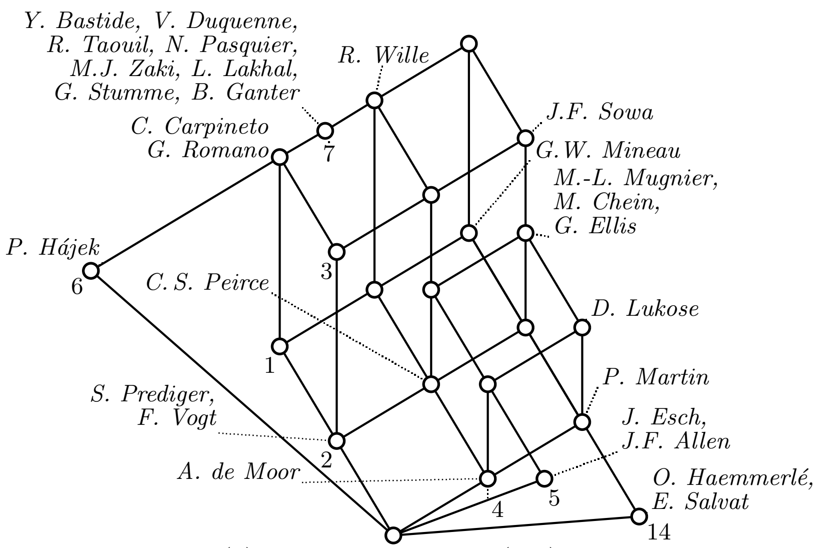
Influence. Finally, we use the author-citation graph $\mathfrak{G}_{\text{cit}}$ to identify key players, i.e., authors that are the most influential or the most central in the graph. Several centrality measures have been proposed (see, e.g., [7]). In Table 4 we present four rankings according to the different measures described in Section 4. One can observe that the different measures show a strong agreement. Note, that the scores are only valid within the investigated community of the three conferences, since we did only consider citations from papers published there. Thus, these figures do not make a general statement about the importance of the authors.

5.3 Publications

In this section, we take a closer look on individual publications and their citations. For each conference the first four rows of Table 5 list cited publications and citation counts and the top most cited publications for each conference and for the set of all sources other than the three conference series.



(a) The concept lattice $\underline{\mathfrak{B}}(\mathbb{K}_{\text{cp}})$.



(b) The concept lattice $\underline{\mathfrak{B}}(\mathbb{K}_{\text{ca}})$.

Fig. 4. The two lattices relate the eight largest clusters from Figure 3 as objects to the most often (in conference papers) cited publications and authors as attributes. The eight clusters are: 1 (P. Valtchev, A. Napoli, A.M.R. Hacene, ...), 2 (R. Wille, P. Eklund, F. Dau, ...), 3 (S.O. Kuznetsov, B. Ganter, S. Obiedkov, ...), 4 (J.F. Sowa, H.S. Delugach, M. Keeler, ...), 5 (G.W. Mineau, B. Moulin, A. Kabbaj, ...), 6 (R. Bělohlávek, V. Vychodil, E. Mephu Nguifo, ...), 7 (S. Ben Yahia, T. Hamrouni, Y. Slimani, ...) and 14 (J.-F. Baget, O. Haemmerlé, M.-L. Mugnier, ...).

The most often cited paper of ICCS at ICCS [61] paved the way for a connection of the two schools of research that are the foundation of ICCS, namely *Formal Concept Analysis* and *Conceptual Graphs*. As a general observation, the most often cited papers from ICCS are theory-minded, the most important papers from ICFCA equally present theory and applications of and for FCA. The most often cited papers from other sources include publications belonging to the foundations of the disciplines FCA [5,36,60] and CG [54].

While the first four rows of the table reveal the most important publications of and from each community, we take a closer look at the theoretical foundations of the conferences in its last row. It contains the most cited publications only from authors that never attended any of the conferences. Naturally, this excludes the well-known foundation papers of Ganter, Wille, Sowa, etc., but it reveals onto which (other) theories the conferences' main results are built. We can see a clear agreement between CLA and ICFCA about the most important foundational publication for both conferences, namely the book by Birkhoff [21]. Furthermore, *association rule mining* was an important topic at both conferences. For the ICCS – as one would assume – three publications of Peirce are the most often cited 'external' publications. Interestingly, the paper that laid the foundation for the *Semantic Web* [20] is the third most important paper in this category. This shows the influence of the Semantic Web community on the ICCS community.

6 Future Work

In this paper, we have analyzed the citation and collaboration behaviour of authors of the three FCA-related conferences ICCS, ICFCA, and CLA. The picture of the FCA community could be completed by adding further publications from journals and books. Finding relevant publications and retrieving their metadata and citations is clearly a first step for future work.

Since we intended to give a broad overview of many different aspects of the community, we naturally chose not to go into too much detail with only one specific aspect of the performed analyses. Each analysis could be extended to a comparison of different settings or methods, e.g., one might try different clustering algorithms to validate the communities found in Section 5.2. Therefore, with respect to space and time constraints, we did only deal with some of the questions relevant for the community and for newcomers. For example, the highly interesting structure of the FCA community that can be read off the co-author graph presented in Section 5.2 could be investigated further. Which kind of subcommunities exist? Which authors are bridges between different communities? Can roles like student, supervisor, etc. be identified? We also plan to validate our ad-hoc assignment of community labels by analyzing the titles and abstracts of the authors' papers. Thereby, it would be possible to explicitly assign authors to topics and thus get a clearer picture of how the community is constituted.

Table 5. The most often cited papers of a certain conference by papers of another conference. The first line of each cell reflects the number of cited papers and the number of citations. The following lines point to the top three citations, the first is given with title.

papers from this conference were cited	papers have been cited at this conference: ICCS	ICFCA	CLA
ICCS	249 publications in 737 citations 26× *Conceptual graphs and formal concept analysis* [61] 19× [53], 16× [50]	66 publications in 192 citations 11× *Pattern structures and their projections* [32] 11× *Conceptual graphs and formal concept analysis* [61] 11× *Boolean concept logic* [62]	33 publications in 51 citations 5× *Pattern structures and their projections* [32] 4× [42], 4× [44]
ICFCA	38 publications in 60 citations 6× *The ToscanaJ suite for implementing conceptual information systems* [19] 5× [59], 4× [27], 4× [26]	60 publications in 120 citations 6× *The ToscanaJ suite for implementing conceptual information systems* [19] 6× *Formal concept analysis for knowledge discovery and data mining: the new challenges* [58] 5× [27], 5× [35], 5× [41], 5× [30], 5× [57]	63 publications in 100 citations 5× *Machine learning and formal concept analysis* [40] 5× *Formal concept analysis for knowledge discovery and data mining: the new challenges* [58] 5× *Analysis of social communities with iceberg and stability-based concept lattices* [37]
CLA	10 publications in 10 citations (at most one citation per paper)	11 publications in 13 citations 2× *What is a fuzzy concept lattice?* [23] 2× *Camelis: Organizing and browsing a personal photo collection with a logical information system* [29]	19 publications in 31 citations 3× *What is a fuzzy concept lattice?* [23] 3× *Towards concise representation for taxonomies of epistemic communities* [52] 3× *The basic theorem on generalized concept lattice* [39] 3× *Parallel recursive algorithm for FCA* [38]
other	4686 publications in 7069 citations 284× *Conceptual structures: information processing in mind and machine* [54] 100× [5], 65× [56]	1877 publications in 3038 citations 139× *Formal concept analysis: mathematical foundations* [5] 32× [60], 26× [36]	1218 publications in 1951 citations 124× *Formal concept analysis: mathematical foundations* [5] 30× [60], 24× [36]
external	3674 publications in 4708 citations 43× *Collected papers* [47] 19× [51], 15× [20], 15× [48]	1304 publications in 1741 citations 25× *Lattice theory* [21] 14× [45], 12× [17]	825 publications in 1041 citations 16× *Lattice theory* [21] 12× [22], 11× [17]

A dimension we could not analyze in the scope of this paper is *time*. Such an analysis would reveal developments and trends of the conferences. It could also allow us to judge the vitality of the communities in the co-author graph.

We would like to invite interested researchers to collectively tackle the above-mentioned challenges. The dataset is freely available,[7] extensions and error corrections are welcome and will be added to the dataset's web page. The meta-data of all publications referenced in this paper is available in BibSonomy at `http://www.bibsonomy.org/group/kde/citedBy:doerfel2012publication`

Acknowledgement. Part of this research was funded by the DFG in the project "Info 2.0 – Informationelle Selbstbestimmung im Web 2.0".

References

1. An, Y., Janssen, J., Milios, E.E.: Characterizing and mining the citation graph of the computer science literature. Knowledge and Information Systems 6(6), 664–678 (2004)
2. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (1998)
3. Gansner, E.R., Hu, Y., Kobourov, S.G.: GMap: Drawing graphs as maps. cs.CG, arXiv:0907.2585v1 (July 2009)
4. Gansner, E.R., North, S.C.: An open graph visualization system and its applications to software engineering. Software Practice & Experience 30(11), 1203–1233 (2000)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
6. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM 46, 604–632 (1999)
7. Koschützki, D., Lehmann, K.A., Peeters, L., Richter, S., Tenfelde-Podehl, D., Zlotowski, O.: Centrality Indices. In: Brandes, U., Erlebach, T. (eds.) Network Analysis. LNCS, vol. 3418, pp. 16–61. Springer, Heidelberg (2005)
8. Newman, M.E.J.: The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences 98(2), 404–409 (2001)
9. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103(23), 8577–8582 (2006)
10. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formal Concept Analysis in Knowledge Discovery: A Survey. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 139–153. Springer, Heidelberg (2010)
11. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G., Kuznetsov, S.O.: Text mining scientific papers: a survey on FCA-based information retrieval research. In: Perner, P. (ed.) Industrial Conference on Data Mining - Poster and Industry Proceedings, pp. 82–96. IBaI Publishing (2011)
12. Rock, T., Wille, R.: Ein TOSCANA–Erkundungssystem zur Literatursuche. FB4-Preprint 1901, TH Darmstadt (1997)
13. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with titanic. Data & Knowledge Engineering 42(2), 189–222 (2002)

[7] `http://www.kde.cs.uni-kassel.de/datasets/`

14. Thijs, B., Glänzel, W.: The influence of author self-citations on bibliometric meso-indicators. The case of european universities. Scientometrics 66(1), 71–80 (2006)
15. Tilley, T., Eklund, P.: Citation analysis using formal concept analysis: A case study in software engineering. In: 18th International Workshop on Database and Expert Systems Applications (DEXA), pp. 545–550. IEEE Computer Society Press (September 2007)
16. Voss, J., Hotho, A., Jäschke, R.: Mapping bibliographic records with bibliographic hash keys. In: Kuhlen, R. (ed.) Information: Droge, Ware oder Commons?, Proceedings of the ISI. Hochschulverband Informationswissenschaft, Verlag Werner Hülsbusch (2009)

References of the Analyzed Publications

17. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994)
18. Barbut, M., Monjardet, B.: Ordre et classification: algèbre et combinatoire. Hachette, Paris (1970)
19. Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 324–348. Springer, Heidelberg (2005)
20. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 284(5), 34–43 (2001)
21. Birkhoff, G.: Lattice Theory, 3rd edn. American Mathematical Society, Providence (1967)
22. Bordat, J.P.: Calcul pratique du treillis de galois d'une correspondance. Informatiques et Sciences Humaines 96, 31–47 (1986)
23. Bělohlávek, R., Vychodil, V.: What is a fuzzy concept lattice? In: CLA 2005, Proceedings of the 3rd International Workshop, Olomouc, vol. 162, pp. 34–45. CEUR-WS.org (2005)
24. Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications. John Wiley & Sons, Chichester (2004)
25. Chein, M., Mugnier, M.-L.: Conceptual graphs: fundamental notions. Revue d'Intelligence Artificielle 6(4), 365–406 (1992)
26. Cimiano, P., Hotho, A., Stumme, G., Tane, J.: Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 189–207. Springer, Heidelberg (2004)
27. Dau, F., Klinger, J.: From Formal Concept Analysis to Contextual Logic. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 81–100. Springer, Heidelberg (2005)
28. Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge University Press, Cambridge (1990)
29. Ferré, S.: Camelis: Organizing and browsing a personal photo collection with a logical information system. In: Proceedings of the Fifth International Conference on Concept Lattices and Their Applications, Montpellier, vol. 331, pp. 112–123. CEUR-WS.org (2007)
30. Freese, R.: Automated Lattice Drawing. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 112–127. Springer, Heidelberg (2004)
31. Ganter, B.: Two basic algorithms in concept analysis. FB4-Preprint 831, TH Darmstadt (1984)

32. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
33. Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis. LNCS (LNAI), vol. 3626. Springer, Heidelberg (2005)
34. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
35. Godin, R., Valtchev, P.: Formal Concept Analysis-Based Class Hierarchy Design in Object-Oriented Software Development. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 304–323. Springer, Heidelberg (2005)
36. Guigues, J.-L., Duquenne, V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires. Mathématiques et Sciences Humaines 95, 5–18 (1986)
37. Jay, N., Kohler, F., Napoli, A.: Analysis of Social Communities with Iceberg and Stability-Based Concept Lattices. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 258–272. Springer, Heidelberg (2008)
38. Krajca, P., Outrata, J., Vychodil, V.: Parallel recursive algorithm for FCA. In: Proceedings of the Sixth International Conference on Concept Lattices and Their Applications, Olomouc, vol. 433, pp. 71–82. CEUR-WS.org (2008)
39. Krajči, S.: The basic theorem on generalized concept lattice. In: Proceedings of the 2nd International Workshop on CLA 2004, Ostrava, pp. 25–33 (2004)
40. Kuznetsov, S.O.: Machine Learning and Formal Concept Analysis. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 287–312. Springer, Heidelberg (2004)
41. Kuznetsov, S.O., Obiedkov, S.: Counting Pseudo-intents and #P-completeness. In: Missaoui, R., Schmidt, J. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3874, pp. 306–308. Springer, Heidelberg (2006)
42. Kuznetsov, S.O., Obiedkov, S., Roth, C.: Reducing the Representation Complexity of Lattice-Based Taxonomies. In: Priss, U., Polovina, S., Hill, R. (eds.) ICCS 2007. LNCS (LNAI), vol. 4604, pp. 241–254. Springer, Heidelberg (2007)
43. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence 14(2-3), 189–216 (2002)
44. Lehmann, F., Wille, R.: A Triadic Approach To Formal Concept Analysis. In: Ellis, G., Rich, W., Levinson, R., Sowa, J.F. (eds.) ICCS 1995. LNCS, vol. 954, pp. 32–43. Springer, Heidelberg (1995)
45. Maier, D.: The Theory of Relational Databases. Computer Science Press, Rockville (1983)
46. Mugnier, M.-L., Chein, M.: Représenter des connaissances et raisonner avec des graphes. Revue d'Intelligence Artificielle 10(1), 7–56 (1996)
47. Peirce, C.S.: Collected Papers, pp. 1931–1935. Harvard University Press, Cambridge
48. Peirce, C.S.: Reasoning and the Logic of Things. Harvard University Press, Cambridge (1992)
49. Prediger, S.: Kontextuelle Urteilslogik mit Begriffsgraphen: ein Beitrag zur Restrukturierung der mathematischen Logik. Shaker, Aachen (1998)
50. Prediger, S.: Simple Concept Graphs: A Logic Approach. In: Mugnier, M.-L., Chein, M. (eds.) ICCS 1998. LNCS (LNAI), vol. 1453, pp. 225–239. Springer, Heidelberg (1998)

51. Roberts, D.: The Existential Graphs of Charles S. Peirce. Mouton, The Hague (1973)
52. Roth, C., Obiedkov, S., Kourie, D.: Towards Concise Representation for Taxonomies of Epistemic Communities. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) CLA 2006. LNCS (LNAI), vol. 4923, pp. 240–255. Springer, Heidelberg (2008)
53. Salvat, E., Mugnier, M.-L.: Sound and Complete Forward and Backward Chainings of Graph Rules. In: Eklund, P., Ellis, G., Mann, G. (eds.) ICCS 1996. LNCS, vol. 1115, pp. 248–262. Springer, Heidelberg (1996)
54. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)
55. Sowa, J.F.: Conceptual graphs summary. In: Eklund, P., Nagle, T., Nagle, J., Gerholz, L. (eds.) Conceptual Structures: Current Research and Practice, pp. 3–51. Ellis Horwood (1992)
56. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole Publishing, Pacific Grove (2000)
57. Tilley, T., Cole, R., Becker, P., Eklund, P.: A Survey of Formal Concept Analysis Support for Software Engineering Activities. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 250–271. Springer, Heidelberg (2005)
58. Valtchev, P., Missaoui, R., Godin, R.: Formal Concept Analysis for Knowledge Discovery and Data Mining: The New Challenges. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 352–371. Springer, Heidelberg (2004)
59. van der Merwe, D., Obiedkov, S., Kourie, D.: AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 372–385. Springer, Heidelberg (2004)
60. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)
61. Wille, R.: Conceptual Graphs and Formal Concept Analysis. In: Lukose, D., Delugach, H., Keeler, M., Searle, L., Sowa, J. (eds.) ICCS 1997. LNCS, vol. 1257, pp. 290–303. Springer, Heidelberg (1997)
62. Wille, R.: Boolean Concept Logic. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 317–331. Springer, Heidelberg (2000)

Understanding the Semantic Structure of Human fMRI Brain Recordings with Formal Concept Analysis

Dominik Endres[1,⋆], Ruth Adam[2,⋆], Martin A. Giese[1], and Uta Noppeney[2]

[1] Sect. Computational Sensomotorics, Dept. Cognitive Neurology, CIN, HIH, BCCN and University Clinic Tübingen, Frondsbergstr. 23, 72070 Tübingen, Germany
dominik.endres@klinikum.uni-tuebingen.de, martin.giese@uni-tuebingen.de

[2] Cognitive Neuroimaging Group, Max Planck Institute for Biological Cybernetics, Tübingen, Germany
{ruth.adam,uta.noppeney}@tuebingen.mpg.de

Abstract. We investigate whether semantic information related to object categories can be obtained from human fMRI BOLD responses with Formal Concept Analysis (FCA). While the BOLD response provides only an indirect measure of neural activity on a relatively coarse spatio-temporal scale, it has the advantage that it can be recorded from humans, who can be questioned about their perceptions during the experiment, thereby obviating the need of interpreting animal behavioral responses. Furthermore, the BOLD signal can be recorded from the whole brain simultaneously. In our experiment, a single human subject was scanned while viewing 72 gray-scale pictures of animate and inanimate objects in a target detection task. These pictures comprise the formal objects for FCA. We computed formal attributes by learning a hierarchical Bayesian classifier, which maps BOLD responses onto binary features, and these features onto object labels. The connectivity matrix between the binary features and the object labels can then serve as the formal context. In line with previous reports, FCA revealed a clear dissociation between animate and inanimate objects with the inanimate category also including plants. Furthermore, we found that the inanimate category was subdivided between plants and non-plants when we increased the number of attributes extracted from the BOLD response. FCA also allows for the display of organizational differences between high-level and low-level visual processing areas. We show that subjective familiarity and similarity ratings are strongly correlated with the attribute structure computed from the BOLD signal.

Keywords: fMRI, inferior temporal cortex, semantic neural decoding.

1 Introduction

Understanding how semantic information is represented in the brain has been an important research focus of neuroscience in the past few years. A large part of

⋆ Equal contribution.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 96–111, 2012.

this research studies object representation in the visual cortex, which we will also concentrate on in this paper. Experimentally, this question has been addressed using physiological and brain imaging techniques, specifically electrophysiological single/multi-cell recordings [16] and fMRI BOLD (functional magnetic resonance imaging, blood-oxygenation-level-dependent) responses [2]. The former have the advantage of providing a direct measure of neural electrical activity. However, one can usually record only from a relatively small population of neurons. Furthermore, the experimental animals cannot easily be questioned about their semantic perceptions. Nevertheless, it was previously shown [6] that formal concept analysis (FCA, [11]) can reveal interpretable semantic information (e.g. specialization hierarchies, or indications of a feature-based representation) from electrophysiological data. Here, we investigate whether similar findings can be obtained from BOLD responses recorded from human subjects. fMRI measures BOLD changes which are indirectly related to neuronal activity. Increased neuronal activity (e.g. due to visual input) in a specific brain area increases blood flow to this area which changes the local ratio between oxygenated (containing oxygen) blood which is diamagnetic and deoxygenated (without oxygen) blood which is paramagnetic. This change in the local magnetic properties of the blood is the BOLD signal detected by fMRI [27]. While the BOLD response provides only an indirect measure of neural activity on a much coarser spatio-temporal scale than electrophysiological recordings, it has the advantage that it can be recorded from humans, which can be questioned about their perceptions during the experiment, thereby obviating the need of interpreting animal behavioral responses. Furthermore, the BOLD signal can be recorded from the whole brain simultaneously.

Our paper is structured as follows: in section 2 we give a brief overview of the organization of the visual system and previous research on the representation of semantic information in the brain. Section 3 introduces the basic ideas of FCA. We describe the experiment in section 4, and the Bayesian feature extractor for computing the formal context from BOLD signals in section 5. Our results are detailed in section 6, and section 7 offers some concluding remarks and avenues of further investigation.

2 Organization of Visual Processing in Humans and Previous Research

This section contains a very brief and incomplete overview of the visual processing pathways in humans and monkeys, for details the reader is referred to [15]. Visual processing begins in the eye. Patterns of light falling onto the retina are converted into electrical signals, which are relayed to the primary visual cortex (V1) by the lateral geniculate nuclei. From the primary visual cortex the information is channeled to visual association cortices and thereafter distributed into two paralleled processing streams: the ventral stream and the dorsal stream [21].

The dorsal (”where”) occipitoparietal stream analyzes object location, guides object-related action and sends information to the parietal cortex. In the ventral (”what”) occipitotemporal stream, which is involved in object identification, information is directed to the inferior temporal (IT) cortex. The human IT contains sub-regions which selectively respond to specific object categories. For example, faces selectively activate the fusiform gyri, whereas landmarks and scenes activate the parahippocampal gyri [23].

However, it is unlikely that there is a specific area in the brain dedicated to every category we encounter in our daily life. Haxby and colleagues could show that activation patterns elicited by various object categories such as faces, cats and shoes were distinct and at the same time overlapping in the IT [13]. Multivoxel pattern analysis decoding techniques applied to the relevant subregions could discriminate between ordinate (basic) levels of a certain category (e.g. beach vs. highway scenes, [28]) as well as between object exemplars (e.g. two different chairs, [5]), showing that those areas also contain information up to the exact object identity. Standard encoding and decoding analyses often compare brain activations evoked by pre-specified object categories (e.g. face vs. house), and are therefore frequently driven as much by result expectations as by the data. Hypothesis-free analyses are especially important for complex stimuli, such as object categories, which cannot be easily grouped *a priori* to account for the entire conceivable feature-space.

One clear advantage of FCA is thus, that it does not require *a priori* grouping of the stimuli. In line with previous findings [13], FCA also allows for the comparison of activation patterns and thus takes into account the distributed and overlapping representation of objects in the brain. Another data-driven approach was applied recently to fMRI data, by computing dissimilarity matrices from fMRI activation patterns. This analysis applied to the IT has revealed hierarchically-organized animate and inanimate clusters [18]. However, we believe that comparing dissimilarity matrices is not sufficient to understand the structure of the representation of visual stimuli in the brain. First, stimulus arrangement is based on pairwise distances and as such does not directly regard the relations between multiple stimuli. Also, pairwise distances are often being further analyzed via hierarchical, tree-structured clustering, while a lattice-based structure may be more appropriate for the study of the cortical representation of complex objects composed of many overlapping features. Second, dissimilarity coefficients are often derived with linear methods, while the brain is known to be a highly non-linear system. Third, dissimilarity analysis does not allow incremental analysis since adding more stimuli or running the analysis with more BOLD data might change the observed dissimilarity pattern. Finally, and most importantly, the connection between stimuli and brain activation pattern observed is not explicitly represented in the dissimilarity matrices. Since FCA provides this connection via concepts and their ordering relation, we therefore decided to investigate if FCA was a suitable tool for elucidating the structure of the representation of (visual) stimuli in the brain.

3 Formal Concept Analysis

We now provide basic definitions and notation used in the following, for a full introduction to Formal Concept Analysis (FCA) see [11]. The *formal context* $K := (G, M, I)$ is comprised of a set of formal objects G, a set of formal attributes M and a binary relation $I \subseteq G \times M$ between members of G and M. The adjective "formal" indicates that these objects and attributes represent abstract entities, although it can be helpful to think of them as actual physical objects and their properties. We will drop "formal" for brevity, except in definitions. In our application, the members of G are visual stimuli, whereas the members of M correspond to binary features computed from a generative model representation of BOLD signals recorded in response to these stimuli (see section 5). If attribute $m \in M$ is used in the representation of the BOLD response to stimulus $g \in G$, then we write $(g, m) \in I$ or gIm. It is customary to represent the context as a cross table (incidence table), where the row(column) headings are the object(attribute) names. For each pair $(g, m) \in I$, the corresponding cell in the cross table has an "x". The table in fig. 1, left, shows a simple example context.

The derivation operator for subsets $X \subseteq G$ is defined as $X' = \{m \in M | \forall g \in X : gIm\}$ i.e. X' is the set of all attributes shared by the objects in X. Likewise, for $Y \subseteq M$ define $Y' = \{g \in G | \forall m \in Y : gIm\}$ i.e. Y' is the set of all objects having all attributes in Y.

Definition 1. *[11] A **formal concept** of the context K is a pair (X, Y) with $X \subseteq G$, $Y \subseteq M$ such that $X' = Y$ and $Y' = X$. X is called the **extent** and Y is the **intent** of the concept (X, Y). $\mathbb{B}(K)$ denotes the set of all concepts of the context K.*

Thus, given the relation I, (X, Y) is a concept if X determines Y and vice versa. X and Y are also called *closed* subsets of G and M with respect to I. For a representation of the relationships between concepts, one defines an order on $\mathbb{B}(K)$:

Definition 2. *[11] If (X_1, Y_1) and (X_2, Y_2) are concepts of a context, (X_1, Y_1) is a **subconcept** of (X_2, Y_2) if $X_1 \subseteq X_2$ (which is equivalent to $Y_1 \supseteq Y_2$). In this case, (X_2, Y_2) is a **superconcept** of (X_1, Y_1) and we write $(X_1, Y_1) \leq (X_2, Y_2)$. The relation $\leq$ is called the **order** of the concepts.*

It can be shown [29,11] that $\mathbb{B}(K)$ and the concept order form a complete lattice. The middle and right panels of fig. 1 depict lattice diagrams corresponding to the context in the left panel. In the diagrams, each node is a concept, the arrows indicate the concept ordering. Full labeling (fig. 1, middle) means that a concept node is drawn with its full extent and intent. A reduced labeled concept lattice (fig. 1, right) shows an object only in the smallest (w.r.t. the concept order of definition 2) concept of whose extent the object is a member. This concept is called the *object concept*, or the concept that *introduces* the object. Likewise, an attribute is shown only in the largest concept of whose intent the attribute is a member, the *attribute concept*, which *introduces* the attribute.

Formal Context

	m0	m1	m2
bear	×		
dog	×		
cabbage		×	
drum			×
pumpkin		×	×

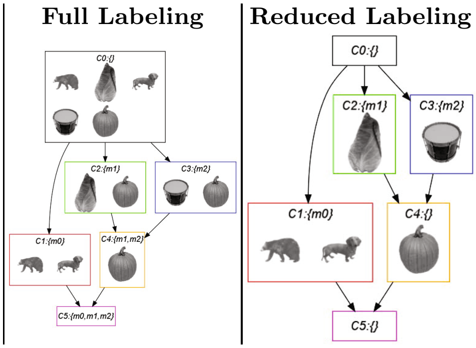


Fig. 1. A simple example context and corresponding lattice diagrams. *Left*: the formal context, represented as a cross-table. The objects (rows) are 5 visual stimuli, each of which can have a subset of 3 attributes (columns) m0,m1,m2, that are computed from (hypothetical) BOLD responses. *Middle*: fully labeled concept lattice. Each rectangle is a concept. The extents are represented by stimulus images, the top of each concept shows concept number and intent, e.g. '$C4 : \{m1, m2\}$' means: concept 4 has intent $\{m1, m2\}$. Concept numbers are computed from the lectic order on the attributes [11]. Arrows indicate concept ordering. Concept 1 comprises the animals, concept 2 contains only vegetables and concept 3 all objects with prominent round parts. Consequently, concept 4 can be thought of as the 'round vegetable' concept. *Right*: concept lattice with reduced labeling. Here, objects are only depicted in the most specific (smallest) concept which contains them, whereas an attribute is only shown in the most general (greatest) concept of whose intent it is a member.

The lattice diagrams is a graphically explicit representation of the ordering relationships between the concepts: concept 2 contains all vegetables, concept 3 comprises the objects with prominent round parts. They have a common child, concept 4, which is the 'round vegetable' (pumpkin) concept. The 'animals' concept (concept 1) is incomparable to any other concept except the top and the bottom of the lattice. Note that these relationships arise as a consequence of the (here hypothetical) BOLD responses. We will show (section 6) that real BOLD responses lead to similarly interpretable structures when one computes attributes from suitable brain regions.

To reiterate, in the following we will denote the set of visual stimuli by G, and the set of attributes computed from BOLD responses by M, and their incidence relation by $I \subseteq G \times M$.

4 fMRI Experiment

4.1 Experimental Methods and Data Preprocessing

Subject. A single right handed, German native speaker, male subject participated in this fMRI study. The subject gave informed written consent prior to

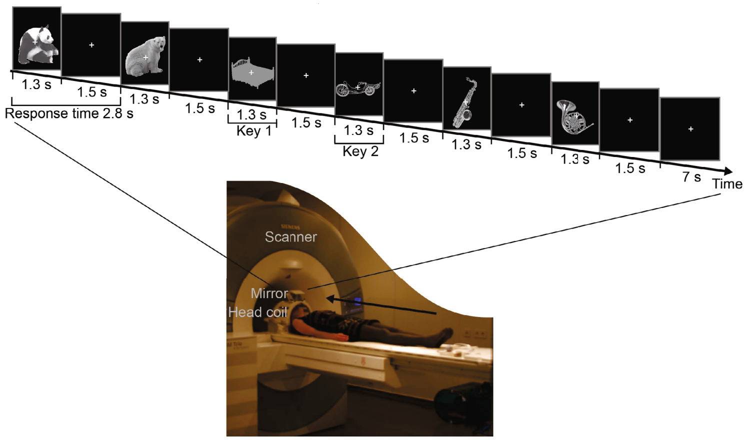


Fig. 2. Experimental setup. **Bottom**: A subject lying in an fMRI scanner. In order to perform the scan, the subject will be moved into the scanner tube. Stimuli are visible via the mirror positioned on the head coil. **Top**: Example run and timing of one experimental block.

the study which was approved by the joint human research review committee of the Max Planck Society and the University of Tübingen.

Stimuli. Stimuli G were $|G| = 72$ gray-scale photographs of real objects taken from Hemera photo objects vol. 1-3. Half of the stimuli were animate objects from the four super-ordinate categories: mammals, birds, vegetables, and flowers. The non-animate objects were taken from the categories: furniture, vehicles, tools, and music instruments. Three ordinate (i.e. at the basic level of taxonomic abstraction) categories were chosen from each super-ordinate category (e.g. bear, dog, and monkey from the mammal category; brush, hammer, scissors from the tools category), and each ordinate category contained three exemplars (e.g. panda bear, brown bear, polar bear from the bear category).

To control for low level visual cues, the luminance of the photographs was equalized according to Knebel et al., 2008 [17]. Second, stimuli were adjusted in size to the same main diagonal length. In addition, 72 silhouettes (filled with the mean luminance value) were created for every object.

Behavioral Ranking of the Stimuli Outside the Scanner. The subject ranked the stimuli in terms of their familiarity, using a 7 point Likert scale (1- not familiar, 7- very familiar). In addition, he judged the similarity between each pair of objects (1-very dissimilar, 7- very similar).

Experimental Procedure in the Scanner. The subject was first familiarized with the photographs in the scanner environment through presentation of all stimuli in random order with each intact photograph followed by its matching silhouette. Stimuli (on a black background with white fixation cross in the center, main diagonal: 3.1 visual angle) were each presented for 1.3s, followed by a white fixation cross for1.5s (see fig. 2, top).

After the familiarization stage, the subject performed the experimental sessions. The experimental paradigm was a target detection task in which the subject had to press one key for a silhouette and another key for an intact image. Each session contained the 72 object stimuli repeated twice (displayed for 1.3s, followed by a 1.5s fixation) and 12 silhouette images, appearing on average every 12 trials. To increase design efficiency, the 6 stimuli from each ordinate category (3 exemplars x 2 repetitions) were presented in a pseudo-randomized order. The subject could respond from the onset of the stimulus until the end of the fixation period, resulting in a response time interval of 2.8s. Instructions emphasized both speed and accuracy of the response, using the index and middle fingers for silhouettes and intact photos respectively.

Blocks of six stimuli (block duration ≈17s) were interleaved with 7s fixation periods. The subject performed 48 sessions (≈10min each) over seven days (max. scanning time: 2h per day). Hence, each object stimulus was presented 96 times and every silhouette image eight times.

Experimental Setup. Stimuli were presented using the Cogent 2000 v1.25 (developed by the Cogent 2000 team at the FIL and the ICN and Cogent Graphics developed by John Romaya at the LON at the Wellcome Department of Imaging Neuroscience, UCL, London, UK) running under MATLAB (Mathworks Inc., Natick, MA, USA) on a Windows PC. The visual stimuli were back-projected onto a Plexiglas screen using a LCD projector (JVC Ltd., Yokohama, Japan) visible to the subject through a mirror mounted on the MR head coil. The subject performed the behavioral task using a MR-compatible custom-built button device connected to the stimulus computer.

fMRI Data Acquisition. A 3 T Siemens Magnetom Trio Tim System (Siemens, Erlangen, Germany) was used to acquire both three-dimensional high-resolution T1-weighted anatomical images (TR=2300ms, TE=2.98ms, TI=1100ms, flip angle=9°, FOV=256mm×240mm×176mm, isotropic spatial resolution 1mm) and T2*-weighted axial echoplanar functional images with BOLD contrast (gradient echo, TR=3080ms, TE=40ms, flip angle=90°, FOV=192mm×192 mm, image matrix 64×64mm, 38 transversal slices acquired sequentially in ascending direction, voxel size=3.0mm×3.0mm ×2.5mm + 0.5mm interslice gap) using a 12-channel head coil (Siemens, Erlangen, Germany). The subject participated in 48 experimental sessions with 212 volume images (whole-brain images) per session, amounting to 10,176 volume images. The first three volumes were discarded to allow for T1 equilibration effects.

Data Preprocessing and GLM Analysis. The functional MRI data was analyzed with statistical parametric mapping (SPM8 software, Wellcome Department of Imaging Neuroscience, London, UK; www.fil.ion.ucl.ac.uk/spm) [10]. According to the common practice we first preprocessed the data to reduce noise such as head motion artifacts. Scans were realigned using the first as a reference, unwarped, slice-time corrected using the middle image as a reference, and spatially normalized into Montreal Neurological Institute (MNI) standard space [7].

To determine the magnitude of the BOLD response in each voxel to a given stimulus, we used a well established mass-univariate approach based on general linear models (GLM). This method defines the explanatory variables/regressors (the stimuli in our case) using a design matrix, and estimates their relative contribution to the observed BOLD activation. The timeseries in each voxel were high-pass filtered to 1/128 Hz. The experiment was modeled in an event related fashion with regressors entered into the design matrix after convolving each event-related unit impulse function (logged to the onset of the visual stimulus) with the canonical hemodynamic response function (see [14] for details about modeling even-related designs). The statistical model included 72 regressors each modeling a particular stimulus and one additional regressor modeling all target stimuli, separately for each session. Nuisance covariates included the realignment parameters (to account for residual motion artifacts). Stimulus-specific effects for each session were estimated from the GLM. The GLM estimate of every explanatory variable (i.e. the parameter weight of this variable) for every voxel was saved in a beta image. All beta images were passed to a second-level analysis as contrasts, in order to allow for random effects analysis and inferences at the population level [9]. This involved creating 73 contrast images for each session and entering them into a second level analysis which evaluated the voxels which are more responsive for visual stimulation compared to fixation.

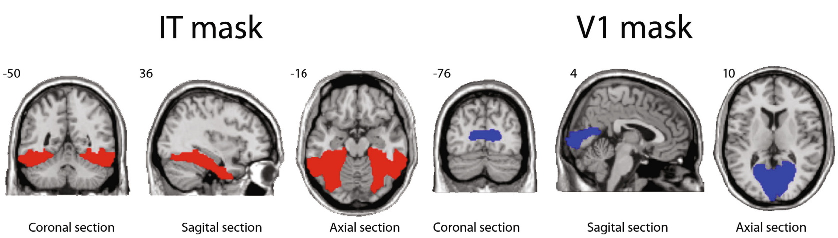


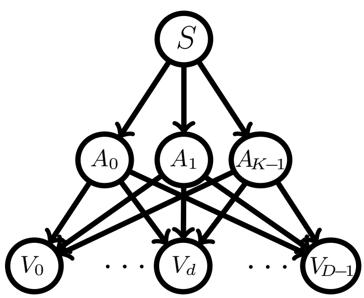
Fig. 3. The brain areas evaluated in this study. Location of the two regions of interests, V1 and IT, displayed on three planes overlaid on a standard brain, numbers are MNI coordinates [7].

4.2 Search Volumes and Voxel Selection

The activation data was extracted from two *a priori* defined anatomical search volumes (region of interests, ROIs, see fig. 3): the inferior temporal cortex (IT mask), and the calcarine sulcus (V1 mask). The IT mask included the bilateral inferior temporal gyri, fusiform gyri and parahippocampal gyri. The V1 mask contained the bilateral calcarine fissure and surrounding cortex which encompasses the primary visual cortex. Those areas were anatomically defined by the AAL library [26] using the MarsBaR toolbox (http://marsbar.sourceforge.net/) [4]. Within each ROI, the 300 most active voxels (the voxels showing the highest absolute activations for the second-level comparison all stimuli > fixation) were selected. From those we selected the 100 voxels that provided the most informative signals (measured by mutual information) about the stimulus identity.

5 Learning the Formal Context with a Hierarchical Bayesian Classifier

To apply FCA, we need to compute attributes from the BOLD signals in the selected voxels (see section 4.2). We have so far experimented with binary attributes only, but note that attribute scaling [11] is possible. However, the results in [6] indicate that binarized responses can summarize most of the conceptually relevant information in neural data. We first experimented with maximally informative thresholding [6] per voxel. In this approach, a threshold is determined for each voxel such that the (binarized) voxel signal allows for the best possible prediction of the stimulus identity. Due to the low signal-to-noise ratios in BOLD signals, this yielded very large and uninterpretable lattices. Therefore, we tried multi-voxel pattern analysis to 'average out' the noise across voxels. We extracted multi-voxel features and associated factors with two standard unsupervised feature extraction techniques, principal component analysis (PCA) [3], and non-negative matrix factorization (NFM) [20]. Both of these methods assume a linear additive generative model of the data, and both try to minimize the error between predicted and actual BOLD pattern. Let $\boldsymbol{V}_i = (V_{0,i}, \ldots, V_{D-1,i})$ be a vector representation of the BOLD activation pattern (D voxels). $i = 0, \ldots, N-1$ is the presentation (or session) index. The vector $\boldsymbol{V}_i$ is decomposed into $K = |M|$ features $\boldsymbol{f}_k$ and associated real-valued factors $A_{k,i}$, such that $K \leq D$ and



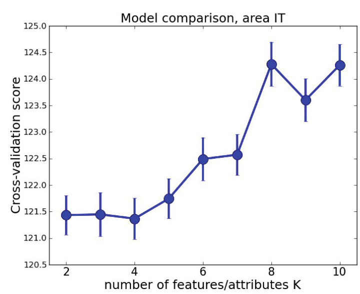


Fig. 4. Left: the feature extractor for learning the formal context, represented as a Bayesian network. Nodes represent random variables, arrows indicate conditional dependencies. A stimulus, represented by a multinomial variable S has $K = |M|$ binary attributes $\boldsymbol{A} = (A_0, \ldots, A_{K-1})$, $A_k \in \{0; 1\}$ which encode the observed BOLD voxel activation pattern $\boldsymbol{V} = (V_0, \ldots, V_{D-1})$. The (binarized) distribution $p(\boldsymbol{A}|S)$ represents the formal context. Voxel activation patterns are described by a distribution $p(\boldsymbol{V}|\boldsymbol{A}) = \prod_d p(V_d|\boldsymbol{A})$ with each voxel computed as a linear combination of non-negative feature vectors $\boldsymbol{f}_k$, i.e. $V_d = \sum_k (\boldsymbol{f}_k)_d \cdot A_k + \eta_d$, where η_d is voxel-specific Gaussian noise. **Right**: comparison of cross-validation scores for models with different K, computed from the $D = 100$ most informative voxels in area IT (see fig. 3). Error bars are SEM, 12-fold cross-validation. The model with $K = 8$ offers (approximately) the best trade-off between good data description and low model complexity. For details, see text.

$$\min_{A_{k,i},\boldsymbol{f}_k} \sum_i \left(\boldsymbol{V}_i - \sum_k \boldsymbol{f}_k A_{k,i} \right)^2 \tag{1}$$

under additional constraints. For NMF, the constraints are positivity of both the $\boldsymbol{f}_k$ and the $A_{k,i}$, whereas PCA requires the $\boldsymbol{f}_k$ to be orthonormal. We then applied maximally informative thresholding on the $A_{k,i}$ averaged over all presentations of a given stimulus to obtain a formal context. While certain basic features were now discernible in the lattices (e.g. a distinction between animate and inanimate objects), there still remained a lot of 'noisy' concepts. To improve the result further, we regularized the feature extraction by stipulating that there be only one configuration of the A_k per stimulus (rather than per stimulus presentation). Moreover, we constrained $A_k \in \{0; 1\}$. The resulting generative model is therefore given by ($S_i \in \{0, \ldots, |G|-1\}$ is a multinomial representation of the stimulus label):

$$\boldsymbol{V}_i = \sum_k \boldsymbol{f}_k A_{k,i} + \boldsymbol{\eta}_i \;\text{ with }\; \boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \Sigma) \tag{2}$$

$$p(A_{k,i}|S_i) \;=\; A_{k,i}^{\boldsymbol{I}_{k,S_i}} (1 - A_{k,i})^{1-\boldsymbol{I}_{k,S_i}} \tag{3}$$

$$p(S_i) \;\sim\; \text{uniform} \tag{4}$$

where $\boldsymbol{\eta}$ is voxel-dependent noise having a Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma)$ with zero mean and diagonal covariance matrix Σ. $\boldsymbol{I}$ is a matrix with $I_{k,s} = 1$ if $A_k = 1$ for $S_i = s$ and 0 otherwise. In other words, $\boldsymbol{I}$ is a binary matrix representation of the context I. To formalize the connection between $\boldsymbol{I}$ and I, choose one-to-one functions $V : G \to \{0; \ldots; |G| - 1\}$ and $W : M \to \{0; \ldots; |M| - 1\}$, then:

$$S_i = s \;\Leftrightarrow\; V(g_i) = s \tag{5}$$

$$A_{k,i} = 1 \Leftrightarrow W^{-1}(k) \in g_i' \text{ and } A_{k,i} = 0 \Leftrightarrow W^{-1}(k) \notin g_i' \tag{6}$$

The model is depicted as a Bayesian network in fig. 4. To learn the parameters, i.e. $\boldsymbol{I}, \Sigma$ and the $\boldsymbol{f}_k$, we employ variational Bayesian expectation maximization (VBEM) [3], with Gamma p(oste)riors on the $\boldsymbol{f}_k$ and the diagonal entries of Σ, and independent Bernoulli p(oste)riors on the entries of $\boldsymbol{I}$. In VBEM, learning is expressed as an maximization problem of a lower bound on the marginal log-likelihood of the $(\boldsymbol{V}_i, S_i)$ data. Let the model parameters be collectively denoted by $\boldsymbol{\Theta} = (\boldsymbol{f_0}, \ldots, \boldsymbol{f_{K-1}}, \Sigma, \boldsymbol{I})$, then this bound is given by

$$L = \left\langle \sum_i \log\left(p(\boldsymbol{V}_i, S_i|\Theta)\right) + \log\left(\frac{p(\Theta)}{q(\Theta)}\right) \right\rangle_{q(\Theta)} \tag{7}$$

where $p(\boldsymbol{V}_i, S_i|\Theta)$ is computed from eqns. 2-4 and $p(\Theta)$ is the parameter prior. $q(\Theta)$ is the variational posterior, which we chose to have the same functional form as the prior, as noted above. The expectation $\langle \ldots \rangle_{q(\Theta)}$ can then be evaluated in closed form. To avoid getting stuck at local maxima in the early phases of the optimization, we precede the VBEM iterations with simulated annealing [24]. One advantage of taking a Bayesian approach to learning is that we can evaluate (at least approximately) which number of attributes/features K offers the best compromise between a good explanation of the data and a low model complexity.

The Gamma priors on $\boldsymbol{f}_k$ enforce NMF-like positivity constraints, which we found to contribute to the interpretability of the results. A possible reason for this is that $A_k = 1$ implies a positive contribution to the BOLD signal under these constraints. In other words, there is an order-preserving mapping from the attribute sets ordered under subset inclusion to brain activity.

6 Results

Model Selection. We learned feature extractors with $K \in \{2; \ldots; 10\}$ features as described in section 5. The VBEM iteration usually began to converge after $\approx$ 50 VBEM steps, preceded by simulated annealing at 10 exponentially decreasing temperatures (1000 samples each) between T_{max} and 1. T_{max} was chosen so that the variational posterior of the entries of $\boldsymbol{I}$ did not differ by more than 0.1 from its prior value 0.5, indicating a high enough temperature to 'smooth out' local maxima. To determine the best K, we performed 12-fold cross validation, the held-out data were always complete sessions (see section 4). The cross-validation score plotted in fig. 4, right, is the variational bound L (eqn. 7) computed on the held-out data after 100 VBEM steps. To model the most informative 100 voxels in area IT, 8 features/attributes appear to be sufficient. Note that this result is conditional on $K \leq 10$.

Lattices. The concept lattices for both ROIs (IT and V1, after 100 VBEM steps) are displayed in fig. 5 and fig. 6 for $K = 2$ and $K = 5$ attributes, respectively. These lattices are drawn with reduced labeling. We did not plot the lattice computed from $K = 8$ attributes, because it would not have fit onto a page ($>$ 200 concepts). However, its main interpretable features are similar to the $K = 5$ lattice, which we describe in the following. The IT lattice with two attributes already shows one of the most prominent semantic features: the distinction between animals and other objects (including plants). Concept C1 introduces 22 stimuli, 18 of which are animals, encompassing all animals in the stimulus set. C0 introduces 50 stimuli, none of which are animals. Indeed, the IT region is known to be specialized in object recognition. Previous studies suggested that categorical representation in the IT are organized in hierarchical fashion that distinguishes animate (including faces and body parts) and inanimate stimuli [18,1]. In contrast, three of the V1 lattice's concepts introduce animals along with other objects. However, the within-concept organization seems to be partly shape-based: C1 introduces mainly thin and elongated stimuli, while the stimuli introduced in C3 are mainly rotund. These observations could be further tested by comparison to lattices computed with low-level shape descriptors as attributes. Similar observations, with somewhat higher 'conceptual resolution',

Table 1. Left: Testing whether the subset ordering of the attributes is correlated with subjective familiarity. $K = |M|$: number of attributes. μ: frequency with which the conditional in eqn. 8 holds across all pairs of stimuli, μ_0, σ_0 are baseline values obtained by randomization. Not all values are significantly above chance ($p < 0.05$), but there is a clear trend towards $z > 0.0$. **Right**: Testing whether attribute set similarity is correlated with subjective similarity. Here, μ is the frequency with which the conditional in eqn. 10 holds, μ_0, σ_0 are corresponding baseline values. All K yield significant results. '0.000' means $p < 0.0005$. For details, see section 6.

Familiarity					Similarity				
K	μ	$\mu_0 \pm \sigma_0$	z	p	K	μ	$\mu_0 \pm \sigma_0$	z	p
2	0.672	0.573 ± 0.023	4.409	0.000	2	0.680	0.614 ± 0.006	11.907	0.000
3	0.682	0.572 ± 0.027	4.062	0.000	3	0.711	0.614 ± 0.007	13.860	0.000
4	0.630	0.572 ± 0.028	2.077	0.019	4	0.713	0.615 ± 0.008	11.749	0.000
5	0.611	0.572 ± 0.034	1.143	0.127	5	0.755	0.617 ± 0.011	12.085	0.000
6	0.653	0.572 ± 0.042	1.919	0.027	6	0.737	0.618 ± 0.013	9.254	0.000
7	0.685	0.572 ± 0.060	1.901	0.029	7	0.735	0.619 ± 0.014	8.022	0.000
8	0.631	0.572 ± 0.051	1.155	0.124	8	0.812	0.623 ± 0.017	11.007	0.000
9	0.657	0.572 ± 0.061	1.379	0.084	9	0.815	0.624 ± 0.016	11.725	0.000
10	0.635	0.572 ± 0.077	0.816	0.207	10	0.821	0.631 ± 0.019	10.132	0.000

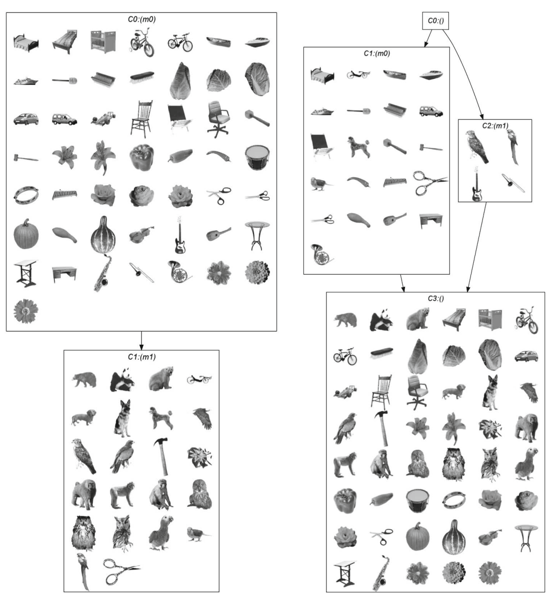


Fig. 5. Left: lattice computed from brain area IT (see fig. 3) with a feature extractor having $K = 2$ features/attributes. Reduced labeling. Concepts are numbered according to the lectic ordering of the intents [11]. E.g. 'C1:{m1}' means: concept number 1, introducing the attribute m1. '{}' denotes the empty set. Images are the introduced objects of each concept. **Right**: lattice computed from brain area V1, also $K = 2$.

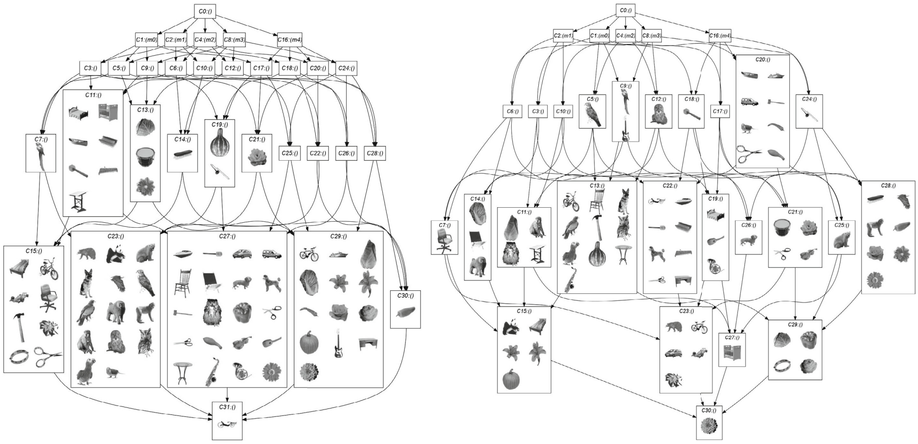


Fig. 6. Left: lattice computed from brain area IT (see fig. 3) with a feature extractor having $K = 5$ features/attributes. Labeling as in fig. 5. **Right**: lattice computed from brain area V1 with $K = 5$.

hold for the lattices with $K = 5$. Here, the IT lattice shows concepts which introduce exclusively animals (C23, 14 animals), mostly plants (C29, 9 plants of 13 stimuli), and non-animates (C11: 7 of 7, C15: 7 of 8). As for $K = 2$, the V1 lattice does not show this sort of semantic organization, but might contain shape-specific concepts (e.g. horizontally elongated objects in C20 and C22). Another noteworthy difference between the V1 and IT lattices is the number of concepts which introduce stimuli: 12 in IT versus 21 in V1. Thus, if one wanted to use the corresponding feature extractor as a simple classifier, one should use signals from V1, since they would yield a higher classification rate.

The most specific concept (C31) in the IT lattice introduces a single stimulus, a recumbent bicycle. This stimulus was ranked by the subject as very unfamiliar and this might explain the high brain activation resulting in this stimulus having all attributes.

Familiarity Ranking Comparison. To substantiate the last observation in a more quantitative fashion, we compared the ordering of the stimuli induced by the attribute sets with the ordering given by the subject's familiarity ranking (7 point likert scale, 1-low, 7-high). Let $g_1, g_2 \in G$ be two stimuli, and $\mathrm{fam}(g)$ the subject's familiarity ranking. If the attribute set inclusion order reflects the familiarity ranking, then the conditional

$$\mathrm{fam}(g_1) \geq \mathrm{fam}(g_2) \text{ given } g_1' \subseteq g_2' \tag{8}$$

should be true in (above chance) many instances. The reason for choosing the direction of the inequality signs is the sparse and efficient coding hypothesis, which is popular in computational neuroscience [22,12,8]: frequently encountered, and thus familiar stimuli should be represented in the brain with less metabolic effort than unusual ones. Thus, due to the positivity constraint on the feature vectors in our model (see section 5), more familiar stimuli should have less attributes. We therefore computed the frequency μ with which the conditional 8 holds, averaged across all stimulus pairs g_1, g_2 and excluding the trivial cases $g_1 = g_2$. To obtain a 'baseline' frequency μ_0 for a given lattice structure, we randomly shuffled the stimuli against the attribute sets. This procedure leaves the lattice structure intact, but randomizes the extents. We repeated the randomization $\approx 10^3$ times for all K to compute a baseline standard deviation σ_0. The results are shown in table 1, left. While not all μ are significantly ($p < 0.05$, one-tailed z-test) above baseline, the trend is clearly towards a higher-than-chance frequency for conditional 8 to hold.

Similarity Comparison. We also evaluated if the subject's pairwise similarity ratings correspond to the (partial) similarity ordering of stimuli induced by the computed attributes. To this end, we used the contrast model by A.Tversky [25] which was formalized and extended in [19]. Let $g_1, g_2, f_1, f_2 \in G$ be stimuli, then

$$\begin{aligned}(g_1, g_2) \geq (f_1, f_2) \Leftrightarrow g_1' \cap g_2' \supseteq f_1' \cap f_2',\ & g_1' \cap \overline{g_2'} \subseteq f_1' \cap \overline{f_2'} \\ \overline{g_1'} \cap g_2' \subseteq \overline{f_1'} \cap f_2',\ & \overline{g_1'} \cap \overline{g_2'} \supseteq \overline{f_1'} \cap \overline{f_2'}\end{aligned} \tag{9}$$

I.e. g_1 is at least as similar to g_2 as f_1 is to f_2 if g_1 and g_2 have more common attributes ($g_1' \cap g_2'$), less separating attributes ($g_1' \cap \overline{g_2'}$ and $\overline{g_1'} \cap g_2'$) and more attributes not shared by either of them ($\overline{g_1'} \cap \overline{g_2'}$). For an in-depth discussion of this definition, see [19]. Let $\mathrm{sim}(g_1, g_2)$ be the subject's similarity rating for stimuli g_1, g_2. We computed the frequency μ with which the following conditional holds:

$$\mathrm{sim}(g_1, g_2) \geq \mathrm{sim}(f_1, f_2) \text{ given } (g_1, g_2) \geq (f_1, f_2) \tag{10}$$

and also evaluated a baseline μ_0, σ_0 by randomization, as described above for the familiarity ranking comparison. The results are shown in table 1, right. All comparisons are highly significant above chance, indicating that the attribute similarity structure is strongly correlated with the subject's similarity ratings.

7 Conclusion

We presented the first (to our knowledge) application of FCA to fMRI data for the elucidation of semantic relationships between visual stimuli. FCA revealed different organization within the two ROIs. While BOLD signals from the primary visual cortical area V1 allow for the construction of a better classifier, the objects in area IT are organized in a high-level semantic fashion. In addition to previous studies, the IT categorical organization separated plants from non-animates. Our current study shows the potential strength of FCA for fMRI data analysis, especially when dealing with a larger stimulus set. Furthermore, subjective familiarity and similarity correlate strongly with attribute-induced orderings of stimuli. In the future, we will investigate what information can be decoded by FCA from other areas of the cortex. For example, we will apply FCA on intermediate brain regions of the ventral stream to investigate how categorical representations are formed in the human brain. We are also planning to check the reproducibility of the lattices by testing additional subjects.

Acknowledgements. This work was supported by EU projects FP7-ICT-215866 SEARISE, FP7-249858-TP3 TANGO, FP7-ICT-248311 AMARSi and the Max-Planck Society. We thank the bwGRiD project for computational resources.

References

1. Bell, A.H., Hadj-Bouziane, F., Frihauf, J.B., Tootell, R.B.H., Ungerleider, L.G.: Object representations in the temporal cortex of monkeys and humans as revealed by functional magnetic resonance imaging. Journal of Neurophysiology 101(2), 688–700 (2009), http://jn.physiology.org/content/101/2/688.abstract
2. Bell, A.H., Malecek, N.J., Morin, E.L., Hadj-Bouziane, F., Tootell, R.B.H., Ungerleider, L.G.: Relationship between functional magnetic resonance imaging-identified regions and neuronal category selectivity. The Journal of Neuroscience 31(34), 12229–12240 (2011), http://www.jneurosci.org/content/31/34/12229.abstract
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
4. Brett, M., Anton, J., Valabregue, R., Poline, J.: Region of interest analysis using an SPM toolbox. Neuroimage 16(2) (2002); 8th International Conference on Functional Mapping of the Human Brain
5. Eger, E., Ashburner, J., Haynes, J.D., Dolan, R.J., Rees, G.: fMRI activity patterns in human LOC carry information about object exemplars within category. J. Cogn. Neurosci. 20(2), 356–370 (2008)
6. Endres, D., Földiák, P., Priss, U.: An application of formal concept analysis to semantic neural decoding. Annals of Mathematics and Artificial Intelligence 57(3-4), 233–248 (2010), doi:10.1007/s10472-010-9196-8
7. Evans, A.C., Marrett, S., Neelin, P., Collins, L., Worsley, K., Dai, W., Milot, S., Meyer, E., Bub, D.: Anatomical mapping of functional activation in stereotactic coordinate space. Neuroimage 1(1), 43–53 (1992)
8. Földiák, P.: Sparse coding in the primate cortex. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks, 2nd edn. MIT Press, Cambridge (2002)
9. Friston, K.J., Holmes, A.P., Price, C.J., Buchel, C., Worsley, K.J.: Multisubject fMRI studies and conjunction analyses. Neuroimage 10(4), 385–396 (1999)
10. Friston, K., Holmes, A., Worsley, K., Poline, J., Frith, C., Frackowiak, R.: Statistical parametric mapping: a general linear approach. Hum. Brain Mapping 2, 189–210 (1995)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical foundations. Springer (1999)
12. Harpur, G.F., Prager, R.W.: Experiments with low-entropy neural networks. In: Baddeley, R., Hancock, P., Földiák, P. (eds.) Information Theory and the Brain, ch. 5, pp. 84–100. Cambridge University Press, New York (2000)
13. Haxby, J., Gobbini, M., Furey, M., Ishai, A., Schouten, J., Pietrini, P.: Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science 5539(293), 2425–2430 (2001)

14. Josephs, O., Turner, R., Friston, K.J.: Event-related fMRI. Human Brain Mapping 5, 243–248 (1997)
15. Kandel, E.R., Schwartz, J.H., Jessell, T.M. (eds.): Principles of Neural Science, ch. 25-29. McGraw-Hill Education (2000)
16. 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), 4296–4309 (2007)
17. Knebel, J.F., Toepel, U., Hudry, J., le Coutre, J., Murray, M.M.: Generating controlled image sets in cognitive neuroscience research. Brain Topogr. 20(4), 284–289 (2008)
18. Kriegeskorte, N., Mur, M., Ruff, D.A., Kiani, R., Bodurka, J., Esteky, H., Tanaka, K., Bandettini, P.A.: Matching categorical object representations in inferior temporal cortex of man and monkey. Neuron 60(6), 1126–1141 (2008)
19. Lengnink, K.: Formalisierungen von Ähnlichkeit aus Sicht der Formalen Begriffsanalyse. Ph.D. thesis, Technische Hochschule Darmstadt, Fachbereich Mathematik (1996)
20. Lin, C.J.: Projected gradient methods for non-negative matrix factorization. Neural Computation 19, 2756–2779 (2007)
21. Mishkin, M., Ungerleider, L.G., Macko, K.A.: Object vision and spatial vision: two cortical pathways. Trends in Neurosciences 6, 414–417 (1983), http://www.sciencedirect.com/science/article/pii/016622368390190X
22. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583), 607–609 (1996)
23. Op de Beeck, H.P., Haushofer, J., Kanwisher, N.G.: Interpreting fMRI data: maps, modules and dimensions. Nat. Rev. Neurosci. 9(2), 123–135 (2008)
24. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes in C++: the art of scientific computing, 3rd edn. Cambridge University Press, New York (2007)
25. Tversky, A.: Features of similarity. Psychological Review 84(4), 327–352 (1977)
26. Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., Joliot, M.: Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the mni MRI single-subject brain. Neuroimage 15(1), 273–289 (2002)
27. Uludag, K., Dubowitz, D., Buxton, R.: Basic principals of functional MRI, pp. 249–287. Elsevier (2005)
28. Walther, D.B., Caddigan, E., Fei-Fei, L., Beck, D.M.: Natural scene categories revealed in distributed patterns of activity in the human brain. J. Neurosci. 29(34), 10573–10581 (2009)
29. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)

Cubes of Concepts: Multi-dimensional Exploration of Multi-valued Contexts

Sébastien Ferré, Pierre Allard, and Olivier Ridoux

IRISA/Université de Rennes 1, Campus de Beaulieu, 35042 Rennes cedex, France
Firstname.Lastname@irisa.fr

Abstract. A number of information systems offer a limited exploration in that users can only navigate from one object to another object, e.g. navigating from folder to folder in file systems, or from page to page on the Web. An advantage of conceptual information systems is to provide navigation from concept to concept, and therefore from set of objects to set of objects. The main contribution of this paper is to push the exploration capability one step further, by providing navigation from set of concepts to set of concepts. Those sets of concepts are structured along a number of dimensions, thus forming a cube of concepts. We describe a number of representations of concepts, such as sets of objects, multisets of values, and aggregated values. We apply our approach to multi-valued contexts, which stand at an intermediate position between many-valued contexts and logical contexts. We explain how users can navigate from one cube of concepts to another. We show that this navigation includes and extends both conceptual navigation and OLAP operations on cubes.

Keywords: formal concept analysis, information systems, data exploration, navigation, multi-valued context, multi-dimensional analysis, OLAP cubes.

1 Introduction

Navigation is a convenient way for exploring data, compared to querying, because it provides guiding to users during their search, and supports exploratory search [16]. At each step of a navigation path, users are located at a navigation place, and a number of navigation links are suggested to them in order to reach neighbour navigation places. However, most existing systems suffer from a number of limitations, e.g., file systems and the Web. A first limitation is that navigation places are objects (e.g., files and folders, Web pages), and users can therefore only jump from object to object. This makes it difficult to group and compare objects. A second limitation is that the navigation graph is manually drawn. As a consequence, the navigation graph is generally very sparse, and requires from users chance or expertise or both to find their way to the searched objects. For example, if photos are organized in a file system first by date then by topic, it is easy enough to find photos by date, difficult to find photos by topic, and impossible to find them by depicted person. A third limitation is

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 112–127, 2012.

that navigation links may lead to empty results, e.g., an empty folder. This is a frustrating experience for users who have to resort to tedious trial-and-error. A fourth limitation is that the navigation path must be linear. At each step, only one navigation link can be chosen. If users want to explore several navigation links, they have to choose and explore one, then to move backward in the navigation history, and choose a second one, and so on.

Conceptual navigation [14,3,15,6,7] overcomes the first three limitations by relying on Formal Concept Analysis (FCA) [13]. Users navigate from concept to concept, and hence from set of objects to set of objects. The navigation graph is the concept lattice, which is automatically derived from data, a formal context, and therefore automatically adapts to changes in data. Finally, it is easy to prevent empty results, by removing the bottom concept from the navigation graph. Regarding the fourth limitation, if the concept lattice is small enough, the line-diagram enables to view all navigation paths at a glance. In practice, however, the concept lattice is much too large to be visualized, and only the current concept and its neighbours are displayed. Existing conceptual navigation systems follow that principle, and only allow for linear navigation paths.

The contribution of this paper is to extend conceptual navigation to non-linear navigation paths. Concretely, this means that users can choose a set of navigation links at some step, leading to a set of concepts. As several multi-choices can be done in sequence, a navigation place becomes a multi-dimensional set of concepts, a *cube of concepts*. We show that our approach covers and extends the multi-dimensional analysis of OLAP [5,20]. We define cubes of concepts on *multi-valued contexts*, which we introduce in this paper. They extend many-valued contexts by allowing several values for the attribute of an object, rather than only one. The advantages of using multi-valued contexts are (1) more general results that directly apply to many-valued contexts and binary contexts, and (2) a data model close to databases [4], OLAP [5,20], the Semantic Web [2], and Logical Information Systems [10].

After introducing useful notations on mappings, multisets, tuples, and FCA in Section 2, Section 3 defines multi-valued contexts, and value domains. Section 4 defines cubes of concepts to be used as navigation places, and cube transformations as navigation links. Those cube transformations are proved safe, i.e. not leading to empty results. Section 5 discusses the representation of cubes of concepts in Abilis. Section 6 compares cubes of concepts and their transformations to OLAP cubes and operators. Section 7 compares our approach to other approaches combining FCA and OLAP.

2 Preliminaries

A *mapping* M from a set A to a set B is a set of couples $a \mapsto b$, where $a \in A$ and $b \in B$, and such that $(a, b_1) \in M \wedge (a, b_2) \in M \Rightarrow b_1 = b_2$. It is a partial function from A to B that is defined in extension. Its domain is noted $dom(M)$, and its range is noted $ran(M)$. A *multiset* M over a set A is a mapping from A to natural numbers $\mathbb{N}$. It generalizes sets by allowing elements to occur several times in a multiset: $a \mapsto n \in M$ means that a has *multiplicity* n in M. For convenience, we define the following notations on a multiset M: $M^\alpha = \{a \mapsto n \in M \mid n \geq \alpha\}$ is the filtering of multiset M by a minimum multiplicity α, $M_Y = \{a \mapsto n \in M \mid a \in Y\}$ is the restriction of the domain of M to Y, and $\#M = \Sigma\{n \mid a \mapsto n \in M\}$ is the cardinal of multiset M.

Given a *tuple* $\overline{x}$ of dimension n, we denote by x_i the i-th component of the tuple. The notation $\overline{x}$ is a shorthand for $x_1 \dots x_n$. We define the following notations on tuples: $\overline{x}+x' = x_1 \dots x_n x'$ is the extension of a tuple $\overline{x}$ by an element x', $\overline{x} - i = x_1 \dots x_{i-1} x_{i+1} \dots x_n$ is the restriction of a tuple $\overline{x}$ by removing the i-th element, $\overline{x}[i \leftarrow x'] = x_1 \dots x_{i-1} x' x_{i+1} \dots x_n$ is the replacement of the i-th element by x', and $dom_i(X) = \{x_i \mid \overline{x} \in X\}$ is the domain of the i-th elements, where X is a set of tuples of same dimension n.

We here recall some basic notations for Formal Concept Analysis (FCA) [13]. However, we adopt a somewhat novel presentation in that *intension* is made a particular case of a more general notion: the *index*. A *formal context* is a triple $K = (O, A, I)$, where O is the set of objects, A is the set of attributes, and $I \subseteq O \times A$ is the incidence relation between objects and attributes. The *extension* of a set of attributes Y is defined as the set of objects that have all the attributes in Y, i.e., $ext(Y) = \{o \in O \mid \forall a \in Y : (o, a) \in I\}$. By abuse of notation, we will denote by $ext(\{a\})$ the expression $ext(a)$ for . The *index* of a set of objects X is defined as the multiset of attributes of objects in X, where the multiplicity of each attribute is the number of its objects in X, i.e.,

$$index(X) = \{a \mapsto n \mid a \in A, n = \#(X \cap ext(a)) \neq 0\}.$$

The *intent* of a set of objects X is then defined as the domain of the index of X filtered by the maximum frequency, i.e., $int(X) = dom(index(X)^{\#X})$. A *concept* c is a pair (X, Y) s.t. $X = ext(Y)$ and $Y = int(X)$: $ext(c) = X$ is called the *extent* of c, and $int(c) = Y$ is called the *intent* of c.

3 Multi-valued Contexts

In order to better represent real datasets and complex data, a number of extensions or generalizations of FCA have been proposed: many-valued contexts [12], logical contexts [9], pattern structures [11], to cite only a few of them. In particular, many-valued contexts are equivalent to tables in relational databases. Logical contexts (and pattern structures) are more general, but miss the distinction between attributes and values, which will be useful in this paper. Conversely, logical contexts enable to describe a photo as depicting both Alice and Bob, which is not possible in a many-valued context where the photo is an object, "depicts" is an attribute, and persons are values: "depicts" is a *multi-valued* attribute. The term "many-valued" means that an attribute can take one among many values, whereas the term "multi-valued" here means that an attribute can take multiple values at once. Multi-valued attributes do not exist in relational databases because those are normalized into several tables (one table for photos, and one table for the "depicts" relation). As FCA generally handles a single table, the context, it is important to allow for multi-valued attributes. This leads us to the definition of *multi-valued contexts* as a set of triples (object,attribute,value), like graphs in the RDF data model [2].

Definition 1 (multi-valued context). *A* multi-valued context *is a quadruple* $K = (O, A, V, T)$*, where* O *is the set of objects,* A *is the set of (valued) attributes,* V *is the set of values, and* $T \subseteq O \times A \times V$ *is a set of triples. Each triple* (o, a, v) *means that* v *is a value taken by the atribute* a *on object* o*.*

As the set $O \times A \times V$ is equivalent to $O \times A \to \mathcal{P}(V)$, a multi-valued context can also be represented like a many-valued context, but allowing for zero, one or several values in each cell. The following table defines a multi-valued context K_e of 6 photos described by the persons they depict, the year they were taken, and their size in pixels. It has 19 triples.

object	person	date	size
1	Alice	2010-03-19	1.1M
2	Bob	2010-07-13	3.1M
3	Charlie	2011-01-30	1.2M
4	Alice, Bob	2011-04-28	3.2M
5	Alice, Charlie	2011-08-20	1.3M
6	Alice	2011-11-11	

This example illustrates the fact that an object can have several values for an attribute (e.g., the depicted persons of photo 4), or no value at all (e.g., the size of photo 6). This example is kept small for illustration purposes, but our approach is designed for datasets with hundreds to thousands of objects, and tens of attributes.

3.1 Value Domains and Attribute-Value Schemas

In practice, there are interdependencies between attributes and values on one hand, and between values on the other hand. Each attribute expects values from a given domain, and the values of a domain can be organized into a generalization ordering (e.g., locations, date intervals). We first define domains of values in order to formalize hierarchies of values, and aspects related to OLAP such as granularity levels, and aggregators.

Definition 2 (value domain). *A* value domain *is a structure* $D = (V, \sqsubseteq, \top, \Lambda, \Gamma)$*, where:*

- V *is the set of values,*
- $\sqsubseteq$ *is a partial ordering (called* subsumption*) that represents the generalization ordering between values, and* $\prec$ *is the corresponding covering relation,*
- $\top$ *is a distinguished value, more general than any value,*
- $\Lambda \subseteq \mathcal{P}(V)$ *is a set of granularity levels, each level* $\lambda \subseteq V$ *being defined as a subset of values,*
- Γ *is a set of aggregators over multi-sets of values.*

The example multi-valued context K_e uses three value domains, respectively for persons, dates, and sizes.

Person. Values are either persons (e.g., Alice, Bob, Charlie) or groups (e.g., family, friends). A person is subsumed by every group it belongs to. Persons and groups constitute the two granularity levels: $\Lambda = \{individual, group\}$. The only applicable aggregators are the count ($\gamma(M) = \#M$) and the distinct count ($\gamma(M) = \#dom(M)$).

Date. Values are dates at four granularity levels: days (e.g., 2010-03-19), months (e.g., 2010-03), years (e.g., 2010), and calendar weeks (e.g., 2010:42). Date values represent intervals of time, and are hierarchically organized by inclusion. Applicable aggregators are, in addition to the count and distinct count, the earliest date, the latest date, and the median date.

Size. Values are natural numbers at various precisions (e.g., 1M, 1.2M, 1234k). There is a granularity level for each level of precision: units (e.g., 1234567), tens (e.g., 1234.56k), ..., millions (e.g., 1M). Size values represent integer intervals, and are hierarchically organized by inclusion. Applicable aggregators are, in addition to those of dates, the sum and the average.

There are different ways to define the hierarchy of values of a value domain. A first way is to use scale contexts [12] when the set of values is finite and fixed. A second way is to define logics as in Logical Concept Analysis (LCA, [10]), which works well for infinite domains (e.g., integers, dates, geographic shapes), and allows for the dynamic and automatic insertion of new values in the hierarchy.

Subsumption can be extended from values to granularity levels: for any two levels λ_1, λ_2, we have $\lambda_1 \sqsubseteq \lambda_2$ iff for every $v_1 \in \lambda_1$, there exists $v_2 \in \lambda_2$ s.t. $v_1 \sqsubseteq v_2$. While levels are generally disjoint and totally ordered, they need not be. A classical example is the date domain, with four disjoint levels: days, months, weeks, and years. The level "day" is subsumed by the levels "month" and "week", both of which are in turn subsumed by the level "year", but neither "month" is subsumed by "week" nor the converse. Sometimes, there are no natural levels, and we want to define them from the topology of the hierarchy. Each level corresponds to a certain depth in the hierarchy, and they may overlap.

Definition 3 (topological levels). *Let $D = (V, \sqsubseteq, \top, \Lambda, \Gamma)$ be a value domain. The granularity level $\lambda(p)$ denotes the topological level at depth p, and is recursively defined as follows for every $p > 0$:*

- $\lambda(0) = \{\top\}$,
- $\lambda(p+1) = \{v \in V \mid \exists v' \in \lambda(p) : v \prec v'\} \cup \{v \in \lambda(p) \mid \nexists v' \in V : v' \prec v\}$.

3.2 Attribute Contexts and Feature Context

A multi-valued attribute is a binary relationship between objects and values. A formal context can therefore be derived for each attribute, with the help of a value domain for handling subsumption between values. It is called an *attribute context*, and it plays the same role as a realized scale.

Table 1. The feature context of the multi-valued context K_e on photos (only some levels of values are shown)

	person			date		size	
photo	Alice	Bob	Charlie	2010	2011	1M	3M
1	×			×		×	
2		×		×			×
3			×		×	×	
4	×	×			×		×
5	×		×		×	×	
6	×				×		

Definition 4 (attribute context). *Let* $K = (O, A, V, T)$ *be a multi-valued context,* $a \in A$ *be an attribute, and* D *a value domain for* a*. The* attribute context *of* a *is defined as* $K_a = (O, V_a, I_a)$*, where* $I_a = \{(o, v) \in O \times V_a \mid (o, a, v') \in T, v' \sqsubseteq_a v\}$*.*

In the attribute context K_a, $ext_a(Y)$ is the set of objects having all values of Y as values of the attribute a (directly or by subsumption), e.g., the set of photos depicting a set of persons together; $index_a(X)$ is the mappings from values in V_a to their frequency as the value of the attribute a over the set of objects X, e.g., the distribution of persons that are depicted in a given set of photos. Finally, $int_a(X)$ is the set of values in V_a occurring in all objects in X through attribute a, e.g., the set of persons depicted by all elements of given set of photos.

The feature context K_F is the apposition (see p. 30 in [13]) of all attribute contexts, distinguishing values by prefixing them with attributes. This is the natural translation of a multi-valued context into a formal context.

Definition 5 (feature context). *Let* $K = (O, A, V, T)$ *be a multi-valued context, and* $(D_a)_{a \in A}$ *be a collection of value domains. The* feature context *is defined as* $K_F = (O, F, I_F)$*, where the set of features* F*, and the incidence relation* I_F *between objects and features is defined by:*

- $F = \{(a, v) \mid a \in A, v \in V_a\}$,
- $I_F = \{(o, (a, v)) \mid (o, a, v') \in T, v' \sqsubseteq_a v\}$.

In the feature context, the extension $ext_F(Y)$ returns the set of objects having a set of features Y; and the index $index_F(X)$ returns the distribution of features over a set of objects X. Table 1 contains a partial representation of the feature context of K_e, showing only one level of values for each attribute (persons, years, and millions of pixels). The FCA operations of the feature context can be related to the FCA operations of attributes contexts as in Lemma 1.

Lemma 1. *The following equations holds for every multi-valued context* K*:*

1. $ext_F((a, v)) = ext_a(v)$,
2. $int_F(X) = \bigcup_{a \in A}\{(a, v) \mid v \in int_a(X)\}$,
3. $index_F(X) = \bigcup_{a \in A}\{(a, v) \mapsto n \mid v \mapsto n \in index_a(X)\}$.

4 Cubes of Formal Concepts as Navigation Places

In conceptual navigation, navigation places are formal concepts. We are here interested in navigating in the concept lattice of the feature context because it contains all the information of a multi-valued context, and its associated value domains. In the following, we assume a multi-valued context $K = (O, A, V, T)$ together with value domains D_a for each $a \in A$, and its derived feature context $K_F = (O, F, I_F)$. In this paper, we start from the framework of Logical Information Systems (LIS, [10]) with a restriction to conjunctive queries, whereas disjunction and negation are generally available in LIS. For the sake of simplicity, we will stick in this paper to this restriction, but the following results easily extent to Boolean queries. In LIS, the current concept c is defined by $ext(c) = ext_F(q)$, where $q \subseteq F$ is the current query. This query is the result of the successive choices of a single feature (*single-choices*) among those suggested by the system, starting from the empty set, and hence from the top concept. We now want to extend conceptual navigation to choices of multiple features (*multi-choices*), in order to escape the linearity limitation, and to provide multi-dimensional analysis. This raises the following questions: What is a natural multi-choice? What is a navigation place after performing a multi-choice? What defines such a navigation place?

Starting from the example multi-valued context K_e, suppose a user wants to look at photos by depicted person. He can select in turn each person, successively visiting the concepts 1456, 24, and 35 (for convenience, we name concepts after their extension: 24 is the concept whose extension is $\{2, 4\}$). This is tedious and unpractical for comparing the three subsets of photos. Alternately, the user could perform a multi-choice of the three persons, leading him to the set of the three concepts, indexed by the three persons. This results in the mapping $\{Alice \mapsto 1456, Bob \mapsto 24, Charlie \mapsto 35\}$. From there, the user wants to focus on photos depicting Alice, and performs a single choice of the feature $(person, Alice)$. The effect of this single-choice applies to each of the three above concepts, leading to the mapping $\{Alice \mapsto 1456, Bob \mapsto 4, Charlie \mapsto 5\}$. Bob and Charlie persist as indices because they are depicted with Alice on some photos. Finally, the user performs a second multi-choice of sizes at the level of millions (1M and 3M). The effect of this multi-choice applies to each concept of the last mapping, producing a mapping of mappings or, in a more compact form, a mapping from couples (person, size) to concepts: $\{(Alice, 1M) \mapsto 15, (Alice, 3M) \mapsto 4, (Bob, 3M) \mapsto 4, (Charlie, 1M) \mapsto 5\}$. Only concepts whose extension is not empty are retained, so that not all combinations of a person and a size are present.

This example scenario shows that a natural multi-choice is a granularity level of some attribute, which we define as an *axis* in analogy with graph axes.

Definition 6 (axis). *An* axis $x = a/\lambda$ *combines an attribute* $a \in A$ *and a granularity level* $\lambda \in \Lambda_a$ *for that attribute. In the following,* $a(x)$ *denotes the attribute of the axis,* $\lambda(x)$ *denotes the level of the axis, and* $index_x(X)$ *is a shorthand for* $(index_a(X))_\lambda$*, the* a*-index over objects* X*, restricted to the level* λ*.*

The example scenario also shows that navigation places are now mappings from tuples of values to concepts of the feature context. Those mappings are equivalent to n-dimensional arrays, where n is the size of tuples. Because the dimension of those arrays can be any natural number, and in analogy with OLAP, we call them *cubes of concepts*. Section 7 compares them to OLAP cubes. What determines such a cube is a sequence of single-choices and multiples-choices. The sequence of single-choices amounts to a set of features, the *query* q, and the sequence of multi-choices amounts to a tuple of axes, the dimension tuple $\overline{d}$.

Definition 7 (cube of concepts). *Given a query $q \subseteq F$, and a tuple $\overline{d}$ of n axes as dimensions, the* cube of concepts *is defined as a mapping from* coordinates *(tuples of values) to concepts:*

$$Cube(q, \overline{d}) = \{\overline{v} \mapsto c \mid \overline{v} \in \prod_{i=1}^{n} \lambda(d_i), ext(c) = ext_F(q) \cap \bigcap_{i=1}^{n} ext((a(d_i), v_i)) \neq \emptyset\}.$$

The above example shows that different concepts in a cube of concepts may overlap, and even that a same concept can appear at different coordinates. This comes from multi-valued contexts, where an object can have several values on a same axis (e.g., Alice and Bob as depicted persons). The example also shows that some coordinates, i.e. some combinations of values, may be missing in the cube. The reason is that either no object matches this combination, or an object has no value on some axis (e.g., photo 6 has no size value). Those are key differences with OLAP cubes (Section 7).

It is possible to generalize some definitions from concepts to cubes.

Definition 8. *The extension of the cube of concepts is the union of the extensions of the concepts. The intent and index of a cube of concepts can be derived from its extension as usual.*

$$ext(C) = \bigcup_{\overline{v} \mapsto c \in C} ext(c) \quad int(C) = int_F(ext(C)) \quad index(C) = index_F(ext(C))$$

Beware that the extension of a cube is not necessarily a concept extension, because concept extensions are not closed under set union. However, it is important not to close it, so that the index $index(ext(C))$ properly reflects the contents of the cube, and not the larger $ext_F(int(C))$ that may contain objects not visible in the cube.

In LIS, navigation links are defined by query transformations [7], rather than by the covering relation of the concept lattice. Query transformations are the addition or the removal of a feature, and combinations such as the replacement of a feature. Adding a feature f to a query q, noted $q + f$, provides downward navigation in the concept lattice. This gives access not only to lower neighbours, but also to concepts deeper in the lattice. All relevant features are suggested as navigation links, and not only those leading to a lower neighbour. Removing a feature f, noted $q - f$, provides upward navigation in the concept lattice. Of course, features can be removed in a different order they were added to the query. Addition and removal can be combined to provide sideward navigation, e.g., shifting from photos of Alice in 2010 to photos of Bob in 2010, and then to photos of Bob in 2011.

In this paper, because navigation places are cubes of concepts, we define navigation links as transformations of cubes of concepts. As a cube of concepts is made of a query q and a tuple of dimensions $\overline{d}$, the above query transformations equally apply to cubes of concepts. A second way to transform a cube is to change the dimensions of the cube. Possible transformations are:

- $\overline{d} + d'$: the addition of a dimension d',
- $\overline{d} - i$: the removal of a dimension d_i,
- $\sigma(\overline{d})$: a permutation σ on $\overline{d}$ to change the ordering of dimensions.

An important property of conceptual navigation is *safeness*, i.e. to suggest only navigation links that lead to concepts whose extension is not empty. This is important to avoid dead-ends, trial-and-error navigation, and hence frustration for users. With cubes of concepts, navigation is safe if it never leads to empty cubes. The following theorem states the conditions under which a cube transformation is *safe*.

Theorem 1. *Let $C = Cube(q, \overline{d})$ be a cube of concepts. The specializing cube transformations are safe under the following conditions:*

- $Cube(q + f, \overline{d})$ *if* $f \in index_F(ext(C))$,
- $Cube(q, \overline{d} + d')$, *if* $index_{d'}(ext(C)) \neq \emptyset$.

The generalizing transformations $Cube(q - f, \overline{d})$ and $Cube(q, \overline{d} - i)$, as well as the permutation transformation $Cube(q, \sigma(\overline{d}))$, are necessarily safe.

Proof. We give the proofs for the specializing transformations. The proof for other transformations are trivial.

- Proof for $Cube(q + f, \overline{d})$:
  $f \in index_F(ext(C))$
  $\Rightarrow ext_F(f) \cap ext(C) \neq \emptyset$
  $\Rightarrow ext_F(f) \cap \bigcup_{\overline{v} \mapsto c \in C} ext(c) \neq \emptyset$ (Definition 8)
  $\Rightarrow \exists \overline{v} \mapsto c \in C : ext_F(f) \cap ext(c) \neq \emptyset$ (Definition 7)
  $\Rightarrow \exists \overline{v} \in \prod_i \lambda(d_i) : ext_F(f) \cap (ext_F(q) \cap \bigcap_i ext((a(d_i), v_i))) \neq \emptyset$
  $\Rightarrow \exists \overline{v} \in \prod_i \lambda(d_i) : ext_F(q + f) \cap \bigcap_i ext((a(d_i), v_i)) \neq \emptyset$
  (because $ext(f) \cap ext(q) = ext(q \cup \{f\}) = ext(q + f)$)
  $\Rightarrow \exists \overline{v} \mapsto c' \in Cube(q + f, \overline{d})$
  $\Rightarrow Cube(q + f, \overline{d}) \neq \emptyset$.
- Proof for $Cube(q, \overline{d} + d')$:
  $index_{d'}(ext(C)) \neq \emptyset$
  $\Rightarrow index_{a(d')}(ext(C))_{\lambda(d')} \neq \emptyset$ (Definition 6)
  $\Rightarrow \exists v' \in \lambda(d') : v' \in index_{a(d')}(ext(C))$
  $\Rightarrow \exists v' \in \lambda(d') : (a(d'), v') \in index_F(ext(C))$
  $\Rightarrow \exists v' \in \lambda(d') : ext_F((a(d'), v')) \cap ext(C) \neq \emptyset$ (Lemma 1)

$\Rightarrow \exists \overline{v} \mapsto c \in C : \exists v' \in \lambda(d') : ext((a(d'), v')) \cap ext(c) \neq \emptyset$ (Definition 8)
$\Rightarrow \exists \overline{v} \in \prod_i \lambda(d_i) : \exists v' \in \lambda(d') : ext((a(d'), v')) \cap (ext_F(q) \cap \bigcap_i ext((a(d_i), v_i))) \neq \emptyset$
$\Rightarrow \exists \overline{v} + v' \in (\prod_i \lambda(d_i)) \times \lambda(d') : ext_F(q) \cap ((\bigcap_i ext((a(d_i), v_i))) \cap ext((a(d'), v'))) \neq \emptyset$
$\Rightarrow \exists \overline{v} + v' \mapsto c' \in Cube(q, \overline{d} + d')$
$\Rightarrow Cube(q, \overline{d} + d') \neq \emptyset.$ □

The condition for the addition of a feature to the query is the same as previously known in LIS. It establishes the feature index $index_F(ext(C))$ of a cube C as the set of features that can be added to the query. The condition for the addition of a dimension involves the indexes for each axis, which are included in the feature index (see Lemma 1). Therefore, the extension of conceptual navigation from concepts to cubes of concepts does not entail any increase in the size of suggested navigation links. A suggested dimension is simply an axis that shares values with the feature context. This is consistent with multi-choices being sets of choices.

5 Representation and Interaction in Abilis

In LIS systems, e.g. Camelis and Abilis[1], the current concept c is represented by the query q that defines it, the extension $ext(c)$ of the concept as a list of objects, and the feature index $index(ext(c))$ over that extension, which includes the intension $int(c)$ of the current concept. The feature index is organized by attribute, like facets in Faceted Search [18], and is displayed as a tree to reflect the generalization ordering between values. The query can be transformed by selecting features in the index. A feature is removed from the query if it belongs to it, otherwise it is added to the query.

Multi-dimensional conceptual navigation with cubes of concepts has been implemented in Abilis, along with rich capabilities to represent cubes of concepts. With a cube of concepts $C = Cube(q, \overline{d})$, we still have a query q, an extension $ext(C)$, and an index $index_F(ext(C))$ over that extension. The index is also the support of interaction by suggesting features and axes to be added or removed from the cube. The important difference is that a navigation place is a set of concepts projected at some coordinates instead of a single concept. Each concept defines a unit of knowledge, and has many possible concrete representations. The two obvious representations of a concept are its extension and its intention. Other useful representations are attribute indexes, and also aggregations of attribute indexes. By analogy with OLAP, we call a possible representation of a concept a *measure*, even if OLAP measures are generally limited to aggregated values.

Definition 9 (measure). *A* measure *is defined as any function from a concept to a piece of data representing some aspect of that concept. Given a multi-valued*

[1] Abilis is a Web interface (try it at `http://ledenez.insa-rennes.fr/abilis/`) on top of Camelis (download it at `http://www.irisa.fr/LIS/ferre/camelis/`)

context K, the measures for a concept c that we have implemented in Abilis are the extension $ext(c)$ (noted ext), the count $\#ext(c)$ (noted count), the index over some axis $index_x(ext(c))$ (noted x), and the aggregated index over some axis $\gamma(index_x(ext(c)))$ (noted $\gamma(x)$).

For example, the concept 1456 has the following representations.

measure	result
ext	$\{1, 4, 5, 6\}$
count	4
person/individual	$\{Alice \mapsto 4, Bob \mapsto 1, Charlie \mapsto 1\}$
date/year	$\{2010 \mapsto 1, 2011 \mapsto 3\}$
size/million	$\{1M \mapsto 2, 3M \mapsto 1\}$
sum(size/million)	$5M$

In Abilis, an extension is represented as a list of objects, which can be displayed only in part if too long. An index over some axis is a multiset of values, and can therefore be represented as a tag cloud, where the font size renders the multiplicity of values. An aggregated index is generally a numerical value, but it could be anything. For example, in a domain where values are geometrical shapes, an aggregated value can be a geometrical shape (e.g., union, centroid, buffer area). If those geometrical shapes are geo-located, they can be rendered on a map.

The definition of a cube of concepts can be refined as a *cube of concept measures*, which is defined by a tuple of measures in addition to the query and dimensions.

Definition 10 (cube of concept measures). *Given a query $q \subseteq F$, a tuple $\overline{d}$ of n axes as dimensions, and a tuple of p measures $\overline{m}$, the* cube of concept measures *is defined as*

$$Cube(q, \overline{d}, \overline{m}) = \{\overline{v} \mapsto (m_j(c))_{j \in 1..p} \mid \overline{v} \mapsto c \in Cube(q, \overline{d})\}.$$

After describing the possible representations of individual concepts, we need to describe the possible representations of cubes of concepts. In other words, how to represent the multi-dimensional structure of a cube in rich and flexible ways. Abilis provides the following structures:

Arrays. Arrays can be used for all dimensions. They use values as row/column labels, and their cells can contains arbitrary representations of dimensions and measures. There are three kinds of arrays: horizontal arrays, vertical arrays, and two-dimensional arrays (spreadsheets). The later represents two dimensions at a time.

Bar Charts. Bar charts can be used to represent the last dimension when the measure is an aggregated numerical value. There are horizontal and vertical bar charts.

Pie Charts. Pie charts can be used in the same conditions as bar charts.

Maps. Maps can be used to represent a geographical dimension.

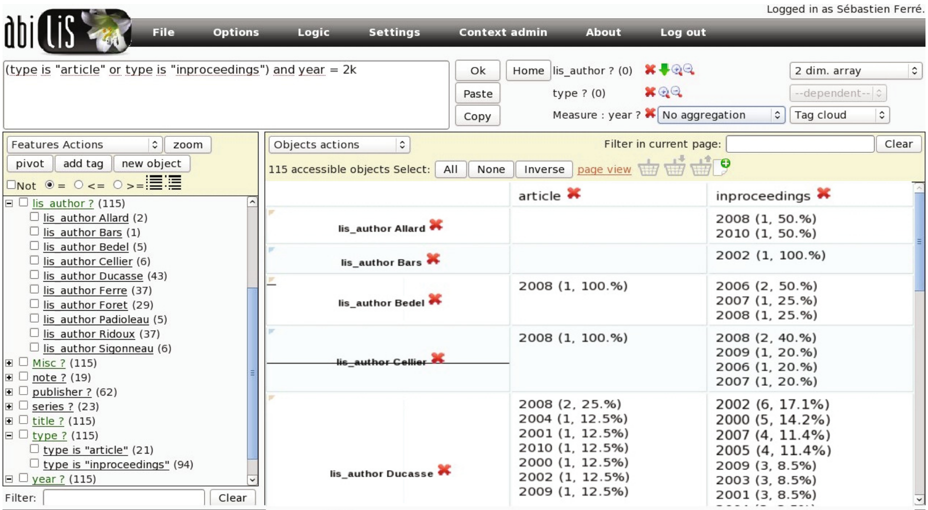


Fig. 1. Screenshot of Abilis showing the distribution over years of publications in journals and conferences since 2000 by author and by type of publication

Figure 1 is a screenshot of our prototype Abilis. It displays a cube of concept measures showing the distribution over years of publications in journals and conferences since 2000 by author and by type of publication. The query is at the top left, the feature index is at the bottom left, the selected dimensions and measure are at the top right along with representation choices, and the cube itself, here a two-dimensional array of tag clouds, is at the bottom right. There are many possible ways to navigate to this view from the initial cube $Cube(\emptyset, (), ext)$, because the definition of the query, dimensions, and measures can be interleaved arbitrarily. Here is a possible scenario. Initially, the list of all 208 publications of the context is displayed in a zero-dimensional cube. Then, this list is grouped by LIS team author by choosing the attribute `lis_author` as a dimension. This results in a vertical array of lists of publications indexed by author. Note that a same publication may appear in several lists (in several cells of the array) because a publication may have several authors (`lis_author` is multi-valued). Then, the results are restricted to journal and conference papers by selecting the two features `type is "article"` and `type is "inproceedings"`. Note that disjunction (and negation) is available in Abilis, like in other LIS systems. Then, by choosing the attribute `year` as a measure, lists of papers are replaced by tag clouds of years in each cell. From there, it would be possible to apply an aggregator such as average, minimum or maximum. Those tag clouds can be restricted to years since 2000 by adding the feature `year = 2k` to the query. Finally, in order to get a finer analysis of the distribution of years of publication, the attribute `type` is selected as an additional dimension, which results in a two-dimensional array of tag clouds. Note how the attribute `type` is used both as a dimension and in the query, and how the attribute `year` is used both as a measure and in the query.

6 Comparison with OLAP

OLAP (On-Line Analytical Processing) [5,20] is an approach to the multi-dimensional exploration of data, and it is part of the domain of business intelligence. OLAP does not add any expressiveness compared to relational databases, and in fact is less expressive, but it makes it much easier and quicker to perform aggregative queries than with SQL. The principle is to let users navigate from view to view, and in this respect, it follows the same goals as conceptual navigation. In this section, we compare our approach to OLAP, and we show that it covers all OLAP representations and operations, and that it allows for more flexibility and expressiveness than OLAP. However, it should be noted that there is a trade-off between expressiveness and efficiency, and that some of the restrictions seen in OLAP are useful to accelerate some computations.

In OLAP, a same structure, called an *OLAP cube*, is used both for representing data and for representing views. An OLAP cube is a multidimensional database that is defined by n dimensions (e.g., date and place), p measures (e.g., sales), and a mapping from tuples of n values (e.g., December 2011 and Rennes) to aggregated values for the measure (e.g., total sales). Therefore, an OLAP cube is equivalent to a cube of concept measures, where measures are all aggregated indexes. In our approach, data is represented as a multi-valued context, from which many different cubes can be defined by varying dimensions and measures. Moreover, elements of the cubes need not be aggregated values, but can also be sets of objects (extension), and multisets of values (indexes). In fact, the objects from which an OLAP cube may have been defined have been lost because they have been aggregated in the preprocessing stage. On the contrary, multi-valued contexts are object-centered.

The object-centered approach allows for more flexibility and heterogeneity in data, such as multi-valued attributes or missing values. Suppose we have the cube of the sum of the size of photos, by person: $\mathit{Cube}(\emptyset, \mathit{person}/\mathit{individual}, \mathit{sum}(\mathit{size}/\mathit{million})) = \{\mathit{Alice} \mapsto 5M, \mathit{Bob} \mapsto 6M, \mathit{Charlie} \mapsto 2M\}$. This is a valid OLAP cube. Now, suppose we want to aggregate those sizes to get the total size of photos. The result in OLAP would be $13M$, whereas the correct result, as given by our approach, is $9M$. This is because a photo can have several persons, and this is why OLAP assumes a strict partition between the values of a dimension.

OLAP defines a number of operators on cubes that play the same role as our cube transformations. We translate each of those operators in our approach:

Slice. The selection of a sub-cube of dimension $n-1$ by fixing the value of some dimension. In our approach, this is equivalent to adding a feature to the query, and removing a dimension. A difference is that any feature can be selected, whether it belongs to a dimension or a measure or none.

Add Dimension. The addition of a dimension. In OLAP, this is restricted to predefined dimensions, while in our approach, this can be any attribute.

Remove Dimension. The removal of a dimension. In OLAP, this necessarily entails an aggregation.

Pivot The swap of two dimensions. This is a particular case of permutation in our approach.

Drill-Down. The change of a dimension level for a finer granularity level (e.g., from years to months or weeks). In our approach, this is equivalent to removing a dimension axis a/λ, and adding the axis a/λ', where $\lambda' \prec \lambda$. Abilis provides a direct navigation link for drill-down.

Roll-Up. The converse of drill-down, i.e. the change of a dimension level for a coarser granularity level (e.g., from months to years).

This demonstrates the flexiblity and expressiveness of our approach. All attributes can be used in the query, dimensions, and measures, and a same attribute can play several roles at the same time. An attribute can even be used twice as a dimension, which makes sense when the attribute is multi-valued. For example, the cube $Cube(\emptyset, (person/individual, person/individual), count)$ displays the number of photos for each couple of persons (and for each person on the diagonal of the array).

Finally, our approach allows for drill-down and roll-up on the measures that are based on an axis, because an axis is parameterized by a level. This is particularly useful when the measure is an index, i.e. a multiset of values. For instance, we can display for each person, the multiset of the dates of their photos, and those dates can be displayed at the level of years or months or weeks or days. This has proved useful for discovering functional dependencies and association rules in multi-valued contexts [1].

7 Related Work

We compare our approach to other approaches combining FCA and OLAP. Stumme [19] describes *conceptual OLAP* for conceptual information systems. His approach is very similar to OLAP, except for the definition of hierarchies of dimension values, and therefore has the same limitations as OLAP (see Section 6). A many-valued context defines the multi-dimensional space. Each attribute defines a dimension, whose hierarchy of values is the conceptual scale derived from a scale context. Those conceptual scales have no defined levels, apart from the default topological levels as in Definition 3. The measures, called variables, are defined out of the context, as functions from objects of the many-valued context to values. Therefore, dimensions and measures are strongly separated, and only aggregated measures are available. The only structures for representing cubes are the line-diagrams of the conceptual scales, which have to be designed in advance for better presentation. Each line-diagram represents one dimension, and nested line-diagrams are used to represent multi-dimensional cubes. Measure values appear as labels of the concepts of the innermost line-diagrams. Nested line-diagrams are an alternative to nested arrays. They need more space but they better expose complex hierarchies of values.

Penkova and Korobko [17] apply standard FCA on cube schemas, instead of on cubes themselves. They start from a formal context where objects are measures, attributes are dimensions, and the incidence relation is the compatibility relation between measures and dimensions. For instance, in a dataset about the activities of a scientific organization, the dimension "journal name" is compatible with the measure "number of published paper" but not with "number of established conferences". A formal concept is a *maximal* cube schema: no dimension can be added without removing measures. The concept lattice can be used to guide users on the addition of dimensions (moving downward) or measures (moving upward). This approach is interesting when different kinds of objects are mixed (e.g., established conferences and published papers), and some dimensions only apply to some kinds of objects. In our approach, the feature index offers the same benefits by showing for each attribute whether it applies to all objects in the current cube, or only to a subset. And this comes without the usual limitations with OLAP: assymetry between dimensions and measures, only aggregated values as measures, etc. Applying our approach to the above examples, objects would be the individual published papers and established conferences, and their number would be obtained by choosing *count* as a measure. Other measures would allow to visualize the objects themselves, or the distribution of authors for papers.

8 Conclusion

The contribution of this paper is to extend conceptual navigation from single-concept views to cubes-of-concepts views. This means that, at each navigation step, the view is not limited to a single concept, but to a set of concepts, organized into a multi-dimensional cube. The single-concept view is a special case of a cube of concepts, having dimension 0. An important difference with OLAP is that each cube cell is a concept, which can be represented by a number of measures: a set of objects (the extension), a multiset of values (an attribute index), or an aggregated value. Finally, the cube need not cover the whole dataset, but can focus on a given subset, which is defined by a query.

Our approach generalizes OLAP cubes and navigation between cubes by relaxing a number of constraints. Cubes are derived from an object-centered multi-valued context, where no distinction is made between dimensions and measures, and where objects are not aggregated *a priori*. Objects can have several values for a same attribute. A same attribute can be used in a query, as a dimension, and as a measure at the same time. Drill-down and roll-up equally apply to dimensions and measures.

Our approach retains some constraints from OLAP relative to relational databases in terms of expressiveness. In particular, in a multi-valued context, the entities of a relational database must clearly be separated between objects and values. In another work, we have extended conceptual navigation and LIS to relational data from the Semantic Web [8]. Our goal is now to join the two extensions into one: multi-dimensional *and* relational conceptual navigation.

References

1. Allard, P., Ferré, S., Ridoux, O.: Discovering functional dependencies and association rules by navigating in a lattice of OLAP views. In: Kryszkiewicz, M., Obiedkov, S. (eds.) Concept Lattices and Their Applications, pp. 199–210. CEUR-WS (2010)
2. Antoniou, G., van Harmelen, F.: A Semantic Web Primer. MIT Press (2004)
3. Carpineto, C., Romano, G.: A lattice conceptual clustering system and its application to browsing retrieval. Machine Learning 24(2), 95–122 (1996)
4. Codd, E.F.: A relational model of data for large shared data banks. Communications of the ACM 13(6), 377–387 (1970)
5. Codd, E., Codd, S., Salley, C.: Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. Codd & Date, Inc., San Jose (1993)
6. Ducrou, J., Eklund, P.: An intelligent user interface for browsing and search MPEG-7 images using concept lattices. Int. J. Foundations of Computer Science 19(2), 359–381 (2008)
7. Ferré, S.: Camelis: a logical information system to organize and browse a collection of documents. Int. J. General Systems 38(4) (2009)
8. Ferré, S.: Conceptual Navigation in RDF Graphs with SPARQL-Like Queries. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 193–208. Springer, Heidelberg (2010)
9. Ferré, S., Ridoux, O.: A Logical Generalization of Formal Concept Analysis. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS(LNAI), vol. 1867, pp. 371–384. Springer, Heidelberg (2000)
10. Ferré, S., Ridoux, O.: An introduction to logical information systems. Information Processing & Management 40(3), 383–419 (2004)
11. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
12. Ganter, B., Wille, R.: Conceptual scaling. In: Roberts, F. (ed.) Applications of Combinatorics and Graph Theory to the Biological and Social Sciences, pp. 139–167. Springer (1989)
13. Ganter, B., Wille, R.: Formal Concept Analysis — Mathematical Foundations. Springer (1999)
14. Godin, R., Missaoui, R., April, A.: Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. International Journal of Man-Machine Studies 38(5), 747–767 (1993)
15. Lindig, C.: Concept-based component retrieval. In: IJCAI Work. Formal Approaches to the Reuse of Plans, Proofs, and Programs. Morgan Kaufmann (1995)
16. Marchionini, G.: Exploratory search: from finding to understanding. Communications of the ACM 49(4), 41–46 (2006)
17. Penkova, T., Korobko, A.: Constructing the integral OLAP-model based on formal concept analysis. In: MIPRO, International Convention, pp. 1544–1548. IEEE (2011)
18. Sacco, G.M., Tzitzikas, Y. (eds.): Dynamic taxonomies and faceted search. The information retrieval. Springer (2009)
19. Stumme, G.: Conceptual on-line analytical processing. In: Tanaka, K., Ghandeharizadeh, S., Kambayashi, Y. (eds.) Information Organization and Databases. The Kluwer International Series in Engineering and Computer Science, vol. 579, pp. 191–203. Springer, US (2001)
20. Vassiliadis, P., Sellis, T.K.: A survey of logical models for OLAP databases. SIGMOD Record 28(4), 64–69 (1999)

Ordinal Factor Analysis

Bernhard Ganter and Cynthia Vera Glodeanu

Institut für Algebra,
Technische Universität Dresden
Bernhard.Ganter@tu-dresden.de,
Cynthia_Vera.Glodeanu@mailbox.tu-dresden.de

Abstract. We build on investigations by Keprt, Snásel, Belohlavek, and Vychodil on Boolean Factor Analysis. Rather than minimising the number of Boolean factors we aim at *many-valued factorisations* with a small number of *ordinal* factors.

Keywords: Factor analysis, Order dimension, Ordinal factor.

1 Introduction

Factor Analysis, in particular Principal Component Analysis, is a popular technique for analysing metric data. It allows for complexity reduction, representing a large part of the given data by a (preferably) low number of unobserved "latent" attributes.

Recently a similar approach was discussed for qualitative data, for data that can be represented in a formal context, and a nice strategy for finding so-called Boolean factors was found.

However, such Boolean factors have limited expressiveness due to their unary nature. It can hardly be expected that much of a complex data set can be captured by only a few Boolean factors.

But even a large factorisation may be useful provided the factors are conceptually "well behaved" and can be grouped into well-structured families, which then may be interpreted as many-valued factors. These are given by the conceptual standard scales of Formal Concept Analysis. Here we focus on the case of (one-dimensional) ordinal scales.

2 Conceptual Factorisation

Definition 1. *A* ***factorisation*** *of a formal context* (G, M, I) *consists of formal contexts* (G, F, I_{GF}) *and* (F, M, I_{FM}) *such that*

$$g\ I\ m \iff g\ I_{GF}\ f \text{ and } f\ I_{FM}\ m \quad \text{for some } f \in F.$$

The elements of F *are called* ***Boolean factors,*** (G, F, I_{GF}) *and* (F, M, I_{FM}) *are the* ***factorisation contexts****. We write*

$$(G, M, I) = (G, F, I_{GF}) \circ (F, M, I_{FM})$$

to indicate a factorisation.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 128–139, 2012.

The investigation of such factorisations and their relation to Formal Concept Analysis goes back to A. Keprt and V. Snásel (see [1] and [2]) and was pushed forward by R. Belohlavek and V. Vychodil (see e.g. [3] and [4]) under the name of “Boolean Factor Analysis” (but see also e.g., P. De Boeck and S. Rosenberg [5]). Extensive information about related work is given in [4]. For better compatibility with the language of Formal Concept Analysis we slightly deviate from these authors' terminology.

To each factorisation there corresponds a **factorising family**

$$\{(A_f, B_f) \mid f \in F\}$$

given by

$$A_f := \{g \in G \mid g\ I_{GF}\ f\} \quad \text{and} \quad B_f := \{m \in M \mid f\ I_{FM}\ m\}.$$

Such families are easy to characterise: A family $\{(A_f, B_f) \mid f \in F\}$ is a factorising family of (G, M, I) iff

$$I = \bigcup_{f \in F} A_f \times B_f.$$

Expressed in words this says that the factorising families are precisely the families of preconcepts of (G, M, I) covering all incidences. As an obvious consequence we get that each factorisation is uniquely determined by its factorising family.

Each preconcept can be enlarged to a formal concept (though not uniquely). Each factorising family therefore may, without increasing the number of Boolean factors, be made to a factorising family of concepts. Such a factorisation will be called **conceptual**.

A key question of the abovementioned investigations concerned *optimal* factorisations, i.e. such with the smallest possible number of Boolean factors. The task of filling a given relation $I \subseteq G \times M$ by as few as possible “rectangles” $A \times B \subseteq I$ had been studied earlier under the name **set dimension** (see [6] and the literature cited there), and is known to be difficult. There is a close connection to the **2-dimension** of the complementary context, which is the number of atoms of the smallest Boolean algebra that admits an order embedding of the concept lattice of the complementary context $\underline{\mathfrak{B}}(G, M, G \times M \setminus I)$. Indeed, the following proposition is an easy consequence of [4] and the dimension theory in [6]:

Proposition 1. *The smallest possible number of Boolean factors of* (G, M, I) *equals the 2-dimension of* $\underline{\mathfrak{B}}(G, M, G \times M \setminus I)$.

Example 1. Consider the formal context in Figure 1, which is part of Wille's data on *Components of musical experience*, see [7]. It can be factored using seven Boolean factors (in several ways, one of which is shown in Figure 3). This is no big deal since there is a trivial factorisation with eight Boolean factors (see below). The concept lattice of the complementary context is depicted in Figure 2. It contains a chain of length six with a narrow M_3. This implies that it cannot be embedded into a 6-dimensional Boolean algebra. Therefore seven is the smallest possible number of Boolean factors.

	m_1 : rounded	m_2 : balanced	m_3 : dramatic	m_4 : transparent	m_5 : well-structured	m_6 : strong	m_7 : lively	m_8 : sprightly	m_9 : rhythmising	m_{10}: fast	m_{11}: playful
g_1	×	×		×	×						×
g_2			×			×			×	×	
g_3			×	×		×	×		×	×	
g_4	×	×		×	×		×	×			×
g_5			×	×	×	×			×	×	
g_6	×	×	×	×	×	×	×		×		
g_7	×	×		×	×	×					
g_8			×		×	×	×			×	

Fig. 1. Components of musical experience (from R. Wille [7], part). The objects are musical pieces, six of them by L.v.Beethoven (g_1: Romance for violin and orchestra F-major, g_2: 9th symphony, 4th movement (presto), g_3: Moonlight sonata, 3rd movement, g_4: Spring sonata, 1st movement, g_5: String quartet op.131, final movement, g_6: Great fuge op. 133) and two by J.S.Bach (g_7: Contrapunctus I, g_8: WTP 1, prelude c minor). Boldface crosses indicate tight incidences.

An incident object-attribute pair $(g, m) \in I$ is called **tight** iff there is no pair $(h, n) \in I$ such that

$$[\gamma h, \mu n] \subsetneq [\gamma g, \mu m].$$

It is easy to see that for a conceptual factorisation it suffices to cover the tight incidences by formal concepts, because the other incidences will then automatically be covered. This is sometimes useful for the computation of the set dimension, since it reduces the size of the corresponding set cover problem. The tight incidences correspond to the double arrows of the complementary context.

Note that in a conceptual factorisation of (G, M, I) the second factorisation context is determined by the first. Indeed, we get from $B_f = A_f^I$ that

$$f \; I_{FM} \; m \iff m \in A_f^I = (f^{I_{GF}})^I.$$

Proposition 2. *For any conceptual factorisation with factor set F the dual attribute order of (G, F, I_{GF}) is the same as the object order of (F, M, I_{FM}).*

Proof. If $f_1^{I_{GF}} \subseteq f_2^{I_{GF}}$ then $A_{f_1} = f_1^{I_{GF}} \subseteq f_2^{I_{GF}} = A_{f_2}$ and thus $B_{f_1} = A_{f_1}^I \supseteq A_{f_2}^I = B_{f_2}$ and therefore $f_1^{I_{FM}} \supseteq f_2^{I_{FM}}$. The converse is similar.

The condition given in this proposition is not in general sufficient. The next theorem characterises the conceptual factorisation contexts. It turns out that a

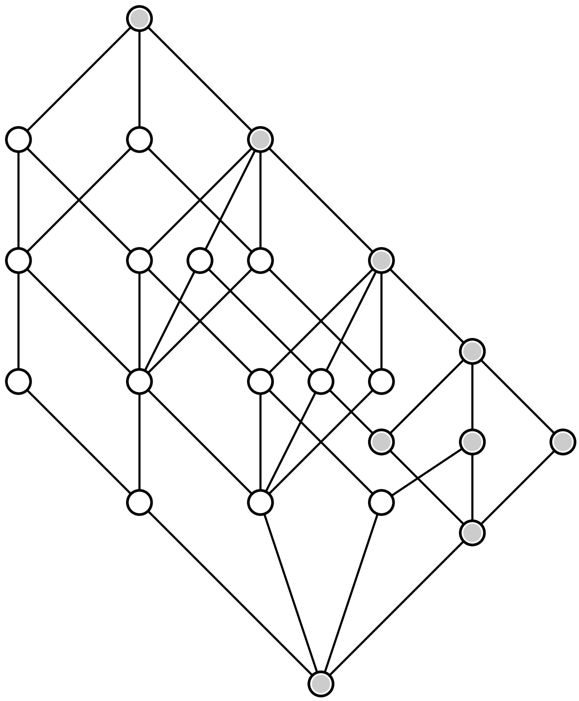


Fig. 2. A diagram of the concept lattice of the formal context $(G, M, G \times M \setminus I)$, where (G, M, I) is given in Figure 1. The shaded substructure does not fit into a Boolean algebra with six atoms.

	f^1	f^2	f^3	f^4	f^5	f^6	f^7
g_1	×						×
g_2			×				
g_3			×		×		
g_4	×	×					×
g_5			×				×
g_6				×	×		×
g_7				×			×
g_8						×	

$\circ$

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}
f^1	×	×		×	×						×
f^2	×	×		×	×		×	×			×
f^3			×			×			×	×	
f^4	×	×		×	×	×					
f^5			×	×			×		×		
f^6			×		×	×	×			×	
f^7				×	×						

Fig. 3. A conceptual factorisation of the formal context from Figure 1

conceptual factorisation links each factor context to the complementary context of the other.

Theorem 1. *(G, F, I_{GF}) and (F, M, I_{FM}) are the factorisation contexts of a conceptual factorisation if and only if*

1. *all intents of (G, F, I_{GF}) are extents of $(F, M, F \times M \setminus I_{FM})$, and*
2. *all extents of (F, M, I_{FM}) are intents of $(G, F, G \times F \setminus I_{GF})$.*

Proof. Start with a conceptual factorisation, and recall that

$$g\, I\, m \iff g^{I_{GF}} \cap m^{I_{FM}} \neq \emptyset,$$

which is equivalent to

$$g\, \not I\, m \iff g^{I_{GF}} \subseteq F \setminus m^{I_{FM}}.$$

For arbitrary $g \in G$ we ask if the object intent $g^{I_{GF}}$ is the intersection of attribute extents of $(F, M, F \times M \setminus I_{FM})$. Suppose not. Then there must be some $f \in F$ which is contained in all attribute extents of $(F, M, F \times M \setminus I_{FM})$ that contain $g^{I_{GF}}$, but which does not belong to $g^{I_{GF}}$. From $g \notin f^{I_{GF}} = A_f$ we infer that $g \notin B_f^I = (f^{I_{FM}})^I$. Consequently there must be some $m \in f^{I_{FM}}$ with $g \not I m$. This is, as stated above, equivalent to $g^{I_{GF}} \subseteq F \setminus m^{I_{FM}}$. But then $F \setminus m^{I_{FM}}$ is an attribute extent of $(F, M, F \times M \setminus I_{FM})$ containing $g^{I_{GF}}$ and not f, a contradiction.

For the converse direction suppose that the conditions are satisfied. In order to show that the factorisation is conceptual, we need to prove that $(f^{I_{GF}})^I \subseteq f^{I_{FM}}$ and dually $(f^{I_{FM}})^I \subseteq f^{I_{GF}}$ hold for all $f \in F$. Now assume that $m \notin f^{I_{FM}}$, which is the same as $f \notin m^{I_{FM}}$. Since $m^{I_{FM}}$ is an extent of (F, M, I_{FM}), there must be, according to the second condition, an object intent of $(G, F, G \times F \setminus I_{GF})$ containing $m^{I_{FM}}$, but not f. In other words, there must be an object $g \in f^{I_{GF}}$ such that $g^{I_{GF}} \cap m^{I_{FM}} = \emptyset$. Thus $g \not I m$, i.e., $m \notin (f^{I_{GF}})^I$.

Note that every formal context (G, M, I) is a conceptual factor context in its **trivial factorisations**, one of them being

$$(G, M, I) = (G, M, I) \circ (M, M, \rightarrow),$$

where $m \rightarrow n :\iff n \in m^{II}$. The other is defined dually. Theorem 1 therefore imposes no restriction on single factorising contexts.

Proposition 3. *If (G, F, I_{GF}) and (F, M, I_{FM}) are conceptual factorisation contexts and $E \subseteq F$, then $(G, E, I_{GF} \cap (G \times E))$ and $(E, M, I_{FM} \cap (E \times M))$ also are conceptual factorisation contexts.*

Proof. Let

$$(G, M, I_E) := (G, E, I_{GF} \cap (G \times E)) \circ (E, M, I_{FM} \cap (E \times M)).$$

Each (A_e, B_e), $e \in E$, is a formal concept of (G, M, I) and, since $I_E \subseteq I$, also of (G, M, I_E).

3 Ordinal Factors

The set F of Boolean factors may be large and should then, for the sake of better interpretability, be divided into conceptually meaningful subsets. An ordinal factor, for example, simply represents a chain of Boolean factors.

Definition 2. *If* (G, F, I_{GF}) *and* (F, M, I_{FM}) *are conceptual factorisation contexts of* (G, M, I) *and* $E \subseteq F$*, then* $(G, E, I_{GF} \cap (G \times E))$ *is called a* ***(many-valued) factor*** *of* (G, M, I)*.*

Many-valued factors are closely related to the **scale measures** described in [6]:

Definition 3. *Let* $\mathbb{K} := (G, M, I)$ *and* $\mathbb{S} := (G_{\mathbb{S}}, M_{\mathbb{S}}, I_{\mathbb{S}})$ *be formal contexts. An* $\mathbb{S}$*-**measure** is a map*

$$\sigma : G \to G_{\mathbb{S}}$$

with the property that the preimage $\sigma^{-1}(E)$ *of every extent* E *of* $\mathbb{S}$ *is an extent of* $\mathbb{K}$*. An* $\mathbb{S}$*-**measure** is called **full**, if every extent of* (G, M, I) *is the preimage of some* $\mathbb{S}$*-extent.*

Proposition 4. $\mathbb{S} := (G, F, I_{GF})$ *is a factor of* (G, M, I) *if and only if the identity map is an* $\mathbb{S}$*-measure.*

Proof. Clearly $\mathbb{S} := (G, F, I_{GF})$ is a factor of (G, M, I) iff each attribute extent $f^{I_{GF}}$ is an extent of (G, M, I).

Definition 4. *A factor* (G, F, I_{GF}) *of* (G, M, I) *is called an* $\mathbb{S}$*-factor if it has a surjective full* $\mathbb{S}$*-measure. If* $\mathbb{S}$ *is an elementary ordinal, or nominal, etc., scale, we speak of an ordinal or nominal factor, etc. Moreover, we say that* (G, M, I) *has an **ordinal** (nominal, etc.) **factorisation** iff it has a first factorising context that can be written as an apposition of ordinal (nominal, etc.) factors.*

In other words: The first factorising context of an ordinal factorisation must be a derived context of a many-valued context with respect to some ordinal scaling.

Example 2. Consider again the formal context of Figure 1. It can be ordinally factored, using 12 Boolean factors, as shown in Figure 4. It becomes apparent that the first factor context is a derived context of an ordinally scaled many-valued context with three many-valued attributes, and the second factor context is the dual of such a derived context, but with reverse scaling. Such many-valued contexts are given in Figure 5. The conceptual scales for the first and the second factor context are shown in Figure 6.

It is somewhat tempting, but highly experimental, to plot the two major factors (which cover 41 of the 46 incidences, i.e., 89% of the data) as it is usual in (numerical) Factor Analysis. Such a diagram is shown in Figure 7. A representation like this may however be misleading, since it displays purely ordinal data in a metric fashion. An additional source of misinterpretation is that the two "dimensions" represent ordinal, not interordinal ("bipolar") data. However, the diagram indicates that ordinal factor analysis, when interpreted correctly, has some expressiveness similar to Factor Analysis based on metric data.

The following proposition is evident:

Proposition 5. *A formal context is an ordinal factor of* (G, M, I) *iff its attribute extents are a linearly ordered family of concept extents of* (G, M, I)*.*

	f_1^1	f_2^1	f_3^1	f_4^1	f_5^1	f_1^2	f_2^2	f_3^2	f_4^2	f_5^2	f_1^3	f_2^3
g_1	×	×	×	×								
g_2						×	×	×	×			
g_3						×	×	×	×	×	×	
g_4	×	×	×	×	×						×	
g_5	×	×				×	×	×	×			
g_6	×	×	×			×	×	×			×	
g_7	×	×	×			×						
g_8	×					×	×				×	×

$\circ$

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}
f_1^1					×						
f_2^1				×	×						
f_3^1	×	×		×	×						
f_4^1	×	×		×	×						×
f_5^1	×	×		×	×		×	×			×
f_1^2						×					
f_2^2			×			×					
f_3^2			×			×			×		
f_4^2			×			×			×	×	
f_5^2			×	×		×	×		×	×	
f_1^3							×				
f_2^3			×		×	×	×			×	

Fig. 4. An ordinal factorisation of the formal context from Figure 1

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}
g_1	×	×		×	×						×
g_2			×			×			×	×	
g_3			×	×		×	×		×	×	
g_4	×	×		×	×		×	×			×
g_5			×	×	×	×			×	×	
g_6	×	×	×	×	×	×	×		×		
g_7	×	×		×	×	×					
g_8			×		×	×	×			×	

$=$

	f^1	f^2	f^3
g_1	4	0	0
g_2	0	4	0
g_3	0	5	1
g_4	5	0	1
g_5	2	4	0
g_6	3	3	1
g_7	3	1	0
g_8	1	2	2

$\circ$

d

	f^1	f^2	f^3
m_1	2	5	2
m_2	2	5	2
m_3	5	1	1
m_4	1	4	2
m_5	0	5	1
m_6	5	0	1
m_7	4	4	0
m_8	4	5	2
m_9	5	2	2
m_{10}	5	3	1
m_{11}	3	5	2

Fig. 5. Symbolic notation of the ordinal factorisation defined in Figure 4. It holds that $g_i \; I_{GF} \; f_j^k \iff f^k(g_i) \geq j$ and that $f_j^k \; I_{FM} \; m_i \iff f^k(m_i) < j$. Thus $g \; I \; m \iff f^k(g) > f^k(m)$ for some k. Reformulated, $g \not\!I \; m \iff f^k(g) \leq f^k(m)$ for all k.

	≥ 1	≥ 2	≥ 3	≥ 4	≥ 5
0					
1	×				
2	×	×			
3	×	×	×		
4	×	×	×	×	
5	×	×	×	×	×

	< 1	< 2	< 3	< 4	< 5
0	×	×	×	×	×
1		×	×	×	×
2			×	×	×
3				×	×
4					×
5					

Fig. 6. Conceptual scales for the two many-valued factor contexts in Figure 5

For an ordinal factorisation there must be a partition $\{F_d \mid d \in D\}$ of the set F of factors such that within each class the attribute order of (G, F, I_{GF}) is linear. According to Proposition 2 the attribute order is dual to that of $(M, F, {I_{FM}}^d)$. This gives the following proposition:

Proposition 6. *For any ordinal factorisation the dual of the second factorisation context also is a derived context of the same many valued context, but with reversely ordered ordinal scales.*

Definition 5. *A relation $R \subseteq G \times M$ is called a **Ferrers relation** iff there are subsets $A_1 \subset A_2 \subset A_3 \ldots \subseteq G$ and $M \supseteq B_1 \supset B_2 \supset B_3 \supset \ldots$ such that $R = \bigcup_i A_i \times B_i$. R is called a Ferrers relation **of concepts** of (G, M, I) iff there are formal concepts $(A_1, B_1) \leq (A_2, B_2) \leq (A_3, B_3) \leq \ldots$ such that $R = \bigcup_i A_i \times B_i$.*

It is well known that a relation $R \subseteq G \times M$ is a Ferrers relation iff the concept lattice $\underline{\mathfrak{B}}(G, M, R)$ is a chain.

Proposition 7. *Any Ferrers relation $R \subseteq I$ is contained in a Ferrers relation of concepts of (G, M, I).*

Proof. If $A_i \times B_i \subseteq I$ then $A_i \times B_i \subseteq A_i'' \times A_i'$. Thus if $R = \bigcup_i A_i \times B_i \subseteq I$ then $R \subseteq \overline{R} := \bigcup_i A_i'' \times A_i' \subseteq I$, and $\overline{R}$ is a Ferrers relation of concepts.

Definition 6. *The **width** of a factorising family $\mathcal{F}$ of concepts is the largest number of pairwise incomparable elements of $\mathcal{F}$. The **ordinal factorisation width** of (G, M, I) is the smallest width of a factorising family of concepts.*

Theorem 2. *The following are equivalent:*

1. *(G, M, I) has ordinal factorisation width $\leq n$.*
2. *(G, M, I) has an ordinal factorisation with $\leq n$ ordinal factors.*
3. *$\underline{\mathfrak{B}}(G, M, G \times M \setminus I)$ has order dimension $\leq n$.*
4. *I can be written as a union of $\leq n$ Ferrers relations.*

Proof. (1) ⇒ (2): (G, M, I) has ordinal factorisation width $\leq n$ iff there is a factorising family $\mathcal{F}$ of concepts, which as an ordered subset of the concept

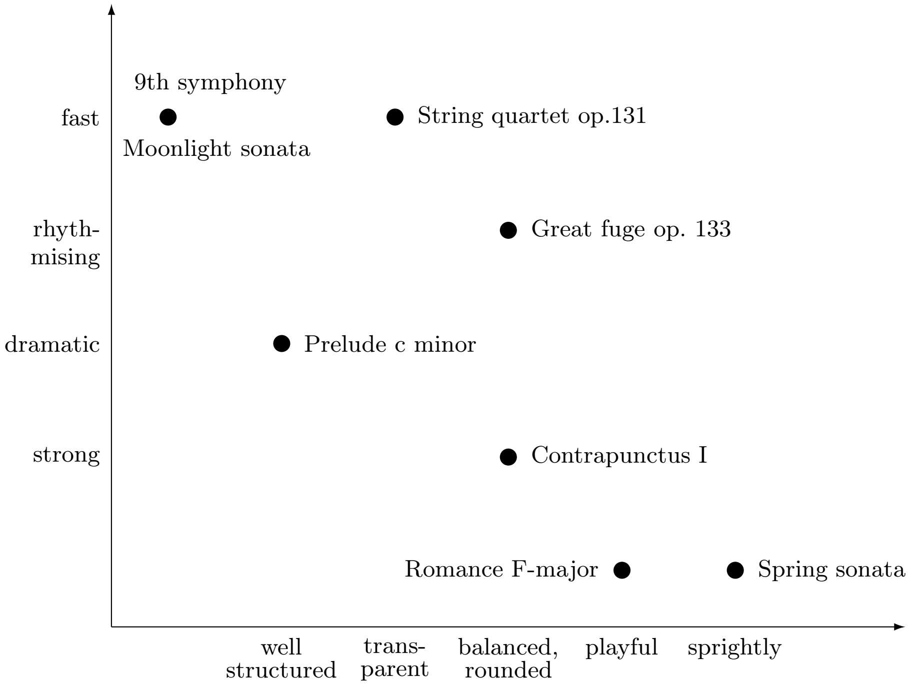


Fig. 7. A "biplot" of the data in Figure 1, based on the first two factors of the ordinal factorisation in Figure 4. Note that no metric information is encoded here. The diagram is based on ordinal data only.

	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}
g_1	1	1		1	1						1
g_2			2			2			2	2	
g_3			2	2		2	3		2	2	
g_4	1	1		1	1		3	1			1
g_5			2	1	1	2			2	2	
g_6	1	1	2	1	1	2	3		2		
g_7	1	1		1	1	2					
g_8			2		1	2	3			3	

Fig. 8. Three Ferrers relations, the union of which is the incidence relation of the formal context in Figure 1

lattice has width $\leq n$. By a classical theorem of R.P. Dilworth this implies that $\mathcal{F}$ can be covered by $\leq n$ chains, i.e., linear ordered families of concepts, each of which induces an ordinal factor. This proves (2).

(2) $\Rightarrow$ (4): The factorising family of an ordinal factor is a chain of concepts, and the incidences occurring in such a chain form a Ferrers relation.

(3) $\Leftrightarrow$ (4) is well known, see e.g. [6].

(4) $\Rightarrow$ (1): If I can be written as a union of $\leq n$ Ferrers relations it can, according to Proposition 7, also be written as a union of $\leq n$ Ferrers relations of concepts. These concepts form a factorising family of width $\leq n$.

Example 3. Consider once more the formal context of Figure 1. Its incidence relation I can indeed be covered by three Ferrers relations, as can be seen from Figure 8.

So the ordinal width of the formal context in Figure 1 equals three (a smaller value is obviously impossible). This was to be expected, since an ordinal factorisation with three ordinal factors was given in Figure 5. Moreover the order dimension of the lattice in Figure 2 is apparently equal to three. The concept

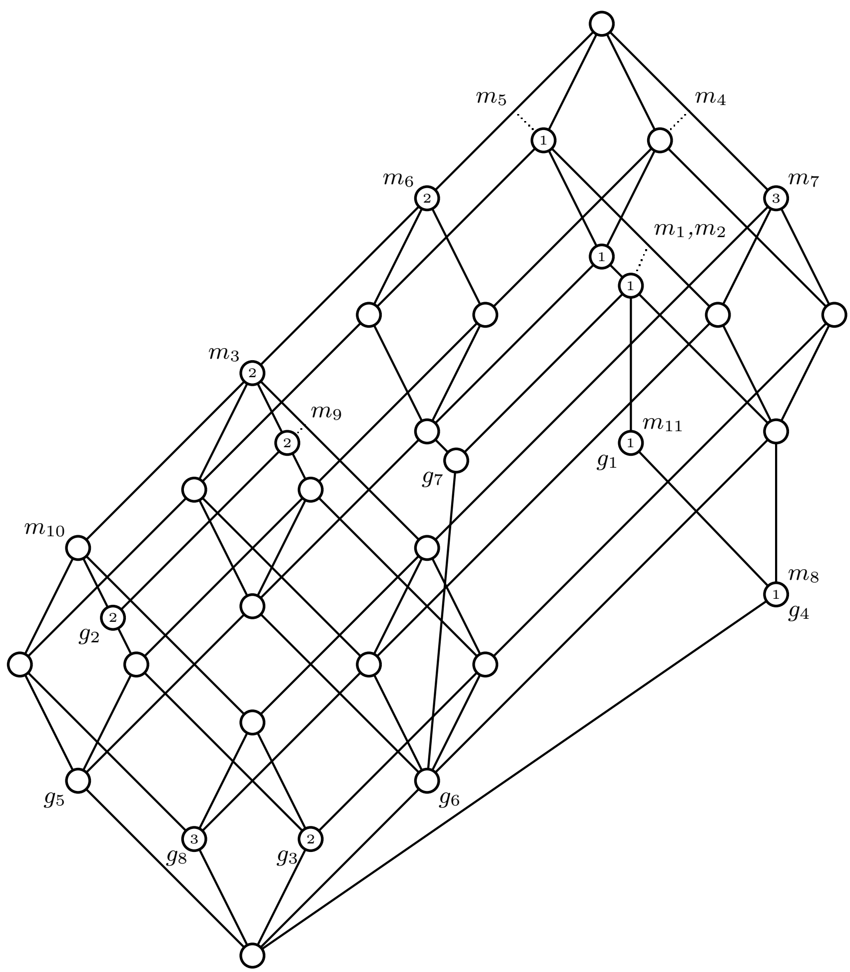


Fig. 9. The concept lattice of the formal context in Figure 1. The encircled numbers mark a factorising family of concepts of width three.

lattice of the formal context in Figure 1 is shown in Figure 9. Three chains are marked in the diagram. These cover all (tight) incidences, i.e., whenever $(g, m) \in I$, then the interval $[\gamma g, \mu m]$ contains some concept from one of these chains. Therefore these concepts form a factorising family of width three.

An immediate consequence of Theorem 2 is that for any $k \geq 3$ the decision problem if a formal context has factorisation width $\leq k$ is $\mathcal{NP}$-complete. This follows from Yannakakis' [8] result that "order dimension $\leq k$" is hard to decide for $k \geq 3$. Another consequence is that one can easily determine the factorisation width of some elementary scales:

Corollary 1. *1. (G, M, I) has ordinal factorisation width 1 iff I is Ferrers.*
2. *The (one-dimensional) contraordinal scale has ordinal factorisation width 2, independent of its size (> 1).*
3. *The interordinal scale has ordinal factorisation width 2, independent of its size (> 1).*

The corollary gives first clues of how algorithmically difficult interordinal and contraordinal factorisation (yet to be developed) will be. The *nominal* scale with n scale values obviously has ordinal factorisation width n.

4 Conclusion

Many-valued factorisations of formal contexts were introduced in this paper as Boolean factorisations where the Boolean factors are grouped into attribute sets of conceptual scales. We have studied to some extent the *ordinal* case, proving that the problem of finding an ordinal factorisation with few factors is equivalent to that of determining the order dimension of the complementary context (and therefore is difficult). For a small example it was demonstrated that ordinal factor analysis of empirical qualitative data may lead to results that are similar to those of numerical factor analysis of metric data.

References

1. Keprt, A., Snásel, V.: Binary factor analysis with help of formal concepts. In: Snásel, V., Belohlávek, R., (eds.) CLA. CEUR Workshop Proceedings, vol. 110. CEUR-WS.org (2004)
2. Keprt, A.: Algorithms for Binary Factor Analysis. PhD thesis (2006)
3. Belohlávek, R., Vychodil, V.: Formal concepts as optimal factors in boolean factor analysis: Implications and experiments. In: Eklund, P.W., Diatta, J., Liquiere, M., (eds.) CLA. CEUR Workshop Proceedings, vol. 331. CEUR-WS.org (2007)
4. Belohlávek, R., Vychodil, V.: Discovery of optimal factors in binary data via a novel method of matrix decomposition. J. Comput. Syst. Sci. 76(1), 3–20 (2010)
5. De Boeck, P., Rosenberg, S.: Hierarchical classes: model and data analysis. Psychometrika 53, 361–381 (1988)

6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
7. Wille, R.: Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 1–33. Springer, Heidelberg (2005)
8. Yannakakis, M.: The complexity of the partial order dimension problem. SIAM Journal on Matrix Analysis and Applications 3(3), 351–358 (1982), http://dx.doi.org/10.1137/0603036

A Macroscopic Approach to FCA and Its Various Fuzzifications

Tim B. Kaiser[1] and Stefan E. Schmidt[2]

[1] SAP AG, Walldorf
tbkaiser@gmx.de
[2] Institut für Algebra, Technische Universität Dresden
midt1@msn.com

Abstract. We promote biresiduation as a fundamental unifying principle in Formal Concept Analysis, including fuzzification and factor analysis. In particular, we show that maximal formal rectangles are exactly formal concepts within the presented framework of biresiduated maps on ordered sets. Macroscopic implications yield the particular derivation operators in specific settings such as Fuzzy Formal Concept Analysis, Factor Analysis, and degree of containment (i.e. degree of being a subset).

Keywords: biresiduation, complete monoids, formal concept analysis, fuzzy formal concept analysis, factor analysis, linear algebra.

1 Introduction

We promote biresiduation as a fundamental unifying principle in Formal Concept Analysis, including fuzzification and factor analysis. In particular, we show that maximal formal rectangles are exactly formal concepts within the presented framework of biresiduated maps on ordered sets. Macroscopic implications yield the particular derivation operators in specific settings such as Fuzzy Formal Concept Analysis, Factor Analysis, and degree of containment, that is, degree of being a subset.

Most of the macros presented in our paper are implicitely already introduced in [Bel11a], Section 5, using lattices as basic structures. Nonetheless, we add value by showing that (1) the maximal rectangle paradigm survives in the general setting and (2) how biadditivity completes the picture and that (3) ordered sets can be used instead of lattices.

Important literature is given by [GW99], [Kra04], [MOARC09], [MOA09], [DP90], regarding factor analysis, especially by [BV09]. Belohlávek gives an overview on approaches to fuzzy concept analysis in [Bel11b] which is an update of [BV05].

We assume the reader to be familiar with ordered sets and formal concept analysis as exposed in [DP90] and [GW99]. Profound informaton on residuation theory can be found, for instance, in [Bly05].

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 140–147, 2012.

2 Biresiduation

One of the key concepts for accessing FCA and its generalizations is that of a biresiduated map. We start by recalling the definition of a residuated map.

Definition 1 (residuated map). *Let P_1 and P_2 be ordered sets. Then $r : P_1 \to P_2$ is a* residuated map *if there exists a map $r^+ : P_2 \to P_1$ such that*

$$r(p_1) \leq p_2 \iff p_1 \leq r^+(p_2).$$

The map r^+ is unique if it exists and is then called the residual of r.

We can consider a binary operation $\otimes : P_1 \times P_2 \to P$ from the cartesian product of two ordered sets P_1 and P_2 into a third ordered set P as unary if we fix one of its two arguments. Let us pick a $p_1 \in P_1$. Then $\otimes(p_1, \cdot) : P_2 \to P$ yields a corresponding unary operation.If $\otimes(p_1, \cdot)$ is residuated for all $p_1 \in P_1$, we call the binary operation residuated in its first argument and denote the corresponding residuals by $\to (p_1, \cdot) : P \to P_2$. We set $\otimes(p_1, p_2) = p_1 \otimes p_2$ and $\to (p_1, p) = p_1 \to p$. For all $p_1 \in P_1$ we have

$$p_1 \otimes p_2 \leq p \iff p_2 \leq p_1 \to p.$$

The analogous procedure applied to the second argument yields the residuals $\leftarrow (\cdot, p_2) : P \to P_1$ and using infix notation we set $\leftarrow (p, p_2) = p \leftarrow p_2$.

The following definition which is central for this paper uses the above construction on both arguments.

Definition 2 (biresiduated map, biresiduation). *Let P_1, P_2, and P be ordered sets. Then*

$$\otimes : P_1 \times P_2 \to P$$

is a biresiduated map, short biresiduation, if $\otimes$ is residuated in both arguments.

If $\otimes$ is a biresiduated map we have

$$p_1 \leq p \leftarrow p_2 \iff p_1 \otimes p_2 \leq p \iff p_2 \leq p_1 \to p$$

for all $p_1 \in P_1$, $p_2 \in P_2$, and $p \in P$ by definition.

Proposition 1. *Let $\otimes : P_1 \times P_2 \to P$ be a biresiduation. Then $\otimes$ is monotone in both arguments.*

Proof. *Let $q, r \in P_2$ and $q \leq r$. Since $p_1 \to (p_1 \otimes \cdot)$ is a closure operator, $q \leq r$ implies $q \leq p_1 \to (p_1 \otimes r)$. The latter is equivalent to $p_1 \otimes q \leq p_1 \otimes r$. By symmetry, a biresiduation is monotone in the first argument as well.*

Example 1. We will show that the classical case embeds itself nicely into our setting. Consider the mapping:

$$\times : 2^G \times 2^M \to 2^{G \times M}, (A, B) \mapsto A \times B.$$

Since $\times$ operates on lattices it is enough to note that it is join-preserving in both arguments to validate that it is a biresiduation. Let $\alpha \in 2^{G\times M}$. We have $A \times B \subseteq \alpha \iff A \subseteq B \to \alpha$ where

$$B \to \alpha = \max\{H \subseteq G \mid H \times B \subseteq \alpha\} = B'.$$

Dually, we get

$$\alpha \leftarrow A = \max\{N \subseteq M \mid A \times N \subseteq \alpha\} = A'.$$

We have recaptured the derivation operators of classical formal concept analysis as residuals of a specific biresiduation, the cartesian product.

Now, it is worth noting the connection between cartesian product and dyadic product as used in linear algebra. We recall the definition of a dyadic product. Given two n-dimensional vectors $\boldsymbol{u}, \boldsymbol{v}$ over a semiring S we can define the dyadic product as

$$\boldsymbol{u} \otimes \boldsymbol{v} := \boldsymbol{u} \cdot \boldsymbol{v}^T$$

where the second multiplication is simply matrix multiplication. If S is the well-known boolean 2-element semiring, the dyadic product resembles exactly the cartesian product. So, in a sense, the dyadic product generalizes the cartesian product. From our point of view, this is key to understanding fuzzy concept analysis in terms of biresiduations.

3 Biadditivity

We complement the approach sketched above (where ordered sets are used as basic structures) by using complete monoids [DKV09] as basic structures.

Definition 3 (complete monoid). *A quadruple $\mathbb{A} := (A, +, 0, \Sigma)$ is called a complete monoid if $(A, +, 0)$ is a commutative monoid and Σ assigns to every $\alpha \in A^I$ (for an arbitrary index set I) an element $\Sigma\alpha =: \Sigma_{i\in I}\alpha(i)$ of A such that*

1. $\Sigma\alpha = 0$ *if* $\alpha(i) = 0$ *for all* $i \in I$
2. $\Sigma\alpha = \alpha(i)$ *if* $I = \{i\}$
3. $\Sigma\alpha = \alpha(i) + \alpha(j)$ *if* $I = \{i, j\}$ *and* $i \neq j$
4. $\Sigma\alpha = \Sigma\beta$ *for every partition* $\mathcal{T}$ *of* I *and* β *given by* $\mathcal{T} \to A, T \mapsto \Sigma\alpha_{|T}$

We define the analogue of biresiduations for complete monoids.

Definition 4 (biadditive map). *Let $\mathcal{A} := (A_1, A_2, A)$ be a triple of complete monoids. Then a biadditive map w.r.t. $\mathcal{A}$ is defined as a map $\otimes : A_1 \times A_2 \to A$ such that*

$$\Sigma\beta \otimes \Sigma\gamma = \Sigma_{(i,j)\in I\times J}\beta(i) \otimes \gamma(j)$$

holds for all $\beta \in A_1^I$ and $\gamma \in A_2^J$ for arbitrary index sets I, J.

The following proposition explains how to construct biadditive maps from given ones.

Proposition 2. *Let $\mathcal{A} := (A_1, A_2, A)$ be a triple of complete monoids and let $\otimes$ be a biadditive map w.r.t. $\mathcal{A}$. For sets G and M define $\otimes_{G,M} : A_1^G \times A_2^M \to A^{G\times M}$ where*

$$(u \otimes_{G,M} w)(g, m) := u(g) \otimes w(m).$$

Then $\otimes_{G,M}$ forms a biadditive map w.r.t. $(A_1^G, A_2^M, A^{G\times M})$ which is also known as the dyadic product.

More generally, we have the following construction.

Proposition 3. *Let $\mathcal{A} := (A_1, A_2, A)$ be a triple of complete monoids and let $\otimes$ be a biadditive map w.r.t. $\mathcal{A}$. For sets G, R, and M define*

$$\otimes_{G,R,M} : A_1^{G\times R} \times A_2^{R\times M} \to A^{G\times M}$$

where

$$(\beta \otimes_{G,R,M} \eta)(g, m) := \Sigma_{r\in R}\beta(g, r) \otimes \eta(r, m).$$

Then $\otimes_{G,R,M}$ forms a biadditive map w.r.t. $(A_1^{G\times R}, A_2^{R\times M}, A^{G\times M})$ – which, as a matter of fact, is the matrix product.

If $\otimes$ is a biadditive map w.r.t. $\mathcal{A} := (A_1, A_2, A)$ and $\alpha \in A$ then a familiy $(u_r, w_r)_{r\in R} \in (A_1 \times A_2)^R$ will be called a sum-decomposition of $(\otimes, \alpha)$ if

$$\alpha = \Sigma_{r\in R} u_r \otimes w_r.$$

An important observation is the following

Proposition 4. *Let $\mathcal{A} := (A_1, A_2, A)$ be a triple of complete monoids and let $\otimes$ be a biadditive map w.r.t. $\mathcal{A}$. Then for sets G, R, M and $\alpha \in A^{G\times M}$ the following holds*

1. *If (β, η) is a decomposition of $(\otimes_{G,R,M}, \alpha)$, that is, $\beta \in A_1^{G\times R}$ and $\eta \in A_2^{R\times M}$ such that $\alpha = \beta \otimes_{G,R,M} \eta$, then $(u_r, w_r)_{r\in R}$ is a sum-decomposition of $(\otimes_{G,M}, \alpha)$ where $u_r := \beta(\cdot, r)$ and $w_r := \eta(r, \cdot)$.*
2. *Conversely, if $(u_r, w_r)_{r\in R}$ is a sum-decomposition of $(\otimes_{G,M}, \alpha)$ then (β, η) is a decomposition of $(\otimes_{G,R,M}, \alpha)$ where $\beta : G \times R \to A_1, (g, r) \mapsto u_r g$ and $\eta : R \times M \to A_2, (r, m) \mapsto w_r m$.*

Example 2. For the last proposition, a class of examples is based on linear algebra: Let $\mathcal{S} = (S, +, \otimes, 0, 1, \Sigma)$ be a complete semiring, that is, $\mathcal{S}_{add} := (S, +, 0, \Sigma)$ is a complete monoid and $\mathcal{S}_{mult} := (S, \otimes, 1)$ is a monoid such that $\otimes$ is biadditive w.r.t. $(\mathcal{S}_{add}, \mathcal{S}_{add}, \mathcal{S}_{add})$. Then, the above proposition can be applied to the situation where $A_1 = A_2 = A = \mathcal{S}_{add}$.

In the next section, we will highlight that residuals of a biresiduation play a crucial role in FCA and its abstractions.

4 Abstract Concepts and Maximal Rectangles

Let $\otimes$ be a biresiduation w.r.t. (P_1, P_2, P). We define

$$\boxed{\mathrm{f}}_{\otimes} := P_1 \times P_2$$

to be its set of *formal rectangles*. If $\alpha \in P$ then

$$\mathcal{K} := (\otimes, \alpha)$$

is called an *abstract context*. Given such an abstract context we can define the set of formal rectangles w.r.t. $\mathcal{K}$ as

$$\boxed{\mathrm{f}}_{\mathcal{K}} := \{(u, w) \in \boxed{\mathrm{f}}_{\otimes} \mid u \otimes w \leq \alpha\}.$$

On a formal rectangle, we can apply our biresiduation operation to yield an *(actual) rectangle*. We define

$$\square_{\mathcal{K}} := \{u \otimes w \mid (u, w) \in \boxed{\mathrm{f}}_{\mathcal{K}}\}$$

to be the set of *(actual) rectangles* w.r.t. $\mathcal{K}$ and

$$\boxed{\mathrm{mf}}_{\mathcal{K}} := \max \boxed{\mathrm{f}}_{\mathcal{K}}$$

to be the set of maximal rectangles regarding the product order on $P_1 \times P_2$. We define the set of abstract concepts as

$$\mathfrak{B}\mathcal{K} := \{(u, w) \in P_1 \times P_2 \mid u \to \alpha = w \;\;\&\;\; \alpha \leftarrow w = u\}.$$

We order the set of maximal rectangles of an abstract context. For $b = (u_b, w_b), c = (u_c, w_c) \in \boxed{\mathrm{mf}}_{\mathcal{K}}$ we set

$$b \leq_{\mathcal{K}} c :\iff u_b \leq_{P_1} u_c \iff w_c \leq_{P_2} w_b.$$

Now we can define an *abstract concept order* as

$$\underline{\mathfrak{B}}\mathcal{K} := (\mathfrak{B}\mathcal{K}, \leq_{\mathcal{K}}).$$

In case (P_1, P_2, P) is a triple of complete lattices, $\underline{\mathfrak{B}}\mathcal{K}$ forms a complete lattice, called the abstract concept lattice of $\mathcal{K}$.

As usual in FCA, let us abbreviate $u \to \alpha$ as u' and $\alpha \leftarrow w$ as w'. We show that even in our rather abstract setting we can talk about maximal rectangles being the abstract concepts.

Proposition 5. *Let $\mathcal{K} := (\otimes, \alpha)$ be an abstract context and define $\gamma : P_1 \to P, x \mapsto (x'', x')$ and $\mu : P_2 \to P, x \mapsto (x', x'')$. Then*

1. $\mathcal{BK} = im(\gamma) = im(\mu)$
2. $\boxed{mf}_{\mathcal{K}} = \mathcal{BK}$

We call $(p_1, p_2) \in \boxed{\text{f}}_{\mathcal{K}}$ a *decomposition of* $\mathcal{K}$ if $p_1 \otimes p_2 = \alpha$. If additionally $(p_1, p_2) \in \boxed{\text{mf}}_{\mathcal{K}}$ we call (p_1, p_2) a *conceptual decomposition of* $\mathcal{K}$.

Corollary 1. *Let $\mathcal{K} = (\otimes, \alpha)$ be an abstract context. If (p_1, p_2) is a decomposition of $\mathcal{K}$ there exists a conceptual decomposition (q_1, q_2) of $\mathcal{K}$ with $p_1 \leq q_1$ and $p_2 \leq q_2$.*

Proof. For instance, set $(q_1, q_2) := \gamma(p_1)$.

5 Macroscopics: Combining Biresiduation and Biadditivity

If L is a complete lattice, let $A(L) := (L, +, 0, \Sigma)$ be the complete monoid where $x + y = sup_L\{x, y\}$ and $0 = sup_L\emptyset$ and $\Sigma\alpha := sup_L\{\alpha(i) \mid i \in I\}$ for all $x, y \in L$ and $\alpha \in L^I$. If $\mathcal{L} := (L_1, L_2, L)$ is a triple of complete lattices then $\mathcal{A}(\mathcal{L}) := (A(L_1), A(L_2), A(L))$.

The following fact will help us to combine biresiduation and biadditivity:

Proposition 6. *Let $\mathcal{L} := (L_1, L_2, L)$ be a triple of complete lattices and let $\otimes : L_1 \times L_2 \to L$ be a map. Then $\otimes$ is biresiduated w.r.t. $\mathcal{L}$ if and only if $\otimes$ is biadditive w.r.t. $\mathcal{A}(\mathcal{L})$.*

Theorem 1. *Let $\mathcal{L} := (L_1, L_2, L)$ be a triple of complete lattices and let $\otimes$ be a biresiduation w.r.t. $\mathcal{L}$. Then for sets G, R, and M the following holds:*

1. *$\otimes_{G,M}$ is a biresiduation w.r.t. $(L_1^G, L_2^M, L^{G\times M})$.*
   Hence, for all $u \in L_1^G$ and $w \in L_2^M$ and $\alpha \in L^{G\times M}$ we have
$$u \leq \alpha \leftarrow w \iff u \otimes_{G,M} w \leq \alpha \iff w \leq u \to \alpha.$$
   Also,
$$(\alpha \leftarrow w)(g) = inf_{L_1}\{\alpha(g, m) \leftarrow w(m) \mid m \in M\}$$
   for all $g \in G$ and
$$(u \to \alpha)(m) = inf_{L_2}\{u(g) \to \alpha(g, m) \mid g \in G\}$$
   for all $m \in M$.
2. *$\otimes_{G,R,M}$ is a biresiduation w.r.t. $(L_1^{G\times R}, L_2^{R\times M}, L^{G\times M})$.*
   Hence, for all $\beta \in L_1^{G\times R}$ and $\eta \in L_2^{R\times M}$ and $\alpha \in L^{G\times M}$ we have
$$\beta \leq \alpha \leftarrow \eta \iff \beta \otimes_{G,R,M} \eta \leq \alpha \iff \eta \leq \beta \to \alpha.$$
   Also,
$$(\alpha \leftarrow \eta)(g, r) = inf_{L_1}\{\alpha(g, m) \leftarrow \eta(r, m) \mid m \in M\}$$
   for all $(g, r) \in G \times R$ and
$$(\beta \to \alpha)(r, m) = inf_{L_2}\{\beta(g, r) \to \alpha(g, m) \mid g \in G\}$$
   for all $(r, m) \in R \times M$.

3. *Let $\alpha \in L^{G\times M}$ and let (β_0, η_0) be a decomposition of $\mathcal{K} = (\otimes_{G,R,M}, \alpha)$. Then there exists a conceptual decomposition (β, η) of $\mathcal{K}$ with $\beta_0 \leq \beta$ and $\eta_0 \leq \eta$. The corresponding sum-decomposition $(\beta(\cdot, r), \eta(r, \cdot))_{r\in R}$ of $\mathcal{K}_0 := (\otimes_{G,M}, \alpha)$ consists of abstract concepts of $\mathcal{K}_0$. Such a sum-decomposition is called* conceptual sum-decomposition*.*
Conversely, if $(x_r, y_r)_{r\in R}$ is a sum-decomposition of $\mathcal{K}_0$ then there exists a conceptual sum-decomposition $(u_r, w_r)_{r\in R}$ of $\mathcal{K}_0$ with $x_r \leq u_r$ and $y_r \leq w_r$. The corresponding decomposition (β, η) of $\mathcal{K}$ defined via
$$\beta : G \times R \to L_1, (g, r) \mapsto u_r(g)$$
and
$$\eta : R \times M \to L_2, (r, m) \mapsto w_r(m)$$
is a conceptual decompsition of $\mathcal{K}$. If (β, η) is a conceptual decomposition of $\mathcal{K}$ then, by definition, $\beta = \alpha \leftarrow \eta$ and $\eta = \beta \to \alpha$.

Proof. Part 1 follows from Propositions 6 and 2. Part 2 follows from Propositions 6 and 3. Part 3 follows from Proposition 6 and 4 together with Corollary 1.

The above theorem extends Theorem 6 from [Bel11a].

Example 3. Referring to example 2, the last theorem is connected with fuzzy formal concept analysis in the following way: a join-complete semiring
$$\mathcal{S} := (S, +, \otimes, 0, 1, \Sigma)$$
is defined as a complete semiring such that $\mathcal{S}$ is idempotent $(x + x = x)$ and $\mathcal{L}(\mathcal{S}) := (S, \leq)$ with $x \leq y :\iff x + y = y$ forms a complete lattice such that $\Sigma = sup_{\mathcal{L}(\mathcal{S})}$. In particluar, $\otimes$ is a biresiduation w.r.t. $(\mathcal{L}(\mathcal{S}), \mathcal{L}(\mathcal{S}), \mathcal{L}(\mathcal{S}))$. Then, for sets G, M and $\alpha \in S^{G\times M}$, (G, M, α) is a fuzzy context over $\mathcal{S}$ having the same concept lattice as the abstract context $(\otimes_{G,M}, \alpha)$. Also the decomposition discussed in the third part of the above theorem applies to this situation.

If we restrict ourselves to the situation of M being a singleton, Theorem 1.1 yields for all $s \in S$ and $u, v \in S^G$
$$us \leq v \iff s \leq u \to v,$$
that is, $u \to v$ can be interpreted as the degree of u being a subset of v.

6 Conclusion

Starting from a given biresiduation (biadditive map) we have shown how to construct two types of biresiduations (biadditive maps): (1) the dyadic product whose residuals can be interpreted as the derivation operators of (Fuzzy) Formal Concept Analysis (talking about formal concepts as being maximal rectangles makes still sense in this abstract framework) and (2) the matrix product, the residuals of which can be interpreted as the factors as in Factor Analysis. As a "by-product", our framework yields a simple proof of the theorem on the universality of formal concepts as factors given by the proof of Theorem 1.3 which entails Theorem 1, [Bel11a].

References

[Bel11a] Belohlavek, R.: Optimal decompositions of matrices with entries from residuated lattices. Journal of Logic and Computation (2011)

[Bel11b] Belohlavek, R.: What is a Fuzzy Concept Lattice? II. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 19–26. Springer, Heidelberg (2011)

[Bly05] Blyth, T.: Lattices and Ordered Algebraic Structures. Universitext, Springer (2005)

[BV05] Belohlávek, R., Vychodil, V.: What is a fuzzy concept lattice? In: Snášel, V., Bělohlávek, R. (eds.) Proc. CLA 2005. CEUR WS, vol. 162, pp. 34–45. Palacký University in Olomouc, VŠB–Technical University of Ostrava (2005)

[BV09] Belohlavek, R., Vychodil, V.: Factor Analysis of Incidence Data via Novel Decomposition of Matrices. In: Ferré, S., Rudolph, S. (eds.) ICFCA 2009. LNCS, vol. 5548, pp. 83–97. Springer, Heidelberg (2009)

[DKV09] Droste, M., Kuich, W., Vogler, H.: Handbook of Weighted Automata. Monographs in Theoretical Computer Science. Springer (2009)

[DP90] Davey, B.A., Priestley, H.A.: Introduction to lattices and order. Cambridge University Press, Cambridge (1990)

[GW99] Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)

[Kra04] Krajci, S.: The basic theorem on generalized concept lattice. In: Snásel, V., Belohlávek, R. (eds.) CLA. CEUR Workshop Proceedings, vol. 110. CEUR-WS.org (2004)

[MOA09] Medina, J., Ojeda-Aciego, M.: On the representation theorem of multi-adjoint concept lattices. In: Carvalho, J.P., Dubois, D., Kaymak, U., da Costa Sousa, J.M. (eds.) IFSA/EUSFLAT Conf., pp. 1091–1095 (2009)

[MOARC09] Medina, J., Ojeda-Aciego, M., Ruiz-Calviño, J.: Formal concept analysis via multi-adjoint concept lattices. Fuzzy Sets and Systems 160(2), 130–144 (2009)

A Connection between Clone Theory and FCA Provided by Duality Theory

Sebastian Kerkhoff

Institut für Algebra, Technische Universität Dresden, Germany
Sebastian.Kerkhoff@tu-dresden.de
http://tu-dresden.de/Members/sebastian.kerkhoff

Abstract. The aim of this paper is to show how Formal Concept Analysis can be used for the benefit of clone theory. More precisely, we show how a recently developed duality theory for clones can be used to dualize clones over bounded lattices into the framework of Formal Concept Analysis, where they can be investigated with techniques very different from those that universal algebraists are usually armed with. We also illustrate this approach with some small examples.

Keywords: clones, duality theory, Formal Concept Analysis, clones of dual operations, coclones, bounded lattices, standard topological contexts.

1 Introduction

In this paper, we show how a duality theory from [Ker11] can be used to connect clone theory with Formal Concept Analysis [GW99].

A clone is a set of (finitary) operations over a set A that is closed under composition and contains all the projection mappings. The interest in clones is driven by the fact that clones represent the behaviour of algebras. However, as long as A contains at least three elements, very little is known about the structure of all clones on A, despite intensive research for several decades.

The principle of Duality is "a very pervasive and important concept in (modern) mathematics" [Haz95] and "an important general theme that has manifestations in almost every area of mathematics" [GBGL08]. When it comes to dualizing clones, the usual approach is to consider a clone as the set of term functions of a suitable algebra and then try to dualize this algebra, which may, or may not, be possible. Another approach, applicable for all clones, was introduced in [Ker11], where clones, inspired by an idea from [Maš06], are dualized by treating them in a more general way as sets of morphisms in a category.

In this paper, we will use the duality theory from [Ker11] (recalled in Section 3 after the preliminaries) and put it to work in Section 4, where we apply it to clones over bounded lattices (also called centralizer clones of bounded lattices), i.e., clones in which every operation is a homomorphism from a finite power of a bounded lattice to the lattice itself. Since the category of bounded lattices can be dualized to the category of standard topological contexts ([Har93], see

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 148–163, 2012.

Subsection 2.3), we can dualize the clones to certain sets of context morphisms. This allows us to investigate the clones from a different angle, namely in the setting of Formal Concept Analysis. To show that this method is in fact a helpful technique to investigate clones over lattices, we choose a few small examples in Section 5 and put the duality to work, producing some concrete results.

2 Preliminaries

In the preliminaries, we will introduce all the ingredients that we need to set up a duality for clones over bounded lattices, except that we will assume the reader to be familiar with the basic notions from Formal Concept Analysis [GW99]. We start with the necessary terminology from category theory, recall the rudimentary basics of clone theory, and end by outlining the dual equivalence for lattices from [Har93] that we are about to incorporate into our clone duality.

2.1 Category Theory

We assume that the reader is familiar with the rudimentary basics of category theory. By that, we mean that the reader should be familiar with the definitions of categories, functors, natural transformations, products and coproducts. In this section, we only introduce our notation and the terminology of duality. For an object $\mathbf{A}$ in a category $\mathcal{C}$, we denote by $\mathbf{A}^n$ the n-th power of $\mathbf{A}$ (provided it exists) and by $\pi_i^n\colon \mathbf{A}^n \to \mathbf{A}$ ($i \in \{1, \ldots, n\}$) the associated projection morphisms. For morphisms $f_1, \ldots, f_n\colon \mathbf{B} \to \mathbf{A}$, we denote by $\langle f_1, \ldots, f_n\rangle\colon \mathbf{B} \to \mathbf{A}^n$ the *tupling* of $f_1, \ldots, f_n$. Dually, for an object $\mathbf{X} \in \mathcal{C}$, we denote by $n \cdot \mathbf{X}$ the n-th copower of $\mathbf{X}$ (provided it exists) and by $\iota_i^n\colon \mathbf{X} \to n \cdot \mathbf{X}$ ($i \in \{1, \ldots, n\}$) the associated injection morphisms. For morphisms $h_1, \ldots, h_n\colon \mathbf{X} \to \mathbf{Y}$, we denote by $[h_1, \ldots, h_n]\colon n \cdot \mathbf{X} \to \mathbf{Y}$ the *cotupling of* $h_1, \ldots, h_n$.[1]

A *dual equivalence* between categories $\mathcal{A}$ and $\mathcal{X}$ is a quadruple $\langle D, E, e, \epsilon\rangle$ where $D\colon \mathcal{A} \to \mathcal{X}$ and $E\colon \mathcal{X} \to \mathcal{A}$ are contravariant functors (i.e., functors that reverse the direction of the morphisms) and $e\colon id_{\mathcal{A}} \to ED$ and $\epsilon\colon id_{\mathcal{X}} \to DE$ are natural isomorphisms. The notion "dual equivalence" is justified since D and E are full, faithful and preserve all purely category-theoretic properties, except that they reverse the direction of the morphisms. For instance, monomorphisms become epimorphisms and products become coproducts. In particular, we have $\mathbf{A}^n \in \mathcal{A}$ if and only if $n \cdot D(\mathbf{A}) \in \mathcal{X}$.

2.2 Clones

Let A be a (not necessarily finite) non-empty set. For $n \in \mathbb{N}_+$ and a set B, we say that the i-th argument of a function $f\colon A^n \to B$ is *nonessential* if

$$f(x_1, \ldots, x_n) \approx f(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n).$$

[1] We will not use the letter g for morphisms since we want to reserve this letter for objects in contexts.

If the i-th argument of f is not nonessential, then it is called *essential*. We say that f is *essentially k-ary* if it has exactly k essential arguments.

Now let $O_A := \bigcup_{n\geq 1} A^{A^n}$ be the set of all finitary, non-nullary operations over A. A subset $C \subseteq O_A$ is a *clone on A* if it contains all the projection mappings

$$\pi_i^n \colon A^n \to A : (x_1, \ldots, x_n) \mapsto x_i \ (1 \leq i \leq n)$$

(also called *trivial operations*) and is closed with respect to superposition of operations in the following sense: For an n-ary operation $f \in C$ and k-ary operations $f_1, \ldots, f_n \in C$, the k-ary operation $f(f_1, \ldots, f_n)$ defined by

$$f(f_1, \ldots, f_n)(x_1, \ldots, x_k) := f(f_1(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k))$$

is also in C. Given an algebra, the set of its non-nullary term functions is a clone. Conversely, every clone can be realized as the set of term functions of a suitable algebra. Hence, clones on a set A represent all possible different behaviours of algebras with carrier set A. Roughly speaking, if one understands all clones on a set A, one understands all algebras on A. This is the main motivation behind clone theory.

The set of all clones on a set A forms a lattice with inclusion, which we denote by $\mathcal{L}_A$. The lattice is countable and completely known for $|A| \leq 2$. However, for $|A| \geq 3$, there are continuum many clones in $\mathcal{L}_A$, and very little is known about the structure of this lattice.

2.3 Hartung's Duality for Lattices

A topological representation theorem for lattices seems to have first appeared in [Urq78]. Since then, there has been put much work into lifting this representation theorem to a dual equivalence of categories (see for example [Geh06], [HD97]). Here, we will look at the duality presented in [Har93], where the dual equivalence is set up between the category of bounded lattices with homomorphisms (i.e., functions that commute with $\vee$ and $\wedge$ and preserve the bottom and the top of the lattice) and the category of so-called standard topological contexts with so-called multivalued standard morphisms, described as in the remainder of this subsection.

A standard topological context is a standard context where the set of objects and the set of attributes are equipped with suitable topologies. To explain this more precisely, let $\mathbb{K}^\tau = ((G, \rho), (M, \sigma), \mathcal{I})$ be a triple where (G, ρ) and (M, σ) are topological spaces and $(G, M, \mathcal{I})$ is a context. A concept $(A, B) \in \mathfrak{B}(G, M, \mathcal{I})$ is said to be *closed* if A and B are closed with respect to ρ and σ, respectively. Denote the set of all closed concepts of $\mathfrak{B}(G, M, \mathcal{I})$ by $\mathfrak{B}^\tau(\mathbb{K}^\tau)$. To define a topological context, recall that, for a topological space $(X, \mathcal{T})$, a subcollection $\mathfrak{S} \subseteq \mathcal{T}$ is said to be a *subbasis of* $(X, \mathcal{T})$ if $\mathcal{T}$ is generated by $\mathfrak{S}$, i.e., if $\mathcal{T}$ is the smallest topology on X containing $\mathfrak{S}$.

Definition 1. *The structure $\mathbb{K}^\tau$ is called a topological context if*

(i) $A \in \rho \Rightarrow A'' \in \rho$ *and* $B \in \sigma \Rightarrow B'' \in \sigma$,

(ii) $\mathfrak{S}_\rho := \{A \subseteq G \mid (A, A') \in \mathfrak{B}^\tau(\mathbb{K}^\tau)\}$ *is a subbasis of* (G, ρ) *and* $\mathfrak{S}_\sigma := \{B \subseteq M \mid (B', B) \in \mathfrak{B}^\tau(\mathbb{K}^\tau)\}$ *is a subbasis of* (M, σ)*.*

A topological context is called a standard topological context if, in addition, the following three conditions hold:

(a) $(G, M, \mathcal{I})$ *is a standard context,*
(b) for every $(g, m) \in \mathcal{I}$*, there exists some* $(A, B) \in \mathfrak{B}^\tau(\mathbb{K}^\tau)$ *such that* $g \in A$ *and* $m \in B$*,*
(c) $(\mathcal{I}^c, (\rho \times \sigma)|_{\mathcal{I}^c})$ *is a compact space*[2] *where* $\mathcal{I}^c := (G \times M) \setminus \mathcal{I}$ *and* $\rho \times \sigma$ *denotes the product topology on* $G \times M$*.*

We will now explain that there is indeed a one-to-one correspondence between bounded lattices and standard topological spaces.

First, let $\mathbb{K}^\tau = ((G, \rho), (M, \sigma), \mathcal{I})$ be a standard topological context, and set $\underline{\mathfrak{B}}^\tau(\mathbb{K}^\tau) := \langle \mathfrak{B}^\tau(\mathbb{K}^\tau), \leq \rangle$ where $\leq$ is the restriction of the usual order-relation on $\mathfrak{B}(G, M, \mathcal{I})$. Then, $\underline{\mathfrak{B}}^\tau(\mathbb{K}^\tau)$ is a bounded lattice. In fact, it is a bounded (but not necessarily complete) sublattice of $\underline{\mathfrak{B}}(G, M, \mathcal{I})$.

For the other direction, we need to introduce the notion of I-maximal filters and F-maximal ideals: For a bounded lattice $\mathbf{A}$, denote by $\mathfrak{F}(\mathbf{A})$ and $\mathfrak{I}(\mathbf{A})$ the set of non-empty (but not necessarily proper) lattice filters and lattice ideals of $\mathbf{A}$, respectively. For $F \in \mathfrak{F}(\mathbf{A})$ and $I \in \mathfrak{I}(\mathbf{A})$, we say that F is I-maximal whenever $F \cap I = \emptyset$ and every proper superfilter $F^* \supsetneq F$ already intersects I. Similarly, we say that I is F-maximal if $F \cap I = \emptyset$ and every proper superideal $I^* \supsetneq I$ already intersects F. Now, set

$$\mathfrak{F}_0(\mathbf{A}) := \{F \in \mathfrak{F}(\mathbf{A}) \mid \exists I \in \mathfrak{I}(\mathbf{A}) : F \text{ is } I\text{-maximal}\},$$
$$\mathfrak{I}_0(\mathbf{A}) := \{I \in \mathfrak{I}(\mathbf{A}) \mid \exists F \in \mathfrak{F}(\mathbf{A}) : I \text{ is } F\text{-maximal}\},$$
$$\mathfrak{R}(\mathbf{A}) := \{(F, I) \in \mathfrak{F}_0(\mathbf{A}) \times \mathfrak{I}_0(\mathbf{A}) \mid F \cap I \neq \emptyset\}.$$

With this notation, we can now define a standard topological context $\mathbb{K}^\tau(\mathbf{A})$ such that $\underline{\mathfrak{B}}^\tau(\mathbb{K}^\tau(\mathbf{A})) \cong \mathbf{A}$. This standard topological context can be defined as follows:

$$\mathbb{K}^\tau(\mathbf{A}) := ((\mathfrak{F}_0(\mathbf{A}), \rho_0), (\mathfrak{I}_0(\mathbf{A}), \sigma_0), \mathfrak{R}(\mathbf{A})),$$

where ρ_0 and σ_0 are given by the subbases

$$\mathfrak{S}_{\rho_0} := \{\{F \in \mathfrak{F}_0(\mathbf{A}) \mid a \in F\} \mid a \in A\},$$
$$\mathfrak{S}_{\sigma_0} := \{\{I \in \mathfrak{I}_0(\mathbf{A}) \mid a \in I\} \mid a \in A\},$$

respectively.

Since we will use this fact in the remainder of this paper, let us note the following (obvious) proposition:

Proposition 2. *For* $X \subseteq \mathfrak{F}_0(\mathbf{A})$*, we have* $g \in X''$ *if and only if* g *is a superfilter of some* $x \in X$*. Similarly, for* $X \subseteq \mathfrak{I}_0(\mathbf{A})$*, we have* $m \in X''$ *if and only if* m *is a superideal of some* $x \in X$*.*

[2] By a compact space, we mean what is sometimes also called a quasicompact space. That is, a topological space in which all open covers have finite subcovers.

Let us now turn our attention to the morphism part of the duality. Therefor, we need to define multivalued standard morphisms and their composition.

A *multivalued function* $F\colon X \to Y$ from a set X to a set Y is a binary relation $F \subseteq X \times Y$ such that $\pi_1(F) = X$. For $x \in X$, $A \subseteq X$ and $B \subseteq Y$, we define

$$\begin{aligned} F(x) &:= \{y \in Y \mid (x,y) \in F\}, \\ F[A] &:= \{y \in Y \mid \exists a \in A : (a,y) \in F\}, \\ F^{[-1]}[B] &:= \{x \in X \mid F(x) \subseteq B\}. \end{aligned}$$

Definition 3. *Let* $\mathbb{K}_1^\tau = ((G_1,\rho_1),(M_1,\sigma_1),\mathcal{I}_1)$, $\mathbb{K}_2^\tau = ((G_2,\rho_2),(M_2,\sigma_2),\mathcal{I}_2)$ *be standard topological contexts. A multivalued standard morphism* $h\colon \mathbb{K}_1^\tau \to \mathbb{K}_2^\tau$ *is a pair* (R_h, S_h) *of multivalued functions* $R_h\colon G_1 \to G_2$ *and* $S_h\colon M_1 \to M_2$ *such that*

(i) $(R_h^{[-1]}[A], S_h^{[-1]}[B]) \in \mathfrak{B}^\tau(\mathbb{K}_1^\tau)$ *for every* $(A,B) \in \mathfrak{B}^\tau(\mathbb{K}_2^\tau)$,
(ii) $R_h(x) = R_h(x)'' = \overline{R_h(x)}$ *for every* $x \in G_1$ *and* $S_h(x) = S_h(x)'' = \overline{S_h(x)}$ *for every* $x \in M_1$.

For $j \in \{1,2,3\}$, let $\mathbb{K}_j^\tau = ((G_j,\rho_j),(M_j,\sigma_j),\mathcal{I}_j)$, be standard topological contexts. We define the composition $h_2 \circ h_1$ of two multivalued standard morphisms $h_1\colon \mathbb{K}_1^\tau \to \mathbb{K}_2^\tau$ and $h_2\colon \mathbb{K}_2^\tau \to \mathbb{K}_3^\tau$ by setting:

$$\begin{aligned} R_{h_2\circ h_1}\colon G_1 \to G_3 &: R_{h_2\circ h_1}(x) := R_{h_2}[R_{h_1}(x)]'', \\ S_{h_2\circ h_1}\colon M_1 \to M_3 &: S_{h_2\circ h_1}(x) := S_{h_2}[S_{h_1}(x)]''. \end{aligned}$$

For two bounded lattices $\mathbf{A}$, $\mathbf{B}$ and a homomorphism $f\colon \mathbf{A} \to \mathbf{B}$, we define the multivalued standard morphism $(R^f, S^f)\colon \mathbb{K}^\tau(\mathbf{B}) \to \mathbb{K}^\tau(\mathbf{A})$ by setting:

$$\begin{aligned} R^f\colon \mathfrak{F}_0(\mathbf{B}) \to \mathfrak{F}_0(\mathbf{A}) &: R^f(x) := \{y \in \mathfrak{F}_0(\mathbf{A}) \mid f^{-1}[x] \subseteq y\}, \\ S^f\colon \mathfrak{I}_0(\mathbf{B}) \to \mathfrak{I}_0(\mathbf{A}) &: S^f(x) := \{y \in \mathfrak{I}_0(\mathbf{A}) \mid f^{-1}[x] \subseteq y\}. \end{aligned}$$

It is important to note that, for f being surjective, the preimage of each $F \in \mathfrak{F}_0(\mathbf{B})$ and each $I \in \mathfrak{I}_0(\mathbf{B})$ is an element of $\mathfrak{F}_0(\mathbf{A})$ and $\mathfrak{I}_0(\mathbf{A})$, respectively. For arbitrary homomorphisms, this is not necessarily true.

Now let $\mathcal{X}$ be the category with standard topological contexts as objects, multivalued standard morphisms as morphisms and $\circ$ as composition. Note that, for a given standard topological context $\mathbf{X} = ((G,\rho),(M,\sigma),\mathcal{I}) \in \mathcal{X}$, the identity morphism $id_{\mathbf{X}}$ is given as follows:

$$\begin{aligned} R_{id_{\mathbf{X}}}\colon G \to G &: R_{id_{\mathbf{X}}}(x) = x'', \\ S_{id_{\mathbf{X}}}\colon M \to M &: S_{id_{\mathbf{X}}}(x) = x''. \end{aligned}$$

Theorem 4 ([Har93]). *Let* $\mathcal{A}$ *be the category of bounded lattices with homomorphisms as morphisms and the usual composition of functions. Then,*

$\mathcal{A}$ and $\mathcal{X}$ are dually equivalent via the two contravariant functors $D\colon \mathcal{A} \to \mathcal{X}$ and $E\colon \mathcal{X} \to \mathcal{A}$ that are given as follows:

$$\begin{aligned} D(\mathbf{A}) &:= \mathbb{K}^\tau(\mathbf{A}) = ((\mathfrak{F}_0(\mathbf{A}), \rho_0), (\mathfrak{I}_0(\mathbf{A}), \sigma_0), \mathfrak{R}(\mathbf{A})), \\ D(f) &:= (R^f, S^f), \\ E(\mathbb{K}^\tau) &:= \underline{\mathfrak{B}}^\tau(\mathbb{K}^\tau), \\ E(h) &:= (R_h^{[-1]}[-_1], S_h^{[-1]}[-_2]) : (A, B) \mapsto (R_h^{[-1]}[A], S_h^{[-1]}[B]). \end{aligned}$$

3 Duality Theory for Clones

In this section, we will explain how we can dualize arbitrary clones. This theory will be the foundation of our work in Section 4, where we will use the machinery to dualize clones over bounded lattice into the framework of Formal Concept Analysis. To obtain this duality theory for clones, we will use a more general notion of a clone:

Definition 5. *Let $n \in \mathbb{N}_+$. A morphism $f\colon \mathbf{A}^n \to \mathbf{A}$ is called an n-ary operation over $\mathbf{A}$. Denote by $O_\mathbf{A}^{(n)}$ the set of all n-ary operations over $\mathbf{A}$, define $O_\mathbf{A} := \bigcup_{n\in\mathbb{N}_+} O_\mathbf{A}^{(n)}$ and, for $F \subseteq O_\mathbf{A}$, set $F^{(n)} := F \cap O_\mathbf{A}^{(n)}$.*

Definition 6. *A subset $C \subseteq O_\mathbf{A}$ is called a clone of operations, written $C \le O_\mathbf{A}$, if C contains all the projection morphisms $\pi_i^n\colon \mathbf{A}^n \to \mathbf{A}$ and, for $f \in C^{(n)}$ and $f_1, \ldots, f_n \in C^{(k)}$, the superposition $f \circ \langle f_1, \ldots, f_n\rangle$ is also in C.*

If $\mathcal{A}$ is the category of sets, then this definition coincides with the usual notion of a clone. It is easy to verify that the clones over an object $\mathbf{A}$ form a complete lattice with respect to inclusion. We call this lattice the *lattice of clones over* $\mathbf{A}$, and we denote it by $\mathcal{L}_\mathbf{A}$. The top element of $\mathcal{L}_\mathbf{A}$ is the *full clone* $O_\mathbf{A}$, and the bottom element is the clone that contains only the projection morphisms.

Since clones are closed under arbitrary intersection, we can define the closure operator Clo that assigns to each subset $F \subseteq O_\mathbf{A}$ the least clone of operations over $\mathbf{A}$ that contains F. It is called the clone *generated by* F. For a single operation f, we write $\mathrm{Clo}(f)$ to mean $\mathrm{Clo}(\{f\})$.

Examples 7.

(i) If $\mathcal{A} = \mathcal{S}et$, then $O_\mathbf{A}$ is the full clone on the set A and $\mathcal{L}_\mathbf{A}$ is the usual clone lattice.
(ii) If $\mathcal{A}$ is a variety (or a quasivariety) of algebras, then $O_\mathbf{A}$ is the centralizer clone of the algebra $\mathbf{A}$ and $\mathcal{L}_\mathbf{A}$ is the lattice of subclones of $O_\mathbf{A}$. Centralizer clones are of particular interest in universal algebra (see [MMT87], for instance).
(iii) For each clone C on a finite set A, we obtain $C = O_\mathbf{A}$ if we define $\mathbf{A}$ to be a relational structure $\langle A, R\rangle$ in a variety of relational structures such that C is the set of polymorphisms of R (that is, the set of operations that preserve each $\sigma \in R$). Such a set of relations R can always be found. In this case, $\mathcal{L}_\mathbf{A}$ is the lattice of subclones of C.

These examples show that one can investigate clones over sets by treating them as clones over objects in (abstract or concrete) categories different from $\mathcal{S}et$.

We can lift every notion from clone theory to our setting as long as we can write it in purely category-theoretic terms. For instance, we can write all kinds of identities. E.g., we can define essential arguments of an operation as follows:

Definition 8. *For $n \in \mathbb{N}_+$ and $i \in \{1, \dots, n\}$, the i-th argument of an operation $f \in O_{\mathbf{A}}^{(n)}$ is said to be nonessential if*

$$f \circ \langle \pi_1^{n+1}, \dots, \pi_n^{n+1} \rangle = f \circ \langle \pi_1^{n+1}, \dots, \pi_{i-1}^{n+1}, \pi_{n+1}^{n+1}, \pi_{i+1}^{n+1}, \dots, \pi_n^{n+1} \rangle.$$

An argument is called essential if it is not nonessential. Moreover, we say that an operation is essentially k-ary if it has exactly k essential arguments.

This definition coincides with the usual definition of (non-)essential arguments as presented at the beginning of Subsection 2.2 whenever the latter is applicable (that is, if the powers of $\mathbf{A}$ are Cartesian powers and the morphisms are set-functions). Having written operations and clones in purely category-theoretic terms, we can dualize all these notions:

Definition 9. *Let $n \in \mathbb{N}_+$. An n-ary dual operation over $\mathbf{X}$ (or cooperation over $\mathbf{X}$) is a morphism from $\mathbf{X}$ to $n \cdot \mathbf{X}$. Denote by $\overline{O}_{\mathbf{X}}^{(n)}$ the set of all n-ary dual operations over $\mathbf{X}$, define $\overline{O}_{\mathbf{X}} := \bigcup_{n \in \mathbb{N}_+} \overline{O}_{\mathbf{X}}^{(n)}$ and, for a set of dual operations $H \subseteq \overline{O}_{\mathbf{X}}$, set $H^{(n)} := H \cap \overline{O}_{\mathbf{X}}^{(n)}$.*

Definition 10. *A subset $C \subseteq \overline{O}_{\mathbf{X}}$ is called a clone of dual operations (or coclone), written $C \leq \overline{O}_{\mathbf{X}}$, if it contains all the injection morphisms and, for $h \in C^{(n)}$ and $h_1, \dots, h_n \in C^{(k)}$, the superposition $[h_1, \dots, h_n] \circ h$ is also in C.*

If $\mathbf{X}$ is a set in the category of sets, then a clone of dual operations over $\mathbf{X}$ is a coclone as introduced in [Csá85].

Definition 11. *For $n \in \mathbb{N}_+$ and $i \in \{1, \dots, n\}$, the i-th argument of a dual operation $h \in \overline{O}_{\mathbf{X}}^{(n)}$ is said to be nonessential if*

$$[\iota_1^{n+1}, \dots, \iota_n^{n+1}] \circ h = [\iota_1^{n+1}, \dots, \iota_{i-1}^{n+1}, \iota_{n+1}^{n+1}, \iota_{i+1}^{n+1}, \dots, \iota_n^{n+1}] \circ h.$$

An argument is called essential if it is not nonessential. Moreover, we say that an operation is essentially k-ary if it has exactly k essential arguments.

Again, clones of dual operations form a complete lattice, which we will denote by $\overline{\mathcal{L}}_{\mathbf{X}}$ and call the *lattice of clones of dual operations over* $\mathbf{X}$.

Analogue to the closure operator Clo on sets of operations, we can define $\overline{\text{Clo}}$: For a set of dual operations $H \subseteq \overline{O}_{\mathbf{X}}$, we denote by $\overline{\text{Clo}}(H)$ the least clone of dual operations that contains H. Again, for a single dual operation, we write $\overline{\text{Clo}}(h)$ instead of $\overline{\text{Clo}}(\{h\})$.

We will now describe how to dualize clones. For this, let $\langle D, E, e, \epsilon\rangle$ be a dual equivalence between two arbitrary categories $\mathcal{A}$ and $\mathcal{X}$, and let $\mathbf{A} \in \mathcal{A}$ such that all finite non-empty powers of $\mathbf{A}$ are also in $\mathcal{A}$. Set $\mathbf{X} := D(\mathbf{A})$. Since $\mathcal{A}$ and $\mathcal{X}$ are dually equivalent, $\mathcal{X}$ contains all finite non-empty copowers of $\mathbf{X}$. The functor D carries $\mathbf{A}$ to $\mathbf{X}$ and reverses the order of the morphisms, so wishful thinking suggests that it should map a morphism $f \in O_\mathbf{A}$ to a morphism in $\overline{O}_\mathbf{X}$. Unfortunately, this is not always the case as D maps f to a morphism from $\mathbf{X}$ to $D(\mathbf{A}^n)$ and the latter is only isomorphic and not necessarily equal to $n \cdot \mathbf{X}$.[3] However, we can get around this technical problem by finding a family of isomorphisms $(\eta_n)_{n\in\mathbb{N}_+}$ such that $f \mapsto \eta_{ar(f)} \circ D(f)$ becomes a clone isomorphism from $O_\mathbf{A}$ to $\overline{O}_\mathbf{X}$ (recall that $ar(f)$ denotes the arity of f).

Lemma 12 ([Ker11]). *There exists a unique family of isomorphisms*

$$(\eta_n \colon D(\mathbf{A}^n) \to n \cdot \mathbf{X})_{n\in\mathbb{N}_+}$$

such that the mapping

$$(-)^\partial \colon O_\mathbf{A} \to \overline{O}_\mathbf{X} : f \mapsto \eta_{\mathrm{ar}(f)} \circ D(f)$$

has the following properties:

(i) $(-)^\partial \colon O_\mathbf{A}^{(n)} \to \overline{O}_\mathbf{X}^{(n)}$ *is a bijection for each* $n \in \mathbb{N}_+$,
(ii) $(\pi_i^n)^\partial = \iota_i^n$ *and* $(f \circ \langle h_1, \ldots, h_n\rangle)^\partial = [h_1^\partial, \ldots, h_n^\partial] \circ f^\partial$ *for all* $n, k \in \mathbb{N}_+$, $f \in O_\mathbf{A}^{(n)}$ *and* $h_1, \ldots, h_n \in O_\mathbf{A}^{(k)}$.

In fact, $\eta_n = [D(\pi_1^n), \ldots, D(\pi_n^n)]^{-1}$.

By this lemma, it follows immediately that C is a clone of operations over $\mathbf{A}$ if and only if C^∂ is a clone of dual operations over $\mathbf{X}$. Moreover, the family $(\eta_n)_{n\in\mathbb{N}_+}$ and hence the construction of $(-)^\partial$ only depends on the choice of the dual equivalence. Thus, the following definition is justified:

Definition 13. *The mapping* $(-)^\partial \colon O_\mathbf{A} \to \overline{O}_\mathbf{X}$ *is called the clone duality with respect to* D*. For* $F \subseteq O_\mathbf{A}$*, set* $F^\partial := \{f^\partial \mid f \in F\}$.

By Lemma 12, we immediately obtain the following theorem:

Theorem 14. $\mathcal{L}_\mathbf{A} \cong \overline{\mathcal{L}}_\mathbf{X}$*, where an isomorphism between* $\mathcal{L}_\mathbf{A}$ *and* $\overline{\mathcal{L}}_\mathbf{X}$ *is given by* $C \mapsto C^\partial$.

Moreover, it is an obvious consequence of Lemma 12 that an identity holds in C if and only if its dualized version holds in C^∂:

Lemma 15. *Let* $f_1 \in O_\mathbf{A}^{(k)}$, $f_2 \in O_\mathbf{A}^{(l)}$*. For* $i_1, \ldots, i_k, j_1, \ldots, j_l \in \{1, \ldots, n\}$*, we have*

$$f_1 \circ \langle \pi_{i_1}^n, \ldots, \pi_{i_k}^n\rangle = f_2 \circ \langle \pi_{j_1}^n, \ldots, \pi_{j_l}^n\rangle \iff [\iota_{i_1}^n, \ldots, \iota_{i_k}^n] \circ f_1^\partial = [\iota_{j_1}^n, \ldots, \iota_{j_l}^n] \circ f_2^\partial.$$

[3] Of course, we could avoid the trouble by defining $n \cdot \mathbf{X} := D(\mathbf{A}^n)$ for all $n \in \mathbb{N}_+$. But then, the copowers of $\mathbf{X}$ might not be canonical and they would depend on the choice of the dual equivalence. One usually wants to avoid both.

In particular, this lemma evidently implies the statement that the i-th argument of some $f \in O_{\mathbf{A}}$ is nonessential if and only if the i-th argument of $f^{\partial} \in \overline{O}_{\mathbf{X}}$ is nonessential.

In [Ker11], clone dualities are used to obtain new results for clones over finite sets and in particular for clones over classical algebraic structures such as Boolean algebras, distributive lattices, median algebras or Boolean groups. In the next section, we will present a new example and discuss how clones over (not necessarily finite) bounded lattices dualize to clones of dual operations in an FCA-framework.

4 Clones over Bounded Lattices

From now on until the end of this paper, let $\mathbf{A} = \langle A, \vee, \wedge, 0, 1\rangle$ be a bounded lattice, and let $\mathcal{A}$ be the category of bounded lattices with all homomorphisms as morphisms. Recall that, in this scenario, $O_{\mathbf{A}}$ is the centralizer clone of the lattice $\mathbf{A}$ (cf. Example 7(ii)). Our goal is to investigate $\mathcal{L}_{\mathbf{A}}$, that is, the lattice of subclones of $O_{\mathbf{A}}$.

We will now construct a clone duality for $O_{\mathbf{A}}$. By Theorem 4, $\mathcal{A}$ is dually equivalent to the category $\mathcal{X}$ of standard topological contexts with multivalued standard morphisms. Recall that the corresponding functor $D\colon \mathcal{A} \to \mathcal{X}$ is given as follows:

$$\begin{aligned} D(\mathbf{A}) &:= \mathbb{K}^{\tau}(\mathbf{A}) = ((\mathfrak{F}_0(\mathbf{A}), \rho_0), (\mathfrak{I}_0(\mathbf{A}), \sigma_0), \mathfrak{R}(\mathbf{A})), \\ D(f) &:= (R^f, S^f). \end{aligned}$$

From now on, let $\mathbf{X} := D(\mathbf{A})$. To obtain the clone duality $(-)^{\partial}\colon O_{\mathbf{A}} \to \overline{O}_{\mathbf{X}}$, we need to observe how the powers of $\mathbf{A}$ dualize under D.

Lemma 16. *For $n \in \mathbb{N}_+$, we have*

$$\begin{aligned} \mathfrak{F}_0(\mathbf{A}^n) &= \left\{A^{i-1} \times x \times A^{n-i} \mid i \in \{1,\ldots,n\}, x \in \mathfrak{F}_0(\mathbf{A})\right\}, \\ \mathfrak{I}_0(\mathbf{A}^n) &= \left\{A^{i-1} \times x \times A^{n-i} \mid i \in \{1,\ldots,n\}, x \in \mathfrak{I}_0(\mathbf{A})\right\}. \end{aligned}$$

Proof. We only show the first equality, since the part for $\mathfrak{I}_0(\mathbf{A}^n)$ is similar.

"$\subseteq$". We will show this direction in three steps. First we show that each $F \in \mathfrak{F}_0(\mathbf{A}^n)$ must be the Cartesian product of n filters $F_1, \ldots, F_n \in \mathfrak{F}(\mathbf{A})$, then we show that exactly $n-1$ of the sets $F_1, \ldots, F_n$ equal A, and finally we show that $F_i \neq A$ implies $F_i \in \mathfrak{F}_0(\mathbf{A})$. For the first part, let $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in F$. Since F is a filter, it is closed under $\wedge$. Thus, for $c_i \in \{a_i, b_i\}$, we have

$$(c_1, \ldots, c_n) \geq (a_1 \wedge b_1, \ldots, a_n \wedge b_n) = (a_1, \ldots, a_n) \wedge (b_1, \ldots, b_n) \in F,$$

and consequently $(c_1, \ldots, c_n) \in F$ since F is also an increasing set. This proves that F can be written as $F_1 \times \ldots \times F_n$ for some $F_1, \ldots, F_n \subseteq A$. If F_i is not an increasing set for some $i \in \{1, \ldots, n\}$, then F is not an increasing set. If F_i is not closed under $\wedge$ for some $i \in \{1, \ldots, n\}$, then F is not closed under $\wedge$. Hence, $F_1, \ldots, F_n \in \mathfrak{F}(\mathbf{A})$.

For the second part, let us first note that we cannot have $F = A^n$, so $F_i \neq A$ holds for at least one $i \in \{1, \ldots, n\}$. Now, let us assume the existence of two integers $i, j \in \{1, \ldots, n\}$, $i \neq j$, such that $F_i \neq A$ and $F_j \neq A$. Without loss of generality we can assume $i = 1$ and $j = 2$. Since $F_1, \ldots, F_n$ are filters, we also have that $F^* := F_1 \times A \times F_3 \times \ldots \times F_n$ and $F^{**} := A \times F_2 \times \ldots \times F_n$ are filters. Moreover, they both properly contain F. Since $F \in \mathfrak{F}_0(\mathbf{A}^n)$, there must exist an ideal $I \in \mathfrak{I}(\mathbf{A}^n)$ that is disjoint to F but intersects F^* as well as F^{**}. Let $x_1 \in F^* \cap I$ and $x_2 \in F^{**} \cap I$. But now, we have $x_1 \vee x_2 \in I$ since I is an ideal, and we have $x_1 \vee x_2 \in F$ by construction of F^* and F^{**}. Thus, $x_1 \vee x_2 \in I \cap F$, which is impossible.

For the third part, let us assume $F_i \neq A$ for some $i \in \{1, \ldots, n\}$. Since we already know that F_i is a filter, we can finish the proof by showing that there exists $I_i \in \mathfrak{I}(\mathbf{A})$ such that F_i is I_i-maximal. Recall that F is I-maximal for some $I \in \mathfrak{I}(\mathbf{A}^n)$. By arguments analogue to above, I can be written as $I_1 \times \ldots \times I_n$ where $I_1, \ldots, I_n \in \mathfrak{I}(\mathbf{A})$. In particular, I_i is an ideal. Let us show that F_i is I_i-maximal. By $F_1 = \ldots = F_{i-1} = F_{i+1} = \ldots = F_n = A$, we can conclude $F_i \cap I_i = \emptyset$ since otherwise it would follow $F \cap I \neq \emptyset$, a contradiction to the I-maximality of F. It remains to show that there cannot exist a proper superfilter $F_i^* \supsetneq F_i$ that is disjoint to I_i: The existence of such $F_i^* \in \mathfrak{F}(\mathbf{A})$ would imply that I does not intersect the filter $A^{i-1} \times F_i^* \times A^{n-i} \supsetneq F$, which would again contradict the I-maximality of F.

"$\supseteq$". Let $i \in \{1, \ldots, n\}$ and $x \in \mathfrak{F}_0(\mathbf{A})$. Clearly, $F := A^{i-1} \times x \times A^{n-i}$ is a filter. Since $x \in \mathfrak{F}_0(\mathbf{A})$, there exists $I \in \mathfrak{I}(\mathbf{A})$ such that x is I-maximal. But now, $A^{i-1} \times I \times A^{n-i}$ is an ideal, and we will finish the proof by showing that F is $(A^{i-1} \times I \times A^{n-i})$-maximal. Clearly, $A^{i-1} \times I \times A^{n-i}$ is disjoint to F. Let $F^* \supsetneq F$ be a proper superfilter. By arguments from above, F^* is of the form $A^{i-1} \times y \times A^{n-i}$ for some filter $y \supsetneq x$. But now, y intersects I, and so F^* intersects $A^{i-1} \times I \times A^{n-i}$. Thus, $F \in \mathfrak{F}_0(\mathbf{A}^n)$. $\square$

It remains to understand how the projection morphisms dualize. For $n \in \mathbb{N}_+$ and $i \in \{1, \ldots, n\}$, we have $(\pi_i^n)^{-1}[x] = A^{i-1} \times x \times A^{n-i}$ for each $x \in \mathfrak{F}_0(\mathbf{A})$ and each $x \in \mathfrak{I}_0(\mathbf{A})$. Thus, the multivalued standard morphism

$$D(\pi_i^n) = (R^{\pi_i^n}, S^{\pi_i^n})\colon D(\mathbf{A}) \to D(\mathbf{A}^n)$$

is given as follows:

$$R^{\pi_i^n}(x) = \{y \in \mathfrak{F}_0(\mathbf{A}^n) \mid A^{i-1} \times x \times A^{n-i} \subseteq y\},$$
$$S^{\pi_i^n}(x) = \{y \in \mathfrak{I}_0(\mathbf{A}^n) \mid A^{i-1} \times x \times A^{n-i} \subseteq y\}.$$

We will now look at the following canonical definition of copowers in $\mathcal{X}$: Let $\mathbf{Y} := ((G_\mathbf{Y}, \rho_\mathbf{Y}), (M_\mathbf{Y}, \sigma_\mathbf{Y}), \mathcal{I}_\mathbf{Y})$ be an object in $\mathcal{X}$. Then, the n-th copower of $\mathbf{Y}$ is defined by setting

$$n \cdot \mathbf{Y} := ((n \cdot G_\mathbf{Y}, \rho_{n\cdot\mathbf{Y}}), (n \cdot M_\mathbf{Y}, \sigma_{n\cdot\mathbf{Y}}), \mathcal{I}_{n\cdot\mathbf{Y}}),$$

where

$$n \cdot G_\mathbf{Y} := \{\langle i, g\rangle \mid i \in \{1, \dots, n\}, g \in G_\mathbf{Y}\},$$
$$n \cdot M_\mathbf{Y} := \{\langle i, m\rangle \mid i \in \{1, \dots, n\}, m \in M_\mathbf{Y}\},$$

$\rho_{n\cdot\mathbf{Y}}$ and $\sigma_{n\cdot\mathbf{Y}}$ are the disjoint union topologies (that is, the finest topologies for which all canonical injections $g \mapsto \langle i, g\rangle$ and $m \mapsto \langle i, m\rangle$ are continuous) and

$$\langle i, g\rangle \mathcal{I}_{n\cdot\mathbf{Y}} \langle j, m\rangle :\Longleftrightarrow i \neq j \text{ or } g\mathcal{I}_\mathbf{Y} m.$$

The associated injection morphisms ι_i^n are given by

$$R_{\iota_i^n}\colon G_\mathbf{Y} \to n \cdot G_\mathbf{Y} \ : R_{\iota_i^n}(y) = \{\langle i, g\rangle \mid g \in y''\} = \langle i, y\rangle'',$$
$$S_{\iota_i^n}\colon M_\mathbf{Y} \to n \cdot M_\mathbf{Y} \ : S_{\iota_i^n}(y) = \{\langle i, m\rangle \mid m \in y''\} = \langle i, y\rangle''.$$

Moreover, for a standard topological context $\mathbf{Z} = ((G_\mathbf{Z}, \rho_\mathbf{Z}), (M_\mathbf{Z}, \sigma_\mathbf{Z}), \mathcal{I}_\mathbf{Z})$ and $h_1, \dots, h_n\colon \mathbf{Y} \to \mathbf{Z}$, the cotupling $[h_1, \dots, h_n]\colon n \cdot \mathbf{Y} \to \mathbf{Z}$ is given as follows:

$$R_{[h_1,\dots,h_n]}\colon n \cdot G_\mathbf{Y} \to G_\mathbf{Z} \ : R_{[h_1,\dots,h_n]}(\langle i, y\rangle) = R_{h_i}(y),$$
$$S_{[h_1,\dots,h_n]}\colon n \cdot M_\mathbf{Y} \to M_\mathbf{Z} \ : S_{[h_1,\dots,h_n]}(\langle i, y\rangle) = S_{h_i}(y).$$

Recall that $\mathbf{X} = D(\mathbf{A}) = ((\mathfrak{F}_0(\mathbf{A}), \rho_0), (\mathfrak{I}_0(\mathbf{A}), \sigma_0), \mathfrak{R}(\mathbf{A}))$. For the copowers of $\mathbf{X}$, we can give a more concrete characterization of the injection morphisms and their cotuplings:

Lemma 17. *Let $n \in \mathbb{N}_+$ and $i \in \{1, \dots, n\}$. Then,*

$$R_{\iota_i^n}(x) = \{\langle i, y\rangle \mid y \in \mathfrak{F}_0(\mathbf{A}), x \subseteq y\},$$
$$S_{\iota_i^n}(x) = \{\langle i, y\rangle \mid y \in \mathfrak{I}_0(\mathbf{A}), x \subseteq y\}.$$

Consequently, for $i_1, \dots, i_k \in \{1, \dots, n\}$, we obtain

$$R_{[\iota_{i_1}^n,\dots,\iota_{i_k}^n]}(\langle j, x\rangle) = \{\langle i_j, y\rangle \mid y \in \mathfrak{F}_0(\mathbf{A}), x \subseteq y\},$$
$$S_{[\iota_{i_1}^n,\dots,\iota_{i_k}^n]}(\langle j, x\rangle) = \{\langle i_j, y\rangle \mid y \in \mathfrak{I}_0(\mathbf{A}), x \subseteq y\}.$$

Proof. As described above, we have $R_{\iota_{i_j}^n}(x) = \{\langle i_j, y\rangle \mid y \in x''\}$, and it is a direct consequence of Proposition 2 that we also have

$$\{\langle i_j, y\rangle \mid y \in x''\} = \{\langle i_j, y\rangle \mid y \in \mathfrak{F}_0(\mathbf{A}), x \subseteq y\}.$$

The part for $S_{\iota_i^n}$ follows in the same way. □

Let us now turn back to constructing our duality. By Lemma 12, there exists a unique family of isomorphism $(\eta_n\colon D(\mathbf{A}^n) \to n\cdot\mathbf{X})_{n\in\mathbb{N}_+}$ with $\iota_i^n = \eta_n \circ D(\pi_i^n)$ for all $n \in \mathbb{N}_+$ and $i \in \{1,\dots,n\}$. Moreover, the lemma states that this family is obtained by setting $\eta_n := [D(\pi_1^n),\dots,D(\pi_n^n)]^{-1}$ for all $n \in \mathbb{N}_+$. In the following proposition, we will describe this family more concretely:

Proposition 18. *For $n \in \mathbb{N}_+$, the unique isomorphism $\eta_n\colon D(\mathbf{A}^n) \to n\cdot\mathbf{X}$ from Proposition 12 is given as follows:*

$$\text{For } x \in \mathfrak{F}_0(\mathbf{A}^n)\text{: } R_{\eta_n}(x) = \{\langle i,y\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A}) \mid x \subseteq A^{i-1}\times y\times A^{n-i}\},$$
$$\text{for } x \in \mathfrak{I}_0(\mathbf{A}^n)\text{: } S_{\eta_n}(x) = \{\langle i,y\rangle \in n\cdot\mathfrak{I}_0(\mathbf{A}) \mid x \subseteq A^{i-1}\times y\times A^{n-i}\}.$$

Proof. In view of Lemma 12, we need to show $\eta_n = [D(\pi_1^n),\dots,D(\pi_n^n)]^{-1}$. For brevity, let us set $h := [D(\pi_1^n),\dots,D(\pi_n^n)]$. Then,

$$R_h\colon n\cdot\mathfrak{F}_0(\mathbf{A}) \to \mathfrak{F}_0(\mathbf{A}^n) : R_h(\langle i,y\rangle) = \{x \in \mathfrak{F}_0(\mathbf{A}^n) \mid A^{i-1}\times y\times A^{n-i} \subseteq x\},$$
$$S_h\colon n\cdot\mathfrak{I}_0(\mathbf{A}) \to \mathfrak{I}_0(\mathbf{A}^n) : S_h(\langle i,y\rangle) = \{x \in \mathfrak{I}_0(\mathbf{A}^n) \mid A^{i-1}\times y\times A^{n-i} \subseteq x\}.$$

On the one hand, for $x \in \mathfrak{F}_0(\mathbf{A}^n)$, we have

$$\begin{aligned} R_{h\circ\eta_n}(x) &= R_h[R_{\eta_n}(x)]'' = R_h[\{\langle i,y\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A}) \mid x \subseteq A^{i-1}\times y\times A^{n-i}\}]'' \\ &= \{z \in \mathfrak{F}_0(\mathbf{A}^n) \mid x \subseteq z\}'' = x'' = R_{id_{D(\mathbf{A}^n)}}, \end{aligned}$$

where the last but one step follows directly from Proposition 2. On the other hand, for $\langle i,y\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A})$, we have

$$\begin{aligned} R_{\eta_n\circ h}(\langle i,y\rangle) &= R_{\eta_n}[R_h(\langle i,y\rangle)]'' = R_{\eta_n}[\{x \in \mathfrak{F}_0(\mathbf{A}^n) \mid A^{i-1}\times y\times A^{n-i} \subseteq x\}]'' \\ &= \{\langle i,z\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A}) \mid y \subseteq z\}'' = \langle i,y\rangle'' = R_{id_{n\cdot\mathbf{X}}}(\langle i,y\rangle), \end{aligned}$$

where the fourth step is again due to Proposition 2. In the same way, it follows that we have $S_{h\circ\eta_n} = S_{id_{D(\mathbf{A}^n)}}$ and $S_{\eta_n\circ h} = S_{id_{n\cdot\mathbf{X}}}$. Thus, $\eta_n = h^{-1}$. □

As outlined in Section 3, we now obtain the clone duality $(-)^\partial\colon O_\mathbf{A} \to \overline{O}_\mathbf{X}$ by setting $f^\partial := \eta_{ar(f)} \circ D(f)$ for $f \in O_\mathbf{A}$. The following proposition states $(-)^\partial$ explicitly:

Proposition 19. *For $f \in O_\mathbf{A}$ with $ar(f) = n$, the multivalued standard morphism $f^\partial \in \overline{O}_\mathbf{X}^{(n)}$ is given as follows:*

$$\text{For } x \in \mathfrak{F}_0(\mathbf{A})\text{: } R_{f^\partial}(x) = \{\langle i,y\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A}) \mid f^{-1}[x] \subseteq A^{i-1}\times y\times A^{n-i}\},$$
$$\text{for } x \in \mathfrak{I}_0(\mathbf{A})\text{: } S_{f^\partial}(x) = \{\langle i,y\rangle \in n\cdot\mathfrak{I}_0(\mathbf{A}) \mid f^{-1}[x] \subseteq A^{i-1}\times y\times A^{n-i}\}.$$

Proof. For $x \in \mathfrak{F}_0(\mathbf{A})$, we have

$$\begin{aligned} R_{f^\partial}(x) &= R_{\eta_n\circ D(f)}(x) \\ &= R_{\eta_n}[R^f(x)]'' \\ &= R_{\eta_n}[\{z \in \mathfrak{F}_0(\mathbf{A}^n) \mid f^{-1}[x] \subseteq z\}]'' \\ &= \{\langle i,y\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A}) \mid f^{-1}[x] \subseteq A^{i-1}\times y\times A^{n-i}\}'' \\ &= \{\langle i,y\rangle \in n\cdot\mathfrak{F}_0(\mathbf{A}) \mid f^{-1}[x] \subseteq A^{i-1}\times y\times A^{n-i}\}. \end{aligned}$$

The part for S_{f^∂} follows in the same way. □

As already noted in the preliminaries, surjective homomorphisms play a special role in the dual equivalence. For them, we can state the following proposition:

Proposition 20. *Let $f \in O_{\mathbf{A}}^{(n)}$ be surjective. Then, for each $x \in \mathfrak{F}_0(\mathbf{A})$ there exist $i \in \{1, \ldots, n\}$, $y \in \mathfrak{F}_0(\mathbf{A})$ such that $R_{f^\partial}(x) = R_{\iota_i^n}(y)$, and similarly, for each $x \in \mathfrak{I}_0(\mathbf{A})$ there exist $i \in \{1, \ldots, n\}$, $y \in \mathfrak{I}_0(\mathbf{A})$ such that $S_{f^\partial}(x) = S_{\iota_i^n}(y)$.*

Proof. Let $x \in \mathfrak{F}_0(\mathbf{A})$. As noted in the preliminaries, f being surjective implies $f^{-1}[x] \in \mathfrak{F}_0(\mathbf{A}^n)$. Thus, there exist $i \in \{1, \ldots, n\}$ and $y \in \mathfrak{F}_0(\mathbf{A})$ such that $f^{-1}[x] = A^{i-1} \times y \times A^{n-i}$. Hence, $R_{f^\partial}(x) = \{\langle i, z \rangle \mid z \in \mathfrak{F}_0(\mathbf{A}), y \subseteq z\} = R_{\iota_i^n}(y)$. As usual, the part for S_{f^∂} follows in the same way. □

Let us summarize the work of this section: We have constructed a clone duality $(-)^\partial$ that dualizes clones over bounded lattices (of arbitrary cardinality) to clones of dual operations that consist of multivalued standard morphisms between standard topological contexts. Thus, we have obtained a technique that allows us to transfer problems from clone theory to the field of Formal Concept Analysis. In the next section, we will put this duality to work and give a small illustration of how this connection can be a useful tool to investigate clones over bounded lattices.

5 A Small Illustration of the Duality

Let us now illustrate that the duality can be used to obtain (new) results for clones over bounded lattices that would be much harder to obtain without the duality. Recall that the categories $\mathcal{A}$ and $\mathcal{X}$, the objects $\mathbf{A}$ and $\mathbf{X}$, the functor D and the clone duality $(-)^\partial \colon O_{\mathbf{A}} \to \overline{O}_{\mathbf{X}}$ still denote what they denoted in the last section (they were all introduced on page 156).

First, we deal with essential arguments. Since the morphisms in our category $\mathcal{A}$ are homomorphisms and therefore set-functions and the products in $\mathcal{A}$ are the Cartesian products, the i-th argument of a morphism $f \in O_{\mathbf{A}}$ is essential in the sense of Definition 8 if and only if the i-th argument of f is essential in the usual sense that we have presented at the beginning of Subsection 2.2. Furthermore, as we have noted in Lemma 15, the i-th argument of f is nonessential if and only if the i-th argument of f^∂ is nonessential. Thus, we can investigate the essentiality of the arguments of an operation $f \in O_{\mathbf{A}}$ by investigating the arguments of its dual $f^\partial \in \overline{O}_{\mathbf{X}}$. To do the latter, we can use the following lemma:

Lemma 21. *Let $n \in \mathbb{N}_+$. For an at least binary multivalued standard morphism $h \in \overline{O}_{\mathbf{X}}^{(n)}$, the following two statements are equivalent:*

(1) the t-th argument of h is nonessential,
(2) $R_h[\mathfrak{F}_0(\mathbf{A})] \subseteq \{\langle i, y \rangle \mid i \in \{1, \ldots, t-1, t+1, \ldots, n\}, y \in \mathfrak{F}_0(\mathbf{A})\}$, and $S_h[\mathfrak{I}_0(\mathbf{A})] \subseteq \{\langle i, y \rangle \mid i \in \{1, \ldots, t-1, t+1, \ldots, n\}, y \in \mathfrak{I}_0(\mathbf{A})\}$.

Proof. Without loss of generality, we can assume $t = 1$.

(1) $\Longrightarrow$ (2). By assumption, h does not depend on its first argument. Hence,

$$[\iota_1^{n+1}, \dots, \iota_n^{n+1}] \circ h = [\iota_{n+1}^{n+1}, \iota_2^{n+1}, \dots, \iota_n^{n+1}] \circ h,$$

and so the claim follows by using Lemma 17.

(2) $\Longrightarrow$ (1). We have $R_h(x) \subseteq \{\langle i, y\rangle \mid i \in \{2, \dots, n\}, y \in \mathfrak{F}_0(\mathbf{A})\}$ for each $x \in \mathfrak{F}_0(\mathbf{A})$. Hence, Lemma 17 yields

$$\begin{aligned} R_{[\iota_1^{n+1},\dots,\iota_n^{n+1}]\circ h}(x) &= R_{[\iota_1^{n+1},\iota_2^{n+1},\dots,\iota_n^{n+1}]}[R_h(x)] \\ &= R_{[\iota_{n+1}^{n+1},\iota_2^{n+1},\dots,\iota_{n+1}^{n+1}]}[R_h(x)] \\ &= R_{[\iota_{n+1}^{n+1},\iota_2^{n+1},\dots,\iota_n^{n+1}]\circ h}(x). \end{aligned}$$

The equation $S_{[\iota_1^{n+1},\dots,\iota_n^{n+1}]\circ h} = S_{[\iota_{n+1}^{n+1},\iota_2^{n+1},\dots,\iota_n^{n+1}]\circ h}$ follows in the same way. $\square$

As this lemma shows, one only needs to look at the images of R_{f^∂} and S_{f^∂} to determine which arguments of an operation $f \in O_{\mathbf{A}}$ are essential and which are nonessential. In most cases, this is much easier than trying to investigate the essentiality of an argument in the usual way. In fact, with this lemma, it becomes remarkably easy to infer many results about the essential arity of operations among $O_{\mathbf{A}}$. For instance, we can now almost trivially deduce the fact that the essential arity of operations among $\overline{O}_{\mathbf{X}}$, and hence $O_{\mathbf{A}}$, is bounded if $\mathbf{X}$ is finite (note that $\mathbf{X}$ is finite if and only if $\mathbf{A}$ is finite). This result is known and usually derived from the fact that lattice-homomorphisms satisfy the strong term condition [McK83], so what we have obtained is an alternative proof, where the lemma above replaces the arguments from universal algebra. A much more ambitious goal would be to use this lemma to obtain a sharp bound on the essential arity of operations over a given finite lattice, which, to the best knowledge of the author, is an open problem. It seems promising that this problem can be solved with the help of the lemma above and some work with the multivalued standard morphisms. However, it would be beyond the scope of this paper.

Let us instead conclude this section with some results about idempotent operations. Recall that a function f is said to be *idempotent* if $f(x, \dots, x) \approx x$. Writing this equivalently in category-theoretic notation, we can say that an operation $f \in O_{\mathbf{A}}^{(n)}$ is idempotent if and only if $f \circ \langle id_{\mathbf{A}}, \dots, id_{\mathbf{A}}\rangle = id_{\mathbf{A}}$. Clearly, by Lemma 15, $f \in O_{\mathbf{A}}$ is idempotent if and only if $f^\partial \in \overline{O}_{\mathbf{X}}$ is a *dual idempotent operation*, that is, $[id_{\mathbf{X}}, \dots, id_{\mathbf{X}}] \circ f^\partial = id_{\mathbf{X}}$. A clone of (dual) operations is called *idempotent* if it contains only idempotent (dual) operations.

We start our small investigation of idempotent operations by providing the following characterization of the dual idempotent operations among $\overline{O}_{\mathbf{X}}$:

Lemma 22. *Let $h \in \overline{O}_{\mathbf{X}}^{(n)}$. The following two statements are equivalent:*

(1) h is idempotent.
(2) For all $x \in \mathfrak{F}_0(\mathbf{A})$, there exists $i \in \{1, \dots, n\}$ such that $R_h(x) = R_{\iota_i^n}(x)$, and for all $x \in \mathfrak{I}_0(\mathbf{A})$ there exists $i \in \{1, \dots, n\}$ such that $S_h(x) = S_{\iota_i^n}(x)$.

Proof. (1) $\Longrightarrow$ (2). As usual, we only need to show the part for R_h since the statement for S_h follows in the same way. First, let $x \in \mathfrak{F}_0(\mathbf{A})$. There exists $f \in O_{\mathbf{A}}$ such that $f^\partial = h$. Since h is idempotent, so is f. This also implies that f is surjective. Therefore, we can apply Proposition 20, and it follows that there exists $i \in \{1, \ldots, n\}$ and $y \in \mathfrak{F}_0(\mathbf{A})$ such that $R_h(x) = R_{\iota_i^n}(y)$. Moreover, the idempotency of h implies $\iota_i^n = [\iota_i^n, \ldots, \iota_i^n] \circ h$. Hence,

$$\begin{aligned} R_{\iota_i^n}(x) &= R_{[\iota_i^n,\ldots,\iota_i^n]\circ h}(x) = R_{[\iota_i^n,\ldots,\iota_i^n]}[R_h(x)]'' = R_{[\iota_i^n,\ldots,\iota_i^n]}[R_{\iota_i^n}(y)]'' \\ &= R_{[\iota_i^n,\ldots,\iota_i^n]\circ\iota_i^n}(y) = R_{\iota_i^n}(y) = R_h(x). \end{aligned}$$

(2) $\Longrightarrow$ (1). We have to show $id_{\mathbf{X}} = [id_{\mathbf{X}}, \ldots, id_{\mathbf{X}}] \circ h$. For each $x \in \mathfrak{F}_0(\mathbf{A})$, there exists $i \in \{1, \ldots, n\}$ such that $R_h(x) = R_{\iota_i^n}(x)$. Hence,

$$\begin{aligned} R_{id_{\mathbf{X}}}(x) &= R_{[id_{\mathbf{X}},\ldots,id_{\mathbf{X}}]\circ\iota_i^n}(x) = R_{[id_{\mathbf{X}},\ldots,id_{\mathbf{X}}]}[R_{\iota_i^n}(x)]'' \\ &= R_{[id_{\mathbf{X}},\ldots,id_{\mathbf{X}}]}[R_h(x)]'' = R_{[id_{\mathbf{X}},\ldots,id_{\mathbf{X}}]\circ h}(x). \end{aligned}$$

Analogously, it follows $S_{id_{\mathbf{X}}} = S_{[id_{\mathbf{X}},\ldots,id_{\mathbf{X}}]\circ h}$, so $id_{\mathbf{X}} = [id_{\mathbf{X}}, \ldots, id_{\mathbf{X}}] \circ h$. □

With this lemma, we can establish a close connection between the dual idempotent operations over $\mathbf{X}$ (and hence the idempotent operations over $\mathbf{A}$) and certain partitions.

Definition 23. *For a dual idempotent operation $h \in \overline{O}_{\mathbf{X}}^{(n)}$, we denote by $\Pi(h)$ the partition of $\mathfrak{F}_0(\mathbf{A}) \cup \mathfrak{I}_0(\mathbf{A})$ obtained by setting $\Pi(h) := \{X_1, \ldots, X_n\} \setminus \{\emptyset\}$ where $X_1, \ldots, X_n$ are defined as follows:*

$$\begin{aligned} \text{For } x \in \mathfrak{F}_0(\mathbf{A})\text{: } x \in X_i &:\Longleftrightarrow R_h(x) = R_{\iota_i^n}(x), \\ \text{for } x \in \mathfrak{I}_0(\mathbf{A})\text{: } x \in X_i &:\Longleftrightarrow S_h(x) = S_{\iota_i^n}(x). \end{aligned}$$

Note that $\Pi(h)$ is a well-defined partition due to Lemma 22. Thus, every dual idempotent operation on $\mathbf{X}$ can be uniquely assigned to a partition of $\mathfrak{F}_0(\mathbf{A}) \cup \mathfrak{I}_0(\mathbf{A})$ (but not necessarily vice versa). Moreover, denoting by $\preccurlyeq$ the finer-than relation for partitions, we can use Lemma 22 to easily deduce the following statement (its proof will be omitted due to limitation of space):

Lemma 24.

(a) For two idempotent dual operations $h_1, h_2 \in \overline{O}_{\mathbf{X}}$, we have $h_2 \in \overline{\mathrm{Clo}}(h_1)$ if and only if $\Pi(h_1) \preccurlyeq \Pi(h_2)$. Consequently, $\overline{\mathrm{Clo}}(h_1) = \overline{\mathrm{Clo}}(h_2)$ if and only if $\Pi(h_1) = \Pi(h_2)$.

(b) Each idempotent $C \leq \overline{O}_{\mathbf{X}}$ is generated by a single dual operation.

Note that the second part of the lemma clearly also holds in its dualized version, that is, each idempotent $C \leq O_{\mathbf{A}}$ is determined by a single operation. The first part of this lemma makes it very easy to decide whether two dual idempotent operations generate each other, and with a little bit more work, we can also establish a close connection between the lattice of partitions of $\mathfrak{F}_0(\mathbf{A}) \cup \mathfrak{I}_0(\mathbf{A})$ (given by $\preccurlyeq$) and the lattice of idempotent clones over $\mathbf{A}$ (one of the ideals of the clone lattice that is of particular interest).

Proposition 25. *The lattice of idempotent clones of operations over* $\mathbf{A}$ *can be embedded into the lattice of partitions of the set* $\mathfrak{F}_0(\mathbf{A}) \cup \mathfrak{I}_0(\mathbf{A})$.

Proof. By Lemma 24, the desired lattice-embedding φ can be obtained by setting $\varphi(C) := \Pi(f^\partial)$ where f is one of the single operations that generate C. □

For future research, it would be an interesting task to further investigate the lattice of idempotent clones of operations over $\mathbf{A}$ by characterizing the sublattice of the lattice of partitions of $\mathfrak{F}_0(\mathbf{A}) \cup \mathfrak{I}_0(\mathbf{A})$ to which it is isomorphic.

6 Conclusion

We used the dual equivalence from [Har93] and the results from [Ker11] to construct a duality between clones over bounded lattices and so-called clones of dual operations over standard topological contexts. We gave some small examples of how this connection between clone theory and Formal Concept Analysis can be used to simplify clone theoretic problems and to produce concrete results. In the process, we also stated some open problems for which an application of the duality seems promising.

References

[Csá85] Csákány, B.: Completeness in coalgebras. Acta Sci. Math. 48, 75–84 (1985)
[GBGL08] Gowers, T., Barrow-Green, J., Leader, I. (eds.): The Princeton companion to mathematics. Princeton University Press, Princeton (2008)
[Geh06] Gehrke, M.: Generalized Kripke frames. Studia Logica 84(2), 241–275 (2006)
[GW99] Ganter, B., Wille, R.: Formal concept analysis. Mathematical foundations. Springer, Berlin (1999); Translated from the 1996 German original by Cornelia Franzke
[Har93] Hartung, G.: An extended duality for lattices. In: General Algebra and Applications, Potsdam. Res. Exp. Math., vol. 20, pp. 126–142. Heldermann, Berlin (1992, 1993)
[Haz95] Hazewinkel, M. (ed.): Encyclopaedia of mathematics. A–Zyg, index, vol. 1–6. Kluwer Academic Publishers, Dordrecht (1995); Translated from the Russian, Reprint of the 1988-1994 English translation
[HD97] Hartonas, C., Dunn, J.M.: Stone duality for lattices. Algebra Universalis 37(3), 391–401 (1997)
[Ker11] Kerkhoff, S.: A general duality theory for clones, Ph.D. thesis, Technische Universität Dresden (2011)
[Maš06] Mašulović, D.: On dualizing clones as lawvere theories. International Journal of Algebra and Computation 16, 675–687 (2006)
[McK83] McKenzie, R.: Finite forbidden lattices. In: Universal Algebra and Lattice Theory, Puebla. Lecture Notes in Math., vol. 1004, pp. 176–205. Springer, Berlin (1982, 1983)
[MMT87] McKenzie, R.N., McNulty, G.F., Taylor, W.: Algebras, lattices, varieties. The Wadsworth & Brooks/Cole Mathematics, vol. I. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey (1987)
[Urq78] Urquhart, A.: A topological representation theory for lattices. Algebra Universalis 8(1), 45–58 (1978)

Formal Concept Discovery in Semantic Web Data

Markus Kirchberg[1], Erwin Leonardi[1], Yu Shyang Tan[1], Sebastian Link[2], Ryan K.L. Ko[1], and Bu Sung Lee[1]

[1] Cloud & Security Lab, Hewlett-Packard Laboratories, Singapore
[2] Dept of Computer Science, Auckland University, New Zealand

Abstract. Semantic Web efforts aim to bring the WWW to a state in which all its content can be interpreted by machines; the ultimate goal being a machine-processable Web of Knowledge. We strongly believe that adding a mechanism to extract and compute concepts from the Semantic Web will help to achieve this vision. However, there are a number of open questions that need to be answered first. In this paper we will establish partial answers to the following questions: 1) Is it feasible to obtain data from the Web (instantaneously) and compute formal concepts without a considerable overhead; 2) have data sets, found on the Web, distinct properties and, if so, how do these properties affect the performance of concept discovery algorithms; and 3) do state-of-the-art concept discovery algorithms scale wrt. the number of data objects found on the Web?

Keywords: Formal Concept Discovery, Semantic Web, Web of Data, Knowledge Extraction, Parallel Algorithms, Performance Evaluation.

1 Introduction

The Semantic Web [1] is the most prominent effort to address limitations underlying the design of the World Wide Web (WWW); the main aim is to bring the WWW to a state in which all its content can also be interpreted by machines. The emphasis is not primarily on putting data on the Web, but rather on creating links in a way that both humans and machines can explore this Web of Data (WoD). The ultimate goal, however, is to create a machine-processable Web of Knowledge. We strongly believe that adding a mechanism to extract and compute formal concepts from the Semantic Web will help to achieve this vision.

In the WoD, facts (commonly referred to as RDF triples or quads) are not partitioned according to their meaning. Therefore, every fact represents a single concept. With a concept abstraction, semantically-related facts are linked together as meaningful units. Formal Concept Analysis (FCA) [2,3] is a toolbox of well-founded, consumer[1]-oriented methods to structure and analyse data. FCA, by means of a lattice representation, enables visualisation of not only data

[1] A consumer being a developer, machine, end-user, etc.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 164–179, 2012.

but also their inherent structures, implications and dependencies. As an additional benefit, concepts are abstractions easier to comprehend by consumers. This assists with the verification of how results have been computed. In turn, this can help with data provenance and with the establishment of trust among users – two additional goals of the Semantic Web.

In this paper, we will briefly outline our vision of supplementing the Web of Data with a concept layer. This is followed by presenting an approach of how FCA tools and algorithms can be applied to the Semantic Web. Once introduced, we will examine differences of properties inherent in Web data-sets and those from traditional FCA data-sets. Mining concepts from diverse (and dynamic) data sources has the potential to produce data inputs that are different from what FCA algorithms have been applied to in the past. We will further study by means of experimentation whether and/or how such differences affect the performance characteristics of FCA algorithms.

2 Related Work

Applying FCA to the Semantic Web domain is not new in itself; there have been several works on ontology alignment, learning and engineering as well as Semantic Web querying, browsing and visualization. In this section, we briefly introduce basic Semantic Web terminology and review the body of existing works at the intersection of FCA and Semantic Web. For an introduction to FCA basics, we refer the reader to [2,3].

2.1 Semantic Web Data

The Resource Description Framework (RDF) [4] is the basic format of data in the Semantic Web; it consists of statements of the form: "*Subject S is in some relation P to object O*". Such statements (S, P, O) are known as RDF triples (or *triples* for short) that can be serialised in N-Triple format. Recently, the notion of provenance (or *context*) has been added to triples, and hence, "*Given context C, Subject S is in some relation P to object O*". Triples with context are called RDF quads (or *quads* for short). Quads can be modelled in N-Quads format, an extension of N-Triples with context. Each quad has the following form:

```
<subject S> <predicate P> <object O> <context C> .
```

2.2 FCA Algorithms and Benchmarks

Over the years, various FCA implementations and benchmarks (e.g., [5]) have been published. Given indicative performance measurements published by Strok et al. [6] as well as the results from the ICCS 2009 FCA Algorithms competition[2], our work mainly utilises implementations from the family of CbO algorithms (i.e., PCbO [7], FCbO [8] and PFCbO [8]) and In-Close (i.e., In-Close2 [9]) algorithms – as they were deemed to be the fastest FCA algorithms.

[2] http://www.upriss.org.uk/fca/fcaalgorithms.html

2.3 FCA and the Semantic Web

Formal Concept Analysis has been applied to various areas of the Semantic Web. Most notable are works on ontology querying, browsing and visualization (e.g., [10]), interactive ontology merging (e.g., [11]), ontology learning from text (e.g., [12]) as well as interactive completion of ontologies (e.g., [13]). Besides ontologies, FCA has been utilised for measuring the similarity of FCA concepts by determining the similarity of concept descriptors (i.e., attributes) via the information content approach [14]; building concept maps for a given set of documents and quantifying the semantic relations between those concepts [15]; facilitating conceptual navigation of RDF graphs whereby concepts are accessed through SPARQL-like queries [16]; and extracting representative questions over a given RDF data-set utilising FCA [17].

3 Towards a Concept Layer for the Semantic Web

The current state of the Semantic Web is centred on the Web of Data (also known as Linked Open Data (LOD) [18]), which is comprised of RDF triples. The ultimate goal of the Semantic Web is to create a machine-processable Web of Knowledge (WoK). Such a WoK would be comprised of services in which the semantics of content are made explicit while content itself is linked to both, other content and services. The gap between the current LOD and the envisioned WoK is still large; for instance LOD's rapid growth, varying data-set compliance wrt. LOD guidelines, complexity of computations over large data-sets, and lack of tools and applications pose significant challenges to reaching the ultimate goal.

How do we select the right LOD data-set(s) that contain(s) the data of interest? Even once suitable LOD data-sets have been found, how can we understand their relationships, schemata (i.e., ontologies), and content? These are just a few of the problems and issues yet to be addressed successfully.

The main barriers for the Web of Knowledge can be categorised into barriers of creation and barriers of usage. The former includes challenges related to the cost involved in developing suitable ontologies as well as annotating Web pages by linking RDF-encoded facts in Web pages to ontologies. The latter refers to difficulty in writing SPARQL [19] queries and semantic rules.

We strongly believe that the aforementioned gap can be bridged by adding an additional abstraction layer of concepts to the Web of Data. The discovery of formal concepts in RDF triples will partition the Web of Data into a Web of concepts. Therefore, a concept layer partitions RDF triples into equivalence classes of semantically related facts. Thus, an additional concept layer better links the mature core layers (i.e., Unicode, Uniform Resource Identifier, RDF and SPARQL) of the Semantic Web Stack [20] with its upper layers; addressing WoK barriers by linking data that are semantically related in a given context.

Concepts can be computed using the well-founded FCA approach. FCA, by means of a lattice representation, enables visualisation of not only data but also their inherent structures, implications and dependencies. Utilising the FCA approach that has been applied to many different areas offers significant benefits and value to Semantic Web efforts:

- Concepts facilitate understanding *without* having ontologies at hand;
- Concepts will lower costs of creating, maintaining, integrating and exchanging ontologies;
- Concepts can be automatically computed from RDF-encoded facts (at scale) and are context-aware;
- Concepts reduce the difficulty in writing SPARQL queries and semantic rules as data can be better understood; and
- Concepts are easier to understand by consumers and allow for verification of data sources by following the path of computation, i.e., supporting trust and provenance.

The notion of concepts has a great potential to remove some of the barriers that currently prevent the full vision of the Semantic Web from becoming reality. We foresee that an additional abstraction layer of concepts can directly address barriers associated with creation (e.g., cost of developing ontologies, linking RDF-encoded facts in Web pages to ontologies at Web-scale) while usage barriers (e.g., writing SPARQL queries and semantic rules) are lower – nevertheless, we further improve usability by making it easier to understand data.

A detailed discussion of the placement and benefits of such a concept layer is beyond the scope of this paper; instead, we will present details on how FCA can be applied to Web-scale data.

4 Applying FCA to Semantic Web Data

4.1 Concept Computation Process Overview

The concept layer mentioned in Section 3 computes Semantic Web concepts using FCA; an overview of the process to compute concepts is depicted in Figure 1. Consumers (i.e., developers, machines, end-users, etc.) first define the context in which concepts are to be explored; therefore, a Concept Recipe is to be specified (Fig. 1, step 1). A *Concept Recipe* is comprised of three parts:

1. The context is specified as a list of data sources; supported data sources are RDF documents and SPARQL queries.
2. The object set is defined by specifying bindings wrt. (RDF) subjects, predicates, objects, or combinations thereof.
3. The attribute set is defined by specifying bindings wrt. (RDF) subjects, predicates, objects, or combinations thereof.

In this paper, we will focus on FCA context extraction from the Web (Fig. 1, step 2) and concept computation (Fig. 1, step 4).

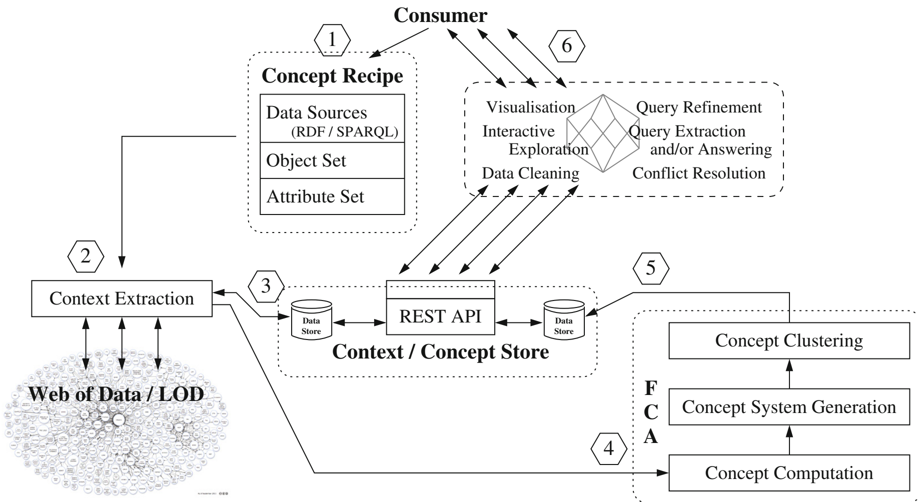


Fig. 1. Concept Computation Process Overview

4.2 Extracting Contexts from the Semantic Web

As an initial example, let us consider a SPARQL-generated context from DBPedia[3]; as data source, we specify a query that returns DBPedia triples with common predicate '*<http://dbpedia.org/ontology/officialLanguage>*'. Thus, (RDF) subjects will be countries while (RDF) objects are languages. A corresponding JSON-based [21] concept recipe with the (FCA) object set made up of countries and the (FCA) attribute set made up of the official languages spoken by these countries would be given as follows:

```
{ "dataSource": [
   { "id": 0,
     "type": "SPARQL_ENDPOINT",
     "endpoint": "http://dbpedia.org/sparql",
     "sparql": "select distinct ?s ?o where {
      ?s <http://dbpedia.org/ontology/officialLanguage> ?o }"
   } ],
  "fcaObjectSet": [
   { "source_id": 0,
     "binding": "?s" } ],
  "fcaAttributeSet": [
   { "source_id": 0,
     "binding": "?o" } ] }
```

[3] DBPedia (http://dbpedia.org/) is a community effort to extract structured information from Wikipedia and to transform it into RDF. Each Wikipedia entry has its corresponding DBPedia URI. DBPedia is one of the most important data sources in LOD as it can be seen as the center of the LOD cloud.

When a Concept Recipe is received by the Context Extraction component (Fig. 1, step 2), the recipe is first parsed to identify the data sources, object set, and attribute set definitions. We define two types of supported data sources, namely, RDF and SPARQL_ENDPOINT. The RDF type data source requires the system to fetch an RDF document from the given URL and store it into a local RDF repository. After it is stored, the system is able to issue the specified SPARQL query against the RDF repository (Fig. 1, step 3). When dealing with the SPARQL_ENDPOINT type data source, the system only needs to send the given SPARQL query to the specified SPARQL endpoint[4]. This is because the data has been stored in an external RDF repository and can be queried via a SPARQL endpoint. The query execution results are materialized in a local relational data store. After processing the data sources, the system generates the object set and attribute set using SQL queries against the relational data store. Similarly, the context matrix is generated by issuing SQL queries to the relational data store.

4.3 Computing Concepts

Given an FCA context, a slightly extended version of an FCA concept computation algorithm (i.e., PCbO, FCbO, PFCbO or In-Close2 in the context of this paper) and customised FCA concept system generation, annotation and clustering scripts are run computing formal concepts for the given context, $\leq$ concept relationships, concept support values, concept lattice edges and their annotations, and values necessary for concept clustering (Fig. 1, step 4). All computational results are stored in a data store (Fig. 1, step 5).

At various points of the processing, data is moved to a local data store (referred to as *Context / Concept Store* in Figure 1). This data store is comprised of an RDF repository (built using the Jena framework [23]) and relational data stores (using both traditional relational database systems as well as column-store database systems); the choice of which data resides in which data store container is based on the efficiency and effectiveness wrt. both storage and access. In addition, using a data store allows us to reuse previously retrieved context portions as well as intermediate computation results if and as feasible.

Given the context, concepts and concept system data are made available through the data store via a RESTful [24] interface (Fig. 1, step 6). Having the information about concepts available this way will drive the development of various applications, such as interactive Semantic Web data exploration, query refinement, concept clustering similarity measurement between RDF triples or facts embedded in them, and detection of data inconsistencies.

Let us revisit our earlier example that extracts a context from DBPedia and forms an object set consisting of countries and an attribute set of the official languages spoken by these countries. Concept extraction returns 497 SPARQL query results which contain 316 unique objects (i.e., countries) and 169 unique

[4] A SPARQL endpoint is a conformant SPARQL protocol service as defined in [22].

attributes (i.e., official languages). Concept computation reveals 187 concepts embedded in the context wrt. the given object set and attribute set.

In order to determine performance implications underlying our approach, we are interested in the overhead incurred when extracting Web content, converting Web content to standard FCA input format (i.e., FIMI or CXT format) and computing Web data concepts. Furthermore, we will examine how Web data differs from those data-sets commonly used in the FCA community up to now. Lastly, we will benchmark current FCA implementations to see how they perform on Web data and whether they have the potential to scale up or not.

4.4 Semantic Web Data Properties

LOD has seen a tremendous rise in popularity over the past few years; as of September 2011, LOD consists of 295 officially acknowledged data-sets; spans domains such as media, geographic, government (largest wrt. triples), publications (largest wrt. number of data-sets), cross-domain, life sciences (largest wrt. out-links) and user-generated content [25]. Mining concepts from such diverse (and dynamic) data sources has the potential to produce data inputs that are different from what FCA algorithms have been applied to in the past. As reference points, we have studied various FCA data-sets made available to the public including the FCA Repository[5] and the Frequent Itemset Mining Dataset Repository[6]. One of the first observations we made was that two data properties seem to differ significantly:

1. Traditional FCA data-sets commonly exhibit a medium to high FCA matrix density while Web data seems to have very low matrix density values – typically well below 1%.
2. Traditional FCA data-sets have a relatively small number of objects, while Web data can have hundreds of thousands, if not millions of objects.

To highlight the former point, we have first obtained a set of meaningful Web data-sets and examined their properties. These Web data-sets are as follows:

– DBPedia Languages: Query the DBPedia SPARQL endpoint for the official languages spoken (`dbpedia-owl:officialLanguage` property) by people living in different countries. The DBPedia URI of each country and the DBPedia URI of each spoken language are the object set and attribute set, respectively.
– DBPedia Drug: Query the DBPedia SPARQL endpoint for the routes of administration (`dbpprop:routesOfAdministration` property) of drugs (e.g., mouth, rectum, intravenous therapy, etc.). The DBPedia URI of each drug and the DBPedia URI of each route of administration are the object set and attribute set, respectively.

[5] http://fcarepository.com/
[6] http://fimi.ua.ac.be/data/

- DBPedia Drug v2: Query the DBPedia SPARQL endpoint for topic (`dcterms:subject` property) of drugs (e.g., Analgesics, Antipyretics, etc.). The DBPedia URI of each drug and the DBPedia URI of each topic are the object set and attribute set, respectively.
- DBPedia Country: Query the DBPedia SPARQL endpoint for the topic (`dcterms:subject` property) of countries (e.g., Republics, Islamic Countries, etc.). The DBPedia URI of each country and the DBPedia URI of each topic are the object set and attribute set, respectively.
- UK Crime Locations[7]: Query the Crime Report UK SPARQL endpoint for the location (`crime:location` property) of each crime report. The value of `crime:location` property becomes the attribute, and the URI of the crime report becomes the object.
- DBPedia Alma-mater: Query the DBPedia SPARQL endpoint for the alma-maters (`dbpedia-owl:almaMater` property) of the persons. The DBPedia URI of each person and the DBPedia URI of each alma-mater are the object set and attribute set, respectively.
- DBPedia Genre: Query the DBPedia SPARQL endpoint for the music genres (`dbpedia-owl:genre` property) of the musical artists. The DBPedia URI of each musical artist and the DBPedia URI of each music genre are the object set and attribute set, respectively.

Table 1 summarises the properties of these data-sets; for comparison, we have included two traditional FCA data-sets (but studied many more): the Adult data-set (a USA Census Income data-set); and the Mushroom data-set (mushrooms being described in terms of physical characteristics; classification: poisonous or edible). Most notable differences include the low FCA matrix density and the small number of concepts inherent in Web data. This leads us to a number of questions we want to address; most prominently:

1. Obtaining data from the Web (instantaneously) causes a considerable overhead. Can the overhead be quantified wrt. downloading the data and converting it to FCA input formats (FIMI versus CXT)? Will the overhead of obtaining and converting data mitigate concept computation time?
2. Considering the significant differences in FCA matrix density, how will common FCA implementations perform on Web data?
3. How good or bad do common FCA implementations scale wrt. the number of objects?

We have conducted various sets of experiments within our context extraction and concept computation framework (depicted in Figure 1 on page 168); selected experiments and results will be discussed in the next section.

[7] Crime Report UK (http://crime.rkbexplorer.com/) is a linked data representation of the street-level crime reports first released for England and Wales in 2011. Each entry represents a crime report enriched by linking to the nearest postcode for the position at which the crime was reported.

Table 1. Data-set Properties (Differences in Density)

Data-set	Data-set Size (in bytes)	No. of Objects	No. of Attributes	Density of Matrix	Matrix Size	No. of Concepts
Adult	n/a	32, 561	124	12.09%	4, 038k	1, 064, 875
Mushroom	n/a	8, 124	119	21.01%	967k	238, 710
DBPedia Languages	82, 958	316	169	0.931%	53k	187
DBPedia Drugs	365, 150	2, 162	459	0.234%	992k	504
DBPedia Drugs v2	2, 923, 649	4, 726	1, 245	0.287%	5, 884k	9, 012
DBPedia Country	2, 948, 530	2, 345	5, 709	0.115%	13, 388k	8, 316
UK Crime Locations	6, 910, 550	31, 936	20, 707	0.005%	661, 299k	20, 708
DBPedia Alma-mater	7, 595, 917	27, 383	5, 407	0.029%	148, 060k	13, 605
DBPedia Genre	10, 553, 958	26, 672	2, 145	0.113%	57, 211k	21, 805

5 Experiments

5.1 Web Data Extraction and Preparation

First, we want to determine the overhead of downloading Web data and converting it to FCA input formats (FIMI and CXT). The general procedure is as discussed in Section 4; individual time measurements are obtained for the following steps: a) Download time; b) Store object set in database; c) Store attribute set in database; d) Store context matrix in database; e) Dump context matrix into FIMI format; and f) Dump context matrix into CXT format.

Figure 2 and Tables 2 and 3 summarise the results for Web data-sets listed in Table 1. Main bottlenecks are download time and generation of CXT-formatted input files (while FIMI-formatted input generation only causes a negligible overhead). The significantly slower CXT-formatted input file generation times for UK Crime Locations, DBPedia Alma-mater and DBPedia Genre data-sets are due to their much larger context matrix sizes (detailed in Table 1).

5.2 FCA Algorithms: Ensuring Fairness

Before we discuss the various experiments in greater detail, it should be highlighted that we have undertaken various efforts to level the playing field for all

Table 2. Web Data Extraction and Preparation Performance (in seconds)

Data-set	Download time	Object set to DB	Attribute set to DB	Context matrix to DB	Generate FIMI format	Generate CXT format
DBPedia Languages	3.56	0.324	0.256	0.515	0.576	0.209
DBPedia Drugs	4.94	0.721	0.348	0.962	0.180	0.249
DBPedia Drugs v2	20.22	0.153	0.644	0.547	0.803	1.694
DBPedia Country	16.48	0.736	0.189	0.504	0.695	4.405
UK Crime Locations	30.72	1.359	0.932	1.336	0.206	161.52
DBPedia Alma-mater	40.18	1.305	0.199	1.468	0.236	39.714
DBPedia Genre	63.66	1.203	0.187	2.351	0.305	15.077

Table 3. Web Data Extraction and Preparation Performance Overall (in seconds)

	DBPedia						UK Crime
	Languages	Drugs	Drugs v2	Country	Genre	Alma-mater	Locations
FIMI Overall	5.231	7.151	22.366	18.604	67.705	43.388	34.554
CXT Overall	4.864	7.220	23.258	22.313	82.478	82.866	195.868

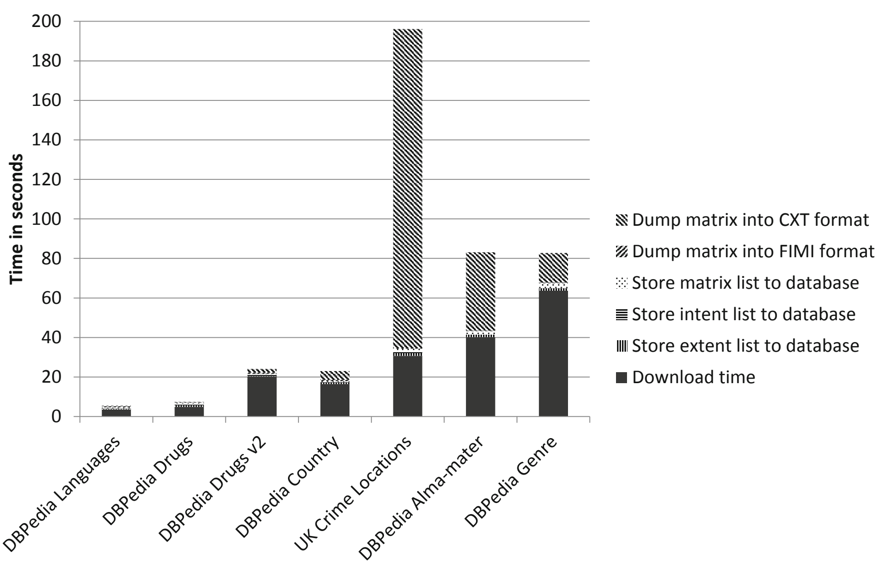


Fig. 2. Web Data Extraction and Preparation Performance Test

utilised algorithms, i.e., PCbO [7], FCbO [8], PFCbO [8], and In-Close2 [9][8]. While PCbO and FCbO only accept FIMI-formatted input files, In-Close2 only supports CXT-formatted input files; PFCbO is the only implementation that supports both input formats. Since CXT input files are significantly larger, we will differentiate between input file reading time and execution time. For concept computation performance evaluation, we will mainly consider execution time. Further differences concern the way results are returned; we have modified all algorithms to only return the intent portion of computed concepts. This will result in almost identical outputs (remaining differences being sorted versus unsorted intent values and whether intent values are counted starting from 0 or 1). As such differences remain, we verify correctness of computed results by ensuring that the number of concepts are the same and that for each computed concept, there is a matching concept with the same number of intent values (i.e., line and word counts of resulting concepts are identical). In addition, the following modifications were applied:

- PCbO: Several memory leaks were closed.
- PFCbO: Several memory leaks were closed, memory management was tweaked to support in-memory data structures holding more than 2GB of data (on 64bit operating systems), and static output buffer memory allocation was adjusted to gather for large data-sets.
- In-Close2: Code was ported from the original Windows implementation to Linux; interactive parts were removed and two memory configurations were prepared to suit the tested data-sets best (as memory allocation is static).

Modified FCA algorithms, data-sets and complete results can be accessed at: http://icfca2012.markuskirchberg.net/.

5.3 FCA Algorithms Performance: Traditional vs. Web Data

Next, we want to determine how common FCA implementations perform on Web data. Experiments start from the respective FCA input formats (FIMI and CXT) that were either generated (in the case of Web data) or obtained from FCA data-set repositories.

Benchmarking Set-up. We conducted this test series on an HP Cluster with the following configuration: Intel 8-core CPU, 2.7GHz, 16GB RAM, 16GB SWAP, and Ubuntu Linux 64bit. All results are average values obtained over 5 complete runs; data-sets and algorithms were matched in round-robin fashion. Default time-out (t/o) setting for each algorithm was $3,600$ seconds.

[8] PCbO and FCbO implementations were obtained from http://fcalgs.sourceforge.net/; PFCbO codes were kindly made available to us by P. Krajca, and the In-Close2 implementation was downloaded from http://sourceforge.net/projects/inclose/.

Table 4. FCA Algorithms Performance (in seconds) for Varying Data-sets

Data-set	In-Close2 (CXT)	PCbO (FIMI)	PCbO -P8 (FIMI)	FCbO (FIMI)	PFCbO (FIMI)	PFCbO -C8 (CXT)	PFCbO -C8 (FIMI)
Adult	1.950	664.83	207.94	6.485	1.328	0.394	**0.382**
Mushroom	0.697	96.530	27.404	1.250	0.376	0.184	**0.173**
DBPedia Language	**0.001**	0.002	0.003	0.003	0.003	0.005	0.004
DBPedia Drugs	0.024	0.035	0.033	0.026	0.023	**0.020**	0.024
DBPedia Drugs v2	**0.093**	1.318	0.575	0.247	0.209	0.150	0.159
DBPedia Country	**0.361**	3.597	2.026	1.489	10.034	8.146	6.966
UK Crime Locations	**24.028**	399.03	401.31	1,188.53	493.41	204.21	209.38
DBPedia Alma-mater	**4.192**	45.855	26.675	24.055	11.144	5.580	5.484
DBPedia Genre	1.704	25.208	9.294	3.312	1.433	**1.063**	1.090

Experiments and Results. For each of the data-sets from Table 1, we have run a variety of FCA algorithms; Table 4 summarises selected results. In-Close2 and PFCbO-based algorithms have the best overall performance; with the exception of the UK Crime Locations data-set for which all CbO-based algorithms including PFCbO perform rather badly. Coincidentally, this is the most sparse data-set that formed a part of our experiments. In-Close2 seems to be very suited for sparse data-sets as it more often than not outperforms PFCbO algorithms. However, the opposite is true for traditional (higher density) FCA data-sets (represented by the Adult and Mushroom data-sets here; the same applies to random FCA data-sets available via the FCA Repository and the Frequent Itemset Mining Dataset Repository).

Table 5. Input File Read Performance (in seconds) for Selected FCA Algorithms

	DBPedia Languages	DBPedia Drugs	DBPedia Drugs v2	DBPedia Country	DBPedia Genre	DBPedia Alma-mater	UK Crime Locations
In-Close2 (CXT)	0.001	0.004	0.017	0.035	0.128	0.278	1.122
PFCbO -C8 (FIMI)	0.000	0.001	0.003	0.005	0.017	0.024	0.076

It should be noted that omitting input file reading times has no significant impact on the results detailed in this subsection (selected reading times are shown in Table 5). However, generating CXT input files versus generating FIMI input files for online Web data processing poses a clear threat to the applicability of

Table 6. Context Extraction and Concept Computation Performance (in seconds)

Data-set	In-Close2 (CXT Input, Reading & Execution)	PFCbO -C8 (FIMI Input, Reading & Execution
DBPedia Languages	**4.866**	5.236
DBPedia Drugs	7.248	**7.176**
DBPedia Drugs v2	23.367	**22.528**
DBPedia Country	**22.709**	25.576
UK Crime Locations	**221.018**	244.008
DBPedia Alma-mater	87.336	**48.896**
DBPedia Genre	84.309	**68.812**
	450.853	422.232

In-Close2. PFCbO, which can deal with FIMI as well as CXT input files equally well, would be the overall winner when summing up corresponding measurements from Tables 3, 4 and 5; as detailed in Table 6.

5.4 FCA Algorithms Performance: Web-Scale Data

Finally, we want to obtain indicative results on how well common FCA implementations scale wrt. the number of objects[9]. For this, we utilise an additional Semantic Web data-set extracted from the 2011 Billion Triple Challenge (BTC)[10] data-set. The RDF-type data-set contains any RDF-typed facts that form a part of the BTC data-set. For our experiments, we consider the first 10K, 50K, 100K, 250K, 500K, and 750K objects, respectively; Table 7 outlines corresponding data-set properties.

Table 7. RDF-type Data-set Properties

Data-set	10K	50K	100K	250K	500K	750K
No. of Objects	10,000	50,000	100,000	250,000	500,000	750,000
No. of Attributes	1,548	1,548	1,548	1,552	1,552	1,552
No. of Concepts	220	449	601	885	1,159	1,416

Benchmarking Set-up. We conducted this test series on an HP Cluster with the following configuration: Intel 8-core CPU, 2.7GHz, 16GB RAM, 16GB SWAP, and Ubuntu Linux 64bit. All results are average values obtained over 5 complete runs; data-sets and algorithms were matched in round-robin fashion. Default time-out (t/o) setting for each algorithm was 10,800 seconds.

[9] Scalability wrt. objects is of more interest as typical Web of Data use cases retrieve streams of similar facts whereby attributes remain almost constant.

[10] http://km.aifb.kit.edu/projects/btc-2011/

Table 8. FCA Algorithms Performance (in seconds) for Web-scale Data Test

	Number of Objects in Context					
	10K	50K	100K	250K	500K	750K
In-Close2 (CXT)	0.511	16.064	71.525	751.31	3,466.19	8,059.15
PCbO (FIMI)	0.375	7.935	31.683	214.62	948.87	2,316.44
PCbO -P8 (FIMI)	0.375	7.863	31.298	211.81	957.41	2,305.76
FCbO (FIMI)	**0.116**	**1.006**	**3.191**	15.339	53.345	106.99
PFCbO (FIMI)	0.591	1.525	4.247	18.387	31.639	52.278
PFCbO -C4 (FIMI)	0.314	2.355	3.690	**13.904**	32.018	42.449
PFCbO -L4 (FIMI)	0.215	1.326	3.349	16.002	**25.883**	**40.901**

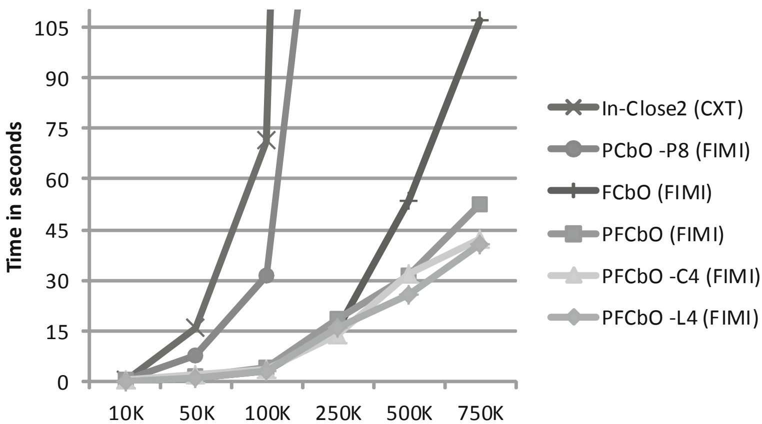


Fig. 3. FCA Algorithms Performance for Web-scale Data Test

Experiments and Results. Figure 3 and Table 8 summarise selected experimental results. The In-Close2 algorithm is the worst performer across the board, while FCbO for small (<= 100K) and PFCbO for large (>= 250K) data-sets perform clearly the best. In addition, CXT-formatted inputs incur a hefty reading time penalty (for example, reading time for the $750K$ CXT data-set exceeds 11.5 seconds while the corresponding FIMI-formatted data-set is read in under 300 milli-seconds). Among the tested algorithms and for the tested data-set, only PFCbO seems to be a feasible solution when moving close to 1 million objects and beyond.

6 Conclusion

In this paper, we have introduced our on-going efforts to apply FCA concepts and algorithms to the Semantic Web. We have shown that FCA context extraction and concept computation times are feasible wrt. online and offline processing.

Notable observations are differences in properties of Web data when compared to traditional FCA data-sets as well as performance measurements for various different types of Web data. With respect to overall performance characteristics, PFCbO is the most suitable state-of-the-art FCA algorithm for Web-scale data. Main drawback of the In-Close2 algorithm are reliance on CXT-formatted input files only and the poor performance in our scalability test. However, as our scalability tests are only indicative, and even PFCbO performed rather poorly for the very low density UK Crime Locations data-set, more in-depth studies are necessary. A particular area of interest is the main memory footprint; some of our ongoing experiments with larger data-sets (beyond 5 million objects) have already resulted in PFCbO running out of main memory (e.g., 16GB RAM plus 16GB SWAP) while In-Close2 has a much smaller in-memory footprint.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
2. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations, 1st edn. Springer-Verlag New York, Inc. (1997)
3. Priss, U.: Formal concept analysis in information science. Annual Review of Information Science and Technology 40(1), 521–543 (2006)
4. Lassila, O., Swick, R.R.: Resource description framework (RDF) model and syntax, version 1, WD-rdf-syntax-971002 (1997)
5. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence 14(2-3), 189–216 (2002)
6. Strok, F., Neznanov, A.: Comparing and analyzing the computational complexity of FCA algorithms. In: Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT), pp. 417–420. ACM (2010)
7. Krajca, P., Outrata, J., Vychodil, V.: Parallel recursive algorithm for FCA. In: Proceedings of the 6th International Conference on Concept Lattices and Their Applications (CLA), vol. 433, pp. 71–82. CEUR WS (2008)
8. Krajca, P., Outrata, J., Vychodil, V.: Advances in algorithms based on CbO. In: Proceedings of the 8th International Conference on Concept Lattices and Their Applications (CLA), vol. 672, pp. 325–337. CEUR WS (2010)
9. Andrews, S.: In-Close2, a High Performance Formal Concept Miner. In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) ICCS-ConceptStruct 2011. LNCS, vol. 6828, pp. 50–62. Springer, Heidelberg (2011)
10. Tane, J., Cimiano, P., Hitzler, P.: Query-Based Multicontexts for Knowledge Base Browsing: An Evaluation. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 413–426. Springer, Heidelberg (2006)
11. Maedche, A., Staab, S.: Ontology learning for the Semantic Web. IEEE Intelligent Systems 16, 72–79 (2001)
12. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using Formal Concept Analysis. Journal of Artificial Intelligence Research 24, 305–339 (2005)

13. Völker, J., Rudolph, S.: Lexico-Logical Acquisition of OWL DL Axioms; An Integrated Approach to Ontology Refinement. In: Medina, R., Obiedkov, S. (eds.) ICFCA 2008. LNCS (LNAI), vol. 4933, pp. 62–77. Springer, Heidelberg (2008)
14. Formica, A.: Concept similarity in Formal Concept Analysis: An information content approach. Knowledge-Based Systems 21, 80–87 (2008)
15. Lee, M.C., Chen, H.H., Li, Y.S.: FCA based concept constructing and similarity measurement algorithms. International Journal of Advancements in Computing Technology (IJACT) 3(1), 97–105 (2011)
16. Ferré, S.: Conceptual Navigation in RDF Graphs with SPARQL-Like Queries. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 193–208. Springer, Heidelberg (2010)
17. d'Aquin, M., Motta, E.: Extracting relevant questions to an RDF dataset using formal concept analysis. In: Proceedings of the 6th International Conference on Knowledge Capture (K-CAP), pp. 121–128. ACM (2011)
18. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web andInformation Systems 5(3), 1–22 (2009)
19. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF, W3C Recommendation (2008)
20. Berners-Lee, T.: Semantic Web - XML2000, Keynote Presentation at XML, Slide 10, Architecture (2000)
21. Crockford, D.: The application/json media type for JavaScript object notation (JSON), IETF, sec. 6, RFC 4627 (2006)
22. Clark, K.G., Feigenbaum, L., Torres, E.: SPARQL protocol for RDF, W3C Recommendation (2008)
23. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: Implementing the Semantic Web recommendations. In: Proceedings of the 13th International World Wide Web (WWW) Conference on Alternate Track Papers & Posters, pp. 74–83. ACM (2004)
24. Fielding, R.T., Taylor, R.N.: Principled design of the modern Web architecture. ACM Transactions on Internet Technology 2(2), 115–150 (2002)
25. Bizer, C., Jentzsch, A., Cyganiak, R.: State of the LOD cloud (2011), http://www4.wiwiss.fu-berlin.de/lodcloud/state/

Concept Lattices of Incomplete Data*

Michal Krupka and Jan Lastovicka

Department of Computer Science
Palacký University in Olomouc 17. listopadu 12,
CZ-77146 Olomouc Czech Republic
{michal.krupka,jan.lastovicka}@upol.cz

Abstract. We present a method of constructing a concept lattice of a formal context with incomplete data. The lattice reduces to a classical concept lattice when the missing values are completed. The lattice also can reflect any known dependencies between the missing values. We show some experiments indicating that in most cases, when the number of missing values is not large, the size of the incomplete concept lattice is not substantially greater than the size of the concept lattice of completed data.

Keywords: Incomplete Data, Concept Lattice, Boolean Algebra, Fuzzy Logic.

1 Introduction

In practice, we sometimes encounter a need to analyze incomplete data. In Formal Concept Analysis (FCA), such data can be represented by a formal context $\langle X, Y, I \rangle$, where I is a multiple-valued relation $X \times Y \to L$, taking values in some structure of truth values L.

Formal contexts with incomplete data were studied first by Burmeiser and Holzer [5]. They used a framework of a three-valued logic, called *Kleene logic*, where $L = \{0, 1, ?\}$, with "?" representing an unknown value. In [5], a notion of incomplete context is introduced and several of its properties are studied. The authors concentrate namely on attribute implications and attribute exploration for incomplete contexts. The authors notice the main drawback of using Kleene logic in this context: in this logic, the implication satisfies $(? \to ?) = ?$. This is appropriate, if the logical value "?" represents different unknown values, but does not satisfy the natural requirement that the implication $a \to a$ should always hold, even if the value of a is unknown.

In this paper, we try to overcome this problem by switching to a multiple-valued logic with several distinct unknown values, where each of the values satisfies $a \to a = 1$. As we show, Boolean algebras are structures of truth values, suitable for our purposes.

Boolean algebras represent a special case of residuated lattices. This allows us to use results of formal concept analysis in fuzzy setting [1,2], where residuated lattices are used as structures of truth values. We also bring some simple new results for FCA in fuzzy setting we need in this paper. It should be emphasized that although we use FCA

* Supported by grant no. P103/10/1056 of the Czech Science Foundation.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 180–194, 2012.

in fuzzy setting, the meaning of our results is different and should not be interpreted as results for fuzzy logic. In fuzzy logic, intermediate truth values (i.e., the values between 0 and 1) indicate a degree to which some condition is satisfied (some proposition is true). However, in our case, these values represent something completely different; they are used as variables (or some expressions, constructed from the variables), holding an unknown value equal to 0 or 1.

Another approach to the above problem has been taken in [8], where a modal logic instead of Kleene logic is used.

In this paper, we concentrate mainly on the problem of constructing a concept lattice of incomplete contexts. The main goal of our research is to enable the possibility to view the conceptual structure of the data, even if the data are incomplete. As it turns out, the concept lattice can be constructed, its size is usually reasonable small (compared to the size of the concept lattice of a completed context), and it contains the information on all the concept lattices, that can be obtained by completions of the data.

The organization of the paper is as follows. Section 2 contains an introduction to concept lattices, Boolean algebras, residuated lattices and fuzzy sets. In Sec. 3 we describe the main reasons for using Boolean algebras as structures of truth values and in Sec. 4 we define incomplete contexts using Boolean algebras.

In Sec. 5 we introduce necessary results from Formal Concept Analysis in fuzzy setting. We also prove some new results we will need in the paper. Section 6 contains our main results. A reader, not familiar with fuzzy sets and Formal Concept Analysis in fuzzy setting can skip these two sections and go directly to Sec. 7.

Section 7 contains a review of main results from previous sections and an illustrative example. Section 8 contains some experimental results on the size of concept lattices of incomplete contexts.

2 Preliminaries

2.1 Concept Lattices

Concept lattices have been introduced in [9], the basic reference is [6]. A *formal context* is a triple $\langle X,Y,I\rangle$ where X is a set of objects, Y a set of attributes and $I \subseteq X \times Y$ a binary relation between X and Y. For $\langle x,y\rangle \in I$ it is said "The object x has the attribute y".

For subsets $A \subseteq X$ and $B \subseteq Y$ we set

$$A^{\uparrow_I} = \{y \in Y \mid \text{for each } x \in A \text{ it holds } \langle x,y\rangle \in I\},$$
$$B^{\downarrow_I} = \{x \in X \mid \text{for each } y \in B \text{ it holds } \langle x,y\rangle \in I\}.$$

If $A^{\uparrow_I} = B$ and $B^{\downarrow_I} = A$, then the pair $\langle A,B\rangle$ is called a *formal concept* of $\langle X,Y,I\rangle$. The set A is called the *extent of* $\langle A,B\rangle$, the set B the *intent of* $\langle A,B\rangle$.

We write $^{\uparrow}$ (resp. $^{\downarrow}$) instead of $^{\uparrow_I}$ (resp. $^{\downarrow_I}$) when I is obvious.

A partial order $\leq$ on the set $\mathscr{B}(X,Y,I)$ of all formal concepts of $\langle X,Y,I\rangle$ is defined by

$$\langle A_1,B_1\rangle \leq \langle A_2,B_2\rangle \quad \text{iff} \quad A_1 \subseteq A_2 \quad (\text{iff } B_2 \subseteq B_1).$$

$\mathscr{B}(X,Y,I)$ together with $\leq$ is a complete lattice and is called the *concept lattice* of $\langle X,Y,I\rangle$.

2.2 Boolean Algebras and Residuated Lattices

Recall [7] that a *Boolean algebra* is an algebra $\mathbf{L} = \langle L, \wedge, \vee, ', 0, 1 \rangle$, with two binary, one unary operation (called *complementation*), and two nullary operations, such that $\langle L, \wedge, \vee \rangle$ is a distributive lattice with the least element 0 and the greatest element 1 and for each $a \in L$,

$$a \wedge a' = 0, \qquad a \vee a' = 1.$$

The basic example of a Boolean algebra is the two-element Boolean algebra $\mathbf{2}$, which is used as a structure of truth values in classical logic, and the Boolean algebra of all subsets of a set U, denoted by $\mathbf{2}^U$. We usually interpret elements of $\mathbf{2}^U$ as mappings from U to $\mathbf{2}$.

A *residuated lattice* [2] is an algebra $\langle L, \wedge, \vee, \otimes, \to, 0, 1 \rangle$, where $\langle L, \wedge, \vee, 0, 1 \rangle$ is a lattice with the least element 0 and the greatest element 1; $\langle L, \otimes, 1 \rangle$ is a commutative monoid (i.e., $\otimes$ is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$); $\otimes$ *(product)* and $\to$ *(residuum)* satisfy the so-called *adjointness property*: $a \otimes b \leq c$ iff $a \leq b \to c$ for each $a, b, c \in L$ (the order $\leq$ on L is defined as usual by $a \leq b$ iff $a \wedge b = a$). A residuated lattice is called *complete* if its underlying lattice is complete. A homomorphism of residuated lattices is called *complete*, if it preserves arbitrary suprema and infima: $\bigvee_{j \in J} f(a_j) = f(\bigvee_{j \in J} a_j)$, $\bigwedge_{j \in J} f(a_j) = f(\bigwedge_{j \in J} a_j)$.

The class of Boolean algebras can be considered a subclass of the class of residuated lattices. Namely, if we set

$$a \otimes b = a \wedge b, \qquad a \to b = a' \vee b, \tag{1}$$

for each a, b in a Boolean algebra $\mathbf{L}$, then $\langle L, \wedge, \vee, \otimes, \to, 0, 1 \rangle$ is a residuated lattice. The complementation can be then computed by $a' = a \to 0$. If $\mathbf{L}$ is a Boolean algebra, then the residuated lattice $\langle L, \wedge, \vee, \otimes, \to, 0, 1 \rangle$ will be again denoted by $\mathbf{L}$.

2.3 L-sets and L-relations

For a residuated lattice $\mathbf{L}$, an $\mathbf{L}$-*set A in universe X* is a mapping $A \colon X \to L$. The set of all $\mathbf{L}$-sets in universe X is denoted by L^X. An $\mathbf{L}$-set $A \in L^X$ is called *crisp*, if $A(X) \subseteq \{0, 1\}$. A *binary* $\mathbf{L}$-*relation between sets X and Y* is an $\mathbf{L}$-set $I \in L^{X \times Y}$.

Operations with $\mathbf{L}$-sets are defined element-wise. For instance, intersection of $\mathbf{L}$-sets $A, B \in L^X$ is an $\mathbf{L}$-set $A \cap B$ in X such that $(A \cap B)(x) = A(x) \wedge B(x)$ for each $x \in X$, etc. An $\mathbf{L}$-set $A \in L^X$ is also denoted $\{{}^{A(x)}/x \mid x \in X\}$. If for all $x \in X$ distinct from $x_1, x_2, \ldots, x_n$ we have $A(x) = 0$, we also write $\{{}^{A(x_1)}/x_1, {}^{A(x_2)}/x_1, \ldots, {}^{A(x_n)}/x_n\}$. We usually write x instead of ${}^1/x$.

For two $\mathbf{L}$-sets $A, B \in L^X$ we say that A is a *subset* of B and write $A \subseteq B$, if for each $x \in X$ it holds $A(x) \leq B(x)$. *The degree $S(A, B)$ to which A is contained in B* is defined by

$$S(A, B) = \bigwedge_{x \in X} A(x) \to B(x). \tag{2}$$

It holds $S(A, B) = 1$ iff $A \subseteq B$.

Mappings $h \colon \mathbf{L}_1 \to \mathbf{L}_2$ of two residuated lattices $\mathbf{L}_1$ and $\mathbf{L}_2$ can transform $\mathbf{L}_1$-sets to $\mathbf{L}_2$-sets: for an $\mathbf{L}_1$-set $A \in (L_1)^X$ the composition $h \circ A$ is an $\mathbf{L}_2$-set, $h \circ A \in (L_2)^X$.

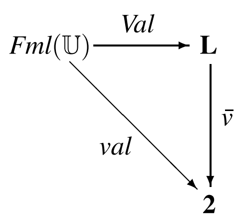
3 Boolean Algebras with Variables

As we mentioned in the introduction, we base our theory of incomplete contexts on a multiple-valued logic with truth values 0 (representing falsity), 1 (representing truth) and other values, representing either variables, containing unknown values from $\{0,1\}$, or values, computed from the variables by means of Boolean operations.

Let $\mathbf{L}$ be a finite Boolean algebra together with a set of *variables* $U = \{u_1,\ldots, u_k\}$ and mapping $\iota : U \to L$, such that $\mathbf{L}$ is generated by $\iota(U)$. We call $\mathbf{L}$ a *Boolean algebra with variables* $u_1,\ldots,u_k$. If $\mathbf{L}$ is freely generated by $\iota(U)$, then the variables $u_1,\ldots,u_k$ are said to be *independent*.

Mappings $v\colon U \to \mathbf{2}$ are called *assignments*. Since $\mathbf{L}$ is generated by $\iota(U)$, then for each assignment v there exists at most one homomorphism $\bar{v}\colon \mathbf{L} \to \mathbf{2}$, such that $v = \bar{v} \circ \iota$. If this homomorphism exists, then the assignment v is called *admissible*. In this way we establish a bijection between the set of all admissible assignments and the set of all homomorphisms from $\mathbf{L}$ to $\mathbf{2}$. The variables $u_1,\ldots,u_k$ are independent if and only if each assignment is admissible. The other cases model situations when there are some known dependencies between the unknown values. For example, if $\iota(u_1) \le \iota(u_2)$, then there is no admissible assignment v such that $v(u_1) = 1$ and $v(u_2) = 0$.

We use the Boolean algebra $\mathbf{L}$ as a structure of truth values. The main reason for this choice can be explained by means of formulas of propositional logic. Let $Fml(\mathbb{U})$ be the set of formulas of propositional logic over a set of propositional variables $\mathbb{U} = \{\mathbb{u}_1,\ldots,\mathbb{u}_k\}$ and with propositional connectives , , and . If $val\colon Fml(\mathbb{U}) \to \mathbf{2}$ is a truth function, such that the assignment v, defined by $v(u_i) = val(\mathbb{u}_i)$, $i = 1,\ldots,k$, is admissible, then val can be factorized through $\mathbf{L}$, as it is shown in the following commutative diagram:



The mapping *Val* is unique and does not depend on v. It satisfies

$$\begin{aligned} Val(\mathbb{u}_i) &= \iota(u_i), \\ Val(F \quad G) &= Val(F) \wedge Val(G), \\ Val(F \quad G) &= Val(F) \vee Val(G), \\ Val(\quad F) &= Val(F)', \end{aligned}$$

for each $i \in \{1,\ldots,k\}$, $F,G \in Fml(\mathbb{U})$. Thus, in our case, when the assignment v is not known, it is reasonable to use the Boolean algebra $\mathbf{L}$ as the structure of truth values and the mapping *Val* as the truth function.

The next theorem shows that this choice satisfies the following two natural requirements:

1. Generality: for every possible dependency between variables there exists an appropriate Boolean algebra $\mathbf{L}$ (together with a mapping $\iota : U \to L$).
2. Efficiency: the Boolean algebra $\mathbf{L}$ is (up to isomorphism) the smallest element within the class of residuated lattices satisfying the generality requirement 1.

Theorem 1. *The following holds for each subset $V \subseteq \mathbf{2}^U$.*

1. Let $\mathbf{L} = \mathbf{2}^V$ and a mapping $\iota : U \to L$ be defined by $(\iota(u))(v) = v(u)$. Then for each $v \in V$ there is exactly one homomorphism $\bar{v} : \mathbf{L} \to \mathbf{2}$ such that $v = \bar{v} \circ \iota$.

2. Let $\mathbf{L}'$ be a residuated lattice and $\iota' : U \to L'$ a mapping such that for each $v \in V$ there is a homomorphism $\bar{v}' : \mathbf{L}' \to \mathbf{2}$ satisfying $v = \bar{v}' \circ \iota'$. Then there exists a surjective homomorphism of residuated lattices $s : \mathbf{L}' \to \mathbf{L}$ such that for each $v \in V$ it holds $\bar{v}' = \bar{v} \circ s$.

Proof. 1. Set for each $a \in L$, $\bar{v}(a) = a(v)$. The mapping $\bar{v}$ is evidently a Boolean algebra homomorphism (for example $\bar{v}(a \wedge b) = (a \wedge b)(v) = a(v) \wedge b(v) = \bar{v}(a) \wedge \bar{v}(b)$) and for each $u \in U$ we have $(\bar{v} \circ \iota)(u) = \bar{v}(\iota(u)) = (\iota(u))(v) = v(u)$, which means that $\bar{v} \circ \iota = v$.

Let $\kappa : V \to L$ be defined by $\kappa(v) = \{v\}$ (i.e., $(\kappa(v))(w) = 1$ iff $w = v$). We will prove that for each $v \in V$ it holds

$$\kappa(v) = \bigwedge_{\substack{u \in U \\ v(u)=1}} \iota(u) \wedge \bigwedge_{\substack{u \in U \\ v(u)=0}} \iota(u)'. \tag{3}$$

Indeed, we have

$$\left(\bigwedge_{v(u)=1} \iota(u) \wedge \bigwedge_{v(u)=0} \iota(u)' \right)(w) = \bigwedge_{v(u)=1} (\iota(u))(w) \wedge \bigwedge_{v(u)=0} (\iota(u))(w)'$$
$$= \bigwedge_{v(u)=1} w(u) \wedge \bigwedge_{v(u)=0} w(u)',$$

which is equal to 1 iff $w = v$. Hence, the right-hand side of (3) is equal to the singleton $\{v\}$.

Now, from (3) and the obvious fact that $\mathbf{L}$ is generated by the set $\kappa(V) \subseteq L$ it follows that $\mathbf{L}$ is generated by the set $\iota(U) \subseteq L$, which proves uniqueness of $\bar{v}$.

2. For each $a \in L'$ set $(s(a))(v) = \bar{v}'(a)$. The mapping s is evidently a homomorphism of residuated lattices (for example, $(s(a \wedge b))(v) = \bar{v}'(a \wedge b) = \bar{v}'(a) \wedge \bar{v}'(b) = (s(a))(v) \wedge (s(b))(v) = (s(a) \wedge s(b))(v)$). For each $u \in U$, $v \in V$ we have $(s(\iota'(u)))(v) = \bar{v}'(\iota'(u)) = v(u) = \bar{v}(\iota(u)) = (\iota(u))(v)$. Thus, $s(\iota'(u)) = \iota(u)$ and since $\mathbf{L}$ is generated by the set $\iota(U)$, then s is surjective.

Finally, for each $a \in L'$ we have $(\bar{v} \circ s)(a) = \bar{v}(s(a)) = (s(a))(v) = \bar{v}'(a)$, which finishes the proof.

Remark 1. The Boolean algebra $\mathbf{L}$ from the above theorem can be also obtained by factorization. We start with the free Boolean algebra $\mathbf{L} = \mathbf{2}^{\mathbf{2}^U}$ over U with the canonical inclusion $\iota : U \to \mathbf{L}$, given by $(\iota(u))(v) = v(u)$, and factorize it by the congruence $\approx$, where $a \approx b$ iff $\bar{v}(a) = \bar{v}(b)$ for each $v \in V$.

Example 1. Let $U = \{u_1, u_2\}$, $V = \{v_1, v_2, v_3\}$, where $v_1 = \emptyset$, $v_2 = \{u_2\}$, $v_3 = \{u_1, u_2\}$ (i.e., $v_1(u_1) = v_1(u_2) = 0$, $v_2(u_1) = 0$, $v_2(u_2) = 1$, and $v_3(u_1) = v_3(u_2) = 1$). The choice of V is equivalent to specifying the following dependency between the variables: $\iota(u_1) \leq \iota(u_2)$ (this dependency removes the remaining possible assignment $\{u_1\}$).

The Boolean algebra $\mathbf{L} = \mathbf{2}^V$ has 8 elements, which we represent as triples: $a = \langle a(v_1), a(v_2), a(v_3)\rangle \in L$. By Theorem 1, the mapping $\iota: U \to L$ is given by $(\iota(u))(v) = v(u)$. Thus, $\iota(u_1) = \langle 0,0,1\rangle$ and $\iota(u_2) = \langle 0,1,1\rangle$. In the table in Fig. 1, we can see all the elements of $\mathbf{L}$ in columns (we write u_i instead of $\iota(u_i)$ in the header row).

	u_1	u_2	0	$u_1' \wedge u_2$	u_2'	$u_1 \vee u_2'$	u_1'	1
$\bar{v}_1$	0	0	0	0	1	1	1	1
$\bar{v}_2$	0	1	0	1	0	0	1	1
$\bar{v}_3$	1	1	0	0	0	1	0	1

Fig. 1. The 8-element Boolean algebra $\mathbf{L}$ from Example 1. Elements of $\mathbf{L}$ are written in columns, their values in the mappings $\bar{v}_1$, $\bar{v}_2$, $\bar{v}_3$ are in rows.

By the proof of Theorem 1, the mappings $\bar{v}_i$, $i = 1,2,3$, are given by $\bar{v}_i(a) = a(v_i)$. Thus, the values of the mappings $\bar{v}_i$ in each of the elements of $\mathbf{L}$ can be read from the rows of the table.

This way we constructed the Boolean algebra $\mathbf{L}$ with variables u_1, u_2, based on the given set of admissible assignments. We could also go the opposite way: if we define a Boolean algebra $\mathbf{L}$ by the table in Fig. 1 and set $u_1 = \langle 0,0,1\rangle$, and $u_2 = \langle 0,1,1\rangle$, then we easily obtain that the admissible assignments are just the assignments v_1, v_2, v_3.

Any $\mathbf{L}$-set $A \in \mathbf{L}^X$, where $\mathbf{L}$ is a Boolean algebra with variables $u_1, \ldots, u_k$, can be interpreted as a set, whose elements depend on values of the variables. Indeed, for each admissible assignment v, $\bar{v} \circ A$ is a mapping from X to $\mathbf{2}$, i.e., a crisp set. We illustrate this in the following example.

Example 2. Let $\mathbf{L}$ be the Boolean algebra with variables u_1, u_2 from Example 1, $X = \{x_1, x_2, x_3\}$ a set. Further, let $A = \{x_1, {}^{u_1 \vee u_2'}/x_2, {}^{u_1'}/x_3\}$ be an $\mathbf{L}$-set in X (for brevity, we write u_i instead of $\iota(u_i)$). For an admissible assignment v, the crisp set $\bar{v} \circ A$ can be computed by assigning values $v(u_1)$, resp. $v(u_2)$ to u_1, resp. u_2. Thus, for the assignments v_1, v_2, and v_3 from Example 1 we obtain $\bar{v}_1 \circ A = \{x_1, x_2, x_3\}$, $\bar{v}_2 \circ A = \{x_1, x_3\}$, and $\bar{v}_2 \circ A = \{x_1, x_2\}$.

4 Incomplete Contexts

Let $\mathbf{L}$ be a Boolean algebra with variables $u_1, \ldots, u_k$, $U = \{u_1, \ldots, u_k\}$ the set of the variables. For simplicity, we suppose that $U \subseteq L$ and ι is the inclusion $U \to L$.

An *incomplete* $\mathbf{L}$*-context* is a triple $\langle X, Y, I\rangle$, where X and Y are sets and $I: X \times Y \to L$ is an $\mathbf{L}$-relation such that $I(X \times Y) \subseteq U \cup \{0, 1\}$. An ordinary formal context $\langle X, Y, J\rangle$ is a *completion* of $\langle X, Y, I\rangle$, if $J = \bar{v} \circ I$ for an admissible assignment $v: U \to \mathbf{2}$.

	y_1	y_2	y_3	y_4	y_5
x_1			×	×	
x_2	u_1	×	u_2	×	
x_3	×	×	×		
x_4		×			

Fig. 2. Incomplete context from Example 3. Empty places represent 0, crosses 1. $u_1, u_2 \in L$ are variables.

Example 3. Let $\mathbf{L}$ and $U \subseteq L$ be from Ex. 1, $X = \{x_1, x_2, x_3, x_4\}$, $Y = \{y_1, y_2, y_3, y_4, y_5\}$, and an $\mathbf{L}$-relation $I \in L^{X \times Y}$ be given by the table in Fig. 2.

In the figure, empty places represent the value 0, crosses 1. The variables u_1, u_2 represent unknown values such that $u_1 \leq u_2$.

In Fig. 3 there are depicted the three possible completions $\langle X, Y, \bar{v}_1 \circ I \rangle$, $\langle X, Y, \bar{v}_2 \circ I \rangle$, $\langle X, Y, \bar{v}_3 \circ I \rangle$ of the incomplete context $\langle X, Y, I \rangle$, corresponding to the admissible assignments $v_1 = \emptyset$, $v_2 = \{u_2\}$, $v_3 = \{u_1, u_2\}$, and their respective concept lattices $\mathscr{B}(X, Y, \bar{v}_1 \circ I)$, $\mathscr{B}(X, Y, \bar{v}_2 \circ I)$, $\mathscr{B}(X, Y, \bar{v}_3 \circ I)$. We use standard labeling of elements of the lattices by objects and attributes.

	y_1	y_2	y_3	y_4	y_5
x_1			×	×	
x_2		×		×	
x_3	×	×	×		
x_4		×			

	y_1	y_2	y_3	y_4	y_5
x_1			×	×	
x_2		×	×	×	
x_3	×	×	×		
x_4		×			

	y_1	y_2	y_3	y_4	y_5
x_1			×	×	
x_2	×	×	×	×	
x_3	×	×	×		
x_4		×			

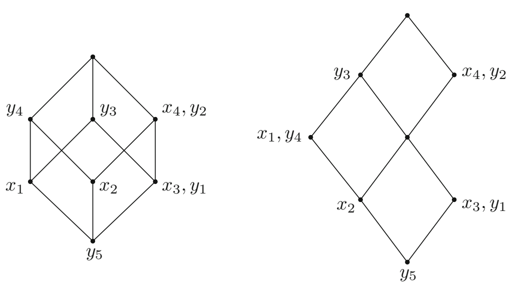


Fig. 3. Three possible completions of the incomplete context from Example 3, corresponding to admissible assignments (from left to right) $v_1 = \emptyset$, $v_2 = \{u_2\}$, $v_3 = \{u_1, u_2\}$ (first row), and their concept lattices (second row)

5 Formal Concept Analysis in Fuzzy Setting

We introduce basics of Formal Concept Analysis in fuzzy setting [1]. Our main reference is [2]. The theory covers classical Formal Concept Analysis as a special case. For crisply generated fuzzy concepts see [3].

Let $\mathbf{L}$ be a complete residuated lattice. By a *formal* $\mathbf{L}$*-context* we understand the triple $\langle X, Y, I\rangle$, where X and Y are sets and I is an $\mathbf{L}$-relation between X and Y, $I\colon X \times Y \to L$. The sets X and Y are interpreted as sets of objects, resp. attributes, and for any $x \in X$, $y \in Y$ the value $I(x,y) \in L$ is interpreted as the degree to which the object x has the attribute y.

For any $\mathbf{L}$-set $A \in L^X$ of objects we define an $\mathbf{L}$-set $A^{\uparrow_I} \in L^Y$ of attributes by

$$A^{\uparrow_I}(y) = \bigwedge_{x\in X} A(x) \to I(x,y). \tag{4}$$

Similarly, for any $\mathbf{L}$-set $B \in L^Y$ of attributes we define an $\mathbf{L}$-set $B^{\downarrow_I}$ of objects by

$$B^{\downarrow_I}(x) = \bigwedge_{y\in Y} B(y) \to I(x,y). \tag{5}$$

The $\mathbf{L}$-set $A^{\uparrow_I}$ is interpreted as the $\mathbf{L}$-set of all attributes shared by objects from A. Similarly, the $\mathbf{L}$-set $B^{\downarrow_I}$ is interpreted as the $\mathbf{L}$-set of all objects having the attributes from B in common. If there is no danger of confusion, we write simply $^{\uparrow}$ and $^{\downarrow}$ instead of $^{\uparrow_I}$ and $^{\downarrow_I}$.

An $\mathbf{L}$*-formal concept* of a formal $\mathbf{L}$-context $\langle X,Y,I\rangle$ is a pair $\langle A,B\rangle \in L^X \times L^Y$ such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The $\mathbf{L}$-set A is called *the extent*, B *the intent* of $\langle A,B\rangle$. The set of all formal concepts of $\langle X,Y,I\rangle$ is denoted $\mathscr{B}(X,Y,I)$ and called the $\mathbf{L}$*-concept lattice of* $\langle X,Y,I\rangle$.

The condition

$$\langle A_1,B_1\rangle \le \langle A_2,B_2\rangle \quad \text{iff} \quad A_1 \subseteq A_2 \quad (\text{iff} \quad B_2 \subseteq B_1) \tag{6}$$

defines a partial ordering on $\mathscr{B}(X,Y,I)$. Together with this ordering, $\mathscr{B}(X,Y,I)$ is a complete lattice with infima and suprema given by

$$\bigwedge_{j\in J} \langle A_j,B_j\rangle = \left\langle \bigcap_{j\in J} A_j, \left(\bigcup_{j\in J} B_j\right)^{\downarrow\uparrow} \right\rangle, \tag{7}$$

$$\bigvee_{j\in J} \langle A_j,B_j\rangle = \left\langle \left(\bigcup_{j\in J} A_j\right)^{\uparrow\downarrow}, \bigcap_{j\in J} B_j \right\rangle, \tag{8}$$

and is called *the* $\mathbf{L}$*-concept lattice of* $\langle X,Y,I\rangle$.

In addition to the partial ordering (6), we have a binary $\mathbf{L}$-relation $\preceq$ on $\mathscr{B}(X,Y,I)$, defined by

$$(\langle A_1,B_1\rangle \preceq \langle A_2,B_2\rangle) = S(A_1,A_2) \quad (= S(B_2,B_1)). \tag{9}$$

The $\mathbf{L}$-relation $\preceq$ satisfies conditions for an $\mathbf{L}$-*order* on $\mathscr{B}(X,Y,I)$ (see [1,2] for details).

A formal $\mathbf{L}$-concept $\langle A,B\rangle \in \mathscr{B}(X,Y,I)$ is called *crisply generated*, if there is a crisp set $B_0 \subseteq Y$ such that $A = (B_0)^{\downarrow}$ (and hence $B = (B_0)^{\downarrow\uparrow}$). The set $\mathscr{B}_c(X,Y,I) \subseteq \mathscr{B}(X,Y,I)$

of all crisply generated **L**-concepts, equipped with the restriction of the ordering $\leq$, is a complete lattice and a $\bigwedge$-semilattice of $\mathscr{B}(X,Y,I)$. More details on crisply generated concepts can be found in [3].

Remark 2. In many applications it is not necessary to work with all formal **L**-concepts from $\mathscr{B}(X,Y,I)$. The restriction to crisply generated concepts has the advantage that the lattice $\mathscr{B}_c(X,Y,I)$ is much smaller than the whole concept lattice $\mathscr{B}(X,Y,I)$. We shall use this advantage later in the paper.

Below are some observations on a correspondence between homomorphisms of residuated lattices and homomorphisms of concept lattices, which we will use in the paper.

Lemma 1. *Let $h\colon \mathbf{L} \to \mathbf{L}'$ be a complete homomorphism of complete residuated lattices, $\langle X,Y,I\rangle$ a formal **L**-context. Then for each $A \in \mathbf{L}^X$ and $B \in \mathbf{L}^Y$ it holds*

$$(h\circ A)^{\uparrow_{h\circ I}} = h\circ A^{\uparrow_I}, \qquad (h\circ B)^{\downarrow_{h\circ I}} = h\circ B^{\downarrow_I}. \tag{10}$$

Proof. For each $y \in Y$ we have

$$\begin{aligned}(h\circ A)^{\uparrow_{h\circ I}}(y) &= \bigwedge_{x\in X}(h\circ A)(x) \to (h\circ I)(x,y) = \bigwedge_{x\in X} h(A(x)) \to h(I(x,y))\\ &= h\left(\bigwedge_{x\in X} A(x) \to I(x,y)\right) = h(A^{\uparrow_I}(y)) = (h\circ A^{\uparrow_I})(y).\end{aligned}$$

Analogously for B.

Theorem 2. *Let $h\colon \mathbf{L} \to \mathbf{L}'$ be a complete homomorphism of complete residuated lattices, $\langle X,Y,I\rangle$ a formal **L**-context. Then for each formal concept $\langle A,B\rangle \in \mathscr{B}(X,Y,I)$ it holds $\langle h\circ A, h\circ B\rangle \in \mathscr{B}(X,Y,h\circ I)$ and the induced mapping $h^{\mathscr{B}(X,Y,I)}\colon \mathscr{B}(X,Y,I) \to \mathscr{B}(X,Y,h\circ I)$ is a complete homomorphism, such that*

$$\left(h^{\mathscr{B}(X,Y,I)}(\langle A_1,B_1\rangle) \preceq h^{\mathscr{B}(X,Y,I)}(\langle A_2,B_2\rangle)\right) = h(\langle A_1,B_1\rangle \preceq \langle A_2,B_2\rangle), \tag{11}$$

for each $\langle A_1,B_1\rangle, \langle A_2,B_2\rangle \in \mathscr{B}(X,Y,I)$. If h is injective, then so is $h^{\mathscr{B}(X,Y,I)}$, if h is surjective, then so is $h^{\mathscr{B}(X,Y,I)}$.

Proof. Let $\langle A,B\rangle \in \mathscr{B}(X,Y,I)$. By Lemma 1, $(h\circ A)^{\uparrow} = h\circ B$ and $(h\circ B)^{\downarrow} = h\circ A$. Thus, $\langle h\circ A, h\circ B\rangle \in \mathscr{B}(X,Y,h\circ I)$.

Let $\langle A_j,B_j\rangle \in \mathscr{B}(X,Y,I)$, $j\in J$. Since h is a complete homomorphism, then for each $x\in X$ we have $(h\circ \bigcap_{j\in J} A_j)(x) = h((\bigcap_{j\in J} A_j)(x)) = h(\bigwedge_{j\in J} A_j(x)) = \bigwedge_{j\in J} h(A_j(x)) = (\bigcap_{j\in J} h\circ A_j)(x)$, showing $h\circ \bigcap_{j\in J} A_j = \bigcap_{j\in J} h\circ A_j$. Now, by (7), the extent of the formal concept $h^{\mathscr{B}(X,Y,I)}(\bigwedge_{j\in J}\langle A_j,B_j\rangle)$ is equal to $h\circ\bigcap_{j\in J} A_j = \bigcap_{j\in J} h\circ A_j$, which is equal to the extent of the formal concept $\bigwedge_{j\in J}\langle h\circ A_j, h\circ B_j\rangle = \bigwedge_{j\in J} h^{\mathscr{B}(X,Y,I)}(\langle A_j,B_j\rangle)$, concluding $h^{\mathscr{B}(X,Y,I)}$ preserves arbitrary infima.

To show $h^{\mathscr{B}(X,Y,I)}$ preserves arbitrary suprema we proceed similarly and use (8).

We have

$$\left(h^{\mathscr{B}(X,Y,I)}(\langle A_1,B_1\rangle) \preceq h^{\mathscr{B}(X,Y,I)}(\langle A_2,B_2\rangle)\right) = (\langle h\circ A_1, h\circ B_1\rangle \preceq \langle h\circ A_2, h\circ B_2\rangle)$$
$$= S(h\circ A_1, h\circ A_2) = \bigwedge_{x\in X} h(A_1(x)) \to h(A_2(x))$$
$$= h\left(\bigwedge_{x\in X} A_1(x) \to A_2(x)\right) = h(S(A_1,A_2)) = h(\langle A_1,B_1\rangle \preceq \langle A_2,B_2\rangle),$$

proving (11).

Let h be injective and suppose $h^{\mathscr{B}(X,Y,I)}(\langle A_1,B_1\rangle) = h^{\mathscr{B}(X,Y,I)}(\langle A_2,B_2\rangle)$. We have $h\circ A_1 = h\circ A_2$, whence $A_1 = A_2$ and, consequently, $\langle A_1,B_1\rangle = \langle A_2,B_2\rangle$, proving $h^{\mathscr{B}(X,Y,I)}$ is injective.

Suppose h is surjective and take $\langle A',B'\rangle \in \mathscr{B}(X,Y,h\circ I)$. Let $B_0 \in \mathbf{L}^Y$ be such that $h\circ B_0 = B'$. Then, by Lemma 1, $A' = B'^{\downarrow} = (h\circ B_0)^{\downarrow} = h\circ B_0^{\downarrow}$ and $B' = A'^{\uparrow} = h\circ B_0^{\downarrow\uparrow}$, showing $h^{\mathscr{B}(X,Y,I)}(\langle B_0^{\downarrow}, B_0^{\downarrow\uparrow}\rangle) = \langle A',B'\rangle$ and proving surjectivity of $h^{\mathscr{B}(X,Y,I)}$.

Theorem 3. *Let* $\mathbf{L}$ *be isomorphic to the direct product* $\mathbf{L}_1\times\mathbf{L}_2$*,* $p_1 : L \to L_1$ *and* $p_2 : L \to L_2$ *be the respective projections. Then* $\mathscr{B}(X,Y,I)$ *is isomorphic to the direct product* $\mathscr{B}(X,Y,p_1\circ I)\times\mathscr{B}(X,Y,p_2\circ I)$ *and the mappings* $p_1^{\mathscr{B}(X,Y,I)}$ *and* $p_2^{\mathscr{B}(X,Y,I)}$ *correspond to the respective Cartesian projections.*

Proof. Let a mapping $h : \mathscr{B}(X,Y,I) \to \mathscr{B}(X,Y,p_1\circ I)\times\mathscr{B}(X,Y,p_2\circ I)$ be defined by $h = \langle p_1^{\mathscr{B}(X,Y,I)}, p_2^{\mathscr{B}(X,Y,I)}\rangle$. By Theorem 2, h is a homomorphism of lattices.

The mapping $f : L \to L_1\times L_2$, $f = \langle p_1,p_2\rangle$, is a complete isomorphism of complete residuated lattices. For each $\langle A_1,B_1\rangle \in \mathscr{B}(X,Y,p_1\circ I)$ and $\langle A_2,B_2\rangle \in \mathscr{B}(X,Y,p_2\circ I)$ set $A(x) = f^{-1}(A_1(x),A_2(x))$, $B(y) = f^{-1}(B_1(y),B_2(y))$. The mapping $\langle\langle A_1,B_1\rangle,\langle A_2,B_2\rangle\rangle \mapsto \langle A,B\rangle$ is clearly the inverse mapping of h.

6 Concept Lattices of Incomplete Contexts

Let $\mathbf{L}$ be a Boolean algebra with variables, V the set of admissible assignments, $\langle X,Y,I\rangle$ an incomplete $\mathbf{L}$-context. Using the results of the previous section, we can construct a concept lattice of $\langle X,Y,I\rangle$. Our first attempt is straightforward: we simply use the theory of $\mathbf{L}$-concept lattices and construct the lattice $\mathscr{B}(X,Y,I)$. The main result on the structure of $\mathscr{B}(X,Y,I)$ is given by the following theorem.

Theorem 4. $\mathscr{B}(X,Y,I)$ *is isomorphic to the direct product* $\prod_{v\in V}\mathscr{B}(X,Y,\bar{v}\circ I)$*. The mappings* $\bar{v}^{\mathscr{B}(X,Y,I)} : \mathscr{B}(X,Y,I) \to \mathscr{B}(X,Y,\bar{v}\circ I)$ *correspond to the respective Cartesian projections.*

Proof. Follows from the fact that $\mathbf{L}$ is isomorphic to $\mathbf{2}^V$ (Theorem 1) and from Theorem 3.

From the above theorem it follows that $\mathscr{B}(X,Y,I)$ is quite large; its size depends exponentially on the number of admissible assignments, which again depends exponentially on the number of variables.

Example 4. The **L**-concept lattice $\mathscr{B}(X,Y,I)$ of the incomplete context from Ex. 3 is isomorphic to the direct product $\mathscr{B}(X,Y,\bar{v}_1 \circ I) \times \mathscr{B}(X,Y,\bar{v}_2 \circ I) \times \mathscr{B}(X,Y,\bar{v}_3 \circ I)$ and has $8 \cdot 8 \cdot 7 = 448$ elements. The individual components of the product are depicted in Fig. 3.

The following theorem shows that instead of the whole **L**-concept lattice $\mathscr{B}(X,Y,I)$ we can use the lattice of crisply generated concepts.

Theorem 5. *For each $v \in V$, the restriction $\bar{v}^{\mathscr{B}_c(X,Y,I)} : \mathscr{B}_c(X,Y,I) \to \mathscr{B}(X,Y,v \circ I)$ of $\bar{v}^{\mathscr{B}(X,Y,I)}$ is a surjective, $\bigwedge$-preserving mapping, such that for each $\langle A_1,B_1\rangle, \langle A_2,B_2\rangle \in \mathscr{B}_c(X,Y,I)$ it holds $\bar{v}^{\mathscr{B}_c(X,Y,I)}(\langle A_1,B_1\rangle) \leq \bar{v}^{\mathscr{B}_c(X,Y,I)}(\langle A_2,B_2\rangle)$ iff $\bar{v}(\langle A_1,B_1\rangle \preceq \langle A_2,B_2\rangle)$ $= 1$.*

Proof. Let $\langle A_0,B_0\rangle \in \mathscr{B}(X,Y,v \circ I)$, $B \in \mathbf{L}^Y$ be the crisp **L**-set in Y such that $v \circ B = B_0$ (if $B_0(y) = 0_\mathbf{2}$ then $B(y) = 0_\mathbf{L}$ and if $B_0(y) = 1_\mathbf{2}$ then $B(y) = 1_\mathbf{L}$). We have $\langle B^\downarrow, B^{\downarrow\uparrow}\rangle \in \mathscr{B}_c(X,Y,I)$ and by Lemma 1, $\bar{v}^{\mathscr{B}_c(X,Y,I)}(\langle B^\downarrow, B^{\downarrow\uparrow}\rangle) = \langle v \circ B^\downarrow, v \circ B^{\downarrow\uparrow}\rangle = \langle (v \circ B)^\downarrow, (v \circ B)^{\downarrow\uparrow}\rangle = \langle A_0,B_0\rangle$.

Since $\mathscr{B}_c(X,Y,I)$ is a $\bigwedge$-semilattice of $\mathscr{B}(X,Y,I)$, then $\bar{v}^{\mathscr{B}_c(X,Y,I)}$ preserves infima. The last assertion follows directly from (11).

7 An Illustrative Example

In the previous section, we showed how to construct a concept lattice $\mathscr{B}_c(X,Y,I)$ of an incomplete **L**-context. Let **L** be a Boolean algebra with variables, $\langle X,Y,I\rangle$ and incomplete **L**-context. Extents and intents of concepts in the lattice $\mathscr{B}_c(X,Y,I)$ are **L**-sets, which can be interpreted as classical sets, whose elements depend on values of the variables (see Example 2). We proved that any choice of values of the variables (by an admissible assignment) transforms the lattice to an ordinary concept lattice of the corresponding completed context. Thus, the lattice contains the information on all the concept lattices, which can be obtained by admissible completions on the incomplete context. Moreover, the information can be easily obtained from the lattice by a simple assignment to the variables. Further, the lattice also contains the information on the ordering of the concepts, which can be again easily computed from any admissible assignment. We illustrate these results in the following example.

Consider the incomplete **L**-context $\langle X,Y,I\rangle$ from Ex. 3. The lattice $\mathscr{B}_c(X,Y,I)$ can be constructed either using an algorithm for lattices of crisply generated concepts (i.e., from [3]), or using a transformation of $\langle X,Y,I\rangle$ to an ordinary formal context and computing its concept lattice (see [4]; in our case, the transformation is based on replacing each row in $\langle X,Y,I\rangle$ with all its admissible completions). Note that in the latter case, an inverse transformation of the lattice would be required to obtain all the information on the concepts (i.e., extents, intents, and their dependence on the values of the variables).

In our case, the lattice $\mathscr{B}_c(X,Y,I)$ has the following 11 elements:

$$\langle\{x_1,x_2,x_3,x_4\},\emptyset\rangle, \langle\{x_1,x_2\},\{{}^{u_2}/y_3,y_4\}\rangle, \langle\{x_1,{}^{u_2}/x_2,x_3\},\{y_3\}\rangle,$$
$$\langle\{x_2,x_3,x_4\},\{y_2\}\rangle, \langle\{x_1,{}^{u_2}/x_2\},\{y_3,y_4\}\rangle, \langle\{x_2\},\{{}^{u_1}/y_1,y_2,{}^{u_2}/y_3,y_4\}\rangle,$$
$$\langle\{{}^{u_2}/x_2,x_3\},\{{}^{u_1\vee u_2'}/y_1,y_2,y_3\}\rangle, \langle\{{}^{u_2}/x_2\},\{{}^{u_1\vee u_2'}/y_1,y_2,y_3,y_4,{}^{u_2'}/y_5\}\rangle,$$

$$\langle\{^{u_1}/x_2,x_3\},\{y_1,y_2,y_3\}\rangle,\langle\{^{u_1}/x_2\},\{y_1,y_2,y_3,y_4,{}^{u_1'}/y_5\}\rangle,\langle\emptyset,\{y_1,y_2,y_3,y_4,y_5\}\rangle.$$

From Theorem 5 it follows that for an admissible assignment v and a concept $c \in \mathscr{B}_c(X,Y,I)$, the corresponding formal concept $\bar{v}^{\mathscr{B}_c(X,Y,I)}(c)$ of the concept lattice $\mathscr{B}(X,Y,v\circ I)$ of the completed context can be computed simply by assigning appropriate values to the variables (see Ex. 2).

For example, let $\langle A,B\rangle = \langle\{^{u_2}/x_2,x_3\},\{^{u_1\vee u_2'}/y_1,y_2,y_3\}\rangle$, $v = v_2$. Since $v_2(u_1) = 0$ and $v_2(u_2) = 1$, then

$$\begin{aligned}\bar{v}_2^{\mathscr{B}(X,Y,I)}(\langle A,B\rangle) &= \bar{v}_2^{\mathscr{B}(X,Y,I)}(\langle\{^{u_2}/x_2,x_3\},\{^{u_1\vee u_2'}/y_1,y_2,y_3\}\rangle)\\ &= \langle\{^1/x_2,x_3\},\{^0/y_1,y_2,y_3\}\rangle = \langle\{x_2,x_3\},\{y_2,y_3\}\rangle,\end{aligned}$$

which is a formal concept of $\langle X,Y,v_2\circ I\rangle$. This way, each $\langle A_0,B_0\rangle \in \mathscr{B}(X,Y,\bar{v}_2\circ I)$ can be obtained as the image of a concept $\langle A,B\rangle \in \mathscr{B}(X,Y,I)$ in the mapping $\bar{v}_2^{\mathscr{B}(X,Y,I)}$.

Moreover, using Theorem 5 we can easily compute the ordering in $\mathscr{B}(X,Y,v\circ I)$. For example, for the concepts $c_1 = \langle\{^{u_2}/x_2,x_3\},\{^{u_1\vee u_2'}/y_1,y_2,y_3\}\rangle$, $c_2 = \langle\{^{u_1}/x_2,x_3\},\{y_1,y_2,y_3\}\rangle$ we have by (9), (2), $c_1 \preceq c_2 = u_1 \vee u_2'$, whence $\bar{v}^{\mathscr{B}_c(X,Y,I)}(c_1) \le \bar{v}^{\mathscr{B}_c(X,Y,I)}(c_2)$ iff $v(u_1)\vee v(u_2)' = 1$. This condition is satisfied for $v = v_1$ and $v = v_3$.

The lattice $\mathscr{B}_c(X,Y,I)$ is depicted in Fig. 4. We use labeling of elements of the lattice based on the main theorem for $\mathscr{B}_c(X,Y,I)$ [3, Theorem 5]. For any $x\in X$, $y \in Y$, let $\langle A_1,B_1\rangle$ be the formal concept labeled by $^a/x$ (for $a = 1$ we use labels x instead of $^1/x$) and $\langle A_2,B_2\rangle$ be the formal concept labeled by y. Then $I(x,y) \ge a$ iff $\langle A_1,B_1\rangle \le \langle A_2,B_2\rangle$. Also, for each formal concept $\langle A,B\rangle$ we have $A(x) \ge a$ and $B(y) = 1$ if and only if $\langle A_1,B_1\rangle \le \langle A,B\rangle$ and $\langle A_2,B_2\rangle \ge \langle A,B\rangle$. This way we can use the labeling to determine the extent and the crisp part of the intent for each formal concept in $\mathscr{B}_c(X,Y,I)$.

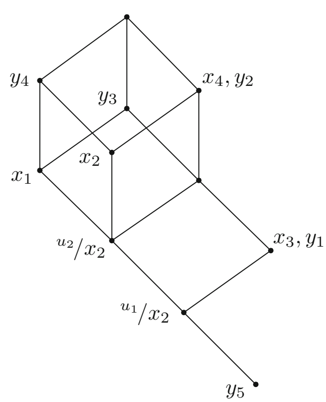


Fig. 4. Lattice of crisply generated concepts from Sec. 7

The concept lattice $\mathscr{B}(X,Y,v\circ I)$ is in a bijection with the factor set of $\mathscr{B}_c(X,Y,I)$ by the equivalence, induced by the mapping $\bar{v}$. In Fig. 5 we can see the classes of this equivalence for each of the admissible assignments v_1,v_2,v_3.

Since the mapping $\bar{v}$ does not change crisp parts of intents (see the proof of Theorem 5), it is possible to obtain the factor sets directly from the labeled diagram by replacing each variable u vith the value $v(u)$ and joining repetitive concepts.

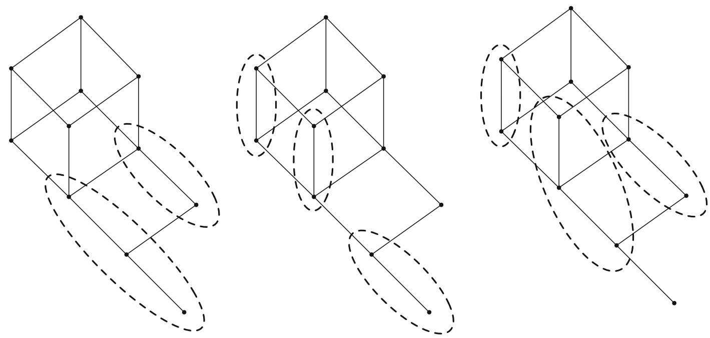


Fig. 5. Factorizations of the lattice of crisply generated concepts from Sec. 7 by means of admissible assignments (from left to right) v_1, v_2, v_3.

8 Experiments

An important question is that of the size of the lattice $\mathscr{B}_c(X,Y,I)$. As the next example shows, in the worst case the size depends exponentially on the number of variables.

Example 5. Let variables $u_1,\dots,u_k$ be independent, $\langle X,Y,I\rangle$ be an incomplete context such that $X=\{x\}$, $Y=\{y_1,\dots,y_n\}$ for $n \geq k$, and the **L**-relation I be given by $I(x,y_j)=u_j$ for $j \leq k$ and $I(x,y_j)=0$ otherwise. Then from the proof of Theorem 5, for each assignment $v\colon U \to \mathbf{2}$, the set $B=\{y_j \mid j \leq k, v(u_j)=1\}$ is an intent of $\langle X,Y,I\rangle$. On the other hand, the concept lattice $\mathscr{B}(X,Y,v\circ I)$ has always at most 2 elements.

The following experiments indicate that for real-world data the situation is more optimistic. It seems that if the number of variables is not large, the size of the concept lattice $\mathscr{B}_c(X,Y,I)$ is not substantially larger than the size of a concept lattice of a similar completed context.

We demonstrate our observation on two classical example contexts, namely the Digits Context and the Tea Ladies Context. The data can be viewed at

`http://www.upriss.org.uk/fca/examples.html`.

Our experiment proceeded as follows. For each $k \in \{1,2,\dots,10\}$ (for the Digits Context) or $k \in \{1,2,\dots,7\}$ (for the Tea Ladies Context) we set **L** to be the Boolean algebra with the set of independent variables $U=\{u_1,\dots,u_k\}$. Since the variables are independent, then the set V of all admissible assignments is equal to 2^U and **L** is isomorphic to $\mathbf{2}^V$. Then we randomly selected k different pairs $\langle x_j,y_j\rangle \in X\times Y$, $j=1,\dots,k$, and computed the number of concepts in the lattice $\mathscr{B}_c(X,Y,\bar{I})$, for an incidence relation with values at positions $\langle x_j,y_j\rangle$ replaced with variables:

$$\bar{I}(x,y)=\begin{cases} u_j & \text{if } \langle x,y\rangle=\langle x_j,y_j\rangle, \\ I(x,y) & \text{otherwise.}\end{cases}$$

We performed this computation 100 times, each time for a different random selection of pairs $\langle x_j, y_j \rangle$. For the resulting values, we computed the arithmetic mean and standard deviation.

The results are summarized in Tbl. 1 and Tbl. 2.

Table 1. Results of experiment for the Digits Context. The number of variables is $k \in \{1, \ldots, 10\}$. For each k we performed 100 measurements. In each measurement, we put k variables at random places in the context, and computed the size of the lattice $\mathscr{B}(X,Y,\bar{I})$. From these 100 measurements we computed the average value and standard deviation. The size of the original concept lattice of the Digits Context is 48.

No. of variables	1	2	3	4	5	6	7	8	9	10
Avg. no. of concepts	50.31	54.31	56.26	59.38	63.49	65.84	68.21	71.01	73.05	77.39
Std. deviation	3.16	6.34	6.51	8.00	8.19	8.67	10.37	11.71	11.31	11.67

Table 2. Results of experiment for the Tea Ladies Context. The number of variables is $k \in \{1, \ldots, 7\}$. For each k we performed 100 measurements. In each measurement, we put k variables at random places in the context, and computed the size of the lattice $\mathscr{B}(X,Y,\bar{I})$. From these 100 measurements we computed the average value and standard deviation. The size of the original concept lattice of the Tea Ladies Context is 65.

No. of variables	1	2	3	4	5	6	7
Avg. no. of concepts	67.88	71.54	74.73	78.3	82.12	86.09	89.67
Std. deviation	2.56	3.79	4.66	5.0	7.53	7.02	8.44

9 Conclusion and Future Research

The experiments in the previous sections indicate that our approach to concept lattices of incomplete contexts could find practical applications. Our future research will concentrate on the folowing topics:

- developing a theory of attribute implications in our framework and comparing the results with the results of Burmeister and Holzer [5], and Obiedkov [8],
- generalizing the theory to FCA in fuzzy setting,
- more experiments and theoretical results on the size of concept lattices of incomplete contexts.

References

1. Belohlavek, R.: Concept lattices and order in fuzzy logic. Ann. Pure Appl. Log. 128(1-3), 277–298 (2004)
2. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic Publishers, Norwell (2002)

3. Bělohlávek, R., Sklenář, V., Zacpal, J.: Crisply Generated Fuzzy Concepts. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 269–284. Springer, Heidelberg (2005)
4. Belohlavek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: Proceedings of FUZZ-IEEE 2005: The 14th IEEE International Conference on Fuzzy Systems, pp. 663–668 (2005)
5. Burmeister, P., Holzer, R.: On the Treatment of Incomplete Knowledge in Formal Concept Analysis. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 385–398. Springer, Heidelberg (2000)
6. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer (1999)
7. Gratzer, G.A.: General lattice theory. Academic Press, New York (1978)
8. Obiedkov, S.: Modal Logic for Evaluating Formulas in Incomplete Contexts. In: Priss, U., Corbett, D., Angelova, G. (eds.) ICCS 2002. LNCS (LNAI), vol. 2393, pp. 314–325. Springer, Heidelberg (2002)
9. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, Boston, pp. 445–470 (1982)

Formal Concept Analysis as a Framework for Business Intelligence Technologies

Juraj Macko

Dept. Computer Science, Palacky University,
Olomouc 17. listopadu 12,
CZ-77146 Olomouc Czech Republic
juraj.macko@upol.cz

Abstract. Numerical datasets in data mining are handled using various methods. In this paper, data mining of numerical data using FCA in combination with some interesting ideas from OLAP technology is proposed. This novel method is an enhancement of FCA, in which measures are assigned to objects and/or attributes and then various numeric operations are applied to these measures (e.g. summarization, aggregation functions etc.). This new approach results in a structure, which is a concept lattice and where the extent and/or intent have aggregated values assigned to them. This structure could be seen as a generalization of OLAP technology. A concept lattice can be constrained by using various closure operators. The new closure operators presented here are based on values with very clear meaning for the user. Finally, a fuzzy OLAP formalization based on FCA in a fuzzy setting and using measures is proposed. Examples are shown for each introduced topic.

Keywords: formal concept analysis, OLAP, fuzzy logic, data mining.

1 Introduction

In Preliminaries the fundamentals of FCA in crisp and in fuzzy settings is described. The *OLAP cube* is introduced and its formal definition is redesigned especially for purposes of this paper. Finally basic notions of classical measures and aggregation operators are shown. In Formal Concept Analysis with Measures, FCA using measures is introduced and a concept lattice with extent (or intent) values is proposed, which results from using aggregations on measures assigned to objects (or attributes respectively). Finally, Applications of Formal Concept Analysis with Measures, consists of applications, in which an attempt to show the contribution of the proposed method is made. Concept lattice with values can be seen as a generalization of OLAP technology. The already developed constraints via the closure operators for concept lattice are enhanced. These are based on values. The concept lattices using values are generalized to fuzzy settings (L-concepts with values) and novel approach to fuzzy *OLAP cube* (L-cube) is shown. This paper is supplemented with comprehensive examples. The final part, summarizes the results and shows a new perspectives in the research.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 195–210, 2012.

2 Preliminaries

Just the fundamentals and basic definitions for the different topics are shown. For all other details, refer to the particular resources. Among important preliminaries can be found Formal Concept Analysis (FCA), FCA in Fuzzy Settings, Online Analytical Processing (OLAP) technology followed by some mathematic foundations, namely classical measure and aggregation operators.

2.1 Formal Concept Analysis in Crisp and Fuzzy Settings

The reader should be familiar with the **Formal Concept Analysis (FCA) in crisp settings**, so only the basic notion will be mentioned. The formal context is denoted by $\langle X, Y, I \rangle$ where $I \subseteq X \times Y$. The formal concept of the formal context $\langle X, Y, I \rangle$ is denoted by $\langle A, B \rangle$, where $A \subseteq X$ and $B \subseteq Y$. $\langle A, B \rangle$ is a formal concept iff $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The concept forming operators $()^{\uparrow}$ and $()^{\downarrow}$ are defined as

$$A^{\uparrow} = \{y \in Y \mid \text{ for each } x \in X : \langle x, y \rangle \in I\},$$
$$B^{\downarrow} = \{x \in X \mid \text{ for each } y \in Y : \langle x, y \rangle \in I\}.$$

The set of all formal concepts of $\langle X, Y, I \rangle$ is denoted by $\mathcal{B}(X, Y, I)$ and equipped with a partial order $\leq$ forms the concept lattice of $\langle X, Y, I \rangle$.
FCA in fuzzy settings uses fuzzy logic. There are a couple of approaches of how to use fuzzy logic in FCA. The approach and notation of Belohlavek will be used. An overview of all approaches including notation can be found in [20]. As an algebra for fuzzy logic the complete residuated lattice $\langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$ will be used, where $\langle L, \wedge, \vee, 0, 1 \rangle$ is a complete lattice with 0 as the least, 1 as the greatest element and $\langle L, \otimes, 1 \rangle$ is a commutative monoid ($\otimes$ is commutative and associative. 1 is a neutral element.). The operator $\otimes$ denotes the truth function of the fuzzy conjunction, the operator $\rightarrow$ denotes the truth function of the fuzzy implication and both operators satisfy the adjointness property: $a \otimes b \leq c$ iff $a \leq b \rightarrow c$ for each element $a, b, c \in L$. Such elements are called truth degrees. Fuzzy set A is defined as $A \in L^U$, where U is the universe and L^U is a collection of all fuzzy sets in the universe U. Fuzzy context (L-context) is defined as $\langle X, Y, I \rangle$, where X and Y are a set of objects and attributes and I is the fuzzy relation (L-relation) defined as $X \times Y \rightarrow L$. $I(x, y)$ means the truth degree of "object x has attribute y". For fuzzy sets $A \in L^X$ and $B \in L^Y$, $A^{\uparrow}(y)$ and $B^{\downarrow}(x)$ are defined as

$$A^{\uparrow}(y) = \bigwedge_{x \in X} (A(x) \rightarrow I(x, y))$$
$$B^{\downarrow}(x) = \bigwedge_{y \in Y} (B(y) \rightarrow I(x, y)).$$

and the fuzzy concept (L-concept) is defined as $\langle A, B \rangle$ such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The collection of all L-concepts equiped with $\leq$ is denoted as $\mathcal{B}(X, Y, I)$ and called an L-lattice. When $L = \{0, 1\}$ the result is classical logic and FCA in crisp settings. For further details see [20].

2.2 Classical Measure [4] and Aggregation Operators [11]

Function $\mu : \mathcal{C} \to \mathbb{R}^+$, where $\mathcal{C} \subseteq 2^X$ is a **classical measure** iff μ is **1. Additive**: iff $\mu(A \cup B) = \mu(A) + \mu(B)$, where $A, B \in \mathcal{C}$, $(A \cup B) \in \mathcal{C}$ and $(A \cap B) = \emptyset$ and iff μ is **2. Monotone**: iff $A, B \in \mathcal{C}$ and $A \subset B$ imply $\mu(A) \leq \mu(B)$. Only a classical measure will be used in this paper (with respect to summarization operator Σ). **Aggregation operator** Θ is a function which assigns a real number y to any n-tuple $(x_1, x_2, ..., x_n)$ of real numbers. Formally:
$\Theta : \bigcup\limits_{n \in N} [0,1]^n \to [0,1]$ which satisfies:

1. $\Theta(x) = x$ - identity when unary
2. $\Theta(0, ..., 0) = 0$ and $\Theta(1, ..., 1) = 1$ - boundary conditions
3. $\Theta(x_1, ..., x_n) \leq \Theta(y_1, ..., y_n)$ if $(x_1, ..., x_n) \leq (y_1, ..., y_n)$ - non decreasing monotonicity

Definition 1 (Ordered Weighted Averaging Operators (OWA)). *$OWA(x_1, ..., x_n) = \sum\limits_{j=1}^{n} w_j x_{\sigma(j)}$ where σ reorders elements of X in following manner $x_{\sigma(1)} \leq x_{\sigma(2)} \leq ... \leq x_{\sigma(n)}$, $w_j \geq 0$ and $\sum\limits_{i=1}^{n} w_i = 1$.*

Particular operators can be defined on OWA with the following conditions:

1. **minimum** $min(X)$: $w_1 = 1$, $w_i = 0$ if $i \neq 1$
2. **maximum** $max(X)$: $w_n = 1$, $w_i = 0$ if $i \neq 1$
3. **median** $med(X)$: $w_{\frac{n+1}{2}} = 1$ if n is odd, $w_{\frac{n}{2}} = \frac{1}{2}$ and $w_{\frac{n}{2}+1} = \frac{1}{2}$ if n is even, $w_i = 0$ in other case
4. **arithmetic mean** $avg(X)$: $w_i = \frac{1}{n}$ for $\forall i$

2.3 On Line Analytical Processing (OLAP)

OLAP is a well known multidimensional data analysis method introduced by E.F. Codd in 1993 (see [2] for details) and is widely spread even among the non-technical users. Typical example of OLAP are the so called "Pivot tables" and "Pivot charts" in MS Office Excel. Mathematical formalization of OLAP have been done by several authors. Although inspiring [8], we used our own formalization in this paper in respect to its use in fuzzy settings. The Database table is formally defined in [5] as the relation r on the relation scheme $R = \{A_1, A_2 \ldots, A_n\}$ as a set of mappings $\{t_1, t_2 \ldots, t_m\}$ from R to $\mathcal{D}$ where $\mathcal{D}$ is a set of all D - domains of attributes A. Note that n is the number of the columns and m the number of the rows in the database table. For the purposes of OLAP definition the sets of domains in $\mathcal{D}$ are divided into two groups: $H_k \in \mathcal{H}$- dimensions and $M_s \in \mathcal{M}$ - measures, where $k \in [1; |\mathcal{H}|]$, $s \in [1; |\mathcal{M}|]$ and $M_s \subseteq \mathbb{R}^+$). The space for the *OLAP cube* could be predefined as a cartesian product

$$C = L^{H_1} \times \cdots \times L^{H_k} \times \cdots \times L^{H_{|\mathcal{H}|}}$$

where $L = \{0, 1\}$. The *OLAP cube* is mapping $\sigma : C \to \mathbb{R}^+$ and is defined as

$$\sigma(h_1, \ldots, h_n) = \bigodot_{i=1}^{m} t_i(M_s) \text{ such that } \{t_i(A_j)\} \supseteq h_j \text{ for all } j \in [1; |\mathcal{H}|]$$

where the symbol $\odot$ stands for the sum operator Σ, the cardinality operator $||$ or the arbitrary aggregation operator Θ and $|\mathcal{H}|$ is the number of OLAP cube dimensions. In Figure 1, part (i) the database table (database relation r) is shown, where attributes $A_1 = \{TradeMark\}$, $A_2 = \{Country\}$ and $A_3 = \{Price\}$ have theirs domains $H_1 = \{BMW, SKODA, FIAT\}$, $H_2 = \{Germany, France\}$ and $M_1 \subseteq \mathbb{R}^+$. Part (ii) shows the 2-dimensional OLAP cube, where $\odot = \Sigma$. Elements of L^{H_k} can be described in the form of $l \in L$ tuples $\langle l_1 \in L, \ldots, l_{|H_k|} \in L\rangle_{H_k}$. E.g. tuple $\langle 1, 0, 0\rangle_{TradeMark}$ represents the car where the $TradeMark = BMW$. Each "cell" in the *OLAP cube* is determined by exactly one tuple from each dimension. The data in Table (ii) in Figure 1 can be interpreted as follows: the cell, which is determined by tuple $\langle 1, 0, 0\rangle_{TradeMark}$ from dimension H_1 and tuple $\langle 0, 1\rangle_{Coutry}$ from dimension H_2, represents the total $Price$ of all BMW cars sold in $France$. Tuple $\langle 0, 0, 0\rangle_{TradeMark}$ has a special meaning. It represents all cars, regardless of the $TradeMark$. The combination of tuples $\langle 0, 0, 0\rangle_{TradeMark}$ and $\langle 0, 0\rangle_{Country}$ represents the total $Price$ of all cars regardless of the $TradeMark$ and $Country$, and thus all cars. By detailed investigation of this table, it is clear that some rows and columns are not defined ("n/a" stands for "not available"). The reason is, that all values from any dimension domain H_k are mutually exclusive, e.g. the $TradeMark$ of the car can be either BMW or $SKODA$, never both. Technically we can put 0 instead of "n/a", because total the $Price$ of such cars can be seen as 0, but for a better illustration "n/a" is used. For this reason in the OLAP cube only these cells are used, which represents either tuple $\langle 0, ..., 0\rangle$ or tuples, where the value $1 \in L$ appears exactly once. All others are useless in the crisp setting of the *OLAP cube* (namely where $L = \{0, 1\}$), but will be used later in the fuzzy setting. The size of the OLAP cube in the crisp setting will be $(|H_1| + 1) \times \cdots \times (|H_{|\mathcal{H}|}| + 1)$.

H_1 TradeMark	H_2 Country	M_1 Price in 000 EUR
BMW	Germany	30
BMW	France	35
SKODA	Germany	20
SKODA	France	25
FIAT	Germany	10
FIAT	France	13

(i) Input data table

			Germany	0	1	0	1
BMW	SKODA	FIAT	France	0	0	1	1
0	0	0		133	73	60	n/a
0	0	1		23	13	10	n/a
0	1	0		45	25	20	n/a
0	1	1		n/a	n/a	n/a	n/a
1	0	0		65	35	30	n/a
1	0	1		n/a	n/a	n/a	n/a
1	1	0		n/a	n/a	n/a	n/a
1	1	1		n/a	n/a	n/a	n/a

(ii) 2-dimensional OLAP Cube using operation Σ

Fig. 1. An example of the OLAP technology (known as "Pivot table" in MS Excel)

3 Formal Concept Analysis with Measures

Formal concept analysis with measures tries to analyze the numerical data. The basic idea is that by assigning the measure to an object and/or attribute, computing formal concepts and aggregating the measures will result in the valuation of the concept. A similar approach was used by the author in [12], where the authors selected important concepts by assigning weight to attributes and computing the value of the concept. The weights were assigned to the attributes artificially, just to get important concepts on output and this numerical approach was compared to other relational approaches. However, FCA with measures goes much deeper. Moreover measures, that were natural, not artificial with respect to the object were assigned. Examples of such naturally assigned measures were the *Price of the Car* or the *Mass of the Planet*. In this chapter the theory is introduced and a comprehensive example is given.

3.1 Formal Concept Analysis with Measures - Theory

Definition 2 (Measure of Object and Attribute). *A Measure of the object is mapping* $\Phi : X \to \mathbb{R}^+$ *and a Measure of the attribute is mapping* $\Psi : Y \to \mathbb{R}^+$.

Definition 3 (Value of Extent and Intent). *The Value of extent is mapping* $v : A_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$ *defined as* $v(A) = \bigodot_{x \in A} \Phi(x)$, *where* $\odot$ *is either the symbol for the sum* Σ *(the "sum" operation) or the symbol for cardinality* $|A|$ *or the arbitrary aggregation function* Θ. *A is an extent of the formal concept* $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$. *Similarly, the value of the intent is mapping* $w : B_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$ *defined as* $w(B) = \bigodot_{y \in B} \Psi(y)$, *where B is an intent of the formal concept* $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$

One more property of aggregation operators must be defined before proceeding:

Definition 4 (Additive Monotonicity and Antitonicity).
$A \subset B$ *imply* $\odot(A) \leq \odot(B)$ *- is an additive monotonicity of* $\odot$ *(denote as* $\odot_{\uparrow}$*)*
$A \subset B$ *imply* $\odot(A) \geq \odot(B)$ *- is an additive antitonicity of* $\odot$ *(denote as* $\odot_{\downarrow}$*)*

In other words, when we add the number to the aggregation and the value of the aggregation will at least increase, the operator is additively monotone, (if the value at least decrease, the operator is additively antitone). It is evident, that a cardinality operator $|A|$ is additively monotone. Operator Σ is additively monotone too, because it is a property of classical measure. With OWA operators, the situation can differ. For additive monotonicity (antitonicity) only the term monotonicity (antitonicity) is used, even though the term "monotone" is used in the definition of the aggregation operators in a different meaning.

Proposition 1 (max(A)). OWA *operator* $max(A)$ *is monotone.*

Proof. $max(A) = \sum_{j=1}^{n} w_j x_{\sigma(j)}$ and $w_n = 1$, $w_i = 0$ if $i \neq 1$, which means $max(A) = x_{\sigma(n)}$ For each $x \in B$ and $x \notin A$ we can have either $x > x_{\sigma(n)}$

and in this case will be $max(A \cup \{x\}) = x_{\sigma(n+1)} = x > max(A)$ or we can have $x \leq x_{\sigma(n)}$ and in this case it will be $max(A \cup \{x\}) = x_{\sigma(n)} = max(A)$, which is exactly $A \subset (A \cup \{x\}) \Rightarrow max(A) \leq max(A \subset (A \cup \{x\}))$.

Proposition 2 (min(A)). *OWA operator $min(A)$ is antitone.*

Proof. $min(A) = \sum_{j=1}^{n} w_j x_{\sigma(j)}$ and $w_1 = 1, w_i = 0$ if $i \neq 1$, which means $min(A) = x_{\sigma(1)}$ For each $x \in B$ and $x \notin A$ we can have either $x < x_{\sigma(1)}$ and in this case will be $min(A \cup \{x\}) = x_{\sigma(1)} = x < min(A)$ or we can have $x \geq x_{\sigma(1)}$ and in this case it will be $min(A \cup \{x\}) = x_{\sigma(1)} = min(A)$, which is exactly $A \subset (A \cup \{x\}) \Rightarrow min(A) \geq min(A \subset (A \cup \{x\}))$.

Proposition 3 (arithmetic mean avg(A)). *The OWA operators arithmetic mean $avg(A)$ and $median(A)$ are neither monotone nor antitone.*

Proof. It is very easy to find such $x \in B$ and $x \notin A$ such that $avg(A \cup \{x\}) \geq avg(A)$ and another $x \in B$ and $x \notin A$ such that $avg(A \cup \{x\}) \leq avg(A)$. For $median(A)$ the proof is similar.

After a short analysis of the monotonicity (antitonicity) of some $\odot$ operators, properties of the extent and intent values within lattice ordering will be examined.

Proposition 4 (monotone aggregations in lattice).
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Rightarrow v(A_1) \geq v(A_2)$ *where* $v(A) = \odot_{\uparrow}(\Phi(A))$
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Rightarrow w(B_1) \leq w(B_2)$ *where* $w(B) = \odot_{\uparrow}(\Psi(B))$

Proof (monotone aggregations in lattice).
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Leftrightarrow A_1 \supseteq A_2 \Rightarrow \odot_{\uparrow}(\Phi(A_1)) \geq \odot_{\uparrow}(\Phi(A_2)) \Leftrightarrow v(A_1) \geq v(A_2)$
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Leftrightarrow B_1 \subseteq B_2 \Rightarrow \odot_{\uparrow}(\Psi(B_1)) \leq \odot_{\uparrow}(\Psi(B_2)) \Leftrightarrow w(B_1) \leq w(B_2)$

Proposition 5 (antitone aggregations in lattice).
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Rightarrow v(A_1) \leq v(A_2)$ *where* $v(A) = \odot_{\downarrow}(\Phi(A))$
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Rightarrow w(B_1) \geq w(B_2)$ *where* $w(B) = \odot_{\downarrow}(\Psi(B))$

Proof (antitone aggregations in lattice).
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Leftrightarrow A_1 \supseteq A_2 \Rightarrow \odot_{\downarrow}(\Phi(A_1)) \leq \odot_{\downarrow}(\Phi(A_2)) \Leftrightarrow v(A_1) \leq v(A_2)$
$\langle A_1, B_1 \rangle \geq \langle A_2, B_2 \rangle \Leftrightarrow B_1 \subseteq B_2 \Rightarrow \odot_{\downarrow}(\Psi(B_1)) \geq \odot_{\downarrow}(\Psi(B_2)) \Leftrightarrow w(B_1) \geq w(B_2)$

For aggregations, which are neither monotone nor antitone (e.g. $avg()$), nothing like the above mentioned propositions can be asserted.

3.2 FCA with Measures - Comprehensive Example

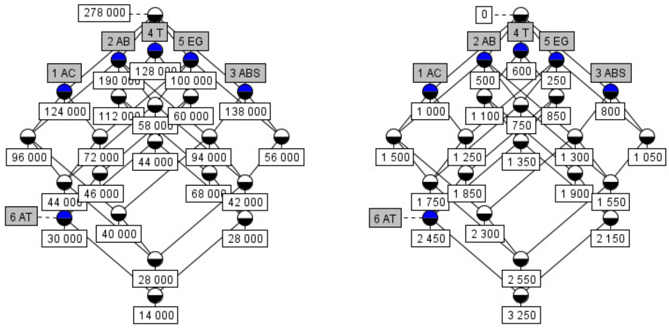
Let $X = \{Car1, \ldots, Car20\}$ be a set of cars and $Y = \{AC, AB, ABS, TMP, EG, AT\}$ be a set of components, namely Air Conditioning (AC), Airbag (AB), Antilock Braking System (ABS), Tempomat (TMP), Extra Guarantee (EG) and Automatic Transmission (AT). Table 1 represents the formal context

Table 1. The formal context of the cars, the additional components, the price of the car and the price of the component

	1. *AC*	2. *AB*	3. *ABS*	4. *TMP*	5. *EG*	6. *AT*	$\Phi(X)$ = Price in EUR
Car1	×	×					16 000
Car2		×	×	×			12 000
Car3		×	×	×	×		14 000
Car4	×			×	×		16 000
Car5	×				×		12 000
Car6	×	×	×				12 000
Car7		×	×	×			12 000
Car8			×				14 000
Car9							16 000
Car10		×					12 000
Car11		×		×			12 000
Car12	×	×	×	×	×	×	14 000
Car13		×		×			16 000
Car14	×	×		×	×	×	16 000
Car15			×		×		14 000
Car16	×	×					12 000
Car17	×	×					12 000
Car18		×	×	×			16 000
Car19			×				16 000
Car20	×	×	×		×		14 000
$\Psi(Y)$ = Price in EUR	1 000	500	800	600	250	100	

Table 2. The formal concepts with the extent value

	Extent Cars	Intent Components	Extent value Total Price
1	X - all cars	$\emptyset$	278 000
2	$\{2, 3, 4, 7, 11, 12, 13, 14, 18\}$	$\{TMP\}$	128 000
3	$\{3, 4, 5, 12, 14, 15, 20\}$	$\{EG\}$	100 000
4	$\{1, 4, 5, 6, 12, 14, 16, 17, 20\}$	$\{AC\}$	124 000
5	$\{1, 2, 3, 6, 7, 10, 11, 12, 13, 14, 16, 17, 18, 20\}$	$\{AB\}$	190 000
6	$\{2, 3, 6, 7, 8, 12, 15, 18, 19, 20\}$	$\{ABS\}$	138 000
7	$\{3, 4, 12, 14\}$	$\{EG, TMP\}$	60 000
8	$\{4, 5, 12, 14, 20\}$	$\{EG, AC\}$	72 000
9	$\{2, 3, 7, 11, 12, 13, 14, 18\}$	$\{AB, TMP\}$	112 000
10	$\{3, 12, 14, 20\}$	$\{AB, EG\}$	58 000
11	$\{3, 12, 15, 20\}$	$\{ABS, EG\}$	56 000
12	$\{1, 6, 12, 14, 16, 17, 20\}$	$\{AC, AB\}$	96 000
13	$\{2, 3, 6, 7, 12, 18, 20\}$	$\{AB, ABS\}$	94 000
14	$\{4, 12, 14\}$	$\{AC, TMP, EG\}$	46 000
15	$\{3, 12, 14\}$	$\{TMP, EG, AB\}$	44 000
16	$\{12, 14, 20\}$	$\{EG, AB, AC\}$	44 000
17	$\{2, 3, 7, 12, 18\}$	$\{AB, ABS, TMP\}$	68 000
18	$\{3, 12, 20\}$	$\{ABS, EG, AB\}$	42 000
19	$\{6, 12, 20\}$	$\{ABS, AC, AB\}$	40 000
20	$\{12, 14\}$	$\{AB, EG, AC, TMP, AT\}$	30 000
21	$\{3, 12\}$	$\{AB, EG, TMP, ABS\}$	28 000
22	$\{12, 20\}$	$\{AB, AC, EG, ABS\}$	28 000
23	$\{12\}$	Y - all components	14 000



(i) The concept lattice with the extent values

(ii) The concept lattice with the intent values

Fig. 2. Examples of the concept lattices with the values

$I \subseteq X \times Y$, which means that the "car has additional component". There are two different measures: measure $\Phi(X)$ which represents the *Price of Car* and measure $\Psi(Y)$ representing the *Price of Component*.

Formal context from Table 1 results to $\mathcal{B}(X, Y, I)$, which consists of 23 concepts (see Table 2). The example of the formal context is $\langle A_{20}, B_{20} \rangle = \langle \{Car12, Car14\}, \{AC, AB, TMP, EG, AT\} \rangle$. Let $\Phi(X)$ be the *Price of the Car*, namely $\Phi(Car12) = 14000$, $\Phi(Car14) = 16000$. Putting $\odot = \Sigma$, then $v(A_{20}) = \sum_{x \in A_{20}} \Phi(x)$ $= \Phi(Car12) + \Phi(Car14) = 14000 + 16000 = 30000$, which is the extent value of concept A_{20}. It can be interpreted as the "Total price of all cars with additional components AC, AB,TMP, EG and AT is 30 000 EUR". This could be interesting information for financial managers. The next step is to calculate the intent value. Let $\Psi()$ be the *Price of the Component*. Putting $\odot = \Sigma$, we have intent value of the formal concept A_{20} calculated as $w(B_{20}) = 1000+500+600+250+100 = 2450$. In Figure 2 the whole concept lattices are set out, with extent or intent values respectively.

More mappings $\Phi(x)$ can be used. In our example $\Phi(x)$ was defined as *Price*, but $\Phi_1(x)$ can be defined as *Price*, $\Phi_2(x)$ as *Costs* and $\Phi_3(x)$ as $Benefit$ - with different measures assigned to an object. In the example Σ was used, but any aggregation operator Θ can be utilized, to achieve various $v_i(A)$ - values for extent e.g. *Average sales*, *Minimal costs* or *Maximal benefit*. Similarly, more attribute measures and corresponding values of intent can be obtained.

4 Applications of Formal Concept Analysis with Measures

4.1 Extent Values and Generalized *OLAP Cube*

Consider the example from the *OLAP cube* definition in Figure 1. The attributes $TradeMark = \{BMW, SKODA, FIAT\}$ and $Country = \{Germany, France\}$

have a non-binary domain, but this can be transformed into binary by nominal scaling. Nominal scaling means, that for each value from the domain we create an extra attribute (for more details see [1]). It will result in the formal context $\langle X, Y, I\rangle$ on Figure 3 (i). Computing concepts and values results in the concept lattice with values in Figure 3 (ii). Now recall the *OLAP cube* and its cells defined by tuples $\langle l_1, \ldots, l_{|H_k|}\rangle_{H_k}$. Intent B of each formal concept $\langle A, B\rangle$ can be used for determining the particular cell in *OLAP cube* as follows: For each attribute of B put 1 into a corresponding tuple. Tuples will determine the cell of the *OLAP cube*. Consider the formal context $\langle A, B\rangle = \langle \{Car1\}, \{BMW, Germany\}\rangle$. From intent B tuple $\langle 1, 0, 0\rangle_{TradeMark}$ and tuple $\langle 0, 1\rangle_{Country}$ are created. Into each cell determined by tuples, the extent value of the concept, is put, in this case $v(\{Car1\}) = 30$. The same *OLAP cube* as shown in Figure 1 is the result. Each concept represents at least one cell. Note, that some cells can remain empty, because the set of attributes, defined by the corresponding tuples should not be a closed set in the given context. For example ignore object 5 in the formal context in Figure 3. The cell defined by tuples $\langle 0, 0, 1\rangle_{TradeMark}$ and $\langle 0, 0\rangle_{Country}$ (representing Total price of all FIATs) remains empty, because set $\{FIAT\}$ is not closed (in formal context excluding object 5). For this reason each cell of the *OLAP cube* has a value, from the closed set of attributes, determined by tuples. There is not only a technical reason to have the *OLAP Cube* complete, but also a semantical reason. If only FIATs are sold in France and no FIATs are sold in any other country, the Total price of all FIATs will be the same as the Total price of all FIATs sold in France. Hence FCA using measures and aggregated values, can be seen as a generalized *OLAP cube*.

4.2 Constraints of Lattice with Values via Closure Operators

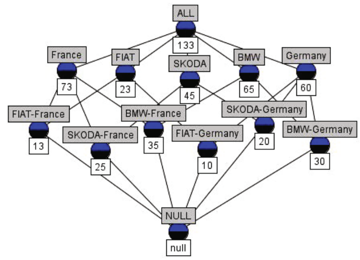
When large datasets are analyzed by FCA an huge amount of concepts is result. Many of them could be insignificant for the user, so there needs to be a way to select those, which can be important. One of the methods for limiting the amount is by setting constraints via a closure operator (see [13], [14]). Advantage of this method is that it is capable of calculating important concepts directly, without calculating the whole lattice.

Definition 5 (C-interesting attributes - [13]). *Let Y be a set of attributes, C be a closure operator in Y. A set $B \subseteq Y$ is called a C-interesting set of attributes (shortly, a set of C-attributes) if $B = C(B)$.*

Definition 6 (C-interesting concepts - [13]). *Let $\langle X, Y, I\rangle$ be a formal context, C be a closure operator in Y. We put $\mathcal{B}_C(X, Y, I) = \{\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ $|B = C(B)\}$. Each $\langle A, B\rangle \in \mathcal{B}_C(X, Y, I)$ is called a C-interesting concept (shortly, a C-concept) in $\langle X, Y, I\rangle$.*

Car nr.	*BMW*	*SKODA*	*FIAT*	*Germany*	*France*	Price in 000 EUR
1	×			×		30
2	×				×	35
3		×		×		20
4		×			×	25
5			×	×		10
6			×		×	13

(i) Formal context with measures



(ii) Concept lattice with extent values

Fig. 3. Example of the concept lattices with the values

Theorem 1 (Closure operator constraints on Concept Lattice - [13]). *Let $\mathcal{B}(X,Y,I)$ be a concept lattice and $K \subseteq \mathcal{B}(X,Y,I)$, then the following are equivalent*

– *K is set of C-concepts for some closure operator C*
– *K equipped with partial order $\leq$ is a complete $\bigvee$-sublattice of $\mathcal{B}(X,Y,I)$*

In [13] there were introduced a couple of examples of closure operators $C(B)$ e.g. "Minimal support" is defined as $C(B) = B$ if $|B^{\uparrow}| \leq s$ or $C(B) = Y$ otherwise. In this paper we present similar closure operators, which will use extent or intent values respectively.

Theorem 2 (Closure operator constraints on Concept Lattice - [13]). *Let $\mathcal{B}(X,Y,I)$ be a concept lattice, C be a closure operator on Y. Then $\mathcal{B}_{\mathcal{C}}(X,Y,I)$ equipped with $\leq$ is a complete $\bigvee$-sublattice of $\mathcal{B}(X,Y,I)$*

Proposition 6 (Maximal intent value using $\odot_{\uparrow}$ as closure operator). *The maximal intent value using monotone aggregation defined as $C(B) = B$ if $w(B) \leq n$ or $C(B) = Y$ otherwise, is a closure operator.*

Proof. We need to prove that: $w(B_k) \leq n \Rightarrow w(\bigcap B_i) \leq n$
$\bigcap B_i \subseteq B_k$ is evident, $w(\bigcap B_i) \leq w(B_k)$ - using monotonicity of aggregation
$w(\bigcap B_i) \leq w(B_k) \leq n \Leftrightarrow w(\bigcap B_i) \leq n$

Proposition 7 (Minimal extent value using $\odot_{\uparrow}$ as closure operator). *The minimal extent value using monotone aggregation defined as $C(B) = B$ if $v(A) = v(B^{\downarrow}) \geq n$ or $C(B) = Y$ otherwise, is a closure operator.*

Proof. We need to prove that: $v(B^{\downarrow}) \geq n \Rightarrow v((\bigcap B_i)^{\downarrow}) \geq n$
$\bigcap B_i \subseteq B_k \Leftrightarrow (\bigcap B_i)^{\downarrow} \supseteq B_k^{\downarrow}$
$v((\bigcap B_i)^{\downarrow}) \geq v(B_k^{\downarrow})$ - using monotonicity of aggregation
$v((\bigcap B_i)^{\downarrow})) \geq v(B_k^{\downarrow}) \geq n \Leftrightarrow v((\bigcap B_i)^{\downarrow})) \geq n$

Proposition 8 (Minimal intent value using $\odot_\downarrow$ as closure operator). *The minimal intent value using antitone aggregation defined as $C(B) = B$ if $w(B) \geq n$ or $C(B) = Y$ otherwise, is a closure operator.*

Proof. We need to prove that: $w(B_k) \geq n \Rightarrow w(\bigcap B_i) \geq n$
$\bigcap B_i \subseteq B_k$ is evident, $w(\bigcap B_i) \geq w(B_k)$ - using antitonicity of aggregation
$w(\bigcap B_i) \geq w(B_k) \leq n \Leftrightarrow w(\bigcap B_i) \geq n$

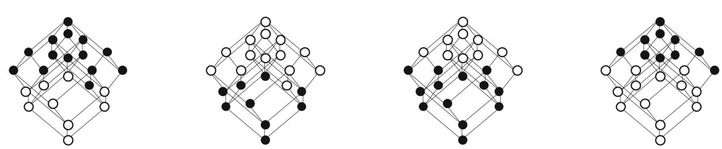
Proposition 9 (Maximal extent value using $\odot_\downarrow$ as closure operator). *The maximal extent value using antitone aggregation defined as $C(B) = B$ if $v(A) = v(B^\downarrow) \leq n$ otherwise, is a closure operator.*

Proof. We need to prove that: $v(B^\downarrow) \leq n \Rightarrow v((\bigcap B_i)^\downarrow) \leq n$
$\bigcap B_i \subseteq B_k \Leftrightarrow (\bigcap B_i)^\downarrow \supseteq B_k^\downarrow$
$v((\bigcap B_i)^\downarrow) \leq v(B_k^\downarrow)$ - using antitonicity of aggregation
$v((\bigcap B_i)^\downarrow)) \leq v(B_k^\downarrow) \leq n \Leftrightarrow v((\bigcap B_i)^\downarrow)) \leq n$

Table 3. Theorems for constraints on the concept lattice with the values

	monotone operator $\odot_\uparrow$		antitone operator $\odot_\downarrow$	
	$\geq$ n	$\leq$ n	$\leq$ n	$\geq$ n
$v(B^\downarrow)$	complete $\bigvee$-sublattice	complete $\bigwedge$-sublattice	complete $\bigvee$-sublattice	complete $\bigwedge$-sublattice
$w\,(B)$	complete $\bigwedge$-sublattice	complete $\bigvee$-sublattice	complete $\bigwedge$-sublattice	complete $\bigvee$-sublattice

All those propositions were constructed for closure operators as defined on set Y. The well known duality of concept the lattice comes to similar dual propositions, where we get $\bigwedge$-sublattice for closure operator C defined on set X. Table 3 is an overview of all possible theorems. Recall, that operator Σ and operator of cardinality $|X|$ together with operator $max()$ belong to monotone operators $\odot_\uparrow$ and operator $min()$ is an antitone operator $\odot_\downarrow$. Based on the previous example with cars, a constrainede lattice can be demonstrated. Figure 4 shows a constrained lattice with different closure operators (• represents C-interesting concepts, ∘ other ones). Moreover such constraining of a concept lattice has a very clear meaning for users, e.g. "We are looking for concepts, where the total price of the car is at least 50 000 EUR".



(i) $C(B) = B$ if $v(A) \geq 50000$, $C(B) = Y$ otherwise
(ii) $C(B) = B$ if $v(A) \leq 50000$, $C(B) = Y$ otherwise
(iii) $C(B) = B$ if $w(B) \leq 1200$, $C(B) = Y$ otherwise
(iv) $C(B) = B$ if $w(B) \geq 1200$, $C(B) = Y$ otherwise

Fig. 4. Concept lattices constrained by closure operators using the values and $\odot = \Sigma$

4.3 FCA in Fuzzy Settings with Measures and Fuzzy OLAP

In fuzzy settings we use fuzzy sets $A \in L^X$ and $B \in L^Y$. Notation $A(x)$ means truth degree in which x belongs to A. Note that $A(x) = 1$ is the same as $x \in A$ in crisp setting and $A(x) = 0$ is the same as $x \notin A$. Definitions of the extent and intent measure will be the same as in the crisp case. The definition of intent and extent value will be slightly different:

Definition 7 (Value of fuzzy Extent and fuzzy Intent). *The value of fuzzy extent is mapping* $v : A_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$ *defined as* $v(A) = \bigodot_{x \in X} (\Phi(x) \times A(x))$, *where* $\times$ *is the symbol for multiplication,* $\odot$ *is either the symbol for sum* Σ *(the "sum" operation) or the arbitrary aggregation function* Θ*.* A *is a fuzzy extent of* L*-concept* $\langle A, B \rangle \in \mathcal{B}(X, Y, I)$*. Similarly, the value of the intent is mapping* $w : B_{\mathcal{B}(X,Y,I)} \to \mathbb{R}^+$ *defined as* $w(B) = \bigodot_{y \in Y} (\Psi(y) \times B(y))$, *where* B *is a fuzzy intent of* L*-concept* $\langle A, B \rangle \in \mathcal{B}(X, Y, I)$

For all our examples in fuzzy settings we will use the Gödel t-norm, for which the adjoint pair of operators is defined as $a \otimes b = min(a, b)$ and $a \to b = 1$ if $a \leq b$ or $a \to b = b$ otherwise. An example of L-context with measure, L-concepts and L-lattice with values can be found in Figure 5 ((i),(iv)) and Figure 6. Attributes from this example can be divided into two dimensions, namely "Size" and "Distance from the Sun". Such dividing enables using the OLAP approach. Space for the *OLAP cube* has already been defined, but the set $L = \{0, 1\}$ was used, because it was the *OLAP cube* in crisp settings. Generally the previous definition for the L-cube can be used for any residuated lattice L. In the example with the planets L is defined as a Gödel chain $L = \{0; \frac{1}{2}; 1\}$. The *OLAP cube* introduces the notation $\langle l_1, \ldots, l_{|H_k|} \rangle_{H_k}$ for L^{H_k}. It is the same as fuzzy set. Now the L-cube in Figure 5 part (ii) can be easily understood, where L-concepts are assigned to the L-cube cells, and part (iii), where they have been assigned to the L-cube corresponding values. Note, that each L-concept has its corresponding cell in the L-cube space according to its fuzzy intent (fuzzy set B). As an example the L-concept number 4 with value 123 353 is assigned to the cell with coordinates $\langle 0; 1 \rangle_{size}$ and $\langle 1; 1 \rangle_{distance}$ corresponding to its fuzzy intent $B = \langle 0; 1; 1; 1 \rangle$. The above mentioned example shows how to organize L-concepts with values in the L-cube. It is the case, where the set of attributes Y, can be organized into dimensions. Generally we can use an L-lattice, with assigned values as it was used in the crisp example with the cars. Independent attributes (attributes, which are not organized into dimensions) in fuzzy settings are not shown, as the reader can easily imagine such examples. Note, that in case of L-cube, the values have only been assigned to the cells, which represents closed fuzzy sets of attributes. Empty cells can also be filled in as in the case of the crisp *OLAP cube* to have the L-cube complete. Before taking this step, it is necessary to consider, if such action makes sense as it did in the crisp case and to discuss the meaning of aggregation functions in the fuzzy setting as well.

Planet	Size: Small	Size: Large	Distance: Far	Distance: Near	Mass in 10^{22}kg
Mercury	1	0	0	1	33
Venus	1	0	0	1	487
Earth	1	0	0	1	597
Mars	1	0	0,5	1	64
Jupiter	0	1	1	0,5	189 860
Saturn	0	1	1	0,5	56 846
Uranus	0,5	0,5	1	0	8 681
Neptune	0,5	0,5	1	0	10 243
Pluto	1	0	1	0	1

(i) Input data table - fuzzy relation and measures

		far	0	0	0	0,5	0,5	0,5	1	1	1
small	large	near	0	0,5	1	0	0,5	1	0	0,5	1
0	0		25	23	18	24	20	13	21	15	7
0	0,5								16		
0	1								10	9	4
0,5	0		22			19				14	
0,5	0,5								8		
0,5	1										
1	0		17		11	12		5	6		2
1	0,5										
1	1								3		1

(ii) L-Cube with corresponding L-concepts using Gödel t-norms

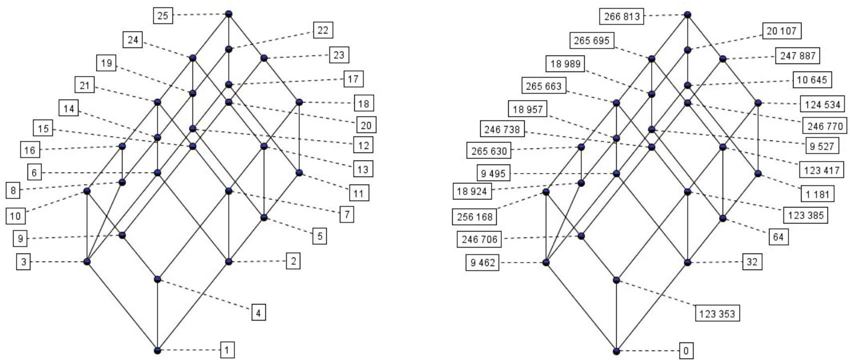
		far	0	0	0	0,5	0,5	0,5	1	1	1
small	large	near	0	0,5	1	0	0,5	1	0	0,5	1
0	0		266 813	247 887	124 534	265 695	246 770	123 417	265 663	246 738	123 385
0	0,5								265 630		
0	1								256 168	246 706	123 353
0,5	0		20 107			18 989				18 957	
0,5	0,5								18 924		
0,5	1										
1	0		10 645		1 181	9 527		64	9 495		32
1	0,5										
1	1								9 462		0

(iii) L-Cube with corresponding L-concepts using Gödel t-norms with extent values using operator Σ on measures assigned to objects

concept nr.	extent	intent	extent value
1	$\langle 0,0,0,0,0,0,0,0,0 \rangle$	$\langle 1;1;1;1 \rangle$	0
2	$\langle 0,0,0,\frac{1}{2},0,0,0,0,0 \rangle$	$\langle 1;0;1;1 \rangle$	32
3	$\langle 0,0,0,0,0,0,\frac{1}{2},\frac{1}{2},0 \rangle$	$\langle 1;1;1;0 \rangle$	9 462
4	$\langle 0,0,0,0,\frac{1}{2},\frac{1}{2},0,0,0 \rangle$	$\langle 0;1;1;1 \rangle$	123 353
5	$\langle 0,0,0,1,0,0,0,0,0 \rangle$	$\langle 1;0;\frac{1}{2};1 \rangle$	64
6	$\langle 0,0,0,\frac{1}{2},0,0,\frac{1}{2},\frac{1}{2},1 \rangle$	$\langle 1;0;1;0 \rangle$	9 495
7	$\langle 0,0,0,\frac{1}{2},\frac{1}{2},\frac{1}{2},0,0,0 \rangle$	$\langle 0;0;1;1 \rangle$	123 385
8	$\langle 0,0,0,0,0,0,1,1,0 \rangle$	$\langle \frac{1}{2};\frac{1}{2};1;0 \rangle$	18 924
9	$\langle 0,0,0,0,1,1,0,0,0 \rangle$	$\langle 0;1;1;\frac{1}{2} \rangle$	246 706
10	$\langle 0,0,0,0,1,1,\frac{1}{2},\frac{1}{2},0 \rangle$	$\langle 0;1;1;0 \rangle$	256 168
11	$\langle 1,1,1,1,0,0,0,0,0 \rangle$	$\langle 1;0;0;1 \rangle$	1 181
12	$\langle 0,0,0,1,0,0,\frac{1}{2},\frac{1}{2},1 \rangle$	$\langle 1;0;\frac{1}{2};0 \rangle$	9 527
13	$\langle 0,0,0,1,\frac{1}{2},\frac{1}{2},0,0,0 \rangle$	$\langle 0;0;\frac{1}{2};1 \rangle$	123 417
14	$\langle 0,0,0,\frac{1}{2},0,0,1,1,1 \rangle$	$\langle \frac{1}{2};0;1;\frac{1}{2} \rangle$	18 957
15	$\langle 0,0,0,\frac{1}{2},1,1,0,0,0 \rangle$	$\langle 0;0;1;\frac{1}{2} \rangle$	246 738
16	$\langle 0,0,0,0,1,1,1,1,0 \rangle$	$\langle 0;\frac{1}{2};1;0 \rangle$	265 630
17	$\langle 1,1,1,1,0,0,\frac{1}{2},\frac{1}{2},1 \rangle$	$\langle 1;0;0;0 \rangle$	10 645
18	$\langle 1,1,1,1,\frac{1}{2},\frac{1}{2},0,0,0 \rangle$	$\langle 0;0;0;1 \rangle$	124 534
19	$\langle 0,0,0,1,0,0,1,1,1 \rangle$	$\langle \frac{1}{2};0;\frac{1}{2};0 \rangle$	18 989
20	$\langle 0,0,0,1,1,1,0,0,0 \rangle$	$\langle 0;0;\frac{1}{2};\frac{1}{2} \rangle$	246 770
21	$\langle 0,0,0,\frac{1}{2},1,1,1,1,1 \rangle$	$\langle 0;0;1;0 \rangle$	265 663
22	$\langle 1,1,1,1,0,0,1,1,1 \rangle$	$\langle \frac{1}{2};0;0;0 \rangle$	20 107
23	$\langle 1,1,1,1,1,1,0,0,0 \rangle$	$\langle 0;0;0;\frac{1}{2} \rangle$	247 887
24	$\langle 0,0,0,1,1,1,1,1,1 \rangle$	$\langle 0;0;\frac{1}{2};0 \rangle$	265 695
25	$\langle 1,1,1,1,1,1,1,1,1 \rangle$	$\langle 0;0;0;0 \rangle$	266 813

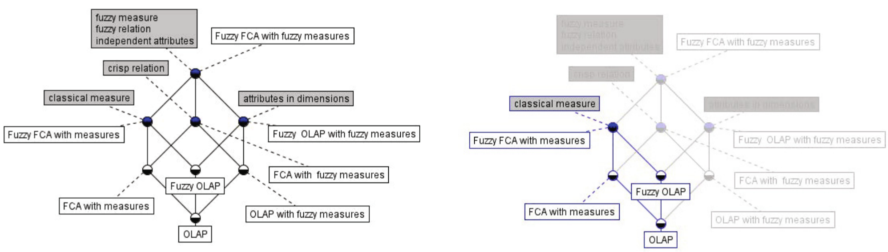
(iv) List of all L-concepts with extent value
Notation: fuzzy set $\{0/y_1; 0/y_2; 1/y_3; \frac{1}{2}/y_4\}$ is denoted as $\langle 0;0;1;\frac{1}{2} \rangle$

Fig. 5. FCA with measures in fuzzy settings, example is taken from [3] without measures, which are taken from NASA



(i) L-lattice - using Gödel t-norm (ii) L-lattice - values of extent

Fig. 6. L-lattice with measures. Sources: [3] (example) and NASA (measures)



(i) An overview of the whole research area (ii) What is done already

Fig. 7. Structure of the research - various use of FCA with measures

5 Conclusion and Future Research

A purpose of this paper is to introduce a novel method of data mining with numeric data. Assigning measures to objects and/or attributes and aggregating the measures into values of extent and/or intent we get a novel structure - concept lattice with values, which can be seen as a generalized *OLAP cube*. Using fuzzy logic in FCA, the L-cube can be defined based on the L-lattice, or the L-lattice itself can be used, regardless of dimensions. FCA in crisp or in fuzzy settings as well, can serve as a mathematical and logical framework for business intelligence technologies based on OLAP. Moreover in the crisp case efficient computing of the concept lattice with values can be done via closure operators based on values. Some constraints with a very clear meaning for the user have been proposed. **Future research** has been designed as follows: In the first step there will be a focus on further generalization. In Figure 7 there are depicted two lattices. The lattice in (i) shows the whole area of research. Note, that the crisp relation is a special case of the fuzzy relation, where $L = \{0, 1\}$, attributes organized into dimensions forms a special case of unorganized attributes and finally the classical measure is a special case of the fuzzy measure. Including fuzzy measures into consideration means also including aggregation operators defined on fuzzy measures. The main goal of future research is to develop a generalized model, i.e. model for the L-lattice with fuzzy measures - the top of the lattice depicted in (i). What has been done in this paper is shown in part (ii). This step includes investigation of t-norms. It is not clear until now, which t-norm fits better for the semantics of the L-cube, i.e. which t-norm fits better for the L-cube in a real application of business intelligence. In the second step we would like to look at the constraints regarding to the L-lattice (or L-cube respectively). Constraints on the concept lattice with values via closure operators are described in this paper, but only in the crisp setting. We want to do the same in fuzzy settings. Moreover, there are some other methods, of how to constrain the L-lattice, using hedges [16] or showing only crisply generated L-concepts [17]. The third step is to develop an efficient algorithm for computing L-lattices with values, using constraints. Step number four will investigate other formal definitions of fuzzy OLAP (e.g. presented in [18] or [19]) and compare it with our approach. According to current investigations it seems that those approaches use fuzzy measures and operations on fuzzy measures. The final step should lead to a real application using the L-lattice (L-cube), which will operate with novel L-SQL language - an enhancement of the classical SQL language. Some of the above mentioned steps will be presented in the extended version of this paper.

Acknowledgements. The author would like to praise the Lord Jesus Christ for His amazing grace and the forces that He has given to the author during the realization of this work. The author dedicates this paper to his beloved wife Marcela. Partly supported by IGA (Internal Grant Agency) of the Palacky University, Olomouc.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer, Berlin (1999)
2. Codd, E.F., Codd, S.B., Salley, C.T.: Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. Codd & Date (1993)
3. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer, Academic/Plenum Publishers, New York (2002)
4. Wang, Z., Klir, G.: Generalized measure theory. Springer, New York (2009)
5. Maier, D.: The theory of relational databases. Computer Science Press, Rockville (1983)
6. Calvo, T., Mayor, G., Mesiar, R. (eds.): Aggregation Operators. New Trends and Applications. Physica-Verlag, Heidelberg (2002)
7. Zadeh, L.A.: Fuzzy sets. Inf. Control 8, 338–353 (1965)
8. Kuznetsov, S.D., Kudryavtsev, A.: A mathematical model of the OLAP cubes. Programming and Computer Software 35(5), 257–265 (2009)
9. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht (1998)
10. Pollandt, S.: Fuzzy Begriffe. Springer, Heidelberg (1997)

11. Calvo, T., Kolesarova, A., Komornikova, M., Mesiar, R.: Aggregation operators: Properties, classes and construction methods. In: Calvo, T., Mayor, G., Mesiar, R. (eds.) Aggregation Operators: New Trend and Applications, pp. 3–106. Physica Verlag, Heidelberg (2002)
12. Belohlavek, R., Macko, J.: Selecting Important Concepts Using Weights. In: Valtchev, P., Jäschke, R. (eds.) ICFCA 2011. LNCS(LNAI), vol. 6628, pp. 65–80. Springer, Heidelberg (2011)
13. Belohlavek, R., Vychodil, V.: Background Knowledge in Formal Concept Analysis: Constraints via Closure Operators. In: ACM SAC 2010, pp. 1113–1114 (2010)
14. Bělohlávek, R., Vychodil, V.: Formal Concept Analysis with Constraints by Closure Operators. In: Schärfe, H., Hitzler, P., Øhrstrøm, P. (eds.) ICCS 2006. LNCS (LNAI), vol. 4068, pp. 131–143. Springer, Heidelberg (2006)
15. Belohlavek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by fuzzy closure operators. In: Int. Conf. Soft Computing and Intelligent Systems & Int. Symposium on Intelligent Systems, SCIS & ISIS 2006, Tokyo, Japan, September 20-24, pp. 309–314 (2006)
16. Belohlavek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: The IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005, Reno, Nevada, USA, May 22-25, pp. 663–668 (2005)
17. Bělohlávek, R., Sklenář, V., Zacpal, J.: Crisply Generated Fuzzy Concepts. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS (LNAI), vol. 3403, pp. 269–284. Springer, Heidelberg (2005)
18. González, C., Tineo, L., Urrutia, A.: Fuzzy OLAP: A Formal Definition. In: Yu, W., Sanchez, E.N. (eds.) Advances in Computational Intelligence. AISC, vol. 61, pp. 189–198. Springer, Heidelberg (2009)
19. Laurent, A., Bouchon-Meunier, B., Doucet, A.: Towards Fuzzy-OLAP Mining
20. Belohlavek, R.: What is a Fuzzy Concept Lattice? II. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 19–26. Springer, Heidelberg (2011)

Good Classification Tests as Formal Concepts

Xenia A. Naidenova

Military Medical Academy, Saint-Petersburg, Russian Federation
ksennaidd@gmail.com

Abstract. The interconnection between the Diagnostic (Classification) Test Approach to Data Analysis and the Formal Concept Analysis (FCA) is considered. The definition of a good classification test is given via Galois's correspondences. Next we discuss the relations between good tests and formal concepts. A good classification test is understood as a good approximation of a given classification on a given set of examples. Classification tests serve as a basis for inferring implicative, functional dependencies and association rules from datasets. This approach gives the possibility to directly control the data analysis process by giving object classifications.

Keywords: Good classification test, Galois lattice, Formal Concept Analysis, Logical rule mining.

1 Introduction

Mining logical rules (association rules, implicative and functional dependencies) is a core and extensively studied problem of data analysis. Databases are becoming increasingly larger, thus requiring a higher computing power to mine logical rules in reasonable time. An increase in the computational power of algorithms leads to the fact that the quantity of extracted logical rules reaches at least hundreds of thousands. To search, in this set of rules, for something useful becomes completely difficult. Meanwhile, taking up data analysis, specialists in any problem domain want to obtain answers to interesting questions that can be formulated in terms of known concepts or goals. Therefore it would be expedient to include the algorithms of logical rule mining in the process of reasoning so as this process would be governed by some consecutively formed goals and sub-goals. Some ideas of ontology-driven association rule mining are advanced in [1], [2], and [3]. The integration of ontology and association rule mining is considered in [4]. Using a Rule Schema for representing user expectations is proposed in [5]. An idea to extract the association rules taking into account the user's objective has also been advanced in [6].

The principle concept of human reasoning is the concept of classification. It is possible to assume that classifications can serve as an instrument of managing logical rules mining. Diagnostic (classification) test approach to data analysis [7-10] can be one of such instrument.

We will consider two ways for giving classifications as it is shown in Fig. 1: 1) by a target attribute *KL* or 2) by a value *v* of target attribute *KL*. A target attribute

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 211–226, 2012.

partitions a given set of examples into disjoint classes the number of which is equal to the number of values of this attribute. A target value of attribute partitions a given set of examples into two disjoint classes: 1) the examples in description of which the target value appears (positive examples); 2) all the other examples (negative examples).

	A	B	C	KL
1	a_1	b_1	c_1	k_1
2	a_2	b_2	c_1	k_1
3	a_1	b_2	c_2	k_2
4	a_1	b_3	c_1	k_3
5	a_3	b_4	c_2	k_3

	A	B	C	D
1	a_1	b_1	c_1	h
2	a_2	b_2	c_1	v
3	a_1	b_2	c_2	v
4	a_1	b_3	c_1	f
5	a_3	b_4	c_2	v

KL
v–
v
v
v–
v

KL – the target attribute; *v* – the target value of attribute

Fig. 1. Two Modes of Giving the Target Classification

We are interested in solving the following tasks:

1. Given attribute *KL*, to infer logical rules of the form:

$A\ B\ C \rightarrow KL$ or

$D\ S \rightarrow KL$ or

or

$A\ S\ Q\ V \rightarrow KL$

where *A*, *B*, *C*, *D*, *Q*, *S*, *V* – the names of attributes.

2. Given value *v* of attribute *KL*, to infer logical rule of the form:

if ((value of attribute *A* = "*a*") &
(value of attribute *B* = "*b*") &
....),
then (value of attribute *KL* = "*v*").

Rules of the first form are functional dependencies as they are determined in relational data base constructing [11]. Rules of the second form are implicative dependencies as they are determined in association rule mining [12]. Left parts of rules can be considered as descriptions of given classifications. In our diagnostic test approach to logical rules mining, left parts of these rules are called diagnostic tests.

A concept of good test as a good approximation of a given classification of objects has been advanced in [8], [13]. A good test must satisfy the greatest number of identifying relations between objects inside the blocks of a given classification and all distinguishing relations between objects from different blocks of a given classification. It means that the best test for a given classification must generate a partition of objects into the smallest number of blocks such that each block of this partition is included in one and only one block of the given classification. This consideration leads to taking into account the set of partitions generated by all possible subsets of a given set of attributes (values) together with the ordering of partitions by set-theoretical inclusion relation. Since the system of all partitions (all equivalence relations) of finite set of objects forms complete algebraic lattice [14], a new algebraic model of diagnostic task has been introduced on the basis of partition lattice construction [15].

The goal of this paper is to give the definition of good diagnostic test in terms of Formal Concept Analysis (FCA) for two ways of object classification representation. For this goal, we will actually use some mathematical techniques that have been applied by S. Kusnetsov [16-17] for establishing a relation between JSM-hypotheses and formal concepts in FCA. The first version of the JSM-method of automated hypotheses generation has been described in [18]. Later versions can be found in [19]. As a method of data analysis, JSM-method is a system of machine learning from positive and negative examples: for given positive and negative examples, one constructs a "generalization" of positive examples that does not "cover" any negative examples.

The rest of the paper is organized as follows. In Section 2, the basic terminology of FCA is given. Section 3 is devoted to defining a concept of good diagnostic (classification) test in terms of FCA. In Section 4, the problem of inferring functional dependencies as a special kind of diagnostic tests approximating a given classification of objects into more than two disjoint blocks is considered. Then a reduction of this task is descrived to a task with only two classes of positive and negative examples. Finally, we give an algebraic model of diagnostic tasks based on the partition lattice. A short conclusion resumes this paper.

2 The Basic Terminology of Formal Concept Analysis

Now we proceed to definitions from FCA [20].

Definition 1. A formal context $K = (G, M, I)$ consists of a set G of objects, a set M of attributes, and binary relation $I \subseteq G \times M$. The notation gIm indicates that $(g, m) \in I$ and denotes the fact that the object g possesses the attribute m.

The Galois connection between the ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$ is given by the following mappings called derivation operators: for $A \subseteq G$ and $B \subseteq M$,

$A' = \{m \in M | \forall g \in A: (gIm)\}$ and $B' = \{g \in G | \forall m \in B: (gIm)\}$.

Definition 2. A **formal concept** of a formal context (G, M, I) is a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The set A is called the **extent**, and the set B is called the **intent** of the concept (A, B).

For $g \in G$ and $m \in M$, $\{g\}'$ is denoted by g' and called **object intent**, and $\{m\}'$ is denoted by m' and called **attribute extent**.

Definition 3. For a context (G, M, I), a concept $X = (A, B)$ is less general than or equal to a concept $Y = (C, D)$ (or $X \leq Y$) if $A \subseteq C$ or, equivalently, $D \subseteq B$.

The set $\mathfrak{R}(K)$ of all concepts of a formal context K together with the partial order $(A, B) \leq (C, D)$ is called *concept* lattice of K.

In the framework of FCA, the definition of implication on attributes is determined as follows.

Definition 4. The implication $A \rightarrow B$, where A, $B \subseteq M$, holds if and only if $A' \subseteq B'$ (or $B \supseteq A''$), i.e., all objects from G that have the set of attribute A also have the set of attributes B.

An equivalence relation θ on the power set $(2^M, \subseteq)$ of M is given as follows: $X\theta Y \Leftrightarrow X' = Y'$. Any concept intent is the largest set of attributes of the equivalence class of θ to which it belongs.

We need the definition of key set of attributes or minimal generator of a class of equivalence of θ.

Definition 5. Let $K = (G, M, I)$ be a formal context and $C \subseteq M$ be a concept intent, i.e., $C'' = C$. The subset $D \subseteq C$ is a minimum generator of C under the closure operator $'$ if $D'' = C$ holds and D is minimal subset with respect to this property, i.e., for all $E \subset D$ we have $E'' \neq C$.

Besides formal contexts defined above (two-valued contexts), so-called many-valued contexts are studied in FCA. For this goal, a many-valued context is reduced to a two-valued one by a scaling procedure [20]. One of the possible types of scaling is used by S. Kuznetsov for representing JSM-hypotheses and classifications in terms of FCA; an example (adopted by us from [16]) of this reduction is given in Tables 1, 2.

Table 1. An example of many-valued dataset (adopted from [16])

G \ M	Color	Frm	Smooth	Form	Target
1 apple	yellow	no	yes	round	+
2 grapefruit	yellow	no	no	round	+
3 kiwi	Green	no	no	oval	+
4 plum	Blue	no	yes	oval	+
5 toycube	Green	yes	yes	cubic	–
6 egg	White	yes	yes	oval	–
7 tennis ball	White	no	no	round	–

Table 2. The result of transforming dataset of Table 1 (adopted from [16])

G \ M	w	y	g	b	f	f-	s	s-	r	r-	Target
1 apple		x				x	x		x		+
2 grapefruit		x				x		x	x		+
3 kiwi			x			x		x		x	+
4 plum				x		x	x			x	+
5 toy cub			x		x		x			x	–
6 egg	x				x		x			x	–
7 tennis ball	x					x		x	x		–

In Table 2, the following abbreviations are used: “w” for white, “y” for yellow, “g” for green, “b” for blue, “s” for smooth, “f” for form, “r” for round, “o” for oval, and “m-” for m ∈ {w, y, g, b, s, f, r, o}.

Now we proceed to definition of classification in term of FCA [16-17].

Let a context $K = (G, M, I)$ be given. In addition to attributes of M, a target attribute $\omega \notin M$ is considered. The set G of all objects is partitioned into three subsets: the set G_+ of those objects that are known as having property ω (these are the positive examples), the set G_- of those objects that are known as not having property ω (the negative examples) and the set G_τ of undetermined examples, i.e., those objects, of which it is unknown whether they have property ω or not. Respectively, we consider three sub-contexts of $K = (G, M, I)$: $K_+ := (G_+, M, I_+)$, $K_- := (G_-, M, I_-)$, and $K_\tau := (G_\tau, M, I_\tau)$, and we have $I_\varepsilon := I \cap (G_\varepsilon \times M)$ for $\varepsilon \in \{+,-,\tau\}$. The corresponding derivation operators are denoted by $(\cdot)^+$, $(\cdot)^-$, $(\cdot)^\tau$, respectively.

A type of hypothesis which is mostly used in practice, namely "no counterexample hypothesis" [19] or positive hypothesis, is a positive formal intent h of K_+ such that $h^+ \neq \varnothing$ and $(h \not\subset g^- \ \&\ h \neq g^-)$ $g^- := \{m \in M \mid (g,m) \in I_-\}$ for any negative example $g \in G_-$. Equivalently, $h^{++} = h$ and $h' \cap G_- = \varnothing$, where $(\cdot)'$ is taken in the whole context $K = (G, M, I)$. An intent of K_+ that is contained in the intent of a negative example is a falsified(+)-generalization. Negative hypotheses and falsified generalizations are defined similarly.

Hypotheses are used for classifying undetermined examples from the set G_τ. If an undetermined example $g_\tau \in G_\tau$ contains a positive hypothesis h_+ (i.e., $\{g_\tau\}^\tau \supseteq h_+$), then it is said that h_+ is a hypothesis in favor of a positive classification of the undetermined example. A hypothesis in favor of a negative classification is defined in a similar way. If there is a hypothesis in favor of a positive classification of g_τ and there is no hypothesis in favor of negative classification of g_τ, then g_τ is classified *positively*. A negative classification of g_τ is realized in a similar way. If $\{g_\tau\}^\tau$ does not contain any negative or positive hypotheses, the classification is unsatisfiable. If $\{g_\tau\}^\tau$ contains both positive and negative hypotheses, then the classification is contradictory.

Return to Table 2. If we have an undetermined example $mango_\tau$ = {y, f-, s, r-}, then it is classified positively, since $mango_\tau$ contains the minimal hypothesis {f-, r-} and does not contain any negative hypothesis. We have also two minimal negative hypotheses: 1. {w} corresponding with examples *egg* and *tennis ball*; 2. {f, s, r-} corresponding with examples *toy cube* and *egg*.

3 Good Diagnostic Test Definition in Terms of FCA

Let $G = \{1, 2,\ldots, N\}$ be the set of objects' indices (objects, for short) and $M = \{m_1, m_2, \ldots, m_j, \ldots m_m\}$ be the set of attributes' values (values, for short). Each object is described by a set of values from M. The object descriptions are represented by rows of a table the columns of which are associated with the attributes taking their values in M (see, please, Table 3).

The definition of good tests is based on correspondences of Galois on $G \times M$ [21] and two relations $G \to M$, $M \to G$. Let $A \subseteq G$, $B \subseteq M$. Denote by B_i, $B_i \subseteq M$, $i = 1,\ldots, N$ the description of object with index i. We define the relations $G \to M$, $M \to G$ as follows: $G \to M$: $A' = \text{val}(A) = \{\text{intersection of all } B_i\text{: } B_i \subseteq M, i \in G\}$ and $M \to G$: $B' = \text{obj}(B) = \{i\text{: } i \in G, B \subseteq B_i\}$. Of course, we have $\text{obj}(B) = \{\text{intersection of all obj}(m)\text{: obj}(m) \subseteq G, m \in B\}$.

Operations val(A), obj(B) are reasoning operations (derivation operators) related to discovering general features of objects or all objects possessing a given set of features. It is worth noticing that, for defining these operators, we do not use any scaling procedure to transform many-valued context to two-valued one.

These operations possess the following properties [22]:

(i) $A_1 \subseteq A_2 \Rightarrow$ val(A_2) $\subseteq$ val(A_1) for all A_1, $A_2 \subseteq G$;
(ii) $B_1 \subseteq B_2 \Rightarrow$ obj(B_2) $\subseteq$ obj(B_1) for all B_1, $B_2 \subseteq M$;
(iii) $A \subseteq$ obj(val(A)) & val(A) = val(obj(val(A))) for all $A \subseteq G$;
(iv) $B \subseteq$ val(obj(B)) & obj(B) = obj(val(obj(B))) for all $B \subseteq M$;
(v) val($\cup A_j$) = $\cap$ val(A_j) for all $A_j \subseteq G$; obj($\cup B_j$) = $\cap$ obj(B_j) for all $B_j \subseteq M$.

The properties (i), (ii) relate to extending subsets A, B. Extending A by some new object j^* leads to receiving a more general feature of objects: ($A \cup j^*$) $\supseteq A$ implies val($A \cup j^*$) $\subseteq$ val(A). It is an elementary step of generalization. Extending B by a new value m leads to decreasing the number of objects possessing the general feature 'Bm' in comparison with the number of objects possessing the general feature 'B': ($B \cup m$) $\supseteq m$ implies obj($B \cup m$) $\subseteq$ obj(B). It is an elementary step of specialization.

Extending B or A is effectively used for finding classification tests, so the property (v) is very important to control the domain of searching for tests. In order to choose a new set ($A \cup j$) such that val($A \cup j$) $\neq \emptyset$ it is necessary to choose j, $j \notin A$, $j \in G$ such that the condition (val(A) $\cap B_j$) $\neq \emptyset$ is satisfied. Analogously, in order to choose a new set ($B \cup m$) such that obj($B \cup m$) $\neq \emptyset$ it is necessary to choose m, $m \notin B$, $m \in M$ such that the condition (obj(B) $\cap$ obj(m)) $\neq \emptyset$ is satisfied.

The properties (iii), (iv) relate to the following generalization operations:

generalization_of(B) = B'' = val(obj(B)); generalization_of(A) =A'' = obj(val(A)).

The generalization operations are actually closure operators [21]. A set A is closed if A = obj(val(A)). A set B is closed if B = val(obj(B)).

These generalization operations are not artificially constructed operations. One can perform, mentally, a lot of such operations during a short period of time. For example, suppose that somebody has seen two films (g) with the participation of Gerard Depardieu (val(g)). After that he tries to know all the films with his participation (obj(val(g))). Assume that one know that Gerard Depardieu acts with Pierre Richard (m) in several films (obj(m)). After that he discovers that these films are the films of the same producer Francis Veber (val(obj(m))).

Namely these generalization operations are used for searching for good diagnostic tests.

For $g \in G$ and $m \in M$, $\{g\}'$ is denoted by g' and called **object intent**, and $\{m\}'$ is denoted by m' and called **value extent**.

Let $K = (G, M, I)$ be a given context and $K = K_+ \cup K_-$, where $K_+ = (G_+, M, I_+)$, $K_- = (G_-, M, I_-)$, $G = G_+ \cup G_-$ ($G_- = G \backslash G_+$).

Diagnostic test is defined as follows.

Definition 6. A diagnostic test for G_+ is a pair (A, B) such that $B \subseteq M$ (A = obj(B) $\neq \emptyset$), $A \subseteq G_+$ and $B \not\subset$ val(g) & $B \neq$ val(g), $\forall g$, $g \in G_-$. Equivalently, obj(B) $\cap G_- = \emptyset$.

In general case, a set B is not closed for diagnostic test (A, B), i. e., a diagnostic test is not obligatory a concept of FCA. This condition is true only for the special class of tests called 'maximally redundant ones'.

Definition 7. A diagnostic test (A, B), $B \subseteq M$ ($A = \text{obj}(B) \neq \varnothing$) for G_+ is **maximally redundant** if $\text{obj}(B \cup m) \subset$ A, for all $m \notin B$ and $m \in M$.

We define also a key or irredundant test as follows.

Definition 8. A diagnostic test (A, B), $B \subseteq M$ ($A = \text{obj}(B) \neq \varnothing$) for G_+ is **irredundant** if any narrowing $B^* = B \backslash m$, $m \in B$ implies that ($\text{obj}(B^*)$, B^*)) is **not a test** for G_+.

Let a pair (A, B) be an irredundant test for G_+, where $A = \text{obj}(B)$. Consider the maximally redundant test (A, B^*) for G_+, i.e., $B^* = \text{val}(A)$ and $\text{obj}(B^*) = A$. Then B is a **minimum generator** of B^* under the closure operator $'$. An irredundant test is a formal concept if and only if it is simultaneously maximally redundant.

Definition 9. A diagnostic test (A, B), $B \subseteq M$ ($A = \text{obj}(B) \neq \varnothing$) for G_+ is **good** if and only if any extension $A^* = A \cup i$, $i \notin A$, $i \in G_+$ implies that (A^*, $\text{val}(A^*)$) is **not a test** for G_+.

If a good test (A, B), $B \subseteq M$ ($A = \text{obj}(B) \neq \varnothing$) for G_+ is irredundant (GIRT), then any narrowing $B^* = B \backslash m$, $m \in B$ implies that ($\text{obj}(B^*)$, B^*)) is **not a test** for G_+.

If a good test (A, B), $B \subseteq M$ ($A = \text{obj}(B) \neq \varnothing$) for G_+ is maximally redundant (GMRT), then any extension $B^* = B \cup m$, $m \notin B$, $m \in M$ implies that ($\text{obj}(B^* \cup m)$, B^*) is **not a good test** for G_+.

Any object description A in a given dataset is a maximally redundant set of values because for any value $m \notin A$, $m \in M$, $\text{obj}(A \cup m)$ is equal to $\varnothing$.

In Table 3, the subset of values '*Blond Blue*' is a good irredundant test for Class-(+) and, simultaneously, it is maximally redundant subset of values. The subset'*Blond Hazel*' is a test for Class-(−) but it is not a good test and, simultaneously, it is maximally redundant subset of values.

Table 3. Example of data classification

Index of example	Height	Color of hair	Color of eyes	Target Class
1	Low	Blond	Blue	+
2	Low	Brown	Blue	–
3	Tall	Brown	Hazel	–
4	Tall	Blond	Hazel	–
5	Tall	Brown	Blue	–
6	Low	Blond	Hazel	–
7	Tall	Red	Blue	+
8	Tall	Blond	Blue	+

The value '*Hazel*' is a good irredundant test for Class-(−). The value '*Red*' is a good irredundant test for Class-(+).

The fact that a GIRT (as minimal generator) is contained in one and only one GMRT implies one of the possible methods for searching for GIRTs for a given class of objects:

- Find all GMRTs for a given class of objects;
- For each GMRT, find all GIRTs (minimal generators) contained in it.

The first algorithm for inferring all GMRTs for a given class of objects has been proposed in [7] and described also in [23]. This algorithm, later called "Background Algorithm" ([24]), uses the vertical format of data, i.e., subsets of indices corresponding to objects, and the level-wise manner for constructing these subsets of size $k + 1$ from subsets of size k, beginning with $k = 1$ and terminating when all extended subsets do not correspond to tests for a given class of objects. "The Background Algorithm" can be also used to extract all the GIRTs contained in a given GMRT [9].

An algorithm for inferring all GMRTs for a given class of objects proposed in [25] is based on a decomposition of the main problem into the subtasks of inferring all GMRTs containing a given subset of values X (maybe, only one value). The subtasks are solved in a depth-first manner with the use of main recursive procedure. The main procedure uses some effective pruning techniques to discard values, which can not be included in any newly constructed GMRTs, and object intents, which do not correspond currently to tests for a given class of objects.

In [26], an algorithm is proposed to improve "the Background algorithm" given in [23]. The improvements are the following: 1) only one lexically ordered set S(test) of k-subsets of object idices, $k = 1,\ldots\ldots n$, is used, where n is the number of objects; 2) S(test) contains only closed subsets of indices corresponding to tests for a given classification; 3) k-subsets are extended via adding indices (objects) one-by-one, and, for each newly obtained subset, its closure is formed and inserted in S(test) if it is the extent of a test but not of a GMRT (if it is the extent of GMRT, then it is stored in the set STGOOD; 4) for each k-subset, indices (objects) admissible to extend it are revealed with the use of STGOOD and S(test).

The algorithm DIAGaRa [9] uses the characteristic W(B) of any subset B of values named by the weight of B: $W(B) = ||\mathrm{obj}(B)||$, where $||s||$ denotes the cardinality of s. The algorithm DIAGaRa searches for all GMRT for a given class of objects such that their weights are equal to or more than WMIN, the minimal permissible value of the weight.

4 Good Diagnostic Tests and Inference Functional Dependencies

The peculiarity of Diagnostic Test Approach consist in giving the basis for mining not only dependencies between values of attributes but also functional dependencies (FD) in the form $X \rightarrow A$, where A is an attribute, X is a set of attributes, $A \notin X$, $A \in M$, $X \subseteq M$, and M is the universe of attributes the values of which appear in descriptions of objects.

Traditionally [27], the following definition of diagnostic test is given: let T be an arbitrary table of n–dimensional pair-wise different vectors partitioned into blocks k_1, k_2, ...k_q, $q > 2$. A collection of coordinates $x_{i1}, \ldots, x_{im}$, $1 \leq m \leq n$ is called diagnostic test with respect to a given partitioning into blocks if the projections of vectors from different blocks defined by $x_{i1}, \ldots, x_{im}$, $1 \leq m \leq n$ are also pair-wise different. Otherwise $x_{i1}, \ldots, x_{im}$, $1 \leq m \leq n$ are to be said non-admissible collection of coordinates.

Functional dependency is defined in the following context. Let $M = \{A_1, \ldots, A_m\}$ be a nonempty set of attributes A_i`s. Let dom(A_i) = $\{a_{i1}, a_{i2}, \ldots\}$ be a finite set of values called the domain of A_i. We assume that dom(A_i) $\cap M = \varnothing$, $i \in \{1, \ldots, m\}$ and dom(A_i) $\cap$ dom(A_j) $= \varnothing$, for $i, j, i \neq j, i, j \in \{1, \ldots, m\}$. Let $T(M)$ be a given set of object descriptions. Usually, description of each object is complete, i.e, contains values of all attributes of M. Assume that $OBJ = \{1, 2, \ldots, n\}$ is the set of object indices and $j \in OBJ$ is associated with description of j-th object, i. e., with $t_j = x_{j1} \ldots x_{jm}$ such that $t_j[A_i] = x_{ji}$ is in dom(A_i) for all A_i in M. Each attribute partitions the set $T(M)$ of objects into disjoint blocks. Let KL be an additional attribute by values of which the set of objects $T(M)$ is also partitioned into disjoint blocks.

A functional dependency (FD) between $X \subseteq M$ and KL is defined as follows: $X \rightarrow KL \Leftrightarrow \forall (i, j), i, j \in OBJ, i \neq j,$

$$t_i[X] = t_j[X] \rightarrow t_i[KL] = t_j[KL], \quad (1)$$

We call this relation *the condition of indistinguishability* between objects in $T(M)$ by values of a subset of attributes X.

Traditional definition of diagnostic test in $T(M)$ w.r.t. classification KL can be rewrite as follows: a subset of attributes $X \subseteq M$ is a diagnostic test for a given classification KL of objects in $T(M)$ if and only if the following condition is satisfied:

$$\forall (i, j), i, j \in OBJ, i \neq j,$$

$$t_i[KL] \neq t_j[KL] \rightarrow t_i[X] \neq t_j[X]. \quad (2)$$

We call this relation *the condition of distinguishability* between objects belonging to different classes in classification KL.

Conditions (1) and (2) are equivalent. But for defining good diagnostic test for a given classification (partitioning) of objects into more then two disjoint blocks, we will use condition (1).

Let Pair(T) be the set of all pairs of objects of $T(M)$. Every partition P of $T(M)$ generates a partition of Pair(T) into two disjoint classes: PairIN(P) and PairBETWEEN(P). PairIN(P) contains all pairs of objects inside the blocks of partition P (these are pairs of objects, connected with the relation of equivalence in partition P). The set PairBETWEEN(P) is the set of all pairs of objects containing objects from different blocks of partition P.

Let us recall the definition of inclusion relation between partitions.

Definition 10. A pair of partitions P_1 and P_2 are said to be in inclusion relation $P_1 \subseteq P_2$ if and only if every block of P_1 is contained in one and only one block of P_2. The relation $\subseteq$ means that P_1 is a sub-partition of P_2.

It follows from Definition 10 that if $P1 \subseteq P2$, then PairIN($P1$) $\subseteq$ PairIN($P2$) and PairBETWEEN($P2$) $\subseteq$ PairBETWEEN($P1$).

Let $P(X)$ be the partition of $T(M)$ generated by $X \subseteq M$. Let PairsIN(X) be the set of object pairs (i, j) inside the blocks of $P(X)$, i. e., $t_i[X] = t_j[X]$.

Definition 11. A set $X \subseteq M$ is a good test or a good approximation of *KL* in *T*(*M*) if and only if the following conditions are satisfied a) *X* is a diagnostic test for *KL*; b) there does not exist a set of attributes *Z*, $Z \subseteq M$, $X \neq Z$ such that *Z* is a diagnostic test for *KL* in *T*(*M*) and PairsIN(*X*) $\subset$ PairsIN(*Z*) $\subseteq$ PairIN(*KL*).

The task of inferring good tests for approximating a given classification (partitioning) of objects into more then two disjoint blocks can be reduced to inferring good tests defined in formal context $K = (G, M, I)$ $K = K_+ \cup K_-$, where $K_+ = (G_+, M, I_+)$, $K_- = (G_-, M, I_-)$, $G = G_+ \cup G_-$ ($G_- = G \setminus G_+$) (see, please, the previous section).

This reduction consists of the following steps.

1. Construct for all pairs $\{i, j\}$, $i, j \in OBJ$ the set $E = \{F_{ij}: 1 \leq i < j \leq n\}$, where $F_{ij} = \{A \in M: t_i[A] = t_j[A]\}$.
2. Construct the partitioning of *E* into two disjoint parts: part IN of attribute sets F_{ij}, such that for corresponding t_i, t_j, $t_i[KL] = t_j[KL]$, and part BETWEEN of attribute sets F_{ij} such that for corresponding t_i, t_j, $t_i[KL] \neq t_j[KL]$.
3. Construct the set test-1(*T*,*KL*) = $\{F_{ij}$: $F_{ij} \in$ IN and $\forall$ F, $F \in$ BETWEEN $F_{ij} \not\subset F'$ & $F_{ij} \neq F'$ $\}$.

Now the indices of elements of test-1(*T*,*KL*) is considered as G_+ and the indices of elements of BETWEEN is considered as G_- of a new constructed formal context $K = (G, M, I)$, where $G = G_+ \cup G_-$, $I = I_+ \cup I_-$, and good, good maximally redundant, and good irredundant tests are defined as in Section 3. But intents of tests will be interpreted as subsets of attributes; consequently dependencies between subsets of attributes and *KL* will be interpreted as FDs. The proof of correctness of this reduction can be found in [26].

Table 4. Example for illustration of good test definition

t	A	B	C	D	E	F	G	KL
t_1	a_1	b_1	c_1	d_1	e_1	f_1	g_1	k_1
t_2	a_2	b_1	c_2	d_2	e_2	f_2	g_1	k_1
t_3	a_1	b_1	c_2	d_1	e_1	f_3	g_2	k_1
t_4	a_1	b_2	c_3	d_2	e_1	f_1	g_1	k_1
t_5	a_2	b_2	c_3	d_2	e_1	f_1	g_2	k_2
t_6	a_3	b_3	c_3	d_3	e_3	f_1	g_3	k_2

In Table 4, we have: the set IN = {*BG*, *ABDE*, *AEFG*, *DG*, *AE*}, the set BETWEEN = {*EF*, *F*, *AD*, *EG*, *BCDEF*, *CF*}, the set test-1(*T*,*KL*) = {*BG(1)*, *ABDE(2)*, *AEFG(3)*, *DG(4)*, *AE(5)*}, $G_+ = \{1, 2, 3, 4, 5\}$.

One of the ways of searching for GMRTs is based on the following theorem [26].

Theorem 1. ([26].) A GMRT for *KL* either belongs to the set test-1(*T*,*KL*) or there exists a number *q*, $2 \leq q \leq nt$, such that this test will be equal to the intersection of exactly *q* elements of the set test -1(*T*,*KL*), where *nt* is the cardinality of test -1(*T*,*KL*).

In our example, the set of good tests = {(obj(*BG*), *BG*), (obj(*AE*), *AE*), (obj(*DG*), *DG*)}. where obj(*BG*) = {1}, obj(*AE*) = {2,3,5}, obj(*DG*) = {4}. In fact, we have the following FDs: $BG \to KL$, $AE \to KL$, and $DG \to KL$ supported by the following inclusion relations $P(BG) \subseteq P(KL)$, $P(DG) \subseteq P(KL)$, $P(AE) \subseteq P(KL)$ for the reason that $P(AE) = \{\{t_1,t_3,t_4\}, \{t_2\}, \{t_5\}, \{t_6\}\}$ $P(BG) = \{\{t_1,t_2\}, \{t_3\}, \{t_4\}, \{t_5\},\{t_6\}\}$, $P(DG) = \{\{t_1\}, \{t_2, t_4\}, \{t_3\}, \{t_5\},\{t_6\}\}$, and $P(KL) = \{\{t_1, t_2, t_3, t_4\}, \{t_5,t_6\}\}$.

The equivalence between diagnostic tests, FDs, and partition dependencies allows constructing an algebraic model of diagnostic task on the basis of the partition lattice.

Now we proceed to some facts from the partition lattice theory.

Theorem 2. The system *PS* of all partitions or all equivalence relations of a set *S* forms a complete algebraic lattice (see the proof in [14].

The unit element of this lattice is the partition *P1* containing only one class – the set *S*, the zero element of this lattice is the partition *P0* in which every class is a single element of *S*.

Algebraic lattice of partitions (partition lattice) can be defined (and generated) by means of two binary operations {+,*} – addition (generalization) and multiplication (refinement). The first of these forms a partition $P_3 = P_1+P_2$ such that $P_1 \subseteq P_3$, $P_2 \subseteq P_3$, and if there exists a partition *P* in *PS* for which $P_1 \subseteq P$ and $P_2 \subseteq P$, then it implies that $P_3 \subseteq P$. Partition P_3 is the least upper bound of partitions P_1 и P_2. The second operation forms a partition $P_4 = P_1 * P_2$ such that $P_4 \subseteq P_1$, $P_4 \subseteq P_2$, and if there exists a partition *P* in *PS* for which $P \subseteq P_1$ and $P \subseteq P_2$, then it implies that $P \subseteq P_4$. Partition P_4 is the greatest lower bound of partitions P_1 and P_2.

Let $\mathfrak{I}(T)$ be the set $\{P(A), A \in M\}$. Consider the set $L(\mathfrak{I}(T))$ of partitions produced by closing atomic partitions of $\mathfrak{I}(T)$ with the use of operations addition + and multiplication * on partitions. $L(\mathfrak{I}(T))$ is the algebraic lattice with constants over *M* ([28]).

The following theorem is given without proof.

Theorem 3. ([28], [29]). Let *T*(*M*) be a table of objects. Let *P*(*X*), *P*(*Y*) be partitions of *T*(*M*) generated by $X, Y \subseteq M$, respectively. Then $T(M) \mid\!- X \to Y$ if and only if $L(\mathfrak{I}(T)) \mid\!- P(X) \subseteq P(Y)$.

Definition 12. ([29]). A set $X \subseteq M$ is a test for a given classification *KL* in table *T*(*M*) of objects, if and only if $P(X) \subseteq P(KL)$ ($P(X) \subseteq P(KL) \equiv P(X)*P(KL) = P(X) \equiv P(X) + P(KL) = P(KL)$).

Definition 13. A set $X \subseteq M$ is a good test or a good approximation of *KL* in *T*(*M*) if the following conditions are satisfied a) *X* is a test for *KL* in *T(M)*; б) there does not exist a set of attributes *Z*, $Z \subseteq U$, $X \neq Z$ such that *Z* is a test for *KL* in *T*(*M*) and $P(X) \subset P(Z) \subseteq P(KL)$.

The concept of the best diagnostic test is defined as follows.

Definition 14. A test *X*, $X \subseteq U$ is the best one for a given classification *KL* in *T*(*M*) if $(\forall Y)\ P(Y) \subseteq P(KL) \Rightarrow \|P(X)\| \leq \|P(Y)\|$.

Let us determine on the power set $(2^M, \subseteq)$ the equivalence relation Θ as follows: $X \Theta Y$ if and only if $P(X) = P(Y)$ ($P(X) \subseteq P(Y)$, $P(Y) \subseteq P(X)$).

Two subsets of attributes X, $Y \subseteq M$ belong to one and the same class of relation Θ if and only if the union of these subsets also belongs to the same class, i. e., $X \equiv Y(\Theta)$ if and only if $X \equiv X \cup Y(\Theta)$ & $Y \equiv X \cup Y(\Theta)$.

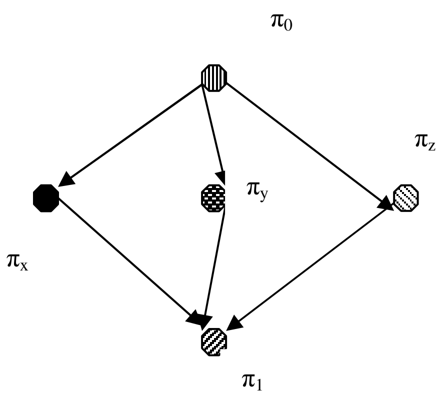
By $[a]\Theta$, where $a \in M$, we denote the equivalence class of relation Θ containing a as $[a]\Theta = \{X: X \subseteq M, X \equiv a(\Theta)\}$.

Fig. 2 gives the diagram of all possible partitions of objects in Table 5.

In table 5, we have $P(A_1) = \{r_1, r_2, r_3\}$, $P(A_2) = \{\{r_1\}, \{r_2\}, \{r_3\}\}$, $P(A_3) = \{\{r_1, r_3\}, \{r_2\}\}$, $P(A_4) = \{\{r_1, r_2\}, \{r_3\}\}$.

Table 5. An example for demonstrating the equivalence relation Θ

N	A_1	A_2	A_3	A_4
r_1	1	1	1	1
r_2	1	2	2	1
r_3	1	3	1	2



$\pi_0 = \{\{\{r_1\}, \{r_2\}, \{r_3\}\}\}$

$\pi_1 = \{\{r_1, r_2, r_3\}\}$

$\pi_x = \{\{r_1, r_3\}, \{r_2\}\}$

$\pi_y = \{\{r_1, r_2\}, \{r_3\}\}\}$

$\pi_z = \{\{r_2, r_3\}, \{r_1\}\}\}$

Fig. 2. The diagram of all possible partitions of the set $\{r_1, r_2, r_3\}$

The following is a direct consequence of the definition of relation Θ.

Proposition 1. If for two subsets of attributes X, $Y \subseteq M$ we have that $P(X)*P(Y) = P(X)$ holds, then $X \cup Y$ and X belong to one and the same equivalence class of relation Θ: $[X \cup Y]\Theta = [X]\Theta$.

For the brevity, we shall omit the sign ∪ an, therefore, $X \cup Y$ will be simply designated as XY.

Proposition 2. If, for X, $Y \subseteq M$, $P(X)*P(Y) = P(X)$ holds, then XY is a subset of maximal set of attributes belonging to the equivalence class $[X]\Theta$ of relation Θ.

Fig. 3 demonstrates the upper semi-lattice with constants over M with equivalence classes of relation Θ on attributes of Table 5. In Fig. 3, we have the following irredundant sets of attributes A_1, A_2, A_3, A_4, and A_3A_4 and the following equivalence classes $[A_1]\Theta$, $[A_2]\Theta = [A_3A_4]\Theta$, $[A_3]\Theta$, $[A_4]\Theta$ of relation Θ.

The set of partition dependencies in this example is:

$\{P(A_1A_4) = P(A_4)$; $P(A_1A_3) = P(A_3)$; $P(A_1A_2) = P(A_2)$; $P(A_2A_3) = P(A_2)$; $P(A_2A_4) = P(A_2)$; $P(A_2A_3A_4) = P(A_2)$; $P(A_2A_3A_4) = P(A_3A_4)$; $P(A_3A_4) \subset P(A_4)$; $P(A_3A_4) \subset P(A_3)\}$. These dependencies can be represented in another form: $A_4 \rightarrow A_1$; $A_3 \rightarrow A_1$; $A_2 \rightarrow A_1$; $A_2 \rightarrow A_3$; $A_2 \rightarrow A_4$; $A_2 \rightarrow A_3A_4$; $A_3A_4 \rightarrow A_2$; $A_3A_4 \rightarrow A_3$; $A_3A_4 \rightarrow A_4$.

We have also four maximally redundant attribute sets including all corresponding irredundant sets of the each equivalence class of relation Θ:

$clo(A_2) = clo(A_3A_4) = [A_1A_2A_3A_4]\Theta$
$clo(A_1) = [A_1]\Theta$
$clo(A_3) = [A_1A_3]\Theta$
$clo(A_4) = [A_1A_4]\Theta$.

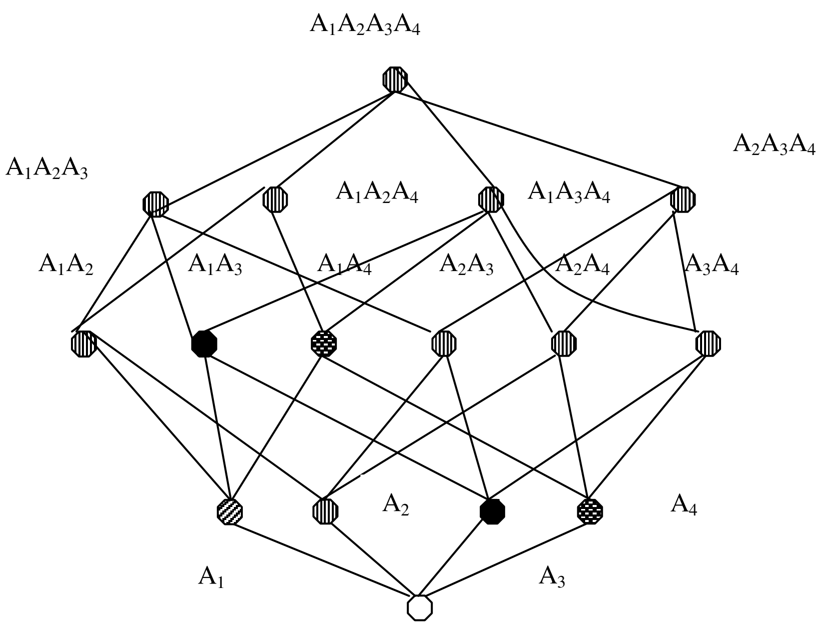


Fig. 3. The upper semi-lattice with constant over M for example in Table 5

The following definitions are important for inferring good classification tests.

Definition 15. A subset of attribute $X \subseteq M$ is said to be **maximally redundant** one if for any attribute $A \notin X$, $A \in M$, subsets AX and X belong to different equivalence classes of relation Θ, i.e., $[XA]\,\Theta \neq [X]\Theta$ for all $A \notin X$, $A \in M$.

A diagnostic test X is said to be maximally redundant one if it is maximally redundant set of attributes. If X is a good maximally redundant test for KL, then after adding to it any attribute $A \notin X$, $A \in M$, we obtain a test for KL but not good one.

Definition 16. A subset of attribute $X \subseteq M$ is said to be irredundant one, if for all Z, $Z \subset X$, Z does not belong to the equivalence class of relation Θ to which X belongs, i.e., $[X]\Theta \neq [Z]\Theta$.

A diagnostic test X is said to be irredundant one if it is irredundant set of attributes. If X is an irredundant test for KL, then deleting any attribute A, $A \in X$ from it leads to the fact that subset $X \backslash A$ is not a test for KL.

In the framework of Diagnostic Test Approach, FDs can be obtained directly by the use of the multiplication operation on partitions in Apriori level-wise manner as it is realized ([30-31]).

5 Conclusion

The concept of good diagnostic (classification) test has been given in terms of FCA and in terms of independently developed Diagnostic Test Analysis (DTA). The link between FCA and DTA approaches to classification of objects has been revealed. An algebraic model of diagnostic task as algebra of object classifications has also been described based on the partition lattice. Integration of FCA and DTA will allow organizing naturally classification-driven logical rule extraction from large datasets with resulting more relevant sets of rules with respect to special goals of data analysis.

References

1. Bellandi, A., Furletti, B., Grossi, V., Romei, A.: Ontology-driven association rule extraction: a case study. In: Proceedings of the Workshop “Context & Ontologies: Representation and Reasoning”, pp. 1–10 (2007)
2. Garriga, G.C., Kralj, P., Lavrac, N.: Closed sets for labeled data. Journal of Machine Learning Research 9, 559–580 (2008)
3. Won, D., McLeod, D.: Ontology-driven rule generalization and categorization for market data. In: Proceedings of the 23rd ICDE Workshop on Data Mining and Business Intelligence (DMBI 2007), pp. 917–923. The IEEE Computer Society (2007)
4. Marinica, C., Guillet, F.: Filtering discovered association rules using ontologies. IEEE Transactions on Knowledge and Data Engineering Journal, Special Issue “Domain-Driven Data Mining” 22(6), 784–797 (2010)
5. Olaru, A., Marinika, C., Guillet, F.: Local mining of association rules with rule schemas. In: Proceeding of the IEEE Symposium on Computational Intelligence and Data Mining. IEEE Symposium Series on Computational Intelligence, pp. 118–124. The IEEE Computer Society (2009)
6. Shen, Y.-D., Zhang, Z., Yang, Q.: Objective-oriented utility-based association mining. In: Proceedings of the IEEE International Conference on Data Mining, pp. 426–433. The IEEE Computer Society (2002)
7. Naidenova, X.A., Polegaeva, J.G.: SISIF – the System of knowledge acquisition from experimental facts. In: Alty, J.L., Mikulich, L.I. (eds.) Industrial Applications of Artificial Intelligence, pp. 87–92. Elsevier Science Publishers B.V., Amsterdam (1991)

8. Naidenova, X.A.: Reducing machine learning tasks to the approximation of a given classification on a given set of examples. In: Proceedings of the 5th National Conference at Artificial Intelligence, Kazan, Tatarstan, vol. 1, pp. 275–279 (1996) (in Russian)
9. Naidenova, X.A.: DIAGARA: an incremental algorithm for inferring implicative rules from examples. Intern. Journal "Information Theories & Applications" 12(2), 171–186 (2005)
10. Naidenova, X.A., Shagalov, V.L.: Diagnostic Test Machine. In: Auer, M. (ed.) Proceedings of the ICL 2009 – Interactive Computer Aided Learning Conference, Austria, CD, pp. 505–507. Kassel University Press (2009)
11. Maier, D.: The theory of relational databases. Computer Science Press (1983)
12. Kotsiantis, S., Kanellopoulos, D.: Association rules mining: a recent overview. GESTS International Transactions on Computer Science and Engineering 32(1), 71–82 (2006)
13. Naidenova, X.A., Polegaeva, J.G.: An algorithm of finding the best diagnostic tests. In: Mintz, G.E., Lorents, P.P. (eds.) The Application of Mathematical Logic Methods, pp. 63–67. Institute of Cybernetics, National Acad. of Sciences of Estonia, Tallinn, Estonia (1986)
14. Ore, O.: Theory of equivalence relations. Duke Mathematical Journal 9(4), 573–627 (1942)
15. Naidenova, X.A.: Automation of experimental data classification based on the algebraic lattice theory. Unpublished doctoral dissertation, Saint-Petersburg, Electro-Technical University (1979) (in Russian)
16. Kuznetsov, S.O.: Machine learning on the basis of Formal Concept Analysis. Automation and Remote Control 62(10), 1543–1564 (2001)
17. Kuznetsov, S.O.: Galois Connections in Data Analysis: Contributions from the Soviet Era and Modern Russian Research. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 196–225. Springer, Heidelberg (2005)
18. Finn, V.K.: On computer-oriented formalization of plausible reasoning in F. Bacon - J. S. Mill Style. Semiotika Inf. 20, 35–101 (1983) (in Russian)
19. Finn, V.K.: Plausible reasoning in intelligent systems of JSM-type. Itogi Nauki Tekh., Ser. Inf. 15, 54–101 (1991)
20. Ganter, B., Wille, R.: Formal Concept Analysis: mathematical foundations. Springer, Heidelberg (1999)
21. Ore, O.: Galois Connexions. Transactions of the American Mathematical Society 55(1), 493–513 (1944)
22. Birkhoff, G.: Lattice theory. Foreign Literature, Moscow (1954) (in Russian)
23. Naidenova, X.A.: Machine learning as a diagnostic task. In: Arefiev, I. (ed.) Knowledge-Dialogue-Solution, Materials of the Short-Term Scientific Seminar, pp. 26–36. State North-West Technical University, Saint-Petersburg (1992)
24. Naidenova, X.A.: Machine learning methods for commonsense reasoning processes. Interactive models. Inference Science Reference, Hershey (2009)
25. Naidenova, X.A., Plaksin, M.V., Shagalov, V.L.: Inductive inferring all good classification tests. In: Valkman, J. (ed.) "Knowledge-Dialog-Solution", Proceedings of International Conference in Two Volumes, vol. 1, pp. 79–84. Kiev Institute of Applied Informatics, Yalta (1995)
26. Naidenova, X.A.: The data-knowledge transformation. In: Soloviev, V. (ed.) Text Processing and Cognitive Technologies, Pushchino, Russia, vol. (3), pp. 130–151 (1999)
27. Juravlev, J.N.: About algebraic approach to solving the pattern recognition and classification tasks. In: Jablonskij, S.V. (ed.) The Problem of Cybernetics, vol. 33, pp. 5–68. Nauka, Moscow (1978)

28. Cosmadakis, S., Kanellakis, P.S., Spiratos, N.: Partition semantics for relations. Computer and System Sciences 33(2), 203–233 (1986)
29. Naidenova, X.A.: Relational model for analyzing experimental data. The Transaction of Acad. Sci. of USSR, Series Technical Cybernetics 4, 103–119 (1982) (in Russian)
30. Huhtala, Y., Karkkainen, J., Porkka, P., Toivonen, Y.: TANE: An Efficient algorithm for discovering functional and approximate dependencies. The Computer Journal 42(2), 100–111 (1999)
31. Megretskaya, I.A.: Construction of natural classification tests for knowledge base generation. In: Pecherskij, Y. (ed.) The Problem of Expert System Application in the National Economy: Reports of the Republican Workshop, pp. 89–93. Mathematical Institute with Computer Centre of Moldova Acad. of Sciences, Kishinev, Moldova (1988)

Modeling Preferences over Attribute Sets in Formal Concept Analysis

Sergei Obiedkov

National Research University Higher School of Economics, Moscow, Russia
sergei.obj@gmail.com

Abstract. In this paper, we consider two types of preferences from preference logic and propose their interpretation in terms of formal concept analysis. We are concerned only with preferences between sets of attributes, or, viewed logically, between conjunctions of atomic formulas. We provide inference systems for the two types of preferences and study their relation to implications.

Keywords: implications, formal concept analysis, preference logic, preferences.

1 Introduction

The formal context on the left-hand side of Fig. 1 shows the menus of four student canteens in Dresden on one particular day.[1] The symbol  stands for a vegetarian meal,  for a non-vegetarian meal without pork,  for a meal containing alcohol, and □ for a meal without any of these properties. The diagram on the right-hand side shows the preferences of an (imaginary) actor over these canteens. It should be read as a line diagram of a partial order: Bergstraße is incomparable to Klinikum, but less preferable than Reichenbachstraße, etc.

Is it possible to derive preferences over menu items from these preferences over the canteens? For example, is a meal with alcohol better than the one without for our actor? Preference logic [13,11,4] gives several ways to answer such questions. The key principle is to extend the preference relation on individual alternatives to sets of alternatives and, based on that, derive preferences between propositions about these alternatives.

In this paper, we consider two common ways to extend the preference relation and, when moving to the level of propositions, restrict ourselves to preferences between conjunctions of atomic propositions, or, to put it in terms of formal concept analysis, to preferences between sets of attributes. We provide contextual interpretations for preferences, develop inference systems, and show how preferences could be computed by translating them into implications. Our aim in this paper is to give a general characterization of preferences rather than to

[1] Jan. 19, 2004. The example is taken from [7]. Figures are produced using the `fca` LaTeX package developed by B. Ganter.

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 227–243, 2012.

				□
Bergstraße	×	×		×
Reichenbachstraße	×		×	
Klinikum	×	×		
Siedepunkt	×			

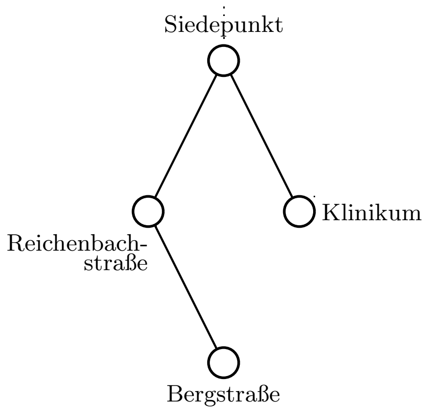


Fig. 1. A preference context

provide a practical method for their extraction from data; for this reason, we are not (yet) particularly concerned about the scalability of our approach.

We assume familiarity with the main notions and notation of formal concept analysis [8]. Here, we define a *preference context* $\mathbb{P} = (G, M, I, \leq)$ as a formal context (G, M, I) supplied with a preference relation $\leq$ on G. We write $g < h$ if $g \leq h$ and $h \not\leq g$. As usual in preference logic, we assume that $\leq$ is a preorder, i.e., it is reflexive and transitive. In general, neither antisymmetry nor totality is required, although it seems natural for a preference relation to have at least one of these properties. The preference relation of the preference context in Fig. 1 is antisymmetric; i.e., it is a partial order.

A preference context $\mathbb{P} = (G, M, I, \leq)$ can be regarded as a combination of two formal contexts: (G, M, I) and $(G, G, \leq)$. Viewed this way, a preference context is a special case of a relational context family in the sense of relational concept analysis (RCA) [12]. However, our goals are different from those generally pursued in RCA, and so is our approach—except for when we define the conceptual existential translation of a preference context (Definition 5) in a way similar to how the wide scaling operator is defined in RCA.

Throughout the text, we use $(\cdot)'$ to denote the derivation operators of (G, M, I) and $(\cdot)^{\leq}$ and $(\cdot)^{\geq}$ to denote the two derivation operators of $(G, G, \leq)$.

2 Universal Preferences

In von Wright's version of preference logic [13], a set Y is preferred to a set X if

$$\forall x \in X \forall y \in Y (x \leq y),$$

that is, every alternative in Y is preferred to every alternative in X. Translating this into preferences over sets of attributes, we obtain the following definition:

Definition 1. *A set of attributes* $B \subseteq M$ *is* universally preferred *to a set of attributes* $A \subseteq M$ *in a preference context* $\mathbb{P} = (G, M, I, \leq)$ *if* $B' \subseteq A'^{\leq}$. *Notation:* $\mathbb{P} \models A \rightarrowtail B$.

That is, $A \rightarrowtail B$ holds (or is valid) in $(G, M, I, \leq)$ if every object with all attributes from B is preferred to every object with all attributes from A. We will call A the premise and B the conclusion of the universal preference $A \rightarrowtail B$. We will sometimes omit curly brackets in premises and conclusions when giving examples of preferences.

Note that, as long as there is no object g with $B \subseteq g'$, we have both $A \rightarrowtail B$ and $B \rightarrowtail A$ for any $A \subseteq M$: in the former case, $\varnothing = B' \subseteq A'^{\leq}$, and, in the latter case, $A' \subseteq B'^{\leq} = \varnothing^{\leq} = G$.

On the other hand, if $B' = G$, the preference $A \rightarrowtail B$ takes place if and only if objects with A are the "worst" objects in G: every object from G (including those from A) is preferred to every object from A. Similarly, $B \rightarrowtail A$ (with $B' = G$) would hold if objects with A are preferable to each other and all other objects.

Example 1.

, , $\square \rightarrowtail \varnothing$

does not hold in the preference context from Fig. 1, for {, , $\square$} $\subseteq$ Bergstraße, but Bergstraße $\not\leq$ Klinikum.

$\varnothing \rightarrowtail$ 

does not hold either, because {  } $\subseteq$ Reichenbachstraße, but Klinikum $\not\leq$ Reichenbachstraße. On the other hand,

$\square \rightarrowtail$ 

does hold, since the only option with  is preferable to the only option with $\square$, i.e., Bergstraße $\leq$ Reichenbachstraße.

The inference system for universal preferences consists of a single rule:

$$\frac{X \rightarrowtail Y}{X \cup U \rightarrowtail Y \cup V}, \tag{1}$$

which allows one to add arbitrary attributes to the premise and conclusion of a valid preference.

Proposition 1. *Rule (1) is sound and complete with respect to universal preferences.*

Proof. **Soundness.** We need to show that

$$\text{if } (G, M, I, \leq) \models X \rightarrowtail Y, \text{ then } (G, M, I, \leq) \models X \cup U \rightarrowtail Y \cup V$$

for any $X, Y, U, V \subseteq M$. From $(G, M, I, \leq) \models X \rightarrowtail Y$, we get $Y' \subseteq X'^{\leq}$. Taking into account $(X \cup U)' \subseteq X'$, we obtain $(Y \cup V)' \subseteq Y' \subseteq X'^{\leq} \subseteq (X \cup U)'^{\leq}$ and $(G, M, I, \leq) \models X \cup U \rightarrowtail Y \cup V$ as claimed.

Completeness. By completeness, we mean that, if $A \rightarrowtail B$ holds for all preference contexts satisfying all the preferences in set Σ, then $A \rightarrowtail B$ can be deduced from Σ using rule (1). In fact, we will prove a stronger claim: in this situation, $A \rightarrowtail B$ can be deduced from a single preference in Σ; in other words, Σ must contain a preference $X \rightarrowtail Y$ with $X \subseteq A$ and $Y \subseteq B$.

Suppose that Σ contains no such preference. Then, consider a preference context $\mathbb{P}$ with only two objects, $g_1 < g_2$, such that $g_1' = B$ and $g_2' = A$. Note that $g_2 \in H^{\leq}$ for any $H \subseteq G$. Obviously, $A \rightarrowtail B$ is invalid in $\mathbb{P}$. Take any $X \rightarrowtail Y \in \Sigma$. If $X \subseteq A$, then $Y \not\subseteq B$ and $Y' \subseteq \{g_2\}$. Consequently, $Y' \subseteq X'^{\leq}$ and $X \rightarrowtail Y$ holds in $\mathbb{P}$. If $X \not\subseteq A$, then $g_2 \notin X'$ and $X'^{\leq} = \{g_1, g_2\}$. Hence, $X \rightarrowtail Y$ holds in this case, too.

Thus, if Σ contains no preference $X \rightarrowtail Y$ with $X \subseteq A$ and $Y \subseteq B$, there is a preference context where every preference from Σ holds, but $A \rightarrowtail B$ does not. □

Universal preferences are very different from implications: even $X \rightarrowtail X$ does not always hold (for example,  $\rightarrowtail$  does not hold in the context from Fig. 1, since incomparable canteens Bergstraße and Klinikum both serve ). Nevertheless, we are now going to describe a formal context $\mathbb{K}_\forall^{\mathbb{P}}$ corresponding to a preference context $\mathbb{P}$ in such a way that valid universal preferences of $\mathbb{P}$ are translated into valid implications of $\mathbb{K}_\forall^{\mathbb{P}}$, while invalid preferences are translated into invalid implications.

Definition 2. *Let* $\mathbb{P} = (G, M, I, \leq)$ *be a preference context. The* universal translation *of* $\mathbb{P}$ *is a formal context* $\mathbb{K}_\forall^{\mathbb{P}} = (G \times G, (M \times \{1,2\}) \cup \{\leq\}, I_\forall)$, *where*

$$\begin{aligned} (g_1, g_2) I_\forall m_1 &\iff g_1 I m_1, \\ (g_1, g_2) I_\forall m_2 &\iff g_2 I m_2, \\ (g_1, g_2) I_\forall \leq &\iff g_1 \leq g_2. \end{aligned}$$

Here, m_1 *and* m_2 *stand for* $(m,1)$ *and* $(m,2)$ *respectively,* $m \in M$. *We denote the derivation operators of* $\mathbb{K}_\forall^{\mathbb{P}}$ *by* $(\cdot)^\forall$.

$T_\forall(A \rightarrowtail B)$, *the* translation of a universal preference $A \rightarrowtail B$, *is the implication*

$$(A \times \{1\}) \cup (B \times \{2\}) \to \{\leq\}$$

of the formal context $\mathbb{K}_\forall^{\mathbb{P}}$.

Example 2. The universal translation of the preference context from Fig. 1 is shown in Fig. 2. Below we give translations of the universal preferences from Example 1 (without curly brackets):

, , $\square \rightarrowtail \varnothing$ $_1$, $_1$, $\square_1 \to \leq$

$\varnothing \rightarrowtail$    $_2 \to \leq$

$\square \rightarrowtail$  $\square_1$, $_2 \to \leq$

It is easy to see that the translations of the first two preferences in Example 2 are not valid in the context shown in Fig. 2, while the third translation is valid.

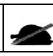
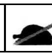
	 $_1$	 $_1$	 $_1$	$\square_1$	$\leq$	 $_2$	 $_2$	 $_2$	$\square_2$
Bergstraße, Bergstraße	×	×		×	×	×	×		×
Bergstraße, Reichenbachstr.	×	×		×	×	×		×	
Bergstraße, Klinikum	×	×		×		×	×		
Bergstraße, Siedepunkt	×	×		×	×	×			
Reichenbachstr., Bergstraße	×		×			×	×		×
Reichenbachstr., Reichenbachstr.	×		×		×	×		×	
Reichenbachstr., Klinikum	×		×			×	×		
Reichenbachstr., Siedepunkt	×		×		×	×			
Klinikum, Bergstraße	×	×				×	×		×
Klinikum, Reichenbachstr.	×	×				×		×	
Klinikum, Klinikum	×	×			×	×	×		
Klinikum, Siedepunkt	×	×			×	×			
Siedepunkt, Bergstraße	×					×	×		×
Siedepunkt, Reichenbachstr.	×					×		×	
Siedepunkt, Klinikum	×					×	×		
Siedepunkt, Siedepunkt	×				×	×			

Fig. 2. The universal translation of the preference context from Fig. 1

This agrees with the validity of the corresponding universal preferences in the original preference context from Fig. 1. More generally:

Proposition 2. *A universal preference* $A \rightarrowtail B$ *is valid in a preference context* $\mathbb{P} = (G, M, I, \leq)$ *if and only if its translation is valid in* $\mathbb{K}_{\forall}^{\mathbb{P}}$*:*

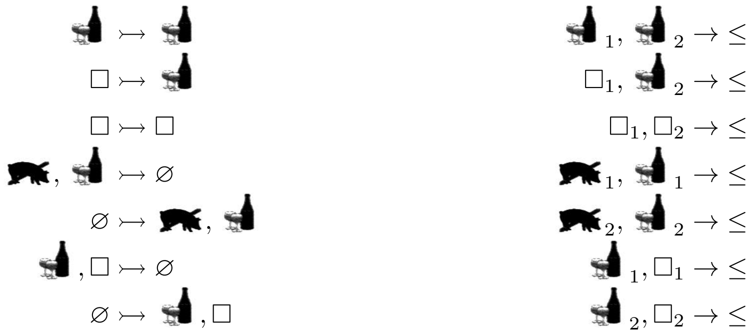
$$\mathbb{P} \models A \rightarrowtail B \quad \iff \quad \mathbb{K}_{\forall}^{\mathbb{P}} \models T_{\forall}(A \rightarrowtail B).$$

Proof. Suppose that $\mathbb{P} \models A \rightarrowtail B$ and $(A \times \{1\}) \cup (B \times \{2\}) \subseteq (g_1, g_2)^{\forall}$ for some $g_1 \in G$ and $g_2 \in G$. Then, $A \subseteq g_1'$ and $B \subseteq g_2'$. Since, $A \rightarrowtail B$ holds in $\mathbb{P}$, we have $g_1 \leq g_2$ and $(g_1, g_2) I_{\forall} \leq$ as required.

Conversely, assume $\mathbb{K}_{\forall}^{\mathbb{P}} \models (A \times \{1\}) \cup (B \times \{2\}) \to \{\leq\}$. We need to show that $B' \subseteq A'^{\leq}$, that is, $g_1 \leq g_2$ whenever $A \subseteq g_1'$ and $B \subseteq g_2'$. Indeed, in the latter case, we have $(A \times \{1\}) \cup (B \times \{2\}) \subseteq (g_1, g_2)^{\forall}$ and, consequently, $(g_1, g_2) I_{\forall} \leq$, i.e., $g_1 \leq g_2$. $\square$

The canonical (Duquenne–Guigues) basis [9] of $\mathbb{K}_{\forall}^{\mathbb{P}}$ in Fig. 2 consists of twelve implications, but, since the translation is not surjective, most of these implications do not correspond to preferences of the preference context in Fig. 1. If we are interested only in preferences, a better representation can be obtained by considering minimal generating sets of $\leq$ in $\mathbb{K}_{\forall}^{\mathbb{P}}$, i.e., minimal attribute sets A such that $\leq \in A^{\forall\forall}$.

Example 3. For our example, this would give us the following set of universal preferences (shown to the left with their translations to the right):



This is the minimal basis of the universal preferences of the preference context in Fig. 1,[2] as the following Proposition shows:

Proposition 3. *Let* $\mathbb{P}$ *be a preference context. The*

$$\Sigma = \{A \rightarrowtail B \mid (A \times \{1\}) \cup (B \times \{2\}) \text{ is minimal}$$
$$\text{w.r.t. } \mathbb{K}^{\mathbb{P}}_{\forall} \models (A \times \{1\}) \cup (B \times \{2\}) \to \{\leq\}\}$$

is the minimal (in the number of preferences) basis of the universal preferences valid in $\mathbb{P}$*.*

Proof. Due to Proposition 2, all universal preferences from Σ are valid in $\mathbb{P}$. Proposition 1 ensures that the only way to obtain a valid universal preference from other valid preferences is to add attributes to their premises and/or conclusions. Taking into account that both premises and conclusions of preferences in Σ are minimal among premises and conclusions of valid preferences, it is easy to see that Σ is indeed the (unique) minimal basis of universal preferences of $\mathbb{P}$. □

Although, as said above, the scalability is not our concern here, a short remark is worth making. If $|G|$ is large, it may not be feasible to work directly with $\mathbb{K}^{\mathbb{P}}_{\forall}$, which contains $|G|^2$ objects. In this case, an approach based on query learning (as opposed to learning from examples) can be used [1]. In this setting, there is a "learner" and a "teacher"; the learner proposes a hypothesis of a certain kind to the teacher, and the teacher either confirms it or provides a counterexample, based on which the learner forms a new hypothesis. Thus, the learner may never see the entire set of examples, but still be able to learn whatever it is learning. A similar approach in formal concept analysis is attribute exploration: the learner starts with a small (possibly empty) context, generates implications valid in this context, and asks the teacher to provide counterexamples, which it then adds to the context [6]. While learning universal preferences, the learner is interested

[2] As should be expected, by saying that Σ is a basis of universal preferences of $\mathbb{P}$, we mean that Σ contains only universal preferences valid in $\mathbb{P}$ and that all other such preferences are valid exactly in contexts where Σ is valid. Σ must also be irredundant with respect to this property.

only in implications of the form $(A \times \{1\}) \cup (B \times \{2\}) \to \{\leq\}$, where $A, B \subseteq M$. To find a counterexample for such an implication, the teacher would identify A' and B' in $\mathbb{P}$ and then check all pairs in $A' \times B'$, eliminating the need to store the entire $\mathbb{K}_{\forall}^{\mathbb{P}}$ explicitly.

If we already have information about implications holding in (G, M, I), a different approach to exploring preferences in $\mathbb{P} = (G, M, I, \leq)$ might be helpful. If $\mathcal{L}$ is a set of implications valid in (G, M, I) (for example, but not necessarily, its Duquenne–Guigues basis), we may compute the canonical basis of $\mathbb{K}_{\forall}^{\mathbb{P}}$ *relative* to the following implications:

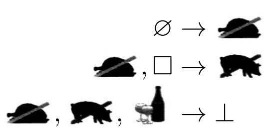
$$\{A \times \{i\} \to B \times \{i\} \mid A \to B \in \mathcal{L}, i \in \{1, 2\}\}.$$

In fact, implications $A \to B \in \mathcal{L}$ with zero support, i.e., with $A' = \varnothing$, give rise to stronger background implications, with $M \times \{1, 2\}$ as the conclusion. So, we may also add the following implications to the background knowledge:

$$\{A \times \{i\} \to M \times \{1, 2\} \mid A \to B \in \mathcal{L}, i \in \{1, 2\} \text{ and } A' = \varnothing\}.$$

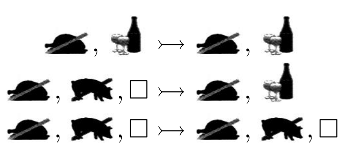
All implications in the relative canonical basis of $\mathbb{K}_{\forall}^{\mathbb{P}}$ will contain the "preference attribute", $\leq$, either in their premises or in their conclusions. However, we are interested only in implications with $\leq$ in the conclusion. It is easy to generate only such implications using Ganter's algorithm [8], which produces implications in the lectic order of their premises with respect to a given linear order on attributes:[3] the trick is to set $\leq$ to be the first attribute in $\mathbb{K}_{\forall}^{\mathbb{P}}$ and terminate the algorithm before it starts generating premises containing $\leq$.

Example 4. The Duquenne–Guigues basis of the formal context in Fig. 1 consists of three implications:



Here, $\perp$ in the conclusion of the third implication stands for M and marks the implication as zero-support. We will see shortly how this can be used.

Computing the canonical basis of $\mathbb{K}_{\forall}^{\mathbb{P}}$ relative to the background implications obtained from these three implications as described above, we get the following preferences (before the algorithm would start generating implications with $\leq$ in the premise):



[3] A set $A \subseteq M$ is lectically smaller than a set $B \subseteq M$, if the first (w.r.t. a given linear order on M) attribute in which A and B differ belongs to B.

Of course, this system of three preferences is equivalent to the system of seven preferences from Example 3 only under the given background knowledge. Obviously, each preference from the smaller set can be derived from one of the seven preferences using rule (1). Inference in the other direction requires Armstrong rules [2] for implications, rule (1) for universal preferences, and three additional rules to connect implications with preferences:

$$\frac{X \to \bot}{\varnothing \rightarrowtail X, \quad X \rightarrowtail \varnothing}, \quad \frac{X \to Y, \quad X \cup Y \rightarrowtail Z}{X \rightarrowtail Z}, \quad \frac{X \to Y, \quad Z \rightarrowtail X \cup Y}{Z \rightarrowtail X}.$$

Example 5. To check that the preference $\square \rightarrowtail$  follows from the hybrid system of implications and universal preferences of Example 4, we first derive two implications using the Armstrong rules:

$\square \to$ , , $\square$ and  $\to$ , .

From these two implications and the universal preference

, , $\square \rightarrowtail$ , 

we obtain $\square \rightarrowtail$  by the last two of the additional rules.

To derive $\varnothing \rightarrowtail$ , $\square$, which is a zero-support preference in the sense that there are no canteens serving both  and $\square$, we first derive the implication

, $\square \to \bot$

and then use the first additional rule to get the desired preference.

To sum up, we described two approaches to the representation of universal preferences: one involves a set of universal preferences together with a single inference rule, while the other combines implications and preferences and requires a more sophisticated inference system. For the preference context in Fig. 1, the first approach results in seven universal preferences, whereas the second approach yields three implications and three preferences. The second approach effectively separates knowledge about preferences from knowledge about the structure of the underlying formal context. For some applications, such separation may be valuable.

The two approaches suggest two ways to design universal preference exploration (by analogy with attribute exploration [6]). One is to go through universal preferences based on minimal generating sets of $\leq$ in $\mathbb{K}_{\forall}^{\mathbb{P}}$ (see Example 3). If a universal preference $A \rightarrowtail B$ valid in the current preference context $\mathbb{P}$ does not hold in general, the user must provide two objects g and h with $A \subseteq g'$, $B \subseteq h'$, and $g \not\leq h$. One of these two objects may already be contained in $\mathbb{P}$; the other must be new.

The other approach is to start with standard attribute exploration of the formal context underlying $\mathbb{P}$ (however, the $\leq$ relation must also be filled for objects added during this stage) and, when that is finished, move to the exploration of universal preferences obtained from the relative basis (see Example 4). Here, as in the first approach, counterexamples are pairs of objects, but they must respect the implications accepted during attribute exploration.

We finish this section by pointing out at a slightly different way to look at universal preferences. Given a preference context $\mathbb{P} = (G, M, I, \leq)$, the formal concepts of $(G, G, \leq)$ summarize the preference relation on object sets that gives rise to universal preferences between attribute sets (see the beginning of this section). Indeed, an object set Y is preferred to an object set X (in the sense of von Wright) if and only if $Y \subseteq X^{\leq}$. Sets X and Y are maximal with respect to this property if and only if (X, Y) is a formal concept of $(G, G, \leq)$. At the same time, if an object set Y is preferred to an object set X, then any subset of Y is preferred to any subset of X. Thus, concepts of $(G, G, \leq)$ provide a complete representation of von Wright's preferences on object sets. We leave it for further work to study the relation between this representation and universal preferences on attribute sets.

3 Existential Preferences

Another type of preferences often used in preference logic derives from the following extension of the $\leq$ relation between individual alternatives to sets of alternatives: a set Y is preferred to a set X if

$$\forall x \in X \exists y \in Y (x \leq y),$$

that is, for every alternative in X, one can find an alternative in Y that is at least as good. If alternatives are described by attribute sets, this reads as follows:

Definition 3. *A set of attributes $B \subseteq M$ is* existentially preferred *to a set of attributes $A \subseteq M$ in a preference context $\mathbb{P} = (G, M, I, \leq)$ if*

$$A' \subseteq \bigcup_{g \in B'} g^{\geq}.$$

Notation: $\mathbb{P} \models A \mapsto B$.

In other words, $A \mapsto B$ is valid if, for every object with all attributes from A, one can find an object with all attributes from B that is at least as good.

It may seem that the terms "strong preferences" and "weak preferences" suggest themselves for universal and existential preferences respectively. We chose not to use them, because, if $A' \neq B' = \varnothing$, then $A \rightarrowtail B$ holds, but $A \mapsto B$ does not.

Example 6. Unlike the corresponding universal preference, the existential preference

, , $\square \mapsto \varnothing$

holds in the preference context from Fig. 1, since $\{$  ,  , $\square\}' = \{\text{Bergstraße}\}$ and Bergstraße $\leq$ Bergstraße $\in \varnothing'$. On the other hand,

$$\varnothing \mapsto$$ 

does not hold (similarly to the corresponding universal preference), because Klinikum $\in \varnothing'$ and $\{$  $\}' =$ Reichenbachstraße, but Klinikum $\not\leq$ Reichenbachstraße. The existential preference

$$\square \mapsto$$ 

holds for the same reason as its universal counterpart.

Note that an existential preference is a generalization of an implication: firstly, every valid implication $A \to B$ gives rise to a valid preference $A \mapsto B$; secondly, if $\leq$ is the identity relation, then valid existential preferences are exactly all the valid implications.

Proposition 4. *Let $\mathbb{P} = (G, M, I, \leq)$ be a preference context.*

1. *If $(G, M, I) \models A \to B$, then $\mathbb{P} \models A \mapsto B$.*
2. *If $\leq$ is the identity relation and $\mathbb{P} \models A \mapsto B$, then $(G, M, I) \models A \to B$.*

Proof. 1. $(G, M, I) \models A \to B$ implies $A' \subseteq B'$, but, since $g \leq g$ for all $g \in G$, we have $B' \subseteq \bigcup_{g \in B'} g^{\geq}$. Therefore, $\mathbb{P} \models A \mapsto B$.

2. If $\leq$ is the identity relation, then $\bigcup_{g \in B'} g^{\geq} = B'$. Hence, $\mathbb{P} \models A \mapsto B$ only when $A' \subseteq B'$, i.e., when $(G, M, I) \models A \to B$.

The inference system for existential preferences is a weakened version of the Armstrong system for implications:

Proposition 5. *The following system of three rules is sound and complete with respect to existential preferences:*

$$\frac{}{X \mapsto X}, \qquad \frac{X \mapsto Y \cup U}{X \cup V \mapsto Y}, \qquad \frac{X \mapsto Y, \quad Y \mapsto Z}{X \mapsto Z}. \tag{2}$$

Proof. **Soundness** is easy; we will concentrate on **completeness**. Suppose that $A \mapsto B$ cannot be derived from a set Σ of existential preferences using rules (2). We will build a preference context $\mathbb{P}_\Sigma = (G, M, I, \leq)$ where all preferences from Σ hold, but $A \mapsto B$ does not.

We start with $G = \{g_0\}$ such that $g_0' = A$. We add new objects to G as follows: if $X \subseteq g'$ for some $X \mapsto Y \in \Sigma$ and $g \in G$, we add to G an object h with $h' = Y$ if such an h is not yet in G. In the end, we set $\leq$ equal to $G \times G$. Obviously, all existential preferences from Σ hold in $\mathbb{P}_\Sigma$. To show that $A \mapsto B$ does not hold, we will prove that $B \not\subseteq g'$ for all $g \in G$.

Let us index objects in G according to the order in which they were added to G. Obviously, $B \not\subseteq g_0' = A$: otherwise, we could derive $A \mapsto B$ using the first two rules. Suppose for contradiction that $B \subseteq g'_k$ for some $k > 0$. Then, there is an existential preference $X \mapsto Y \in \Sigma$ with $B \subseteq Y = g'_k$ and $X \subseteq g'_j$ for some $j < k$. Therefore, we can derive $X \mapsto B$ from Σ using the second rule. If $j = 0$, then $X \subseteq A = g'_0$ and $A \mapsto B$ is derivable from Σ (again using the second rule and augmenting the premise of $X \mapsto B$). Otherwise, there must be $i < j$ and $U \mapsto V \in \Sigma$ such that $X \subseteq V = g'_j$ and $U \subseteq g'_i$. From $U \mapsto V$ we derive $U \mapsto X$ and then, using $X \mapsto B$ and the third rule, $U \mapsto B$. Continuing like this, we will eventually derive $W \mapsto B$ for some $W \subseteq A = g'_0$, from which $A \mapsto B$ is derived by the second rule. However, $A \mapsto B$ is not derivable from Σ. Therefore, our assumption that $B \subseteq g'_k$ is wrong, and $B \not\subseteq g'$ for all $g \in G$.

Now, $g_0 \in A'$, but $\bigcup_{g \in B'} g^{\geq} = \bigcup \varnothing = \varnothing$. Therefore, $A \mapsto B$ does hold in $\mathbb{P}_\Sigma$, as claimed. □

Next, we develop a translation of existential preferences into implications similarly to how this was done for universal preferences in Definition 2.

Definition 4. *Let* $\mathbb{P} = (G, M, I, \leq)$ *be a preference context. The* existential translation *of* $\mathbb{P}$ *is a formal context* $\mathbb{K}^{\mathbb{P}}_{\exists} = (G, \mathfrak{P}(M), I_{\exists})$*, where* $\mathfrak{P}(M)$ *is the power set of* M *and*

$$gI_{\exists}A \iff g^{\leq} \cap A' \neq \varnothing.$$

We denote the derivation operators of $\mathbb{K}^{\mathbb{P}}_{\exists}$ *by* $(\cdot)^{\exists}$*.*

$T_{\exists}(A \mapsto B)$*, the* translation of an existential preference $A \mapsto B$*, is the implication*

$$\{A\} \to \{B\}$$

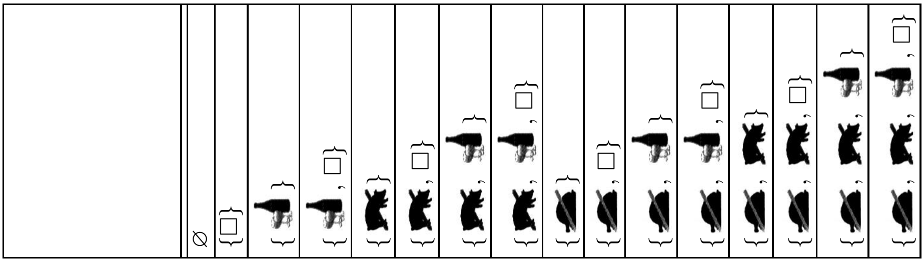
of the formal context $\mathbb{K}^{\mathbb{P}}_{\exists}$*.*

Example 7. The existential translation of the preference context from Fig. 1 is shown in Fig. 3. Below we give translations of the existential preferences from Example 6 (here, we omit curly brackets in preferences, but keep them in the translations to emphasize that implications are between sets of sets):

From Fig. 3, it is easy to see that the first and third implications hold in $\mathbb{K}^{\mathbb{P}}_{\exists}$, while the second one does not.

Proposition 6. *An existential preference* $A \mapsto B$ *is valid in a preference context* $\mathbb{P} = (G, M, I, \leq)$ *if and only if its translation is valid in* $\mathbb{K}^{\mathbb{P}}_{\exists}$*:*

$$\mathbb{P} \models A \mapsto B \quad \iff \quad \mathbb{K}^{\mathbb{P}}_{\exists} \models T_{\exists}(A \mapsto B).$$



Bergstraße	×	×	×		×	×			×	×	×		×	×		
Reichenbachstr.	×		×						×		×					
Klinikum	×				×				×				×			
Siedepunkt	×								×							

Fig. 3. The existential translation of the preference context from Fig. 1

Proof. Suppose that $\mathbb{P} \models A \mapsto B$ and $A \in g_1^{\exists}$ for some $g_1 \in G$. Then, $g_1^{\leq} \cap A' \neq \varnothing$, i.e., there is $g_2 \in A'$ with $g_1 \leq g_2$. Since $A \mapsto B$ holds in $\mathbb{P}$, there is $g_3 \in B'$ such that $g_2 \leq g_3$. Due to transitivity of $\leq$, we have $g_1 \leq g_3$ and $g_1 I_{\exists} B$ as required.

Conversely, assume $\mathbb{K}_{\exists}^{\mathbb{P}} \models \{A\} \to \{B\}$. We need to show that $A' \subseteq \bigcup_{g \in B'} g^{\geq}$, that is, $g_1 \leq g_2$ for some $g_2 \in B'$ whenever $A \subseteq g_1'$. Indeed, in the latter case, we have $g_1 I_{\exists} A$ and, consequently, $g_1 I_{\exists} B$, which holds exactly when $g_1 \leq g_2$ for some $g_2 \in B'$. $\square$

The canonical basis [9] of $\mathbb{K}_{\exists}^{\mathbb{P}}$ in Fig. 3 consists of 16 implications, but, as it was the case with $\mathbb{K}_{\forall}^{\mathbb{P}}$, most of them are of little interest from the viewpoint of computing preferences, since existential preferences correspond to implications of $\mathbb{K}_{\exists}^{\mathbb{P}}$ with single-element premises. More relevant is the system of existential preferences given by the following set:

$$\{A \mapsto B \mid A \text{ is minimal and } B \text{ is maximal w.r.t. } \mathbb{K}_{\exists}^{\mathbb{P}} \models \{A\} \to \{B\}\}. \qquad (3)$$

To see that this system is complete, note that every valid existential preference can be derived from this set using the second rule of (2)—except for trivial preferences $X \mapsto X$, which can be derived with the first rule. However, the third rule, transitivity, is not necessary with this system. Given that the third rule is necessary in general, it should be clear that this system may be redundant. Indeed, it may contain preferences $\{a\} \mapsto \{b\}$, $\{b\} \mapsto \{c\}$, and $\{a\} \mapsto \{c\}$, even though the third preference is a consequence of the first two.[4] Therefore, some reduction may be needed to obtain a minimal representation from (3).

Example 8. Here is the system of existential preferences defined by (3) for the preference context in Fig. 1 (shown to the left with their translations to the right):

[4] The preferences $\{a\} \mapsto \{b\}$ and $\{a\} \mapsto \{c\}$ cannot be replaced by $\{a\} \mapsto \{b, c\}$, since the latter does not follow from them.

$\varnothing \mapsto$ 🦃	$\{\varnothing\} \to \{\{$🦃$\}\}$
$\square \mapsto$ 🦃, 🍷	$\{\square\} \to \{\{$🦃, 🍷$\}\}$
$\square \mapsto$ 🦃, 🐖, $\square$	$\{\square\} \to \{\{$🦃, 🐖, $\square\}\}$
🍷 $\mapsto$ 🦃, 🍷	$\{\{$🍷$\}\} \to \{\{$🦃, 🍷$\}\}$
🍷, $\square \mapsto$ 🦃, 🐖, 🍷, $\square$	$\{\{$🍷, $\square\}\} \to \{\{$🦃, 🐖, 🍷, $\square\}\}$
🐖 $\mapsto$ 🦃, 🐖	$\{\{$🐖$\}\} \to \{\{$🦃, 🐖$\}\}$
🐖, 🍷 $\mapsto$ 🦃, 🐖, 🍷, $\square$	$\{\{$🐖, 🍷$\}\} \to \{\{$🦃, 🐖, 🍷, $\square\}\}$

Note that, in Example 8, we need the preference 🐖 $\mapsto$ 🦃, 🐖 even in the presence of $\varnothing \mapsto$ 🦃: it may happen that there is "the best" object with 🦃, but no object with both 🦃 and 🐖. However, in our example $\varnothing \mapsto$ 🦃 is not only an existential preference, but also an implication. This means that, if there is an object with 🐖, this object has both 🦃 and 🐖. In other words, the existential preference 🐖 $\mapsto$ 🦃, 🐖 follows from the implication $\varnothing \to$ 🦃. Thus, as with universal preferences, it may be useful to consider a hybrid system of implications and existential preferences.

A set $\mathcal{L}$ of implications valid in (G, M, I) gives rise to the following background implications of $\mathbb{K}_{\exists}^{\mathbb{P}}$ for every $A \subseteq M$:

$$\{A\} \to \{\mathcal{L}(A)\}$$

where $\mathcal{L}(A)$ is the closure of A with respect to the implications in $\mathcal{L}$. In addition, we use the following background implications for every $A \subseteq M$:

$$\{A\} \to \{A \setminus \{a\} \mid a \in A\},$$

as well as

$$\varnothing \to \{\varnothing\}.$$

Example 9. Setting $\mathcal{L}$ to be the canonical basis of the formal context $\mathbb{P}$ in Fig. 1 (see Example 4) and computing the implications of $\mathbb{K}_{\exists}^{\mathbb{P}}$ with one-element premises relative to the background implications just defined, we obtain a single existential preference:

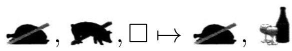
🦃, 🐖, $\square \mapsto$ 🦃, 🍷.

Compared to Example 8, this is a noticeable improvement on the size of the representation (three implications and one preference instead of seven preferences), but it requires a hybrid inference system for implications and existential preferences. Such system consists of Armstrong rules for implications, rules (2) for existential preferences, and one additional rule relating implications and preferences, which is based on Proposition 4:

$$\frac{A \to B}{A \mapsto B}.$$

Example 10. The existential preference $\square \mapsto$  can be derived in the hybrid system as follows. First, we derive the implication

from the canonical basis (see Example 4) with Armstrong rules. Then, we use the additional rule to transform this implication into a preference:

From this and , we finally obtain

using the third rule from (2).

The formal context in Fig. 3 strikes as rather uneconomical: for example, it consists of two identical parts, which is due to the fact that the original context satisfies the implication $\varnothing \to$ . We will now describe an alternative translation, which makes a better use of the structure of the original context. The idea is to use only concept intents (rather than all subsets of M) as attributes of the formal context into which the translation is done.

Definition 5. *Let* $\mathbb{P} = (G, M, I, \leq)$ *be a preference context. The* conceptual existential translation *of* $\mathbb{P}$ *is a formal context* $\mathbb{C}_\exists^\mathbb{P} = (G, \underline{\mathfrak{B}}(G, M, I), I_\exists)$, *where* $\underline{\mathfrak{B}}(G, M, I)$ *is the concept set of the formal context* (G, M, I) *and*

$$gI_\exists(A, B) \iff g^{\leq} \cap A \neq \varnothing.$$

We denote the derivation operators of $\mathbb{K}_\exists^\mathbb{P}$ *by* $(\cdot)^\exists$.

$T_\exists^\mathbb{C}(A \mapsto B)$, *the* conceptual translation of an existential preference $A \mapsto B$, *is the implication*

$$\{(A', A'')\} \to \{(B', B'')\}$$

of the formal context $\mathbb{C}_\exists^\mathbb{P}$.

Readers familiar with relational concept analysis will notice that a similar construction is used there under the name of the wide scaling operator [12]: the context resulting from such scaling applied to $\mathbb{P}$ is the apposition of (G, M, I) and $\mathbb{C}_\exists^\mathbb{P}$. The wide scaling operator is intended to capture the semantics of existential role restrictions as defined in description logics [3], and, indeed, it is easy to see that an existential preference $A \mapsto B$ corresponds to the terminiological axiom $A \sqsubseteq \exists R_\leq.B$ (where $R_\leq$ is the role interpreted by the preference relation $\leq$).[5]

Example 11. The conceptual existential translation of the preference context from Fig. 1 is shown in Fig. 4. Below we give conceptual translations (in an abbreviated form) of the existential preferences from Example 6:

[5] Note that a universal preference $A \rightarrowtail B$ corresponds not to the axiom $A \sqsubseteq \forall R_\leq.B$, but to the axiom $A \sqsubseteq \forall\neg R_\leq.\neg B$.

As can be seen from the line diagram of the concept lattice of $\mathbb{C}^{\mathbb{P}}_{\exists}$ in Fig. 4, the first and the third implications are valid, while the second one is not.

	(BRKS,)	(R,)	(BK,)	(B, □)	(∅, □)
Bergstraße	×	×	×	×	
Reichenbachstraße	×	×			
Klinikum	×		×		
Siedepunkt	×				

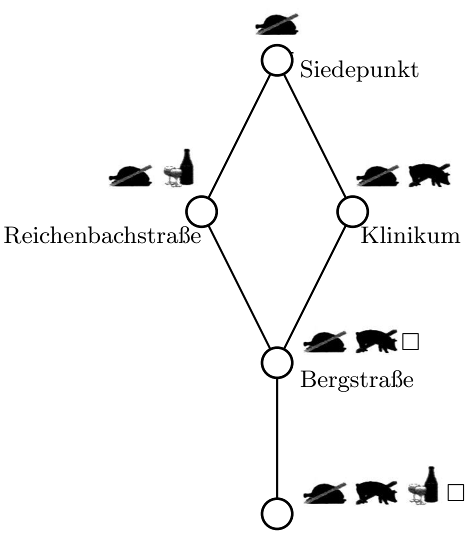


Fig. 4. The conceptual existential translation of the preference context from Fig. 1 and its concept lattice. The attributes are the formal concepts of the formal context underlying the original preference context; only concept intents are used as attribute labels in the diagram.

Proposition 7. *An existential preference $A \mapsto B$ is valid in a preference context $\mathbb{P} = (G, M, I, \leq)$ if and only if its conceptual translation is valid in $\mathbb{C}_{\exists}^{\mathbb{P}}$:*

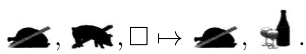
$$\mathbb{P} \models A \mapsto B \quad \iff \quad \mathbb{C}_{\exists}^{\mathbb{P}} \models T_{\exists}^{\mathbb{C}}(A \mapsto B).$$

Proof. The proof is almost identical to the proof of Proposition 6. □

The previously considered translations are purely syntactical, but the conceptual translation of existential preferences requires access to the derivation operator of the original context. Therefore, the information contained in $\mathbb{C}_{\exists}^{\mathbb{P}}$ must be regarded relative to the information in the original formal context. In particular, a complete set of existential preferences valid in $\mathbb{P} = (G, M, I, \leq)$ relative to the implications of (G, M, I) is as follows:

$$\{A \mapsto B \mid \mathbb{C}_{\exists}^{\mathbb{P}} \models \{(A', A)\} \to \{(B', B)\} \text{ and } B \not\subseteq A\}. \tag{4}$$

Example 12. For our example, this gives the same existential preference as before:



Thus, conceptual translation is not only more economical in terms of the size of the representation (identifying the closed rather than all subsets of M with new attributes), but it also allows more straightforward computation of relative preferences without explicitly using background implications and, thus, avoiding the computation of the canonical basis, which is known to be a hard problem [10,5].

As with universal preferences, we described two ways to represent existential preferences: a pure system of existential preferences and a hybrid system of implications and existential preferences. On the other hand, organizing exploration of existential preferences seems less straightforward than it was the case for universal preferences, since adding a counterexample—an object—to an existentially translated context requires also adding new attributes.

4 Conclusion

In this paper, we studied two approaches to deriving preferences over attribute sets from preferences over individual objects: universal preferences and existential preferences. For each of the two types of preferences, we developed two inference systems: one for a set of pure preferences of the respective type and the other for a mixed set of implications and preferences. We showed how preferences can be translated into implications of certain formal contexts, which makes it possible—at least, in principle—to compute preferences with existing algorithms for implications; however, we leave a thorough treatment of algorithmic issues for further research. It also remains to see whether there is any nontrivial relation between universal and existential preferences that could be used to build a hybrid system for these two types of preferences.

References

1. Angluin, D.: Queries and concept learning. Machine Learning 2, 319–342 (1988)
2. Armstrong, W.W.: Dependency structure of data base relationships. In: Proc. IFIP Congress, pp. 580–583 (1974)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications, 2nd edn. Cambridge University Press, New York (2010)
4. van Benthem, J.: Modal Logic for Open Minds. CSLI lecture notes, Center for the Study of Language and Information (2010)
5. Distel, F., Sertkaya, B.: On the complexity of enumerating pseudo-intents. Discrete Applied Mathematics 159(6), 450–466 (2011)
6. Ganter, B.: Attribute exploration with background knowledge. Theoretical Computer Science 217(2), 215–233 (1999)
7. Ganter, B., Obiedkov, S.: Implications in Triadic Formal Contexts. In: Wolff, K.E., Pfeiffer, H.D., Delugach, H.S. (eds.) ICCS 2004. LNCS (LNAI), vol. 3127, pp. 186–195. Springer, Heidelberg (2004)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
9. Guigues, J.-L., Duquenne, V.: Famille minimale d'implications informatives résultant d'un tableau de données binaires. Mathématiques et Sciences Humaines 24(95), 5–18 (1986)
10. Kuznetsov, S.O., Obiedkov, S.: Some decision and counting problems of the Duquenne–Guigues basis of implications. Discrete Applied Mathematics 156(11), 1994–2003 (2008)
11. Liu, F.: Changing for the Better. Preference Dynamics and Agent Diversity. Ph.D. thesis, Universiteit van Amsterdam (2008)
12. Rouane Hacene, M., Huchard, M., Napoli, A., Valtchev, P.: A Proposal for Combining Formal Concept Analysis and Description Logics for Mining Relational Data. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS (LNAI), vol. 4390, pp. 51–65. Springer, Heidelberg (2007)
13. von Wright, G.H.: The Logic of Preference. Edinburgh University Press (1963)

Finding Top-N Colossal Patterns Based on Clique Search with Dynamic Update of Graph

Yoshiaki Okubo and Makoto Haraguchi

Graduate School of Information Science and Technology
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
{yoshiaki,mh}@ist.hokudai.ac.jp

Abstract. In this paper, we discuss a method for finding *top-N colossal frequent patterns*. A colossal pattern we try to extract is a maximal pattern with top-N largest length. Since colossal patterns can be found in relatively lower areas of an itemset (concept) lattice, an efficient method with some effective pruning mechanisms is desired.

We design a depth-first branch-and-bound algorithm for finding colossal patterns with top-N length, where a notion of *pattern graph* plays an important role. A pattern graph is a compact representation of the class of frequent patterns with a designated length. A colossal pattern can be found as a *clique* in a pattern graph satisfying a certain condition. From this observation, we design an algorithm for finding our target patterns by examining cliques in a graph defined from the pattern graph. The algorithm is based on a depth-first branch-and-bound method for finding a maximum clique. It should be noted that as our search progresses, the graph we are concerned with is dynamically updated into a sparser one which makes our task of finding cliques much easier and the branch-and-bound pruning more powerful. To the best of our knowledge, it is the first algorithm tailored for the problem which can *exactly* identify top-N colossal patterns. In our experimentation, we compare our algorithm with famous maximal frequent itemset miners from the viewpoint of computational efficiency for a synthetic and a benchmark dataset.

Keywords: colossal patterns, maximal patterns, pattern graph, cliques.

1 Introduction

For a given *transaction database* $\mathcal{D}$, a problem of *Frequent Pattern/Itemset Mining* is to enumerate every *pattern* of items, also called an *itemset*, which *frequently* appears in $\mathcal{D}$ [1]. It has been one of the fundamental computation tasks in the field of *Data Mining* for recent decades.

As is well known, since we in general suffer from a huge number of frequent patterns, we are usually concerned with some particular class of patterns. Paying our attention to the structure of possible pattern space, called an *itemset lattice*, we are often interested in *maximal* frequent patterns (e.g. [3,6]) and *closed*

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 244–259, 2012.

frequent patterns (e.g. [4,5,6]), where a closed pattern is related to the notion of *Formal Concept* [11]. Besides, several measures of *interestingness*, e.g. *Bond* [8], have been proposed from the viewpoint of quality of patterns.

In this paper, we are concerned with maximal frequent patterns. Especially, we try to extract maximal frequent patterns with larger length. Such a pattern is often called a (frequent) *long pattern* and several excellent algorithms for finding them have already been proposed (e.g. [2]). In general, the frequency (or support) of a longer pattern tends to be lower. In spite of this tendency, if a longer pattern is found to be frequent, then we might expect such a pattern to provide us some valuable insight. For example, in the field of *Bioinformatics*, long patterns are actually regarded as important ones to be mined. In this sense, extracting long patterns would be worth investigating.

In [12], Zhu, et al. have originally named a pattern with larger length a *Colossal Pattern* in order to distinguish such a pattern from just frequent patterns. They have discussed the problem of finding a designated number (say, K) of frequent colossal patterns and proposed an algorithm based on a method of *Pattern Fusion*. In the algorithm, assuming a *pool* of (frequent) patterns with smaller length, we obtain a set of larger patterns by combining several patterns, called *core patterns*, in the pool. Then, the pattern pool is updated into a new pool consisting of the newly obtained patterns. Starting with an initial pool of frequent patterns with length 3, such a process of *core pattern fusion* is iterated until we get a pool with K (larger) patterns as the final output. Core patterns to be combined are cleverly selected so that each resultant pattern can *probably* get closer to some of correct K colossal patterns. Therefore, we can avoid examining many useless ones which cannot constitute correct colossal patterns.

In [13], Xie and Yu have recently tried to solve a similar problem. Their problem is to find top-K *maximal* frequent patterns, where top-K means K patterns with length larger than those of the other maximal ones. They have proposed an algorithm, named *Max-Clique*, which works in top-down manner. In their method, a notion of *Pattern Graph* has been introduced and effectively used to restrict candidate patterns to be examined. A pattern graph is an edge-weighted undirected graph with items as vertices and can be viewed as a compact representation of frequent patterns with a designated length. A key observation in the pattern graph is that a frequent pattern with length ℓ can be obtained as a *clique* in the graph in which any edge-weight is no less than some value determined by ℓ. In their algorithm, a set of maximal cliques with larger size in the pattern graph is first extracted and then each of them is refined to several maximal frequent patterns, where the refinement process has two phases, *item removal phase* and *candidate extension phase*. For a maximal clique, in the former phase, several items are identified with some *randomization factor* and then removed from the clique. In general, since some necessary items might be *wrongly* removed in the phase, they are recovered in the latter phase with a traditional frequent pattern miner.

Exactly speaking, the problems in [12] and [13] are different in a mathematical sense. In most cases, however, the difference will never cause serious matters.

A fact to be noted here is that both of their algorithms are based on *heuristic* approaches. They cannot always detect top-K colossal/maximal frequent patterns with larger length *exactly*. In general, we might miss some of the target patterns to be extracted. Needless to say, such an incompleteness would be surely undesirable because those missing patterns could provide valuable information and knowledge. It would be therefore worth investigating a *complete* algorithm which can exactly discover the targets.

In this paper, we try to design a complete algorithm for efficiently extracting all maximal frequent patterns with *top-N largest length*. Unlike the previous problems of mining K colossal/maximal patterns with larger length in [12,13], our parameter N for top-N restricts the length of patterns to be detected, not the number of extracted patterns. We consider that it is more intuitive for users to provide N rather than K because we are interested in *colossality* of patterns. For example, if we try to obtain all patterns with the largest length, we need to provide an adequate value of K by which all the target patterns can be covered completely. We, however, have no idea to guess how many patterns actually have the largest length. In our problem, on the other hand, we can simply provide $N = 1$ for the task. Thus, our problem in this paper can be regarded as a natural modification of the previous problems.

We present a *depth-first branch-and-bound algorithm* for our problem. Especially, it has been designed based on a branch-and-bound method for finding a *maximum clique* [15,16]. In the algorithm, we first construct a pattern graph, as *Max-Clique* does [13]. From the pattern graph, we then extract a subgraph G satisfying a simple constraint on edge-weight, where our target patterns can be extracted as cliques in G.

In order to obtain a maximal frequent pattern, we recursively expand a clique (a pattern) into larger cliques in a *depth-first* manner. During the process, we manage a *tentative* list of maximal patterns with top-N largest length obtained so far. It is noted here that for each clique C, we can estimate an upper bound, $\tilde{u}_C$, on the size of any C's expansion. Therefore, if we find $\tilde{u}_C$ is less than the minimum length of patterns in the tentative list, we do not need to expand C any more because no expansion of C with top-N largest length can be obtained. Thus, such a C and its possible expansions can be pruned without losing completeness, that is, a branch-and-bound pruning is available.

In our computation, the tentative top-N list is iteratively updated, monotonically increasing the minimum length of patterns in the list. Simultaneously, the graph G can be also updated into a more sparse one. As the result, as our search progresses, the task of finding cliques in G becomes much easier and the branch-and-bound pruning more powerful. Our algorithm, therefore, can efficiently and completely find all maximal patterns with top-N largest length. To the best of our knowledge, it is the first algorithm which can *exactly* identify the target patterns, not approximately.

For a synthetic dataset and a famous benchmark dataset, our experimental results show that the computational performance of our algorithm is better than famous fast algorithms for extracting maximal frequent itemsets, *MAFIA* [3], *LCM* [6] and *ABS* [7].

2 Preliminaries

Let $\mathcal{I}$ be a set of *items*. A subset of $\mathcal{I}$, $X \subseteq \mathcal{I}$, is called an *itemset*. We often call an itemset a *pattern*. If an itemset X consists of k items, that is, $|X| = k$, then X is particularly called a *k-itemset/pattern*, where k is the *length* of X.

A *transaction* is given as an itemset $T \subset \mathcal{I}$. A *transaction database*, $\mathcal{D}$, is a finite collection (a multiple set) of transactions.

For a transaction database $\mathcal{D}$ and a pattern X, the set of transactions in $\mathcal{D}$ containing X is referred to as $\mathcal{D}(X)$, that is, $\mathcal{D}(X) = \{T \in \mathcal{D} \mid X \subseteq T\}$. The number of transactions in $\mathcal{D}(X)$, $|\mathcal{D}(X)|$, is called the *frequency* of X.

For an integer σ, a pattern X is said to be *σ-frequent* if $|\mathcal{D}(X)| \geq \sigma$ holds. The set of σ-frequent patterns is referred to as $\mathcal{F}_\sigma$. Given σ as a threshold on the *minimum frequency*, *Frequent Itemset/Pattern Mining* [1] is defined as the problem of identifying $\mathcal{F}_\sigma$.

Based on pattern length, $\mathcal{F}_\sigma$ can be partitioned into $\mathcal{F}_\sigma^0, \mathcal{F}_\sigma^1, \mathcal{F}_\sigma^2, \ldots$ and $\mathcal{F}_\sigma^L$, where $\mathcal{F}_\sigma^\ell$ is the set of σ-frequent ℓ-patterns and L is the maximum length of patterns in $\mathcal{F}_\sigma$.

A (undirected) graph is denoted by $G = (V, E)$, where V is a set of *vertices*, $E \subseteq V \times V$ a set of *edges*. In case each edge is assigned some weight, a graph is denoted by $G = (V, E, w)$, where $w : E \to \boldsymbol{N}$ is a *weight function* for edges.

For a vertex $v \in V$, the set of vertices adjacent to v is denoted by $N_G(v)$, where $|N_G(v)|$ is called the *degree* of v. The degree of v is often referred to as $deg_G(v)$. If it is clear from the context, they are simply denoted by $N(v)$ and $deg(v)$, respectively.

3 Pattern Graph

A *pattern graph*, introduced in [13], is an approximate compact representation of the frequent itemsets with a designated length and is formally defined as follows.

Definition 1. (ℓ-Item Pattern Graph)
Let $\mathcal{D}$ be a transaction database with the set of items $\mathcal{I}$ and σ a threshold on the minimum frequency. For a length ℓ, the *ℓ-item pattern graph* is an edge-weighted undirected graph $G_{\sigma,\ell} = (\mathcal{I}, E, w)$, where $(x, y) \in E \subseteq \mathcal{I} \times \mathcal{I}$ iff there exists a σ-frequent ℓ-itemset $X \in \mathcal{F}_\sigma^\ell$ such that $\{x, y\} \subseteq X$, and for $(x, y) \in E$, $w((x, y)) = |\ \{X \in \mathcal{F}_\sigma^\ell \mid \{x, y\} \subseteq X\}\ |$. ■

From the definition, if a pair of items x and y co-occur in *at least* one σ-frequent ℓ-itemset, then they are connected in the graph. Moreover, the edge (x, y) is assigned an weight defined as the number of σ-frequent ℓ-itemsets containing both x and y.

In the pattern graph, we can observe an interesting theoretical property.

Lemma 1. [13]
Let $G_{\sigma,\ell}$ be the ℓ-item pattern graph constructed for a threshold on the minimum frequency σ. If a k-itemset X is σ-frequent, then X forms a clique in $G_{\sigma,\ell}$ in which the weight of each edge is at least $\binom{k-2}{\ell-2}$. ■

That is, the lemma states *necessary conditions* on frequent patterns with a certain length. Based on them, in [13], an algorithm for mining top-K maximal frequent patterns, named *Max-Clique*, has been proposed, where top-K means "K patterns with larger length". From the lemma, any σ-frequent pattern can be extracted by examining only cliques in the pattern graph. In fact, *Max-Clique* first extracts a given number (more than K) of larger maximal cliques in *Clique Detection Phase* and then, in *Candidate Refinement Phase*, tries to refine each maximal clique into several maximal patterns with larger length. Since the latter phase, however, takes a randomized method, *Max-Clique* is not a complete algorithm for the problem.

As has been discussed in [13], the value of ℓ has a direct influence on our computation. A larger ℓ can make the graph $G_{\sigma,\ell}$ sparser and both of the computation phases easier with less approximation error. However, the task of identifying $\mathcal{F}_\sigma^\ell$ necessary for constructing the graph becomes harder. From an empirical point of view, *Max-Clique* takes $\ell = 3$.

In the following section, we define an optimization problem of finding *top-N colossal frequent patterns*, where top-N means "*patterns with top-N largest length*", that is, we try to *maximize* the length of patterns to be extracted. We then present a complete algorithm for the optimization problem.

4 Top-*N* Colossal Frequent Pattern Mining

As has been seen in many actual domains, it would be worth discussing a problem of finding all frequent patterns with top-N largest length, where N is a relatively small integer specified by users based on their interest. We call this mining task *Top-N colossal frequent pattern mining* and formalize it as follows.

Definition 2. (Top-N Colossal Frequent Pattern Mining)
Let $\mathcal{D}$ be a transaction database, σ a threshold on the minimum frequency and N an integer. Then, *Top-N Colossal Frequent Pattern Mining* is a problem of identifying the set of σ-frequent patterns with top-N largest length, denoted by $\mathcal{CLS}_{\sigma,N}$, that is, $\mathcal{CLS}_{\sigma,N} = \mathcal{F}_\sigma^L \cup \mathcal{F}_\sigma^{L-1} \cup \cdots \cup \mathcal{F}_\sigma^{L-N+1}$, where L is the maximum length of σ-frequent patterns. ■

In the next section, we present a depth-first branch-and-bound algorithm for exactly identifying $\mathcal{CLS}_{\sigma,N}$ with the help of a pattern graph.

5 Finding Top-*N* Colossal Patterns with Pattern Graph

Given a transaction database $\mathcal{D}$ with a set of items $\mathcal{I}$ and a threshold on the minimum frequency σ, let us assume we have the ℓ-item pattern graph $G_{\sigma,\ell} = (\mathcal{I}, E, w)$. The observations stated in Lemma 1 bring us an idea for finding colossal frequent patterns in $\mathcal{D}$ with top-N largest length.

5.1 Fundamental Idea

Extracting Patterns as Cliques in Graph with Dynamic Update: From Lemma 1, we can notice the following simple fact.

Observation 1
For a length k, a σ-frequent k-pattern can be obtained as a clique in a graph $G^W_{\sigma,\ell}$ defined from the pattern graph $G_{\sigma,\ell}$ as $G^W_{\sigma,\ell} = (\mathcal{I}, E^W)$, where $W = \binom{k-2}{\ell-2}$ and $E^W = \{e \in E \mid w(e) \geq W\}$. ■

Note here that E^W is obtained by just deleting every edge in E with the weight less than W, that is, $\binom{k-2}{\ell-2}$. It is easy to see that for any k' such that $k' \geq k$, every σ-frequent k'-pattern can be also found as a clique in $G^W_{\sigma,\ell}$. This implies that examining possible cliques in $G^W_{\sigma,\ell}$ would be a basic procedure for extracting all frequent patterns with the length *no less than* k.

Since, in general, we have no idea to identify the maximum length of frequent patterns, L, without any computation, we are forced to begin our computation under $W = 1$ given by $\ell = 3$ and $k = 3$ [1]. We try to extract cliques in $G^1_{\sigma,3} = (\mathcal{I}, E^1)$ each of which is examined whether it is *actually* a frequent maximal pattern or not. If a clique Q is found to be a frequent maximal pattern, then Q is stored in a list $\mathcal{L}_N$ which keeps patterns with top-N largest length found so far. Once the list $\mathcal{L}_N$ is filled with patterns with *tentative* top-N length, our task is now to detect frequent patterns (cliques) with the length no less than s, where s is the minimum length of the patterns in $\mathcal{L}_N$. We can therefore update the graph $G^1_{\sigma,3}$ into $G^{s-2}_{\sigma,3}$ and continue to extract cliques. Whenever a pattern is found to be longer than those with the minimum length in $\mathcal{L}_N$, the list is updated so that it correctly keeps patterns with top-N length at that point. Furthermore, since the minimum length of the patterns in the list might increase from s to say s' accordingly, the graph $G^{s-2}_{\sigma,3}$ can be also updated to $G^{s'-2}_{\sigma,3}$. It should be emphasized here that from the definition of $G^W_{\sigma,\ell}$, the higher W becomes, the sparser $G^W_{\sigma,\ell}$ becomes. Thus, as our search progresses, the task of extracting cliques becomes easier. Such a procedure is repeated until no clique remains to be examined and finally, the list $\mathcal{L}_N$ is output as our target patterns with top-N largest length.

Depth-First Branch-and-Bound Method for Finding Patterns with Larger Length: Finding cliques in a graph has been one of the fundamental problems in computer science. In these decades, many researchers have developed excellent algorithms for efficiently finding a *maximum* clique and enumerating *maximal* cliques (e.g. [14,17]). Since our primary task is to extract cliques in a graph, those methods would be helpful in designing our algorithm.

Let G be a graph with a set of items, $\mathcal{I} = \{x_1, \ldots, x_{|\mathcal{I}|}\}$, as the vertices. Consider a *total ordering* on $\mathcal{I}$, $\prec$, simply defined as $x_i \prec x_j$ iff $i < j$. We assume that for each subset $X \subseteq \mathcal{I}$, the elements in X is always ordered based on $\prec$.

[1] Although it has not been explicitly stated in [13], we reasonably assume in this paper $k \geq \ell > 2$.

For a set X, the first element is referred to as $head(X)$ and the last one as $tail(X)$. Furthermore, the set of first k elements is referred to as $prefix(X,k)$, where $prefix(X,0)$ is defined as $\emptyset$.

We introduce here a *partial ordering* $\prec_s$ on $2^{\mathcal{I}}$. Let X_i and X_j be subsets of $\mathcal{I}$ such that $X_i \neq X_j$. Then we have $X_i \prec_s X_j$ iff $X_i = prefix(X_j, |X_i|)$. It can be easily observed that the partially ordered set $(2^{\mathcal{I}}, \prec_s)$ forms a *tree* with the root node $\emptyset$ which is well-known as a *set enumeration tree*.

Since each clique in G is a subset of $\mathcal{I}$ and any subset of a clique is also a clique, the ordering $\prec_s$ is still valid for cliques and the cliques in G also form a tree, called a *clique enumeration tree*.

For a problem of finding a maximum clique in a given graph, several efficient algorithms have been proposed in [15,16]. Basically speaking, these algorithms commonly explore the clique enumeration tree in a *depth-first manner* with a simple *branch-and-bound pruning* in order to efficiently find a maximum clique, that is, a clique with the largest size. The underlying idea is expected to be useful for our problem of finding colossal patterns (as cliques) with top-N largest length.

In a clique enumeration tree, an immediate successor of a clique can be generated by adding a certain vertex (item), called an *extensible candidate*, to the clique.

Definition 3. (Extensible Candidates for Clique)
Let Q be a clique in a graph $G = (\mathcal{I}, E)$. A vertex v adjacent to any vertex in Q is called an *extensible candidate* (or simply a candidate) for Q. The set of extensible candidates is denoted by $cand_G(Q)$, that is, $cand_G(Q) = \{v \in \mathcal{I} \mid \forall x \in Q, \quad (v,x) \in E\} = \bigcap_{x \in Q} N_G(x)$. If it is clear from the context, it is simply denoted by $cand(Q)$. ■

Since it is easy from the definition to see that for any $v \in cand(Q)$, $Q \cup \{v\}$ always becomes a clique, we can easily generate an immediate successor of Q by just adding $v \in cand(Q)$ such that $tail(Q) \prec v$. Thus, the clique enumeration tree can be *systematically* generated without any duplications.

We can observe the following simple but important theoretical property.

Observation 2
For cliques Q and Q' such that $Q \subseteq Q'$, $Q \cup cand(Q) \supseteq Q' \cup cand(Q')$. ■

From this observation, it is easy to see that for a clique Q, each successor of Q is a subset of $Q \cup cand(Q)$, more exactly, a subset of $Q \cup exp\text{-}cand(Q)_{tail(Q)}$, where for an item α, $exp\text{-}cand(Q)_\alpha = \{x \in cand(Q) \mid \alpha \prec x\}$. This fact provides us a branch-and-bound pruning mechanism.

Let G be a graph in which we try to find colossal patterns (as cliques) with top-N largest length. Moreover, let us assume that a list $\mathcal{L}_N$ already keeps maximal patterns with top-N length found so far, where the minimum length of the patterns in $\mathcal{L}_N$ is referred to as $minlen(\mathcal{L}_N)$, that is, $minlen(\mathcal{L}_N) = \min_{X \in \mathcal{L}_N}\{|X|\}$.

Observation 3
Let Q be a clique such that $|Q \cup exp\text{-}cand(Q)_{tail(Q)}| < minlen(\mathcal{L}_N)$. Then, for any successor Q' of Q, $|Q'| < minlen(\mathcal{L}_N)$. ■

As a direct consequence, we have the following pruning rule.

Pruning 1
For a clique Q, if $|Q \cup exp\text{-}cand(Q)_{tail(Q)}| < minlen(\mathcal{L}_N)$ holds, then Q and its successors do not have to be examined. ■

If the condition holds, we can safely prune all possible successors of Q as useless patterns without loss of completeness. In order to enjoy this pruning effect as much as possible, we examine cliques in a depth-first manner.

More concretely speaking, a (current) clique Q can be expanded in several ways by adding an extensible candidate v selected from $exp\text{-}cand(Q)_{tail(Q)}$. For an immediate successor $Q_v = Q \cup \{v\}$, if $|Q_v \cup exp\text{-}cand(Q_v)_v| \geq minlen(\mathcal{L}_N)$ holds, then we try to further expand Q_v with $exp\text{-}cand(Q_v)_v$ in the same manner. Conversely, if $|Q_v \cup exp\text{-}cand(Q_v)_v| < minlen(\mathcal{L}_N)$, then we can immediately discard Q_v and try the next successor $Q_{v'} = Q \cup \{v'\}$ with another $v' \in exp\text{-}cand(Q)_{tail(Q)}$ in the order of $\prec$. Starting with $Q = \emptyset$ and $cand(Q) = \mathcal{I}$, this expansion process is recursively iterated until we have no Q to be expanded.

5.2 More about Prunings of Useless Cliques

Our targets to be extracted must be frequent patterns. During our search, therefore, if a clique Q is found to be infrequent, then we can immediately backtrack to the next clique in order to skip all of the successors of Q.

In addition to this ordinary pruning, we can exclude many redundant cliques based on the notion of *closed patterns* [4], equivalently, *formal concepts* [11]. Since we are interested in (frequent) maximal patterns with top-N largest length, paying our attention to closed patterns is sufficient for our purpose.

Given a transaction database $\mathcal{D}$ with $\mathcal{I}$, for a pattern $X \subseteq \mathcal{I}$, we can uniquely compute its *closure*, denoted by $closure(X)$ [2], defined as $closure(X) = \bigcap_{T \in \mathcal{D}(X)} T$. In general, since several patterns have their identical closure, examining those patterns would be undesirable for efficient computation. We can observe the following simple property of closed patterns which is useful for excluding such redundant patterns.

Observation 4
Let X be a pattern and x an item such that $tail(X) \prec x$. If an item $\alpha \in closure(X \cup \{x\}) \backslash closure(X)$ such that $\alpha \prec x$ can be found, then there exists a pattern Y such that $closure(Y) = closure(X \cup \{x\})$ and Y is examined prior to $X \cup \{x\}$ in depth-first traversal of the set enumeration tree. ■

[2] As a conventional notation, $closure(X)$ is often denoted by X'' in *Formal Concept Analysis* [11].

As has been seen before, since a clique enumeration tree is a part of a set enumeration tree, the above property can be also observed in the clique tree and as a result, it provides us the following pruning rule to exclude redundant cliques in our depth-first search.

Pruning 2
Let Q be a clique and $Q \cup \{x\}$ an immediate successor of Q. If there exists $\alpha \in closure(Q \cup \{x\}) \setminus closure(Q)$ such that $\alpha \prec x$, then $Q \cup \{x\}$ and its successors do not have to be examined. ∎

5.3 Expanding Clique with Pruning Rules

Our procedure of expanding cliques with the prunings is described in detail.

Assume we already have a list, $\mathcal{L}_N$, of maximal patterns with top-N length found so far. Thus, the graph we are currently concerned with is $G_{\sigma,3}^{minlen(\mathcal{L}_N)-2}$. We now try to expand a clique (as a closed frequent pattern) Q in the graph.

Checking Maximality of Pattern. For each item x in $cand(Q)$, we first compute the set of transactions containing $Q \cup \{x\}$, $\mathcal{D}(Q \cup \{x\})$. It can be done efficiently by the technique called *occurrence-deliver* [6].

If $|\mathcal{D}(Q \cup \{x\})| < \sigma$ for each $x \in cand(Q)$, then Q is a frequent maximal pattern because any superset of Q is not frequent. In this case, the list $\mathcal{L}_N$ is adequately updated with Q as follows, where $s = minlen(\mathcal{L}_N)$.

- $|Q| < s$**:** Although Q is frequent and maximal, it is out of our target due to its short length. We, therefore, have nothing to do for $\mathcal{L}_N$ and just backtrack.
- $|Q| = s$**:** Q is simply added to $\mathcal{L}_N$ and then we backtrack.
- $|Q| > s$**:** If there already exists a pattern in $\mathcal{L}_N$ with the length $|Q|$, Q is simply added to $\mathcal{L}_N$. If such a pattern does not exist, Q is added to $\mathcal{L}_N$ and then the patterns with length s in $\mathcal{L}_N$ are removed. In this case, since $minlen(\mathcal{L}_N)$ is incremented from s to say s', the current graph $G_{\sigma,3}^{s-2}$ can be also updated to $G_{\sigma,3}^{s'-2}$. Then, we backtrack.

In case of Q is not maximal, Q is tried to expand with several particular items in $cand(Q)$ as follows.

Expanding Clique with Hopeful Extensible Candidates. Q is divided into two parts, $generator(Q)$ and $implied(Q)$. The former is the set of items in Q *actually* added at the previous expansion steps to get Q from $\emptyset$, and the latter the set of items in Q which are *implied* by $generator(Q)$, that is, $implied(Q) = closure(generator(Q)) \setminus generator(Q)$.

Here we consider a subset of $cand(Q)$, $freq\text{-}cand(Q)$, defined as

$$freq\text{-}cand(Q) = \{x \in cand(Q) \mid |\mathcal{D}(Q \cup \{x\})| \geq \sigma \ \wedge \ deg_G(x) \geq s - 1\},$$

where G is the current graph we are concerned with and $s = minlen(\mathcal{L}_N)$. It is easy to see that expanding Q with each item in $freq\text{-}cand(Q)$ is sufficient for our purpose because our target must be frequent and a clique in G with the size no less than s. More precisely, we can try only items $x \in freq\text{-}cand(Q)$ such that $tail(generator(Q)) \prec x$.

Let us assume Q is tried to expand by adding an item $\tilde{x} \in freq\text{-}cand(Q)$ such that $tail(generator(Q)) \prec \tilde{x}$. $freq\text{-}cand(Q)$ can be separated into $chk\text{-}cand(Q)_{\tilde{x}}$ and $exp\text{-}cand(Q)_{\tilde{x}}$, where $chk\text{-}cand(Q)_{\tilde{x}}$ is the set of items $x \in freq\text{-}cand(Q)$ such that $x \prec \tilde{x}$ and $exp\text{-}cand(Q)_{\tilde{x}}$ the set of those such that $\tilde{x} \prec x$. Based on Pruning 1, we first check whether $\mid Q \mid + |exp\text{-}cand(Q)_{\tilde{x}}| < s$ holds or not. If it is true, we can immediately stop expanding Q and backtrack. This means we never try to expand Q with $x' \in freq\text{-}cand(Q)$ such that $\tilde{x} \prec x'$. If it is not the case, we enter to the following two phases, *checking-phase* and *expansion-phase*.

Checking-Phase: We check whether there exists an item $x \in chk\text{-}cand(Q)_{\tilde{x}}$ implied by $Q \cup \{\tilde{x}\}$. It can be done by verifying whether $\mathcal{D}(Q \cup \{x\}) \supseteq \mathcal{D}(Q \cup \{\tilde{x}\})$ or not. If it is the case for some x, x is implied by $Q \cup \{\tilde{x}\}$, but not by Q. Based on Pruning 2, therefore, the expansion with $\tilde{x}$ is stopped and the immediate successor of $\tilde{x}$ in $freq\text{-}cand(Q)$ is then tried as the next $\tilde{x}$. If there is no such x, the expansion-phase is carried out.

Expansion-Phase: Let us denote the expansion of Q with $\tilde{x}$ by $Q_{\tilde{x}}$. For each $x \in exp\text{-}cand_{\tilde{x}}$, we check whether $\mathcal{D}(Q \cup \{x\}) \supseteq \mathcal{D}(Q \cup \{\tilde{x}\})\}$ holds or not. If it is true, that is, x is implied by $Q \cup \{\tilde{x}\}$, then x becomes an element of $implied(Q_{\tilde{x}})$. Thus, $Q_{\tilde{x}}$ is given as $Q_{\tilde{x}} = generator(Q_{\tilde{x}}) \cup implied(Q_{\tilde{x}})$, where

$$generator(Q_{\tilde{x}}) = Q \cup \{\tilde{x}\}$$

and

$$implied(Q_{\tilde{x}}) = implied(Q) \cup \{x \in exp\text{-}cand(Q)_{\tilde{x}} | \mathcal{D}(Q \cup \{x\}) \supseteq \mathcal{D}(Q \cup \{\tilde{x}\})\}.$$

Moreover, the set of extensible candidates for $Q_{\tilde{x}}$, $cand(Q_{\tilde{x}})$, consists of the items in $freq\text{-}cand(Q)$ each of which is adjacent to any item in $Q_{\tilde{x}}$, that is,

$$cand(Q_{\tilde{x}}) = freq\text{-}cand(Q) \cap \left(\bigcap_{x \in Q_{\tilde{x}}} N(x) \right).$$

By recursively applying this expansion procedure to the newly obtained $Q_{\tilde{x}}$, we can obtain top-n colossal frequent patterns in a depth-first manner.

5.4 Algorithm for Finding Top-N Colossal Patterns

Pseudo-code: The above discussion can be summarized in a pseudo-code presented in Figure 1. In the figure, we assume $tail(\emptyset) = \perp$, where the symbol $\perp$ is a (virtual) minimum element in any ordering. Moreover, for a list of patterns $\mathcal{L}$, the function $minlen(\mathcal{L})$ returns the minimum length of the patterns in $\mathcal{L}$ if the list tentatively contains patterns with top-N largest length found so far. Otherwise, it returns 3 which corresponds to the minimum length of patterns to be extracted as our assumption.

procedure MAIN($\mathcal{D}$, σ, N):
[Input] $\mathcal{D}$: a transaction database with a set of items $\mathcal{I}$.
σ: a threshold on the minimum frequency.
N: an integer for top-N.
[Output] $\mathcal{CLS}_{\sigma,N}$: the set of top-N colossal σ-frequent patterns.
[Global Variables] $\mathcal{L}$: a list of colossal frequent patterns.
G: an edge-weighted undirected graph.
$\mathcal{D}$, σ and N.
begin
compute $\mathcal{F}_\sigma^3$; construct $G_{\sigma,3}^1$ from $\mathcal{F}_\sigma^3$; $G \leftarrow G_{\sigma,3}^1$; $\mathcal{L} \leftarrow \emptyset$;
FINDTOPNCOLOSSALPATTERNS($\emptyset$, $\emptyset$, $\mathcal{I}$);
return $\mathcal{L}$;
end

---

procedure FINDTOPNCOLOSSALPATTERNS(Gen, Imp, $Cand$):
begin
$Q \leftarrow Gen \cup Imp$;
for each $x \in Cand$ **do** compute $\mathcal{D}(Q \cup \{x\})$;
$FrecCand \leftarrow \{x \in Cand \mid |\mathcal{D}(Q \cup \{x\})| \geq \sigma \ \wedge \ deg_G(x) \geq minlen(\mathcal{L}) - 1\}$;
if $FrecCand = \emptyset$ **then begin** UPDATETOPNLIST(Q); return; **end** // Q is maximal
$ChkCand \leftarrow \{x \in FrecCand \mid x \prec tail(Gen)\}$; $ExpCand \leftarrow \{x \in FrecCand \mid tail(Gen) \prec x\}$;
while $ExpCand \neq \emptyset$ **do**
begin
if $|Q| + |ExpCand| < minlen(\mathcal{L})$ **then** return; // Pruning 1
else
begin
$\tilde{x} \leftarrow head(ExpCand)$; $ExpCand \leftarrow ExpCand \setminus \{\tilde{x}\}$;
$NewImp \leftarrow Imp$; $NewCand \leftarrow \emptyset$; $pruning\text{-}flag \leftarrow 0$;
for each $x \in ChkCand$ **do** // Cheking-Phase
begin
if $\mathcal{D}(Q \cup \{\tilde{x}\}) \subseteq \mathcal{D}(Q \cup \{x\})$ **then** // x is implied by $\tilde{x}$
begin $pruning\text{-}flag \leftarrow 1$; break; **end** // Pruning 2
else if x and $\tilde{x}$ are adjacent in G **then** $NewCand \leftarrow NewCand \cup \{x\}$;
end
if $pruning\text{-}flag = 1$ **then**
begin $ChkCand \leftarrow ChkCand \cup \{\tilde{x}\}$; $ExpCand \leftarrow ExpCand \setminus \{\tilde{x}\}$; continue; **end**
for each $x \in ExpCand$ **do** // Expansion-Phase
begin
if $\mathcal{D}(Q \cup \{\tilde{x}\}) \subseteq \mathcal{D}(Q \cup \{x\})$ **then** $NewImp \leftarrow NewImp \cup \{x\}$; // x is implied by $\tilde{x}$
else if x and $\tilde{x}$ are adjacent in G **then** $NewCand \leftarrow NewCand \cup \{x\}$;
end
FindTopNColossalPatterns($Gen \cup \{\tilde{x}\}$, $NewImp$, $NewCand$);
if Q is a clique in G **then** // G might have been updated
begin
$ChkCand \leftarrow \{x \in ChkCand \cap cand_G(Q) \mid deg_G(x) \geq minlen(\mathcal{L}) - 1\}$;
$ExpCand \leftarrow \{x \in ExpCand \cap cand_G(Q) \mid deg_G(x) \geq minlen(\mathcal{L}) - 1\}$;
if $\tilde{x} \in cand_G(Q) \wedge deg_G(\tilde{x}) \geq minlen(\mathcal{L}) - 1$ **then** $ChkCand \leftarrow ChkCand \cup \{\tilde{x}\}$;
end
else return; // Q is no longer a clique due to update of G
end
end
end

---

procedure UPDATETOPNLIST(Q):
begin
$\mathcal{L} \leftarrow \mathcal{L} \cup \{Q\}$;
if $|\{|P| \mid P \in \mathcal{L}\}| \geq N$ **then** // $\mathcal{L}$ contains patterns with tentatively top-N largest length
begin
remove all patterns with M-th largest length from $\mathcal{L}$ such that $N < M$;
$G \leftarrow G_{\sigma,3}^{minlen(\mathcal{L})-2}$; // updating graph
end
end

Fig. 1. Algorithm for Finding Top-N Colossal Frequent Patterns

Running Example

We are given a transaction database $\mathcal{D}$ in Figure 2, where the ordering on the items, $\prec$, is given as alphabetical order. We try to find top-1 colossal 2-frequent patterns in $\mathcal{D}$.

We first identify $\mathcal{F}_2^3$, the set of 2-frequent 3-patterns. There exist 9 those patterns shown in Figure 3 (a). For each pair of items, we then count the number of 2-frequent 3-patterns in which the items co-occur. Figure 3 (b) shows the result of counting in a matrix form. Based on the frequency matrix, we can construct a 3-item pattern graph $G_{2,3}$ in Figure 3 (c) from which the initial graph $G_{2,3}^1$ in Figure 4 is obtained.

ID	Transaction
1	c d e
2	a c d f
3	a b c d
4	a b c d f
5	b c e f
6	a b d e
7	a b e

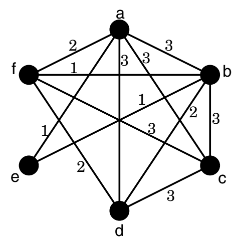
Fig. 2. Trans. DB. $\mathcal{D}$

$\mathcal{F}_2^3$
a b c
a b d
a b e
a c d
a c f
a d f
b c d
b c f
c d f

(a) 3-pat. with freq. 2

	a	b	c	d	e	f
a	-	3	3	3	1	2
b	3	-	3	2	1	1
c	3	3	-	3	0	3
d	3	2	3	-	0	2
e	1	1	0	0	-	0
f	2	1	3	2	0	-

(b) Freq. table of item-pairs



(c) $G_{2,3}$

Fig. 3. 3-Item Pattern Graph $G_{2,3}$ for $\mathcal{D}$

Then, the function FindTopNColossalPatterns is initially called with the arguments $Gen_0 = \emptyset$, $Imp_0 = \emptyset$ and $Cand_0 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$, respectively [3], where the function call is referred to as **C0**.

As $Q_0 = Gen_0 \cup Imp_0 = \emptyset$, for each $x \in Cand_0$, $\mathcal{D}(Q_0 \cup \{x\})$ is computed. It gives $FrecCand_0 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ and then $ChkCand_0 = \emptyset$ and $ExpCand_0 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. Q_0 is now tried to expand with each $x \in ExpCand_0$ in order.

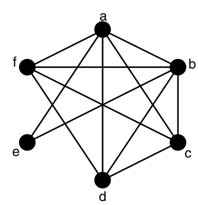


Fig. 4. $G_{2,3}^1$

For $\tilde{x}_{01} = \mathtt{a}$, we have $ExpCand_0 = \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. Through Checking-Phase and Expansion-Phase, $NewImp_{01} = \emptyset$ and $NewCand_{01} = \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ are obtained and then FindTopNColossalPatterns is recursively called as the function call **C1** with the arguments $Gen_0 \cup \{\tilde{x}_{01}\} = \{\mathtt{a}\}$, $NewImp_{01}$ and $NewCand_{01}$.

As $Gen_1 = Gen_0 \cup \{\tilde{x}_{01}\}$, $Imp_1 = NewImp_{01}$ and $Cand_1 = NewCand_{01}$, we have $Q_1 = \{\mathtt{a}\}$ and identify $\mathcal{D}(Q_1 \cup \{x\})$ for each $x \in Cand_1$. Since $FrecCand_1 = \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$, we have $ChkCand_1 = \emptyset$ and $ExpCand_1 = \{\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ and then try to expand Q_1 with each $x \in ExpCand_1$.

[3] We use proper subscripts to distinguish individual variables in each recursive call of FindTopNColossalPatterns.

For $\tilde{x}_{11} = \mathtt{b} \in ExpCand_1$, $ExpCand_1 = \{\mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ is obtained. Checking-Phase and Expansion-Phase give $NewImp_{11} = \emptyset$ and $NewCand_{11} = \{\mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$. They are passed to FindTopNColossalPatterns together with $Gen_1 \cup \{\tilde{x}_{11}\} = \{\mathtt{a}, \mathtt{b}\}$ as the first argument, making the function call **C2**.

As $Gen_2 = Gen_1 \cup \{\tilde{x}_{11}\}$, $Imp_2 = NewImp_{11}$ and $Cand_2 = NewCand_{11}$, we have $Q_2 = \{\mathtt{a}, \mathtt{b}\}$ and identify $\mathcal{D}(Q_2 \cup \{x\})$ for each $x \in Cand_2$. Since $|\mathcal{D}(\{\mathtt{a}, \mathtt{b}, \mathtt{f}\})| \not\geq 2$, $\mathtt{f}$ cannot be contained in $FrecCand_2$. From $FrecCand_2 = \{\mathtt{c}, \mathtt{d}, \mathtt{e}\}$, we have $ChkCand_2 = \emptyset$ and $ExpCand_2 = \{\mathtt{c}, \mathtt{d}, \mathtt{e}\}$, and try to expand Q_2 with each $x \in ExpCand_2$.

For $\tilde{x}_{21} = \mathtt{c} \in ExpCand_2$, we have $ExpCand_2 = \{\mathtt{d}, \mathtt{e}\}$. Since $ChkCand_2 = \emptyset$, Checking-Phase is skipped. In Expansion-Phase, it is found that $\mathtt{d}$ is implied by $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. We, therefore, obtain $NewImp_{21} = \{\mathtt{d}\}$. Moreover, since $\mathtt{e}$ is not adjacent to $\mathtt{c}$ in the current graph $G = G^1_{2,3}$, $\mathtt{e}$ cannot belong to $NewCand_{21}$. Thus, we get $NewCand_{21} = \emptyset$. FindTopNColossalPatterns is called with $Gen_2 \cup \{\tilde{x}_2\}$, $NewImp_{21}$ and $NewCand_{21}$, making the function call **C3**.

As $Gen_3 = Gen_2 \cup \{\tilde{x}_{21}\}$, $Imp_3 = NewImp_{21}$ and $Cand_3 = NewCand_{21}$, we have now $Q_3 = Gen_3 \cup Imp_3 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. Since $Cand_3 = \emptyset$, that is, $FreqCand_3 = \emptyset$, Q_3 is a maximal frequent pattern. Then, UpdateTopNList is called with the argument Q_3. Since $minlen()$ is incremented to $|Q_3| = 4$ and our task is now to extract cliques/patterns with the size no less than 4, the graph we are currently concerned with can also be updated into $G = G^{4-2}_{2,3} = G^2_{2,3}$ shown in Figure 5. The function call, **C3** is now completed and we return to the while loop in the previous call **C2**.

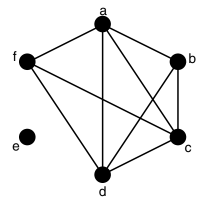


Fig. 5. $G^2_{2,3}$

Before trying to expand Q_2 with the next $\tilde{x}_{22} \in ExpCand_2$, we need to check whether Q_2 is a clique in the updated graph G. Since $Q_2 = \{\mathtt{a}, \mathtt{b}\}$ is still a clique, $ChkCand_2$ and $ExpCand_2$ are also modified so that they are valid for the current G. By taking intersection with $cand_G(Q_2) = \{\mathtt{c}, \mathtt{d}\}$, we have $ChkCand_2 = \emptyset$ and $ExpCand_2 = \{\mathtt{d}\}$. Since $\tilde{x}_{21} = \mathtt{c} \in cand_G(Q_2)$, $\mathtt{c}$ is then added to $ChkCand_2$, that is, $ChkCand_2 = \{\mathtt{c}\}$. As the result, in the while loop, we see $|Q_2| + |ExpCand_2| = |\{\mathtt{a}, \mathtt{b}\}| + |\{\mathtt{d}\}| = 3 < 4$. From Pruning 1, therefore, the function call **C2** is now completed.

Returning to the while loop in the function call **C1**, we can similarly continue our procedure. As the result, one more maximal frequent pattern with length 4, $\{\mathtt{a}, \mathtt{c}, \mathtt{d}, \mathtt{f}\}$, can be detected. Finally, the patterns in the list $\mathcal{L}$, $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ and $\{\mathtt{a}, \mathtt{c}, \mathtt{d}, \mathtt{f}\}$, are presented as our target, top-1 colossal frequent patterns.

6 Experimental Results

For our experimentations, our system has been implemented in language *C* and compiled by `gcc-4.4.3` with the optimization of `O3` on Ubuntu-10.04. For comparison, fast maximal frequent itemset miners, *MAFIA* [3], *LCM* [6] and *ABS* [7], have also been compiled. The systems have been executed on a PC with Intel® Core™-i3 M380 (2.53GHz) CPU and 8GB main memory.

We have prepared a synthetic dataset and a famous benchmark dataset, $ExtDiag_n$ and *Accidents*. The former is for comparison with *Pattern Fusion Method* in [12] and the latter with *Max-Clique* in [13]. Since these systems are not publicly available, it is hard to make a direct comparison with them. As the authors in [12,13] have compared their system with *LCM*, we attempt some indirect comparison via *LCM*. It is noted that computation times by our system include those for constructing pattern graphs.

$Diag_n$ is an $n \times (n-1)$ matrix, where i-th raw as a transaction consists of $(n-1)$-items of integers 1 to n without i. It can be extended to $ExtDiag_n$ by adding $n/2$ identical raws each of which consists $(n-1)$-items of integers $n+1$ to $2n-1$. Thus, $Ext - Diag_n$ has $n + (n/2)$ transactions each of which consists of $n-1$ (integer) items. In [12], $ExtDiag_n$ has been discussed as a simple example for which it is quite hard to find frequent patterns with larger length. In case of $n = 40$ and the minimum frequency threshold $\sigma = 20$, $ExtDiag_{40}$ has $\binom{40}{20}$ maximal (closed) frequent patterns with length 20 and only one with length 39. It is easily imagined that most maximal/closed frequent pattern miners cannot finish their computations for the pattern with the largest length because they take exponential time for the patterns with length 20.

Accident is publicly available in the FIMI repository [4] and is widely used as a benchmark dataset for frequent pattern mining. It consists of 340,183 traffic accident records in which 468 distinct items appear. In [13], although *LCM* takes long time for the dataset under various parameter settings, *Max-Clique* can find patterns with larger length with reasonable time.

For $ExtDiag_n$, under several values for n, we have observed computation times by *LCM*, *MAFIA*, *ABS* and our system, where the minimum frequency threshold is set to $\sigma = n/2$ and N for top-N 1. The results are summarized in Figure 6. In the figure, as the value of n increases, computation times by *LCM*, *MAFIA* and *ABS* grow exponentially, where trials for $ExtDiag_n$ have been aborted after 73,200 seconds (about 20 hours) past. On the other hand, our system has took just a few seconds for each value of n because once the unique pattern with the largest length is found, the other maximal patterns with less length can be skipped in our search. Although $ExtDiag_n$ might be considered as an extreme instance, not a few datasets in real domains seem to have a similar characteristic, that is, only a few patterns uncommonly have larger length and hence they are interesting. The authors expect that our algorithm would be particularly useful for such a case.

For *Accident*, varying the minimum frequency threshold σ, we have observed computation times by *LCM*, *MAFIA*, *ABS* and ours, where N for top-N is set to 1. We present the results in Figure 7. As we can observe, in the higher range of σ, *LCM* and *MAFIA* show much better performances than ours. However, the times needed grow very rapidly in the range of lower σ. On the other hand, the times taken by our system are relatively stable even in the lower range of σ. Recently, some kinds of patterns with lower frequency have been paid attention as *rare patterns/concepts* [9,10,18,19]. It is well known that extracting those patterns is a hard task because in general there exist a huge number of patterns belonging to lower

[4] Frequent Itemset Mining Implementations Repository (`http://fimi.ua.ac.be/data/`).

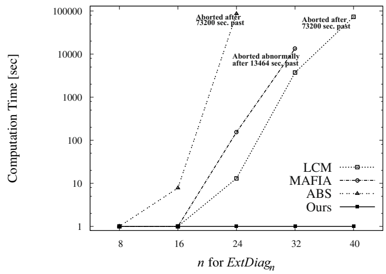


Fig. 6. Computation Times for $ExtDiag_n$

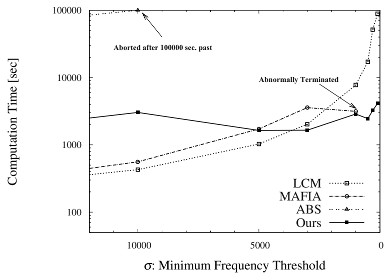


Fig. 7. Computation Times for *Accident*

range of frequency. Our system would be a promising candidate which can work well even in such a hard situation. This is a remarkable advantage of our system.

7 Concluding Remarks

In this paper, we have discussed a problem of finding top-N colossal frequent patterns. Particularly, we have proposed a method for completely and exactly detecting them. For our efficient computation, a depth-first branch-and-bound algorithm has been designed. In the algorithm, our target patterns can be found as cliques in a graph constructed based on a k-item pattern graph. It should be emphasized again that as our search progresses, the graph dynamically becomes sparser. Therefore, it makes our primary task for finding cliques more efficient and our pruning effect more powerful. In our experimentation, it has been observed that our method can efficiently find top-N colossal patterns for datasets $ExtDiag_n$ and *Accident* from which it is hard to exactly identify those patterns by previously proposed algorithms.

Usefulness of our algorithm has to be proved for more datasets including those in real application domains, e.g., Bioinformatics. Improving efficiency of the algorithm is also required for large scale real datasets. For this purpose, several techniques useful for efficiently finding maximum/maximal cliques would be helpful. Concretely speaking, an adequate ordering on vertices (items) to be added in the clique expansion process can drastically improve efficiency of our computation, as has been previously reported in [14,15]. Moreover, *approximate coloring of vertices* can provide us more powerful branch-and-bound pruning. These techniques must be incorporated into our algorithm. It would be also worth investigating colossal patterns (concepts) from the viewpoint of semantics.

References

1. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent Pattern Mining - Current Status and Future Directions. Data Mining and Knowledge Discovery 15(1), 55–86 (2007)
2. Bayardo Jr., R.J.: Efficiently Mining Long Patterns from Databases. In: Proc. of the 1998 ACM SIGMOD Int'l. Conf. on Management of Data, pp. 85–93 (1998)
3. Burdick, D., Calimlim, M., Flannick, J., Gehrke, J., Yiu, T.: MAFIA: A Maximal Frequent Itemset Algorithm. IEEE Transactions on Knowledge and Data Engineering 17(11), 1490–1504 (2005)

4. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. Information Systems 24(1), 25–46 (1999)
5. Wang, J., Han, J., Pei, J.: CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining - KDD 2003, pp. 236–245 (2003)
6. Uno, T., Kiyomi, M., Arimura, H.: LCM ver. 2: Efficient Mining Algorithm for Frequent/Closed/Maximal Itemsets. In: Proc. of IEEE ICDM 2004 Workshop - FIMI 2004 (2004), http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-126/
7. Flouvat, F., Marchi, F.D., Petit, J.: ABS: Adaptive Borders Search of Frequent Itemsets. In: Proc. of IEEE ICDM 2004 Workshop - FIMI 2004 (2004), http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-126/
8. Omiecinski, E.R.: Alternative Interest Measures for Mining Associations in Databases. IEEE Transactions on Knowledge and Data Engineering 15(1), 57–69 (2003)
9. Szathmary, L., Napoli, A., Valtchev, P.: Towards Rare Itemset Mining. In: Proc. of the 19th IEEE Int'l Conf. on Tools with Artificial Intelligence - ICTAI 2007, pp. 305–312 (2007)
10. Troiano, L., Scibelli, G., Birtolo, C.: A Fast Algorithm for Mining Rare Itemsets. In: Proc. of the 2009 9th Int'l Conf. on Intelligent Systems Design and Applications – ISDA 2009, pp. 1149–1155 (2009)
11. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations, p. 284. Springer (1999)
12. Zhu, F., Yan, X., Han, J., Yu, P.S., Cheng, H.: Mining Colossal Frequent Patterns by Core Pattern Fusion. In: Proc. of the 23rd IEEE Int'l Conf. on Data Engineering - ICDE 2007, pp. 706–715 (2007)
13. Xie, Y., Yu, P.S.: Max-Clique: A Top-Down Graph-Based Approach to Frequent Pattern Mining. In: Proc. of the 2010 IEEE Int'l Conf. on Data Mining - ICDM 2010, pp. 1139–1144 (2010)
14. Tomita, E., Akutsu, T., Matsunaga, T.: Efficient Algorithms for Finding Maximum and Maximal Cliques: Effective Tools for Bioinformatics. In: Laskovski, A.N. (ed.) Biomedical Engineering, Trends in Electronics, Communications and Software, pp. 625–640. InTech (2011)
15. Tomita, E., Kameda, T.: An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique with Computational Experiments. Journal of Global Optimization 37(1), 95–111 (2007)
16. Balas, E., Yu, C.S.: Finding a Maximum Clique in an Arbitrary Graph. SIAM Journal on Computing 15(4), 1054–1068 (1986)
17. Eppstein, D., Strash, D.: Listing All Maximal Cliques in Large Sparse Real-World Graphs. In: Pardalos, P.M., Rebennack, S. (eds.) SEA 2011. LNCS, vol. 6630, pp. 364–375. Springer, Heidelberg (2011)
18. Okubo, Y., Haraguchi, M., Nakajima, T.: Finding Rare Patterns with Weak Correlation Constraint. In: Proceedings of the 2010 IEEE International Conference on Data Mining Workshops - ICDMW 2010, pp. 822–829 (2010)
19. Okubo, Y., Haraguchi, M.: An Algorithm for Extracting Rare Concepts with Concise Intents. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS(LNAI), vol. 5986, pp. 145–160. Springer, Heidelberg (2010)

Quantitative Concept Analysis

Dusko Pavlovic

Royal Holloway, University of London, and University of Twente
dusko.pavlovic@rhul.ac.uk

Abstract. Formal Concept Analysis (FCA) begins from a context, given as a binary relation between some objects and some attributes, and derives a lattice of concepts, where each concept is given as a set of objects and a set of attributes, such that the first set consists of all objects that satisfy all attributes in the second, and vice versa. Many applications, though, provide contexts with quantitative information, telling not just whether an object satisfies an attribute, but also quantifying this satisfaction. Contexts in this form arise as rating matrices in recommender systems, as occurrence matrices in text analysis, as pixel intensity matrices in digital image processing, etc. Such applications have attracted a lot of attention, and several numeric extensions of FCA have been proposed. We propose the framework of *proximity sets (proxets)*, which subsume partially ordered sets (posets) as well as metric spaces. One feature of this approach is that it extracts from quantified contexts quantified concepts, and thus allows full use of the available information. Another feature is that the categorical approach allows analyzing any universal properties that the classical FCA and the new versions may have, and thus provides structural guidance for aligning and combining the approaches.

Keywords: concept analysis, enriched category, semantic completion, universal property.

1 Introduction

Suppose that the users $U = \{\text{Abby, Dusko, Stef, Temra, Luka}\}$ provide the following star ratings for the items $J = \{$"Nemo", "Crash" , "Ikiru", "Bladerunner"$\}$

	"Nemo"	"Crash"	"Ikiru"	"Bladerunner"
Abby	★ ★ ★ ★	★ ★ ★ ★ ★	★ ★	★ ★ ★ ★
Dusko	★ ★	★ ★	★ ★ ★ ★	★ ★ ★ ★ ★
Stef	★ ★	★ ★ ★ ★ ★	★ ★ ★	★ ★
Temra	★	★ ★ ★	★ ★ ★	★ ★ ★ ★
Luka	★ ★ ★ ★ ★	★	★	★ ★

This matrix $\Phi = (\Phi_{iu})_{J\times U}$ contains some information about the relations between these users' tastes, and about the relations between the styles of the items (in this case movies) that they rated. The task of data analysis is to extract that information. In particular, given a *context* matrix $\Phi : J \times U \to R$ like in the above table, the task of *concept* analysis is to detect, on one hand, the latent concepts of *taste*, shared by some of the

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 260–277, 2012.

users in U, and on the other hand the latent concepts of *style*, shared by some of the items in J. In Formal Concept Analysis (FCA) [35,15,9,14,33], the latent concepts are expressed as sets: a taste t is a set of users, i.e. a map $U \xrightarrow{t} \{0,1\}$, whereas a style s is a set of items, i.e. a map $J \xrightarrow{s} \{0,1\}$. We explore a slightly refined notion of concept, which tells not just whether two users (resp. two items) share the same taste (resp. style) or not, but it also quantifies the degree of *proximity* of their tastes (resp. styles). This is formalized by expressing a taste as a map $U \xrightarrow{\tau} [0,1]$, and a style as a map $J \xrightarrow{\sigma} [0,1]$. The value τ_u is thus a number from the interval [0,1], telling how close is the taste τ to the user u; whereas the value σ_i tells how close is the item i to the style σ. These concepts are *latent*, in the sense that they are not given in advance, but mined from the context matrix, just like in FCA, and similarly like in Latent Semantic Analysis (LSA) [10]. Although the extracted concepts are interpreted differently for the users in U and for the items in J (i.e. as the tastes and the styles, respectively) it turns out that the two obtained concept structures are isomorphic, just like in FCA and LSA. However, our approach allows initializing a concept analysis session by some prior concept structures, which allow building upon the results of previous analyses, from other data sets, or specified by the analyst. This allows introducing different conceptual backgrounds for the users in U and for the items in J.

Related Work and Background. The task of capturing quantitative data in FCA was recognized early on. The simplest approach is to preprocess any given numeric data into relational contexts by introducing thresholds, and then apply the standard FCA method [13,15]. This basic approach has been extended in several directions, e.g. Triadic Concept Analysis [17,18,26] and Pattern Structures [12,19,20], and refined for many application domains. A different way to introduce numeric data into FCA is to allow *fuzzy* contexts, as binary relations evaluated in an abstract lattice of truth values L. The different ways to lift the FCA constructions along the inclusion $\{0,1\} \hookrightarrow L$ have led to an entire gamut of different versions of fuzzy FCA [3,4,7,8,23], surveyed in [5]. With one notable exception, all versions of fuzzy FCA input quantitative data in the form as fuzzy relations, and output qualitative concept lattices in the standard form. The fact that numeric input data are reduced to the usual lattice outputs can be viewed as an advantage, since the outputs can then be presented, and interpreted, using the available FCA visualization tools and methods. On the other hand, only a limited amount of information contained in a numeric data set can be effectively captured in lattice displays. The practices of spectral methods of concept analysis [1,10,22], pervasive in web commerce, show that the quantitative information received in the input contexts can often be preserved in the output concepts, and effectively used in ongoing analyses. Our work has been motivated by the idea that suitably refined FCA constructions could output concept structures with useful quantitative information, akin to the concept eigenspaces of LSA. It turns out that the steps towards quantitative concepts on the FCA side have previously been made by Bělohlávek in [4], where fuzzy concept lattices derived from fuzzy contexts were proposed and analyzed. This is the mentioned notable exception from the other fuzzy and quantitative approaches to FCA, which all derive just qualitative concept lattices from quantitative contexts. Bělohlávek's basic definitions turn out to be remarkably close to the definitions we start from in the present paper, in spite of the fact that his goal is to generalize FCA using carefully chosen fuzzy structures, whereas we use enriched categories with the ultimate goal to align FCA with the spectral methods for concept analysis, such as LSA. Does this confirm that the structures obtained in both cases naturally arise from the shared FCA foundations, rather than from either the fuzzy or the categorical approach? The ensuing analyses, however, shed light on these structures from essentially different angles, and open up completmentary views: while Bělohlávek provides a detailed analysis of the internal structure of fuzzy concept lattices, we provide a high level view of their universal properties, from which some internal features follow, and which offers guidance through the maze of the available structural choices. Combining the two methods seems to open interesting alleys for future work.

Our motivating example suggests that our goals might be related to those of [11], where an FCA approach to recommender systems was proposed. However, the authors of [11] use FCA to tackle the problem of *partial* information (the missing ratings) in recommender systems, and they abstract away the *quantitative* information (contained in the available ratings); whereas our goal is to capture this quantitative information, and we leave the problem of partial information aside for the moment.

Outline of the Paper. In Sec. 2 we introduce *proximity sets (proxets)*, the mathematical formalism supporting the proposed generalization of FCA. Some constructions and notations used throughout the paper are introduced in Sec. 2.2. Since proxets generalize posets, in Sec. 3 we introduce the corresponding generalizations of infimum and supremum, and spell out the basic completion constructions, and the main properties of the infimum (resp. supremum) preserving morphisms. In Sec. 4, we study context matrices over proximity sets, and describe their decomposition, with a universal property analogous to the Singular Value Decomposition of matrices in linear algebra. Restricting this decomposition from proxets to discrete posets (i.e. sets) yields FCA. The drawback of this quantitative version of FCA is that in it a finite context generally allows an infinite proxet of concepts, whereas in the standard version of FCA, of course, finite contexts lead to finite concept lattices. This problem is tackled in Sec. 5, where we show how the users and the items, as related in the context, induce a finite generating set of concepts. Sec. 6 provides a discussion of the obtained results and ideas for the future work.

2 Proxets

2.1 Definition, Intuition, Examples

Notation. Throughout the paper, the order and lattice structure of the interval $[0, 1]$ are denoted by $\leq$, $\wedge$ and $\vee$, whereas $\cdot$ denotes the multiplication in it.

Definition 2.1. *A* proximity *over a set A is a map* $(\vdash) : A \times A \to [0, 1]$ *which is*

- *reflexive:* $(x \vdash x) = 1$,
- *transitive:* $(x \vdash y) \cdot (y \vdash z) \ \leq \ (x \vdash z)$, *and*
- *antisymmetric:* $(x \vdash y) = 1 = (y \vdash x) \implies x = y$

If only reflexity and transitivity are satisfied, and not antisymmetry, then we have an intensional *proximity map. The antisymmetry condition is sometimes called* extensionality. *A(n intensional)* proximity set*, or* proxet*, is a set equipped with a(n intensional)* *proximity map. A* proximity *(or* monotone*) morphism between the proxets A and B is a function* $f : A \to B$ *such that all* $x, y \in A$ *satisfy* $(x \vdash y)_A \le (fx \vdash fy)_B$. *We denote by* Prox *the category of proxets and their morphisms.*

Categorical View. A categorically minded reader can understand intensional proxets as categories enriched [21] over the poset [0, 1] viewed as a category, with the monoidal structure induced by the multiplication. In the presence of reflexivity and transitivity, $(x \vdash y) = 1$ is equivalent with $\forall z.\ (z \vdash x) \le (z \vdash y)$, and with $\forall z.\ (x \vdash z) \ge (y \vdash z)$. A proximity map is thus asymmetric if and only if $(\forall z.\ (z \vdash x) = (z \vdash y)) \Rightarrow x = y$, and if and only if $(\forall z.\ (z \vdash x) = (z \vdash y)) \Rightarrow x = y$. This means that extensional proxets correspond to *skeletal* [0, 1]-enriched categories.

Examples. The first example of a proxet is the interval [0, 1] itself, with the proximity

$$(x \vdash y)_{[0,1]} = \begin{cases} \frac{y}{x} & \text{if } y < x \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

Note that $(\ \vdash\ ) : [0, 1] \times [0, 1] \to [0, 1]$ is now an *operation* on [0,1], satisfying

$$(x \cdot y) \le z \iff x \le (y \vdash z) \tag{2}$$

A wide family of examples follows from the fact that proximity sets (proxets) generalize partially ordered sets (posets), in the sense that any poset S can be viewed as a proxet $\mathsf{W}S$, with the proximity induced by the partial ordering $\underset{S}{\sqsubseteq}$ as follows:

$$(x \vdash y)_{\mathsf{W}S} = \begin{cases} 1 & \text{if } x \underset{S}{\sqsubseteq} y \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The proxet $\mathsf{W}S$ is intensional if and only if S is just a *preorder*, in the sense that the relation $\underset{S}{\sqsubseteq}$ is just transitive and reflexive. The other way around, any (intensional) proxet A induces two posets (resp. preorders), ΥA and ΛA, with the same underlying set and

$$x \underset{\Upsilon A}{\sqsubseteq} y \iff (x \vdash y)_A = 1 \qquad\qquad x \underset{\Lambda A}{\sqsubseteq} y \iff (x \vdash y)_A > 0$$

Since the constructions W, Υ and Λ, extended on maps, preserve monotonicity, a categorically minded reader can easily confirm that we have three functors, which happen to form two adjunctions $\Lambda \dashv \mathsf{W} \dashv \Upsilon : \mathsf{Prox} \to \mathsf{Pos}$. Since $\mathsf{W} : \mathsf{Pos} \hookrightarrow \mathsf{Prox}$ is an embedding, Pos is thus a reflective and correflective subcategory of Prox. This means that $\Lambda\mathsf{W}S = S = \Upsilon\mathsf{W}S$ holds for every poset S, so that posets are exactly the proxets where the proximities are evaluated only in 0 or 1; and that ΛA and ΥA are respectively the initial and the final poset induced by the proxet A, as witnessed by the obvious morphisms $\mathsf{W}\Upsilon A \to A \to \mathsf{W}\Lambda A$. The same universal properties extend to a correspondence between intensional proxets and preorders.

A different family of examples is induced by metric spaces: any metric space X with a distance map $d : X \times X \to [0, \infty]$ can be viewed as a proxet with the proximity map

$$(x \vdash y) = 2^{-d(x,y)} \tag{4}$$

Proxets are thus a common generalization of posets and metric spaces. But the usual metric distances are symmetric, i.e. satisfy $d(x,y) = d(y,x)$, whereas the proximities need not be. The inverse of (4) maps any proximity to a *quasi*-metric $d(x,y) = -\log(x \vdash y)$ [36], whereas intensional proximities induce *pseudo*-quasi-metrics [37]. For a concrete family of examples of quasi-metrics, take any family of sets $\mathcal{X} \subseteq \wp X$, and define

$$d(x,y) = |y \setminus x|$$

The distance of x and y is thus the number of elements of y that are not in x. This induces the proximity $(x \vdash y) = 2^{-|y\setminus x|}$. If $\mathcal{X}$ is a set of documents, viewed as bags (multisets) of terms, then both constructions can be generalized to count the difference in the numbers of the occurrences of terms in documents, and the set difference becomes multiset subtraction.

Proximity or Distance? The isomorphism $-\log x : [0,1] \rightleftarrows [0,\infty] : 2^{-x}$ is easily seen to lift to an isomorphism between the category of proxets, as categories enriched over the multiplicative monoid $[0,1]$ and the category of generalized metric spaces, as categories enriched over the additive monoid $[0,\infty]$. Categorical studies of generalized metric spaces were initiated in [25], continued in denotational semantics of programming languages [34,6,24], and have recently turned out to be useful for quantitative distinctions in ecology [27]. The technical results of this paper could equivalently be stated in the framework of generalized metric spaces. While this would have an advantage of familiarity to certain communities, the geometric intuitions that come with metrics turn out to be misleading when imposed on the applications that are of interest here. The lifting of infima and suprema is fairly easy from posets to proxets, but leads to mysterious looking operations over metrics. In any case, the universal properties of matrix decompositions do not seem to have been studied in either framework so far.

2.2 Derived Proxets and Notations

Any proxets A, B give rise to other proxets by following standard constructions:

- the *dual* (or *opposite*) proxet $\overline{A}$, with the same underlying set and the proximity $(x \vdash y)_{\overline{A}} = (y \vdash x)_A$;
- the *product* proxet $A \times B$ over the cartesian product of the underlying sets, and the proximity $(x,u \vdash y,v)_{A\times B} = (x \vdash y)_A \wedge (u \vdash v)_B$
- the *power* proxet B^A over the monotone maps, i.e. $\mathsf{Prox}(A,B)$ as the underlying set, with the proximity $(f \vdash g)_{B^A} = \bigwedge_{x\in A} (fx \vdash gx)_B$.

There are natural correspondences of proxet morphisms

$$\mathsf{Prox}(A,B) \times \mathsf{Prox}(A,C) \cong \mathsf{Prox}(A, B\times C) \quad \text{and} \quad \mathsf{Prox}(A\times B, C) \cong \mathsf{Prox}(A, C^B)$$

Notations. In any proxet A, it is often convenient to abbreviate $(x \vdash y)_A = 1$ to $x \underset{A}{\leq} y$. For $f, g : A \to B$, it is easy to see that $f \underset{B^A}{\leq} g$ if and only if $fx \underset{B}{\leq} gx$ for all $x \in A$.

3 Vectors, Limits, Adjunctions

3.1 Upper and Lower Vectors

Having generalized posets to proxets, we proceed to lift the concepts of the least upper bound and the greatest lower bound. Let $(S, \sqsubseteq)$ be a poset and let $L, U \subseteq S$ be a lower set and an upper set, respectively, in the sense that

$$(x \sqsubseteq y \text{ and } y \in L) \Rightarrow x \in L \qquad\qquad (x \in U \text{ and } x \sqsubseteq y) \Rightarrow y \in U$$

Then an element denoted $\bigsqcup L$ is supremum of L, and $\bigsqcap U$ is the infimum of U, if all $x, y \in A$ satisfy

$$\bigsqcup L \leq y \iff \forall x.\ (x \in L \Rightarrow x \sqsubseteq y) \tag{5}$$

$$x \leq \bigsqcap U \iff \forall y.\ (y \in U \Rightarrow x \sqsubseteq y) \tag{6}$$

We generalize these definitions to proxet limits in (7-8). To generalize the lower sets, over which the suprema are taken, and the upper sets for infima, observe that any upper set $U \subseteq S$ corresponds to a monotone map $\overrightarrow{U} : S \to \{0, 1\}$, whereas every lower set L corresponds to an antitone map $\overleftarrow{L} : \overline{S} \to \{0, 1\}$, where $\overline{S}$ is the dual proxet defined in Sec. 2.2.

Definition 3.1. *An* upper *and a* lower vector *in a proxet A are the monotone maps* $\overrightarrow{\upsilon} : A \to [0, 1]$ *and* $\overleftarrow{\lambda} : \overline{A} \to [0, 1]$. *The sets of vectors* $\Uparrow A = \overline{[0, 1]^A}$ *and* $\Downarrow A = [0, 1]^{\overline{A}}$ *form proxets, with the proximity* $\left(\overrightarrow{\upsilon} \vdash \overrightarrow{\tau}\right)_{\Uparrow A} = \bigwedge_{x \in A} \left(\overrightarrow{\tau}_x \vdash \overrightarrow{\upsilon}_x\right)_A$ *and* $\left(\overleftarrow{\lambda} \vdash \overleftarrow{\mu}\right)_{\Downarrow A} = \bigwedge_{x \in A} \left(\overleftarrow{\lambda}_x \vdash \overleftarrow{\mu}_x\right)_A$ *computed by infima in* $[0, 1]$.

Remark. Note that $(x \vdash y) \leq \left(\overrightarrow{\upsilon}_x \vdash \overrightarrow{\upsilon}_y\right)$ is equivalent with $\overrightarrow{\upsilon}_x \cdot (x \vdash y) \leq \overrightarrow{\upsilon}_y$, and $(x \vdash y) \leq \left(\overleftarrow{\lambda}_x \vdash \overleftarrow{\lambda}_y\right)$ with $(x \vdash y) \cdot \overleftarrow{\lambda}_y \leq \overleftarrow{\lambda}_x$.

3.2 Limits

Definition 3.2. *The* upper limit *or* supremum $\coprod \overleftarrow{\lambda}$ *of the lower vector* $\overleftarrow{\lambda}$ *and the* lower limit *or* infimum $\prod \overrightarrow{\upsilon}$ *of the upper vector* $\overrightarrow{\upsilon}$ *are the elements of A that satisfy for every* $x, y \in A$

$$\left(\coprod \overleftarrow{\lambda} \vdash y\right)_A = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \vdash y)_A \tag{7}$$

$$\left(x \vdash \prod \overrightarrow{\upsilon}\right)_A = \bigwedge_{y \in A} \overrightarrow{\upsilon}_y \vdash (x \vdash y)_A \tag{8}$$

The proxet A is complete *under infima (resp. suprema) if every upper (resp. lower) vector has an infimum (resp. supremum), which thus yield the operations* $\prod : \Uparrow A \to A$ *and* $\coprod : \Downarrow A \to A$

Remarks. Condition (7) generalizes (5), whereas (8) generalizes (6). Note how proximity operation $\vdash$ over [0,1], defined in (1), plays in (7–8) the role that the implication $\Rightarrow$ over $\{0, 1\}$ played in (5–6). This is justified by the fact that $\vdash$ is adjoint to the multiplication in $[0, 1]$, in the sense of (2), in the same sense in which $\Rightarrow$ is adjoint to the meet in $\{0, 1\}$, or in any Heyting algebra, in the sense of $(x \wedge y) \leq z \iff x \leq (y \Rightarrow z)$.

An element w of a poset S is an upper bound of $L \subseteq S$ if it satisfies just one direction of (5), i.e. $(w \sqsubseteq y) \Longrightarrow \forall x.\ (x \in L \Rightarrow x \sqsubseteq y)$. Ditto for the lower bounds. In a proxet A, u is an upper bound of $\overleftarrow{\lambda}$ and ℓ is a lower bound of $\overrightarrow{\upsilon}$ if all $x, y \in A$ satisfy

$$(u \vdash y)_A \leq \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \vdash y)_A$$

$$(x \vdash \ell)_A \leq \bigwedge_{y \in A} \overrightarrow{\upsilon}_y \vdash (x \vdash y)_A$$

Using (2) and instantiating y to u in the first inequality, and x to ℓ in the second one, these conditions can be shown to be equivalent with $\overleftarrow{\lambda}_x \leq (x \vdash u)_A$ and $\overrightarrow{\upsilon}_y \leq (\ell \vdash y)_A$, which characterize the upper and the lower bounds in proxets.

3.3 Completions

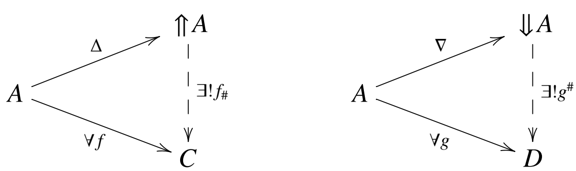
Each element a of a proxet A induces two *representable* vectors

$$\begin{array}{rclcrcl} \Delta a & : & A \to [0,1] & \qquad & \nabla a & : & \overline{A} \to [0,1] \\ & & x \mapsto (a \vdash x)_A & & & & x \mapsto (x \vdash a)_A \end{array}$$

It is easy to see that these maps induce proximity morphisms $\Delta : A \to \Uparrow A$ and $\nabla : A \to \Downarrow A$, which correspond to the categorical *Yoneda embeddings* [29, Sec. III.2]. They make $\Uparrow A$ into the lower completion, and $\Downarrow A$ into the upper completion of the proxet A.

Proposition 3.3. *$\Uparrow A$ is upper complete and $\Downarrow A$ is lower complete. Moreover, they are universal, in the sense that*

- *any monotone $f : A \to C$ into a complete proxet C induces a unique $\prod$-preserving morphism $f_\# : \Uparrow A \to C$ such that $f = f_\# \circ \Delta$;*
- *any monotone $g : A \to D$ into a cocomplete proxet D induces a unique $\coprod$-preserving morphism $g^\# : \Downarrow A \to D$ such that $g = g^\# \circ \nabla$.*



3.4 Adjunctions

Proposition 3.4. *For any proximity morphism* $f : A \to B$ *holds* $(a) \iff (b) \iff (c)$ *and* $(d) \iff (e) \iff (f)$*, where*

(a) $f\left(\coprod \overleftarrow{\lambda}\right) = \coprod f\left(\overleftarrow{\lambda}\right)$
(b) $\exists f_* : B \to A\ \forall x \in A\ \forall y \in B.\ (fx \vdash y)_B = (x \vdash f_* y)_A$
(c) $\exists f_* : B \to A.\ \mathrm{id}_A \le f_* f \wedge f f_* \le \mathrm{id}_B$
(d) $f\left(\prod \overrightarrow{\upsilon}\right) = \prod f\left(\overrightarrow{\upsilon}\right)$
(e) $\exists f^* : B \to A\ \forall x \in A\ \forall y \in B.\ (f^* y \vdash x)_B = (y \vdash fx)_A$
(f) $\exists f^* : B \to A.\ f^* f \le \mathrm{id}_A \wedge \mathrm{id}_B \le f f^*$

The morphisms f^* *and* f_* *are unique, whenever they exist.*

Definition 3.5. *An* upper adjoint *is a proximity morphism satisfying (a-c) of Prop. 3.4; a* lower adjoint *satisfies (d-f). A* (proximity) adjunction *between proxets A and B is a pair of proximity morphisms* $f^* : A \rightleftarrows B : f_*$ *related as in (b-c) and (e-f).*

3.5 Projectors and Nuclei

Proposition 3.6. *For any adjunction* $f^* : A \rightleftarrows B : f_*$ *holds* $(a) \iff (b)$ *and* $(c) \iff (d)$*, where*

(a) $\forall xy \in B.\ (f_* x \vdash f_* y)_A = (x \vdash y)_B$
(b) $f^* f_* = \mathrm{id}_B$
(c) $\forall xy \in A.\ (f^* x \vdash f^* y)_B = (x \vdash y)_A$
(d) $f_* f^* = \mathrm{id}_A$

Definition 3.7. *An adjunction satisfying (a-b) of Prop. 3.6 is an* upper projector*; an adjunction satisfying (c-d) is a* lower projector. *The upper (resp. lower) component of an upper (resp. lower) projector is called the upper (lower)* projection. *The other component (i.e. the one in (a), resp. (c)) is called the upper (lower)* embedding.

Proposition 3.8. *Any upper (lower) adjoint factors, uniquely up to isomorphism, through an upper (lower) projection followed by an upper (lower) embedding through the proxet*

$$\wr f \int = \{\langle x, y\rangle \in A \times B \mid f^* x = y \wedge x = f_* y\}$$

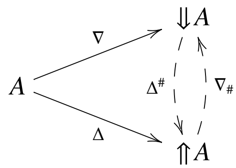
Definition 3.9. *A* nucleus *of the adjunction* $f^* : A \rightleftarrows B : f_*$ *consists of a proxet* $\wr f \int$ *together with*

- *embeddings* $A \overset{e_*}{\hookleftarrow} \wr f \int \overset{e^*}{\hookrightarrow} B$
- *projections* $A \overset{p^*}{\twoheadrightarrow} \wr f \int \overset{p_*}{\twoheadleftarrow} B$

such that $f^* = e^* p^*$ *and* $f_* = e_* p_*$.

3.6 Cones and Cuts

The *cone operations* are the proximity morphisms $\Delta^{\#}$ and $\nabla_{\#}$



These morphisms are induced by the universal properties of the Yoneda embeddings ∇ and Δ as completions, stated in Prop. 3.3. Since by definition $\Delta^{\#}$ preserves suprema, and $\nabla_{\#}$ preserves infima, Prop. 3.4 implied that each of them is an adjoint, and it is not hard to see that they form the adjunction $\Delta^{\#} : \Downarrow A \rightleftarrows \Uparrow A : \nabla_{\#}$. Spelling them out yields

$$\left(\Delta^{\#}\overleftarrow{\lambda}\right)_a = \bigwedge_{x\in A} \overleftarrow{\lambda}_x \vdash (x \vdash a) \qquad\qquad \left(\nabla_{\#}\overrightarrow{\upsilon}\right)_a = \bigwedge_{x\in A} \overrightarrow{\upsilon}_x \vdash (a \vdash x)$$

Intuitively, $\left(\Delta^{\#}\overleftarrow{\lambda}\right)_a$ is the proximity of $\overleftarrow{\lambda}$ to a as its upper bound, as discussed in Sec. 3.2. Visually, $\left(\Delta^{\#}\overleftarrow{\lambda}\right)_a$ thus measures the *cone* from $\overleftarrow{\lambda}$ to a, whereas $\left(\nabla_{\#}\overrightarrow{\upsilon}\right)a$ measures the cone from a to $\overrightarrow{\upsilon}$.

Proposition 3.10. *For every $\overleftarrow{\lambda} \in \Downarrow A$ every $\overrightarrow{\upsilon} \in \Uparrow A$ holds*

$$\begin{array}{lcl} \overleftarrow{\lambda} \leq \nabla_{\#}\Delta^{\#}\overleftarrow{\lambda} & \textit{and} & \overleftarrow{\lambda} \geq \nabla_{\#}\Delta^{\#}\overleftarrow{\lambda} \iff \exists \overrightarrow{\upsilon}.\ \overleftarrow{\lambda} = \Delta^{\#}\overrightarrow{\upsilon} \\ \overrightarrow{\upsilon} \leq \Delta^{\#}\nabla_{\#}\overrightarrow{\upsilon} & \textit{and} & \overrightarrow{\upsilon} \geq \Delta^{\#}\nabla_{\#}\overrightarrow{\upsilon} \iff \exists \overleftarrow{\lambda}.\ \overrightarrow{\upsilon} = \nabla_{\#}\overleftarrow{\lambda} \end{array}$$

The transpositions make the following subproxets isomorphic

$$\begin{aligned} (\Downarrow A)_{\nabla_{\#}\Delta^{\#}} &= \left\{\overleftarrow{\lambda} \in \Downarrow A \mid \overleftarrow{\lambda} = \nabla_{\#}\Delta^{\#}\overleftarrow{\lambda}\right\} \\ (\Uparrow A)_{\Delta^{\#}\nabla_{\#}} &= \left\{\overrightarrow{\upsilon} \in \Uparrow A \mid \overrightarrow{\upsilon} = \Delta^{\#}\nabla_{\#}\overrightarrow{\upsilon}\right\} \end{aligned}$$

Definition 3.11. *The vectors in $(\Downarrow A)_{\nabla_{\#}\Delta^{\#}}$ and $(\Uparrow A)_{\Delta^{\#}\nabla_{\#}}$ are called* cones. *The associated cones $\overleftarrow{\gamma} \in (\Downarrow A)_{\nabla_{\#}\Delta^{\#}}$ and $\overrightarrow{\gamma} \in (\Uparrow A)_{\Delta^{\#}\nabla_{\#}}$ such that $\overleftarrow{\gamma} = \nabla_{\#}\overrightarrow{\gamma}$ and $\overrightarrow{\gamma} = \Delta^{\#}\overleftarrow{\gamma}$ a* cut $\gamma = \langle\overleftarrow{\gamma}, \overrightarrow{\gamma}\rangle$ *in proxet A. Cuts form a proxet $\Updownarrow A$, isomorphic with $(\Downarrow A)_{\nabla_{\#}\Delta^{\#}}$ and $(\Uparrow A)_{\Delta^{\#}\nabla_{\#}}$, with the proximity*

$$(\gamma \vdash \varphi)_{\Updownarrow A} = \left(\overleftarrow{\gamma} \vdash \overleftarrow{\varphi}\right)_{\Downarrow A} = \left(\overrightarrow{\gamma} \vdash \overrightarrow{\varphi}\right)_{\Uparrow A}$$

Lemma 3.12. *The $\Updownarrow A$-infima are constructed in $\Downarrow A$, and $\Updownarrow A$ suprema are constructed in $\Uparrow A$.*

Corollary 3.13. *A proxet A has all suprema if and only if it has all infima.*

Dedekind-MacNeille Completion is a Special Case. If A is a poset, viewed by (3) as the proxet $\mathsf{W}A$, then $\Updownarrow \mathsf{W}A$ is the Dedekind-MacNeille completion of A [28]. The above construction extends the Dedekind-MacNeille completion to the more general framework of proxets, in the sense that it satisfies in the universal property of the Dedekind-MacNeille completion [2]. The construction seems to be novel in the familiar frameworks of metric and quasi-metric spaces. However, Quantitative Concept Analysis requires that we lift this construction to matrices.

4 Proximity Matrices and Their Decomposition

4.1 Definitions, Connections

Definition 4.1. *A* proximity matrix *Φ from proxet A to proxet B is a vector $\Phi : \overline{A} \times B \to [0,1]$. We write it as $\Phi : A \looparrowright B$, and write its value $\Phi(x,y)$ at $x \in A$ and $y \in B$ in the form $(x \models y)_\Phi$. The matrix composition of $\Phi : A \looparrowright B$ and $\Psi : B \looparrowright C$ is defined*

$$(x \models z)_{(\Phi\,;\Psi)} = \bigvee_{y \in B} (x \models y)_\Phi \cdot (y \models z)_\Psi$$

With this composition and the identity matrices $\mathrm{Id}_A : A \times A \to [0,1]$ *where* $\mathrm{Id}_A(x,x') = (x \vdash x')_A$*, proxets and proxet matrices form the category* Matr.

Remark. Note that the defining condition $(u \vdash x) \cdot (y \vdash v) \le ((x \models y)_\Phi \vdash (u \models v)_\Phi)$, which says that Φ is a proximity morphism $\overline{A} \times B \to [0,1]$, can be equivalently written

$$(u \vdash x) \cdot (x \models y)_\Phi \cdot (y \vdash v) \;\le\; (u \models v)_\Phi \tag{9}$$

Definition 4.2. *The* dual $\Phi^\ddagger : B \looparrowright A$ *of a matrix $\Phi : A \looparrowright B$ has the entries*

$$(y \models x)_{\Phi^\ddagger} = \bigwedge_{\substack{u \in A \\ v \in B}} (u \models v)_\Phi \vdash ((u \vdash x)_A \cdot (y \vdash v)_B)$$

A matrix $\Phi : A \looparrowright B$ where $\Phi^{\ddagger\ddagger} = \Phi$ is called a suspension.

Remarks. It is easy to see by Prop. 3.10 that $(x \vdash y)_\Phi \le (x \vdash y)_{\Phi^{\ddagger\ddagger}}$ holds for all $x \in A$ and $y \in B$, and that Φ is a suspension if and only if there is some $\Psi : B \looparrowright A$ such that $\Phi = \Psi^\ddagger$. It is easy to see that $\Phi \le \Psi \Rightarrow \Phi^\ddagger \ge \Psi^\ddagger$, and thus $\Phi \le \Phi^{\ddagger\ddagger}$ implies $\Phi^\ddagger = \Phi^{\ddagger\ddagger\ddagger}$.

Definition 4.3. *The matrices $\Phi : A \looparrowright B$ and $\Psi : B \looparrowright A$ form a* connection *if* $\Phi\,;\Psi \le \mathrm{Id}_A$ *and* $\Psi\,;\Phi \le \mathrm{Id}_B$.

Proposition 4.4. *$\Phi : A \looparrowright B$ and $\Phi^\ddagger : B \looparrowright A$ always form a connection.*

Definition 4.5. *A matrix $\Phi : A \looparrowright B$ is* embedding *if $\Phi\,;\Phi^\ddagger = \mathrm{Id}_A$; and a* projection *if* $\Phi^\ddagger\,;\Phi = \mathrm{Id}_B$.

Definition 4.6. *A* decomposition *of a matrix $\Phi : A \looparrowright B$ consists of a proxet D, with*

– *projection matrix* $P : A \looparrowright D$, *i.e.* $(d \vdash d')_D = \bigvee_{x \in A} (d \models x)_{P^\ddagger} \cdot (x \models d')_P$,
– *embedding matrix* $E : D \looparrowright B$, *i.e.* $(d \vdash d')_D = \bigvee_{y \in B} (d \models y)_E \cdot (y \models d')_{E^\ddagger}$,

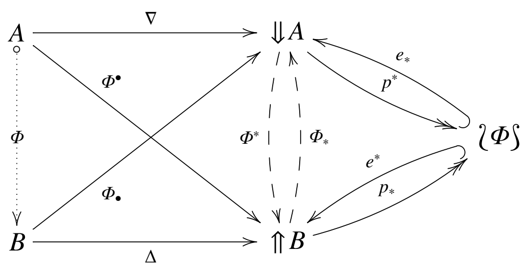
such that $\Phi = P \,;E$, *i.e.* $(x \models y)_\Phi = \bigvee_{d \in D} (x \models d)_P \cdot (d \models y)_E$.

Matrices as Adjunctions. A matrix $\Phi : A \looparrowright B$ can be equivalently presented as either of the proximity morphisms $\Phi_\bullet$ and $\Phi^\bullet$, which extend to Φ_* and Φ^* using Thm. 3.3

$$\dfrac{\overline{A} \times B \xrightarrow{\Phi} [0,1]}{\dfrac{A \xrightarrow{\Phi_\bullet} \Uparrow B \qquad\qquad B \xrightarrow{\Phi^\bullet} \Downarrow A}{\Downarrow A \xrightarrow{\Phi^*} \Uparrow B \qquad\qquad \Uparrow B \xrightarrow{\Phi_*} \Downarrow A}}$$

$$\left(\Phi^* \overleftarrow{\lambda}\right)_b = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (x \models b)_\Phi \qquad\qquad \left(\Phi_* \overrightarrow{\upsilon}\right)_a = \bigwedge_{y \in B} \overrightarrow{\upsilon}_y \vdash (a \models y)_\Phi \tag{10}$$

Both extensions, and their nucleus, are summarized in diagram (11).



(11)

The adjunction $\Phi^* : \Downarrow A \rightleftarrows \Uparrow B : \Phi_*$ means that

$$\left(\Phi^* \overleftarrow{\lambda} \vdash \overrightarrow{\upsilon}\right)_{\Uparrow B} = \bigwedge_{y \in B} \overrightarrow{\upsilon}_y \vdash (\Phi^* \overleftarrow{\lambda})_y = \bigwedge_{x \in A} \overleftarrow{\lambda}_x \vdash (\Phi_* \overrightarrow{\upsilon})_x = \left(\overleftarrow{\lambda} \vdash \Phi_* \overrightarrow{\upsilon}\right)_{\Downarrow A}$$

holds. The other way around, it can be shown that any adjunction between $\Downarrow A$ and $\Uparrow B$ is completely determined by the induced matrix from A to B.

Proposition 4.7. *The matrices* $\Phi \in \mathsf{Matr}(A, B)$ *are in a bijective correspondence with the adjunctions* $\Phi^* : \Downarrow A \rightleftarrows \Uparrow B : \Phi_*$.

4.2 Matrix Decomposition through Nucleus

Prop. 3.10 readily lifts to matrices.

Proposition 4.8. *For every* $\overleftarrow{\alpha} \in \Downarrow A$ *every* $\overrightarrow{\beta} \in \Uparrow B$ *holds*

$$\overleftarrow{\alpha} \le \Phi_* \Phi^* \overleftarrow{\alpha} \qquad \text{and} \qquad \overleftarrow{\alpha} \ge \Phi_* \Phi^* \overleftarrow{\alpha} \iff \exists \overrightarrow{\beta} \in \Uparrow B.\ \overleftarrow{\alpha} = \Phi^* \overrightarrow{\beta}$$
$$\overrightarrow{\beta} \le \Phi^* \Phi_* \overrightarrow{\beta} \qquad \text{and} \qquad \overrightarrow{\beta} \ge \Phi^* \Phi_* \overrightarrow{\beta} \iff \exists \overleftarrow{\alpha} \in \Downarrow A.\ \overrightarrow{\beta} = \Phi_* \overleftarrow{\alpha}$$

The adjunction $\Phi^* : A \rightleftarrows B : \Phi_*$ induces the isomorphisms between the following proxets

$$\wr\Phi\int_A = \left\{\overleftarrow{\alpha} \in \Downarrow A \mid \overleftarrow{\alpha} = \Phi_*\Phi^*\overleftarrow{\alpha}\right\}$$
$$\wr\Phi\int_B = \left\{\overrightarrow{\beta} \in \Uparrow B \mid \overrightarrow{\beta} = \Phi^*\Phi_*\overrightarrow{\beta}\right\}$$
$$\wr\Phi\int = \left\{\gamma = \langle\overleftarrow{\gamma}, \overrightarrow{\gamma}\rangle \in \Downarrow A \times \Uparrow B \mid \overleftarrow{\gamma} = \Phi_*\overrightarrow{\gamma} \wedge \Phi^*\overleftarrow{\gamma} = \overrightarrow{\gamma}\right\}$$

with the proximity

$$(\gamma \vdash \varphi)_{\wr\Phi\int} = \left(\overleftarrow{\gamma} \vdash \overleftarrow{\varphi}\right)_{\Downarrow A} = \left(\overrightarrow{\gamma} \vdash \overrightarrow{\varphi}\right)_{\Uparrow B}$$

Definition 4.9. $\wr\Phi\int$ *is called the* nucleus *of the matrix Φ. Its elements are the Φ*-cuts.

Theorem 4.10. *The matrix $\Phi : A \looparrowright B$ decomposes through $\wr\Phi\int$ into*

- *the projection $P^* : A \looparrowright \wr\Phi\int$ with $\left(x \models \langle\overleftarrow{\alpha}, \overrightarrow{\beta}\rangle\right)_{P^*} = \overleftarrow{\alpha}_x$, and*
- *the embedding $E^* : \wr\Phi\int \looparrowright B$ with $\left(\langle\overleftarrow{\alpha}, \overrightarrow{\beta}\rangle \models y\right)_{E^*} = \overrightarrow{\beta}_y$*

4.3 Universal Properties

Any proxet morphism $f : A \to B$ induces two matrices, $\Omega f : A \looparrowright B$ and $\mho f : B \looparrowright A$ with

$$(x \models y)_{\Omega f} = (fx \vdash y)_B \qquad (y \models x)_{\mho f} = (y \vdash fx)_B$$

Definition 4.11. *A* proximity matrix morphism *from a matrix $\Phi : F_0 \looparrowright F_1$ to $\Gamma : G_0 \looparrowright G_1$ consists of pair of monotone maps $h_0 : F_0 \to G_0$ and $h_1 : F_1 \to G_1$ such that*

- $\Omega h_0 ; \Gamma = \Phi ; \Omega h_1$,
- *h_0 preserves any $\coprod$ that may exist in F_0,*
- *h_1 preserves any $\prod$ that may exist in F_1.*

Let MMat *denote the category of proxet matrices and matrix morphisms. Let* CMat *denote the full subcategory spanned by proximity matrices between complete proxets.*

Proposition 4.12. CMat *is reflective in* MMat *along* $\wr - \int$: MMat $\rightleftarrows$ CMat : U

Posets and FCA. If A and B are posets, a $\{0, 1\}$-valued proxet matrix $\Phi : A \looparrowright B$ can be viewed as a subposet $\Phi \subseteq A \times B$, lower closed in A and upper closed in B. The adjunction $\Phi^* : A \rightleftarrows B : \Phi_*$ is the Galois connection induced by Φ, and the posetal nucleus $\wr\Phi\int$ is now the complete lattice such that

- $A \overset{\nabla}{\to} \Downarrow A \twoheadrightarrow \wr\Phi\int$ is $\vee$-generating and $\wedge$-preserving,
- $B \overset{\nabla}{\to} \Uparrow B \twoheadrightarrow \wr\Phi\int$ is $\wedge$-generating and $\vee$-preserving.

When A and B are discrete posets, i.e. with all elements incomparable, then any binary relation $R \subseteq A \times B$ can be viewed as a proxet matrix between them. Restricting to the vectors that take their values in 0 and 1 yields $\Downarrow A \cong (\wp A, \subseteq)$ and $\Uparrow B \cong (\wp B, \supseteq)$. The concept lattice of FCA then arises from the Galois connection $R^* : \Downarrow A \rightleftarrows \Uparrow B : R_*$ as the concept lattice $\wr R \wr$. Restricted to $\{0, 1\}$-valued matrices between discrete sets A and B, Prop. 4.12 thus yields a universal construction of a lattice $\vee$-generated by A and $\wedge$-generated by B. The FCA concept lattice derived from a context Φ is thus its posetal nucleus $\wr \Phi \wr$. This universal property is closely related with the methods and results of [2,16].

Lifting the Basic Theorem of FCA. The Basic Theorem of FCA says that every complete lattice can be realized as a concept lattice, namely the the one induced by the context of its own partial order. For quantitative concept analysis, this is an immediate consequence of Prop.4.12, which implies a proxet A is complete if and only if $\mathrm{Id}_A = \wr \mathrm{Id}_A \wr$. Intuitively, this just says that nucleus, as a completion, preserves the structure that it completes, and must therefore be idempotent, as familiar from the Dedekind-MacNeille construction. It should be noted that this property does not generalize beyond proxets.

5 Representable Concepts and Their Proximities

5.1 Decomposition without Completion

The problem with factoring matrices $\Phi : A \looparrowright B$ through $\wr \Phi \wr$ in practice is that $\wr \Phi \wr$ is a large, always infinite structure. The proxet $\wr \Phi \wr$ is the completion of the matrix $\Phi : A \looparrowright B$ in the sense that it is

- the subproxet of the $\coprod$-completion $\Downarrow A$ of A, spanned by the vectors $\overleftarrow{\alpha} = \Phi_* \Phi^* \overleftarrow{\alpha}$,
- the subproxet of the $\prod$-completion $\Uparrow B$ of B, spanned by the vectors $\overrightarrow{\beta} = \Phi^* \Phi_* \overrightarrow{\beta}$.

Since there are always uncountably many lower and upper vectors, and the completions $\Downarrow A$ and $\Uparrow B$ are infinite, $\wr \Phi \wr$ follows suit. But can we extract a small set of generators of $\wr \Phi \wr$, still supporting a decomposition of the matrix Φ.

Definition 5.1. *The* representable concepts *induced Φ are the elements of the completion $\wr \Phi \wr$ induced the representable vectors, i.e.*

- lower representable concepts $\nabla \Phi = \{\langle \Phi_* \Phi^* \nabla a, \Phi^* \nabla a \rangle \mid a \in A\}$
- upper representable concepts $\Delta \Phi = \{\langle \Phi_* \Delta b, \Phi^* \Phi_* \Delta b \rangle \mid b \in B\}$
- representable concepts $\Diamond \Phi = \nabla \Phi \cup \Delta \Phi$

Notation. The elements of $\Diamond \Phi$ are written in the form $\Diamond x = \langle \overleftarrow{\Diamond x}, \overrightarrow{\Diamond x} \rangle$, and thus

$$\overleftarrow{\Diamond a} = \Phi_* \Phi^* \nabla a \qquad \overrightarrow{\Diamond a} = \Phi^* \nabla a$$

$$\overleftarrow{\Diamond b} = \Phi_* \Delta b \qquad \overrightarrow{\Diamond b} = \Phi^* \Phi_* \Delta b$$

Theorem 5.2. *For any proxet matrix $\Phi : A \looparrowright B$, the restriction of the decomposition $A \stackrel{P^*}{\looparrowright} \wr\Phi\wr \stackrel{E^*}{\looparrowright} B$ from Thm. 4.10 along the inclusion $\Diamond\Phi \hookrightarrow \wr\Phi\wr$ to the representable concepts yields a decomposition $A \stackrel{P}{\looparrowright} \Diamond\Phi \stackrel{E}{\looparrowright} B$ which still satisfies Def. 4.6. More precisely, the matrices*

- $P : \overline{A} \times \Diamond\Phi \hookrightarrow \overline{A} \times \wr\Phi\wr \xrightarrow{P^*} [0,1]$
- $E : \overline{\Diamond\Phi} \times B \hookrightarrow \overline{\wr\Phi\wr} \times B \xrightarrow{E^*} [0,1]$

are such that $P : A \looparrowright \Diamond\Phi$ is a projection, $E : \Diamond\Phi \looparrowright B$ is an embedding, and $P\,;E = \Phi$.

5.2 Computing Proximities of Representable Concepts

To apply these constructions to the ratings matrix from Sec. 1, we first express the star ratings as numbers between 0 and 1.

	n	c	i	b
a	$\frac{4}{5}$	1	$\frac{2}{5}$	$\frac{4}{5}$
d	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	1
s	$\frac{2}{5}$	1	$\frac{3}{5}$	$\frac{2}{5}$
t	$\frac{1}{5}$	$\frac{3}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
l	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$

where we also abbreviated the user names to $U = \{A, D, S, T, L\}$ and the item names to $J = \{n, c, i, b\}$. Now we can compute the representable concepts $\Diamond\varphi \in \Diamond\Phi$ according to Def. 5.1, using (10):

$$(\overleftarrow{\Diamond j})_u = \left(\bigwedge_{\ell\in J} (\Delta j)_\ell \vdash (u \models \ell)\right) = (u \models j) \qquad (\overrightarrow{\Diamond j})_k = \left(\bigwedge_{x\in U} (\overrightarrow{\Diamond j})_x \vdash (x \models k)\right) = \left(\bigwedge_{x\in U} (x \models j) \vdash (x \models k)\right)$$

$$(\overrightarrow{\Diamond u})_j = \left(\bigwedge_{x\in U} (\nabla u)_x \vdash (x \models j)\right) = (u \models j) \qquad (\overleftarrow{\Diamond u})_v = \left(\bigwedge_{\ell\in J} (\overleftarrow{\Diamond u})_\ell \vdash (v \models \ell)\right) = \left(\bigwedge_{\ell\in J} (u \models \ell) \vdash (v \models \ell)\right)$$

Since $\overleftarrow{\Diamond\varphi} = \Phi_*\overrightarrow{\Diamond\varphi}$ and $\Phi^*\overleftarrow{\Diamond\varphi} = \overrightarrow{\Diamond\varphi}$, it suffices to compute one component of each pair $\Diamond\varphi = \langle\overleftarrow{\Diamond\varphi}, \overrightarrow{\Diamond\varphi}\rangle$, say the first one. So we get

$$\overleftarrow{\Diamond n} = \begin{pmatrix}\frac{4}{5} & \frac{2}{5} & \frac{2}{5} & \frac{1}{5} & 1\end{pmatrix} \qquad \overleftarrow{\Diamond c} = \begin{pmatrix}1 & \frac{2}{5} & 1 & \frac{3}{5} & \frac{1}{5}\end{pmatrix} \qquad \overleftarrow{\Diamond \imath} = \begin{pmatrix}\frac{2}{5} & \frac{4}{5} & \frac{3}{5} & \frac{3}{5} & \frac{1}{5}\end{pmatrix}$$

$$\overleftarrow{\Diamond b} = \begin{pmatrix}\frac{4}{5} & 1 & \frac{2}{5} & \frac{4}{5} & \frac{1}{5}\end{pmatrix} \qquad \overleftarrow{\Diamond a} = \begin{pmatrix}1 & \frac{2}{5} & \frac{1}{2} & \frac{1}{4} & \frac{1}{5}\end{pmatrix} \qquad \overleftarrow{\Diamond d} = \begin{pmatrix}\frac{1}{2} & 1 & \frac{2}{5} & \frac{1}{2} & \frac{1}{4}\end{pmatrix}$$

$$\overleftarrow{\Diamond s} = \begin{pmatrix}\frac{2}{3} & \frac{2}{5} & 1 & \frac{1}{2} & \frac{1}{5}\end{pmatrix} \qquad \overleftarrow{\Diamond t} = \begin{pmatrix}\frac{2}{3} & \frac{2}{3} & \frac{1}{2} & 1 & \frac{1}{3}\end{pmatrix} \qquad \overleftarrow{\Diamond l} = \begin{pmatrix}\frac{4}{5} & \frac{2}{5} & \frac{2}{5} & \frac{1}{5} & 1\end{pmatrix}$$

The proximities between all representable concepts can now be computed in the form

$$(x \vdash y)_{\Diamond\Phi} \;=\; (\Diamond x \vdash \Diamond y)_{\Diamond\Phi} = \bigwedge_{u\in U} \overleftarrow{\Diamond x}_u \vdash \overleftarrow{\Diamond y}_u$$

since the proximity in $\Diamond\Phi$ is just the proximity in $\nabla\Phi$, which is a subproxet ot $\Downarrow U$, so its proximity is by Def. 3.1 the pointwise minimum. Hence

$\vdash$	n	c	i	b	a	d	s	t	l
n	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	1
c	$\frac{1}{3}$	1	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{5}{12}$	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{1}{3}$
i	$\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{2}{3}$	$\frac{5}{12}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{5}{6}$	$\frac{1}{3}$
b	$\frac{1}{4}$	$\frac{2}{5}$	$\frac{1}{2}$	1	$\frac{5}{16}$	$\frac{5}{8}$	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{1}{4}$
a	$\frac{4}{5}$	1	$\frac{2}{5}$	$\frac{4}{5}$	1	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{4}{5}$
d	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{4}{5}$	1	$\frac{2}{5}$	1	$\frac{2}{5}$	$\frac{2}{3}$	$\frac{2}{5}$
s	$\frac{2}{5}$	1	$\frac{3}{5}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{2}{5}$	1	$\frac{1}{2}$	$\frac{2}{5}$
t	$\frac{1}{5}$	$\frac{3}{5}$	$\frac{3}{5}$	$\frac{4}{5}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{5}$
l	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	1

The bottom five rows of this table display the values of the representable concepts themselves

$$(u \vdash j)_{\Diamond\Phi} = (u \models j)_{\Phi} \tag{12}$$

$$(u \vdash v)_{\Diamond\Phi} = \bigwedge_{\ell\in J} (v \models \ell)_{\Phi} \vdash (u \models \ell)_{\Phi} \tag{13}$$

for $u, v \in U$ and $j \in J$, because $\left(\overleftarrow{\Diamond u} \vdash \overleftarrow{\Diamond x}\right)_{\Diamond\Phi} = \overleftarrow{\Diamond x}_u$ follows from the general fact that $\left(\nabla a \vdash \overleftarrow{\lambda}\right)_{\Downarrow A} = \overleftarrow{\lambda}_a$. The upper four rows display the values

$$(j \vdash k)_{\Diamond\Phi} = \bigwedge_{x\in U} (x \models j)_{\Phi} \vdash (x \models k)_{\Phi} \tag{14}$$

$$(j \vdash u)_{\Diamond\Phi} = \bigwedge_{x\in U} (x \models j)_{\Phi} \vdash (x \vdash u)_{\Diamond\Phi} = \bigwedge_{\ell\in J} (u \models \ell)_{\Phi} \vdash (j \vdash \ell)_{\Diamond\Phi} \tag{15}$$

for $u \in U$ and $j, k \in J$. Intuitively, these equations can be interpreted as follows:

- (13) *the proximity* $(u \vdash v)$ *measures how well* $(v \models \ell)$ *approximates* $(u \models \ell)$:
  - u's liking $(u \models \ell)$ of any movie ℓ is at least $(u \vdash v) \cdot (v \models \ell)$.
- (14) *the proximity* $(j \vdash k)$ *measures how well* $(x \models j)$ *approximates* $(x \models k)$
  - any user x's rating $(x \models k)$ is at least $(x \models j) \cdot (j \vdash k)$,
- (15) *the proximity* $(j \vdash u)$ *measures how well* j*'s style approximates* u*'s taste*
  - any x's proximity $(x \vdash u)$ to u is at least $(x \models j) \cdot (j \vdash u)$,
  - j's proximity $(j \vdash \ell)$ to any ℓ is at least $(j \vdash u) \cdot (u \models \ell)$.

Since $(a \vdash l) = \frac{4}{5}$, it would make sense for Abby to accept Luka's recommendations, but not the other way around, since $(l \vdash a) = \frac{1}{5}$. Although Temra's rating of "Ikiru" is just $(t \vdash i) = \frac{3}{5}$, "Ikiru" is a good test of her taste, since her rating of it is close to both Dusko's and Stefan's ratings.

Latent Concepts? While the proximities between each pair of users and items, i.e. between the induced *representable* concepts, provide an interesting new view on their relations, the task of determining the *latent* concepts remains ahead. What are the dominant tastes around which the users coalesce? What are the dominant styles that connect the items? What will such concepts look like? Formally, a dominant concept is a highly biased cut: in a high proximity of some of the representable concepts, and distant from the others. One way to find such cuts is to define the concepts of *cohesion* and *adhesion* of a cut along the lines of [30], and solve the corresponding optimization problems. Although there is no space to expand the idea in the present paper, some of the latent concepts can be recognized already by inspection of the above proximity table (recalling that each cut is both a supremum of users' and an infimum of items' representations).

6 Discussion and Future Work

What has been achieved? We generalized posets to proxets in Sec. 2 and 3, and lifted in Sec. 4 the FCA concept lattice construction to the corresponding construction over proxets, that allow capturing quantitative information. Both constructions share the same universal property, captured by the nucleus functor in Sec. 4.3. In both cases, the concepts are captured by *cuts*, echoing Dedekind's construction of the reals, and MacNeille's minimal completion of a poset. But while finite contexts yield finite concept lattices in FCA, in our analysis they yield infinitely many quantitative concepts. This is a consequence of introducing the infinite set of quantities [0,1]. The same phenomenon occurs in LSA [10], which allows the entire real line of quantities, and the finite sets of users and items span real vector spaces, that play the same role as our proxet completions. The good news is that the infinite vector space of latent concepts in LSA comes with a canonical basis of finitely many singular vectors, and that our proxet of latent concepts also has a finite generator, spelled out in Sec. 5. The bad news is that the generator described there is not a canonical basis of dominant latent concepts, with the suitable extremal properties, but an ad hoc basis determined by the given sets of users and items. Due to a lack of space, the final step of the analysis, finding the basis of dominant latent concepts, had to be left for a future paper. This task can be reduced to some familiar optimization problems.

More interestingly, and perhaps more effectively, this task can also addressed using qualitative FCA and its concept scaling methods [13]. The most effective form of concept analysis may thus very well be a combination of quantitative and qualitative analysis tools. Our analysis of the numeric matrix, extracted from the given star ratings, should be supplemented by standard FCA analyses of a family of relational contexts scaled by various thresholds. We conjecture that the resulting relational concepts will be the projections of the dominant latent concepts arising from quantitative analysis. If that is the case, then the relational concepts can be used to guide computation of quantitative concepts.

This view of the quantitative and the qualitative concept analyses as parts of a putative general FCA toolkit raises an interesting question of their relation with LSA and the spectral methods of concept analysis [10,1], which seem different. Some preliminary discussions on this question can be found in [31,32]. While FCA captures a *particle* view of network traffic, where the shortest path determines the proximity of two network nodes, LSA corresponds to the *wave* view of the traffic, where the proximity increases with the number of paths. Different application domains seem to justify different views, and call for a broad view of all concept mining methods as parts of the same general toolkit.

Acknowledgements. Anonymous reviewers' suggestions helped me to improve the paper, and to overcome some of my initial ignorance about the FCA literature. I am particularly grateful to Dmitry Ignatov, who steered the reviewing process with a remarkable patience and tact. I hope that my work will justify the enlightened support, that I encountered in these first contacts with the FCA community.

References

1. Azar, Y., Fiat, A., Karlin, A., McSherry, F., Saia, J.: Spectral analysis of data. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC 2001, pp. 619–626. ACM, New York (2001)
2. Banaschewski, B., Bruns, G.: Categorical characterization of the MacNeille completion. Archiv der Mathematik 18(4), 369–377 (1967)
3. Bělohlávek, R.: Fuzzy relational systems: foundations and principles, vol. 20. Plenum Publishers (2002)
4. Bělohlávek, R.: Concept lattices and order in fuzzy logic. Annals Pure Appl. Logic 128(1-3), 277–298 (2004)
5. Belohlavek, R.: What is a Fuzzy Concept Lattice? II. In: Kuznetsov, S.O., Ślęzak, D., Hepting, D.H., Mirkin, B.G. (eds.) RSFDGrC 2011. LNCS, vol. 6743, pp. 19–26. Springer, Heidelberg (2011)
6. Bonsangue, M.M., van Breugel, F., Rutten, J.J.M.M.: Generalized metric spaces: completion, topology, and power domains via the yoneda embedding. Theor. Comput. Sci. 193(1-2), 1–51 (1998)
7. Burusco, A., Fuentes-González, R.: Construction of the L-fuzzy concept lattice. Fuzzy Sets and systems 97(1), 109–114 (1998)
8. Burusco, A., Fuentes-González, R.: The study of the L-fuzzy concept lattice. Mathware & Soft Computing 1(3), 209–218 (2008)
9. Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications. John Wiley & Sons (2004)
10. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)
11. du Boucher-Ryan, P., Bridge, D.G.: Collaborative recommending using Formal Concept Analysis. Knowl.-Based Syst. 19(5), 309–315 (2006)
12. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) ICCS 2001. LNCS (LNAI), vol. 2120, pp. 129–142. Springer, Heidelberg (2001)
13. Ganter, B., Wille, R.: Conceptual scaling. Institute for Mathematics and Its Applications 17, 139 (1989)
14. Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis. LNCS (LNAI), vol. 3626. Springer, Heidelberg (2005)
15. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
16. Gehrke, M.: Generalized kripke frames. Studia Logica 84(2), 241–275 (2006)
17. Kaytoue, M., Kuznetsov, S.O., Macko, J., Meira Jr., W., Napoli, A.: Mining biclusters of similar values with triadic concept analysis. In: Proceedings of CLA 2011. CLA (2011)
18. Kaytoue, M., Kuznetsov, S.O., Napoli, A.: Pattern mining in numerical data: Extracting closed patterns and their generators. Research Report RR-7416, INRIA (October 2010)

19. Kaytoue, M., Kuznetsov, S.O., Napoli, A.: Revisiting numerical pattern mining with formal concept analysis. In: Proceedings of IJCAI 2011, pp. 1342–1347. AAAI (2011)
20. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci. 10(181), 1989–2001 (2011)
21. Kelly, G.M.: Basic Concepts of Enriched Category Theory. London Mathematical Society Lecture Note, vol. 64, pp. 1–136. Cambridge University Press (1982); Reprinted in Theory and Applications of Categories, vol. 10, pp.1–136 (2005)
22. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer 42(8), 30–37 (2009)
23. Krajči, S.: A generalized concept lattice. Logic Journal of IGPL 13(5), 543–550 (2005)
24. Künzi, H.P., Schellekens, M.P.: On the yoneda completion of a quasi-metric space. Theor. Comput. Sci. 278(1-2), 159–194 (2002)
25. William Lawvere, F.: Metric spaces, generalised logic, and closed categories. Rendiconti del Seminario Matematico e Fisico di Milano 43, 135–166 (1973)
26. Lehmann, F., Wille, R.: A Triadic Approach to Formal Concept Analysis. In: Ellis, G., Rich, W., Levinson, R., Sowa, J.F. (eds.) ICCS 1995. LNCS, vol. 954, pp. 32–43. Springer, Heidelberg (1995)
27. Leinster, T., Cobbold, C.: Measuring diversity: the importance of species similarity. Ecology (to appear, 2012)
28. MacNeille, H.M.: Extensions of partially ordered sets. Proc. Nat. Acad. Sci. 22(1), 45–50 (1936)
29. Mac Lane, S.: Categories for the Working Mathematician. Graduate Texts in Mathematics, vol. 5. Springer (1971); 2nd edn. (1997)
30. Pavlovic, D.: Network as a Computer: Ranking Paths to Find Flows. In: Hirsch, E.A., Razborov, A.A., Semenov, A., Slissenko, A. (eds.) CSR 2008. LNCS, vol. 5010, pp. 384–397. Springer, Heidelberg (2008)
31. Pavlovic, D.: On quantum statistics in data analysis. In: Bruza, P. (ed.) Quantum Interaction 2008. AAAI (2008), arxiv.org:0802.1296
32. Pavlovic, D.: Quantifying and Qualifying Trust: Spectral Decomposition of Trust Networks. In: Degano, P., Etalle, S., Guttman, J. (eds.) FAST 2010. LNCS, vol. 6561, pp. 1–17. Springer, Heidelberg (2011)
33. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: Formal Concept Analysis in Knowledge Discovery: A Survey. In: Croitoru, M., Ferré, S., Lukose, D. (eds.) ICCS 2010. LNCS, vol. 6208, pp. 139–153. Springer, Heidelberg (2010)
34. Wagner, K.R.: Liminf convergence in omega-categories. Theor. Comput. Sci. 184(1-2), 61–104 (1997)
35. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Dan Reidel, Dordrecht (1982)
36. Wilson, W.A.: On quasi-metric spaces. Amer. J. Math. 52(3), 675–684 (1931)
37. Kim, Y.W.: Pseudo quasi metric spaces. Proc. Japan Acad. 10, 1009–1012 (1968)

Some Notes on Managing Closure Operators

Sebastian Rudolph

Karlsruhe Institute of Technology, Germany
rudolph@kit.edu

Abstract. It is widely known that closure operators on finite sets can be represented by sets of implications (also known as inclusion dependencies) as well as by formal contexts. In this paper we survey known results and present new findings concerning time and space requirements of diverse tasks for managing closure operators, given in contextual, implicational, or black-box representation. These tasks include closure computation, size minimization, finer-coarser-comparison, modification by "adding" closed sets or implications, and conversion from one representation into another.

Keywords: Formal Concept Analysis, closure operators, complexity issues.

1 Introduction

Closure operators and closure systems are a basic notion in algebra and occur in various computer science scenarios such as logic programming or functional dependencies in databases. One central task when dealing with closure operators is to represent them in a succinct way while still allowing for their efficient computational usage. Formal concept analysis (FCA) naturally provides two complementary ways of representing closure operators: by means of *formal contexts* on one side and *implication sets* on the other. Although being complementary, these two representation modes share a lot of properties:

- Both allow for tractable closure computation.
- Both kinds of data structures do not uniquely represent the corresponding closure operator, but in either case, there is a well-known minimal "normal form" which is unique up to isomorphism: row-reduced formal contexts and canonical bases.
- In both cases, this normal form can be computed with polynomial effort.

For a given closure operator, the space needed to represent it in one or the other way may differ significantly: there are closure operators whose minimal implicational representation is exponentially larger than their minimal contextual one and vice versa (see Section 3).

Thus, it seems worthwhile to modify algorithms which store and manipulate closure operators (as many FCA algorithms do) such that they can switch between the two representations depending on which is more memory-efficient. To this end, algorithms performing basic operations on closure operators need to be available for both representations. Moreover, conversion methods from one representation to the other are needed

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 278–291, 2012.

and their computational complexity needs to be analyzed. Thereby, it is not only interesting to determine the required resources related to the size of the input, but also to the size of the output. This is to account for the fact, that (in order to be "fair") an algorithm creating a larger output should be allowed to take more time and use more memory.

Next to surveying well-known complexity results for tasks related to closure operators in different representations, this paper's noteworthy original contributions are the following:

- We clarify the complexities for comparing closure operators in different representations in terms of whether one is a refinement of the other. Interestingly, some of the investigated comparison tasks are tractable (i.e. time-polynomial), others are not.
- We show how to compute an implication set which realizes the closure operator of a given context and has polynomial size compared to the size of the context. This is achieved by virtue of auxiliary attributes. Note that this contrasts results according to which without such auxiliary attributes, a worst-case exponential blow-up is unavoidable.
- Exploiting this polynomial representation, we propose an alternative algorithm for computing the Duquenne-Guigues base of a given context.

The paper is organized as follows. After recalling some basics about closure operators and formal concept analysis in Section 2, we note that no representation is generally superior to the other in terms of the size needed to store it in Section 3. Finally, Section 4 provides algorithms and complexity results before we conclude in Section 5.

2 Preliminaries

We start providing a condensed overview over the notions used in this paper.

2.1 Closure Operators

Definition 1. *Let M be an arbitrary set. A function $\varphi : 2^M \to 2^M$ is called a* closure operator *on M if it is*

1. extensive*, i.e., $A \subseteq \varphi(A)$ for all $A \subseteq M$,*
2. monotone*, i.e., $A \subseteq B$ implies $\varphi(A) \subseteq \varphi(B)$ for all $A, B \subseteq M$, and*
3. idempotent*, i.e., $\varphi(\varphi(A)) = \varphi(A)$ for all $A \subseteq M$.*

A set $A \subseteq M$ is called closed *(or φ-*closed *in case of ambiguity), if $\varphi(A) = A$. The set of all closed sets $\{A \mid A = \varphi(A) \subseteq M\}$ is called* closure system.

It is easy to show that for an arbitrary closure system $\mathcal{S}$, the corresponding closure operator φ can be reconstructed by

$$\varphi(A) = \bigcap_{B \in \mathcal{S},\ A \subseteq B} B.$$

Hence, there is a one-to-one correspondence between a closure operator and the according closure system.

Definition 2. *Given two closure operators φ and ψ on M, φ is called* finer than *ψ (written $\varphi \leq \psi$, alternatively we also say ψ is* coarser than *φ) if every φ-closed set is also ψ-closed. We call φ and ψ* equivalent *(written $\varphi \equiv \psi$), if $\varphi(A) = \psi(A)$ for all $A \subseteq M$.*

It is well-known that the set of all closure operators together with the "finer than" relation constitutes a complete lattice.

2.2 Contexts

Following the normal line of argumentation of FCA [8], we use formal contexts as data structure to encode closure operators.

Definition 3. *A* formal context *$\mathbb{K}$ is a triple (G, M, I) with an arbitrary set G called* objects*, an arbitrary set M called* attributes*, and a relation $I \subseteq G \times M$ called* incidence relation*. The* size *of $\mathbb{K}$ (written: $\#\mathbb{K}$) is defined as $|G| \cdot |M|$, i.e. the number of bits to store I.*

This basic data structure can then be used to define operations on sets of objects or attributes, respectively.

Definition 4. *Let $\mathbb{K} = (G, M, I)$ be a formal context. We define a function $(\cdot)^I : 2^G \to 2^M$ with $\widetilde{G}^I := \{m \mid gIm \text{ for all } g \in \widetilde{G}\}$ for $\widetilde{G} \subseteq G$. Furthermore, we use the same notation to define the function $(\cdot)^I : 2^M \to 2^G$ where $\widetilde{M}^I := \{g \mid gIm \text{ for all } m \in \widetilde{M}\}$ for $\widetilde{M} \subseteq M$. For convenience, we sometimes write g^I instead of $\{g\}^I$ and m^I instead of $\{m\}^I$.*

Applied to an object set, this function yields all attributes common to these objects; by applying it to an attribute set we get the set of all objects having those attributes. The following facts are consequences of the above definitions:

- $(\cdot)^{II}$ is a closure operator on G as well as on M.
- For $A \subseteq G$, A^I is a $(\cdot)^{II}$-closed set and dually
- for $B \subseteq M$, B^I is a $(\cdot)^{II}$-closed set.

In the following, we will focus only on the closure operator on attribute sets and exploit the fact that this closure operator is independent from the concrete object set G; it suffices to know the set of the context's object intents. Thus, we will directly use intent sets, that is: families $\mathcal{F}$ of subsets of M to represent formal contexts.

Definition 5. *Given a family $\mathcal{F} \subseteq 2^M$, we let $\mathbb{K}(\mathcal{F})$ denote the formal context (G, M, I) with $G = \mathcal{F}$ and, for an $A \in \mathcal{F}$, we let AIm exactly if $m \in A$. Given $B \subseteq M$, we use the notation $B^{\mathcal{F}}$ to denote the attribute closure B^{II} in $\mathbb{K}(\mathcal{F})$ and let $\#\mathcal{F} = \#\mathbb{K}(\mathcal{F}) = |\mathcal{F}| \cdot |M|$.*

For the sake of simplicity we will from now on to refer to $\mathcal{F}$ as contexts (on M). We recall the first basic complexity result:

Proposition 1. *For any context $\mathcal{F}$ on a set M and any set $A \subseteq M$, the closure $A^{\mathcal{F}}$ can be computed in $O(\#\mathcal{F}) = O(|\mathcal{F}| \cdot |M|)$ time and $O(|M|)$ space.*

Given an arbitrary context $\mathcal{F}$ representing some closure operator φ on some set M, the question whether there exists another $\mathcal{F}'$ representing φ and satisfying $\#\mathcal{F}' < \#\mathcal{F}$ – and if so, how to compute it – is straightforwardly solved by noting that this coincides with the question if $\mathbb{K}(\mathcal{F})$ is row-reduced and how to row-reduce it. Hence we obtain:

Proposition 2. *Given a context $\mathcal{F}$ on M, a size-minimal context $\mathcal{F}'$ with $(\cdot)^{\mathcal{F}} \equiv (\cdot)^{\mathcal{F}'}$ can be computed in $O(|\mathcal{F}| \cdot \#\mathcal{F}) = O(|\mathcal{F}|^2 \cdot |M|)$ time and $O(|M|)$ space.*

Algorithm 1 displays the according method cast in our representation via set families.

We close this section by noting that for a given closure operator φ, the minimal $\mathcal{F}$ with $\varphi \equiv (\cdot)^{\mathcal{F}}$ is uniquely determined. We will denote it by $\mathcal{F}(\varphi)$.

2.3 Implications

Given a set of attributes, *implications* on that set are logical expressions that can be used to describe certain attribute correspondences which are valid for all objects in a formal context.

Definition 6. *Let M be an arbitrary set. An* implication *on M is a pair (A, B) with $A, B \subseteq M$. To support intuition we write $A \to B$ instead of (A, B). We say an implication $A \to B$* holds *for an attribute set C (also: C* respects *$A \to B$), if $A \nsubseteq C$ or $B \subseteq C$. Moreover, an implication $\mathfrak{i}$* holds *(or: is* valid*) in a formal context $\mathbb{K} = (G, M, I)$ if it holds for all sets g^I with $g \in G$. We then write $\mathbb{K} \models \mathfrak{i}$. The size of an implication set $\mathfrak{I}$ (written: $\#\mathfrak{I}$) is defined as $|\mathfrak{I}| \cdot |M|$. Given a set $A \subseteq M$ and a set $\mathfrak{I}$ of implications on M, we write $A^{\mathfrak{I}}$ for the smallest set that contains A and respects all implications from $\mathfrak{I}$. (Since those two requirements are preserved under intersection, the existence of a smallest such set is assured).*

It is obvious that for any set $\mathfrak{I}$ of implications on M, the operation $(\cdot)^{\mathfrak{I}}$ is a closure operator on M. Furthermore, it can be easily shown that an implication $A \to B$ is valid in a formal context $\mathbb{K} = (G, M, I)$ exactly if $B \subseteq A^{II}$.

The following result is an often noted and straightforward consequence from [16].

Proposition 3. *For any attribute set $B \subseteq M$ and set $\mathfrak{I}$ of implications, $B^{\mathfrak{I}}$ can be computed in $O(\#\mathfrak{I}) = O(|\mathfrak{I}| \cdot |M|)$ time and $O(|M|)$ space.*

Like in the case of the contextual encoding, also here it is natural to ask for a size-minimal set of implications that corresponds to a certain closure operator.

Although there is in general no unique minimal implication set for a given closure operator φ, the so-called Duquenne-Guigues base or stem base [10] is often used as a (minimal) canonical representation. We follow this practice and denote it by $\mathfrak{I}(\varphi)$.

Algorithm 2 (cf. [1,22,19]) provides a well-known way to turn an arbitrary implication set into an equivalent Duquenne-Guigues base. Thus we can note the following complexity result.

Proposition 4 (Day 1992). *Given a set $\mathfrak{I}$ of implications on M, a size-minimal $\mathfrak{I}'$ with $(\cdot)^{\mathfrak{I}} \equiv (\cdot)^{\mathfrak{I}'}$ can be computed in $O(|\mathfrak{I}| \cdot \#\mathfrak{I}) = O(|\mathfrak{I}|^2 \cdot |M|)$ time and $O(|\mathfrak{I}| \cdot |M|)$ space.*

Algorithm 1. `minimizeContext`

Input: context $\mathcal{F}$ on M
Output: size-minimal context $\mathcal{F}'$ such that $(\cdot)^{\mathcal{F}} \equiv (\cdot)^{\mathcal{F}'}$

$\mathcal{F} := \mathcal{F}'$
for each $A \in \mathcal{F}'$ **do**
  if $A = A^{\mathcal{F}' \setminus \{A\}}$ **then**
    $\mathcal{F}' := \mathcal{F}' \setminus \{A\}$
  end if
end for
output $\mathcal{F}'$

Algorithm 2. `minimizeImpSet`

Input: implication set $\mathfrak{I}$ on M
Output: size-minimal implication set $\mathfrak{I}'$ such that $(\cdot)^{\mathfrak{I}} \equiv (\cdot)^{\mathfrak{I}'}$

$\widetilde{\mathfrak{I}} := \emptyset$
for each $A \to B \in \mathfrak{I}$ **do**
  $\widetilde{\mathfrak{I}} := \widetilde{\mathfrak{I}} \cup \{A \to (A \cup B)^{\mathfrak{I}}\}$
end for
$\mathfrak{I}' := \emptyset$
for each $A \to B \in \widetilde{\mathfrak{I}}$ **do**
  delete $A \to B$ from $\widetilde{\mathfrak{I}}$
  $C := A^{\widetilde{\mathfrak{I}} \cup \mathfrak{I}'}$
  if $C \neq B$ **then**
    $\mathfrak{I}' := \mathfrak{I}' \cup \{C \to B\}$
  end if
end for
output $\mathfrak{I}'$

A closer look at the algorithm reveals that the $O(|\mathfrak{I}| \cdot |M|)$ space bound comes about by the necessity of a 2-pass processing of the implication set. Note that both passes can be performed *in situ* (i.e., by overwriting the input with the output) which would require only $O(|M|)$ additional memory.

3 Size Comparisons

Given these two encodings which are very alike with respect to the complexities of computing closures and minimization, a question which arises naturally is whether one encoding is superior to the other in terms of memory required to store it. First of all, note that for a given M, the size of both representations is bounded by $2^{|M|} \cdot |M|$, i.e. at most exponential in the size of M.

The following proposition shows that for some φ, $\#\mathcal{F}(\varphi)$ is exponentially larger than $\#\mathfrak{I}(\varphi)$.

Proposition 5. *There exists a sequence $(\varphi_n)_{n\in\mathbb{N}}$ of closure operators such that $\#\mathcal{F}(\varphi_n) \in \Theta(2^n)$ whereas $\#\mathfrak{I}(\varphi_n) \in \Theta(n^2)$.*

Proof. We define φ_n as the closure operator on the set $M_n = \{1, \ldots, 2n\}$ that corresponds to the implication set $\mathfrak{I}_b$ containing the implication $\{2i-1, 2i\} \to M_n$ for every $i \in \{1, \ldots, n\}$. Then, we obtain $\#\mathfrak{I}(\varphi_n) = 2n^2$. On the other hand, $\mathcal{F}(\varphi_n) = \{\{2k - a_k | 1 \leq k \leq n\} \mid \langle a_1, \ldots, a_n \rangle \in \{0,1\}^n\}$ (as schematically displayed in Fig. 1) whence we obtain $\#\mathcal{F}(\varphi_n) = 2^n \cdot 2n$. □

On the other hand, as a consequence of a result on the number of pseudo-intents [13,17], we know that the converse is true as well: for some φ, $\#\mathfrak{I}(\varphi)$ is exponentially larger than $\#\mathcal{F}(\varphi)$.

	1	2	...	$2n-3$	$2n-2$	$2n-1$	$2n$
g_1	×		...	×		×	
g_2	×		...	×			×
g_3	×		...		×	×	
g_4	×		...		×		×
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
g_{2^n-1}		×	...		×	×	
g_{2^n}		×	...		×		×

Fig. 1. Example for a context that is exponential in the size of its stem base

Proposition 6 (Kuznetsov 2004, Mannila & Räihä 1992). *There exists a sequence $(\varphi_n)_{n\in\mathbb{N}}$ of closure operators such that $\#\mathcal{F}(\varphi_n) \in \Theta(n^2)$ but $\#\Im(\varphi_n) \in \Theta(2^n)$.*

This result seems to imply that in general, one cannot avoid the exponential blowup if a contextually represented closure operator is to be represented by means of implications.

However, as the following definition and theorem show, this only holds if M is supposed to be fixed. If we allow for a bit more freedom in terms of the used attribute set, this blowup can be avoided.

Definition 7. *Given a context $\mathcal{F}$ on a set M, let M^+ denote the set M extended by a one new attribute m_F for each $F \in \mathcal{F}$. Then we define $\Im_{\mathcal{F}}$ as implication set on M^+ containing for every $m \in M$ the two implications $\{m\} \to \{m_F \mid F \in \mathcal{F}, m \notin F\}$ and $\{m_F \mid F \in \mathcal{F}, m \notin F\} \to \{m\}$.*

Theorem 7. *Given a context $\mathcal{F}$ on a set M, the following hold*

1. $\#\Im_{\mathcal{F}} = 2 \cdot |M| \cdot |M^+| = 2 \cdot |M| \cdot (|M| + |\mathcal{F}|) \leq (\#\mathcal{F})^2$.
2. $(\cdot)^{\mathcal{F}} \equiv (\cdot)^{\Im_{\mathcal{F}}}|_M$, *that is,* $A^{\mathcal{F}} = A^{\Im_{\mathcal{F}}} \cap M$ *for all* $A \subseteq M$.

Proof. The first claim is obvious.

For the second claim, we first show that for an arbitrary set $A \subseteq M$ holds $A^{\Im_{\mathcal{F}}} = B \cup C$ with $B := \{m_F \mid F \in \mathcal{F}, A \nsubseteq F\}$ and $C := \{m \mid \{m_F \mid F \in \mathcal{F}, m \notin F\} \subseteq B\}$. To show $A^{\Im_{\mathcal{F}}} \subseteq B \cup C$ we note that $A \subseteq B \cup C$ and that $B \cup C$ is $\Im_{\mathcal{F}}$-closed: $B \cup C$ satisfies all implications of the type $\{m_F \mid F \in \mathcal{F}, m \notin F\} \to \{m\}$ by definition of C. To check implications of the second type, $\{m\} \to \{m_F \mid F \in \mathcal{F}, m \notin F\}$, we note that

$$\begin{aligned} C &:= \{m \mid \{m_F \mid F \in \mathcal{F}, m \notin F\} \subseteq B\} \\ &= \{m \mid \{m_F \mid F \in \mathcal{F}, m \notin F\} \subseteq \{m_F \mid F \in \mathcal{F}, A \nsubseteq F\}\} \\ &= \{m \mid \forall F \in \mathcal{F} : m \notin F \to A \nsubseteq F\} \end{aligned}$$

Now, picking an $m \in C$, we find that every m_F for which $m \notin F$ must also satisfy $A \nsubseteq F$ and therefore $m_F \in B$ so we find all implications of the second type satisfied.

Further, we show $B \cup C \subseteq A^{\Im_{\mathcal{F}}}$, by proving $B \subseteq A^{\Im_{\mathcal{F}}}$ and $C \subseteq A^{\Im_{\mathcal{F}}}$ separately.

We obtain $B = \{m_F \mid F \in \mathcal{F}, A \nsubseteq F\} \subseteq A^{\Im_{\mathcal{F}}}$ due to the following: given an $F \in \mathcal{F}$ with $A \nsubseteq F$, we find an $m \in A$ with $m \notin F$ and thus an implication $m \to \{m_F, \ldots\}$ contained in $\Im_{\mathcal{F}}$, therefore $A^{\Im_{\mathcal{F}}}$ must contain m_F.

We then also obtain $C := \{m \mid \{m_F \mid F \in \mathcal{F}, m \notin F\} \subseteq B\} \subseteq A^{\mathfrak{I}_\mathcal{F}}$ by the following argument: picking an $m \in C$, we find the implication $\{m_F \mid F \in \mathcal{F}, m \notin F\} \to \{m\}$ contained in $\mathfrak{I}_\mathcal{F}$. On the other hand, we already know $B \subseteq A^{\mathfrak{I}_\mathcal{F}}$ and $B \supseteq \{m_F \mid F \in \mathcal{F}, m \notin F\}$, hence $m \in A^{\mathfrak{I}_\mathcal{F}}$.

Finally, we obtain $A^{\mathfrak{I}_\mathcal{F}}|_M = A^{\mathfrak{I}_\mathcal{F}} \cap M = C = \{m \mid \forall F \in \mathcal{F} : m \notin F \to A \nsubseteq F\} = \{m \mid \forall F \in \mathcal{F} : A \subseteq F \to m \in F\} = \bigcap_{F \in \mathcal{F}, A \subseteq F} F = A^{\mathcal{F}}$ for any $A \subseteq M$. □

Thus, we obtain a polynomially size-bounded implicational representation of a context. In our view this is a remarkable – although not too intricate – insight as it seems to challenge the practical relevance of computationally hard problems w.r.t. pseudo-intents (recognizing, enumerating, counting), on which theoretical FCA research has been focusing lately [14,19,15,21,20,2].

4 Algorithms for Managing Closure Operators

4.1 Finer or Coarser?

Depending on how closure operators are represented, there are several ways of checking if one is finer than the other. As the general case, we consider the situation where both closure operators are given in a "black-box" manner, i.e. as opaque functions that we can call and that come with runtime guarantees.

Theorem 8. *Let φ and ψ be closure operators on a set M for which computing of closures can be performed in time t_φ and t_ψ, respectively and space s_φ and s_ψ, respectively. Moreover, let $cl_\varphi = |\{\varphi(A) \mid A \subseteq M\}|$.*

Then, $\varphi \leq \psi$ can be decided in $O(cl_\varphi \cdot (|M| \cdot t_\varphi + t_\psi))$ and in $O(2^{|M|} \cdot (t_\varphi + t_\psi))$. The space complexity is bounded by $O(s_\varphi + s_\psi)$.

Proof. For the $O(cl_\varphi \cdot (|M| \cdot t_\varphi + t_\psi))$ time bound, we employ Ganter's NextClosure algorithm [6,7] for enumerating the closed sets of φ. We note that (i) it only makes use of the closure operator in a black-box manner (that is, it does not depend on a its specific representation) by calling it as a function and (ii) it has polynomial delay, more precisely the time between two closed sets being output is $O(|M| \cdot t_\varphi)$. For each delivered closed set, we have to additionally check if it is ψ-closed, hence the overall time needed per φ-closed set is $O(|M| \cdot t_\varphi + t_\psi)$.

The $O(2^{|M|} \cdot (t_\varphi + t_\psi))$ bound can be obtained by naïvely checking all subsets of M for φ-closedness and ψ-closedness.

In both cases, no intermediary information needs to be stored between testing successive sets which explains the space complexity. □

Note that all known black-box algorithms require exponentially many closure computations w.r.t. to $|M|$. This raises the question whether this bound can be improved if one or both of the to-be-compared closure operators are available in a specific representation. The subsequent theorem captures the cases where polynomially many calls suffice.

Theorem 9. *Let φ be a closure operator on a set M for which computing of closures can be performed in t_φ time and s_φ space. Then, the following hold:*

Algorithm 3. `finerThanContext`

Input: closure operator φ on set M, context $\mathcal{F}$
Output: YES if $\varphi \leq (\cdot)^{\mathcal{F}}$, NO otherwise

```
for each A ∈ F do
    if A ≠ φ(A) then
        output NO
        exit
    end if
end for
output YES
```

Algorithm 4. `coarserThanImpSet`

Input: closure operator φ on set M, implication set $\mathfrak{I}$
Output: YES if $(\cdot)^{\mathfrak{I}} \leq \varphi$, NO otherwise

```
for each A → B ∈ ℑ do
    if B ⊈ φ(A) then
        output NO
        exit
    end if
end for
output YES
```

- *for a context $\mathcal{F}$ on M, the problem $\varphi \leq (\cdot)^{\mathcal{F}}$ can be decided in $|\mathcal{F}| \cdot t_\varphi$ time and s_φ space and*
- *for an implication set $\mathfrak{I}$ on M, the problem $(\cdot)^{\mathfrak{I}} \leq \varphi$ can be decided in $|\mathfrak{I}| \cdot t_\varphi$ time and s_φ space.*

Proof. Algorithm 3 provides a solution for the first case. It verifies that every element (in other words: every object intent) of $\mathcal{F}$ is φ-closed, this suffices to guarantee that all $\mathcal{F}$-closed sets are φ-closed since every $\mathcal{F}$-closed set is an intersection of elements of $\mathcal{F}$ and φ-closed sets are closed under intersections (since this holds for every closure operator).

Algorithm 4 provides a solution for the second case. To ensure that every φ-closed set is also $(\cdot)^{\mathfrak{I}}$-closed, it suffices to show that every φ-closed set respects all implications from $\mathfrak{I}$. If every φ-closed set respects an implication $A \to B \in \mathfrak{I}$ can in turn be verified by checking if $B \subseteq \varphi(A)$. □

The results established in the above theorem give rise to precise polynomial complexity bounds for three of the four possible comparisons of closure operators which are contextually or implicationally represented.

Corollary 10. *Given contexts $\mathcal{F}, \mathcal{F}'$ and implication sets $\mathfrak{I}, \mathfrak{I}'$ on some set M, it is possible to check*

- $(\cdot)^{\mathcal{F}} \leq (\cdot)^{\mathcal{F}'}$ *in time* $O(|\mathcal{F}| \cdot |\mathcal{F}'| \cdot |M|)$,
- $(\cdot)^{\mathfrak{I}} \leq (\cdot)^{\mathfrak{I}'}$ *in time* $O(|\mathfrak{I}| \cdot |\mathfrak{I}'| \cdot |M|)$, *and*
- $(\cdot)^{\mathcal{F}} \leq (\cdot)^{\mathfrak{I}}$ *in time* $O(|\mathcal{F}| \cdot |\mathfrak{I}| \cdot |M|)$.

Surprisingly, the ensuing question – whether it is possible to establish a polynomial time complexity bound for the missing comparison case – has to be denied assuming $P \neq NP$, as the following theorem shows.[1]

[1] As indicated by a reviewer, this result in a slightly different formulation is already known in other communities, cf. [9].

Theorem 11. *Given a context $\mathcal{F}$ and an implication set $\mathfrak{I}$ on some set M, deciding if $(\cdot)^{\mathfrak{I}} \leq (\cdot)^{\mathcal{F}}$ is coNP-complete.*

Proof. To show coNP membership, we note that $(\cdot)^{\mathfrak{I}} \not\leq (\cdot)^{\mathcal{F}}$ if and only if there is a set A and which is $(\cdot)^{\mathfrak{I}}$-closed but not $(\cdot)^{\mathcal{F}}$-closed. Clearly, we can guess such a set and check the above properties in polynomial time.

We show coNP hardness, by a polynomial reduction of the problem to 3SAT [11]. Given a set $C = \{C_1, \ldots, C_k\}$ of 3-clauses (i.e. $|C_i| = 3$) over a set of literals $L = \{p_1, \neg p_1, \ldots p_\ell, \neg p_\ell\}$, we let $M = L$ and define

$$\mathfrak{I} := \{\{p_i, \neg p_i\} \to M \mid p_i \in L\}$$

as well as

$$\mathcal{F} := \{M \setminus (C_i \cup \{m\}) \mid C_i \in C, m \in M\}.$$

We now show that there is a set A with $A^{\mathfrak{I}} = A$ but $A^{\mathcal{F}} \neq A$ exactly if there is a valuation on $\{p_1, \ldots, p_\ell\}$ for which C is satisfied.

For the "if" direction assume $val : \{p_1, \ldots, p_\ell\} \to \{true, false\}$ to be that valuation and define $A := \{p_i \mid val(p_i) = true\} \cup \{\neg p_i \mid val(p_i) = false\}$. Obviously, A is $(\cdot)^{\mathfrak{I}}$-closed. On the other hand, since by definition A must contain one element from each $C_i \in C$, we have that $F \not\subseteq A$ for all $F \in \mathcal{F}$ and hence $A^{\mathcal{F}} = M \neq A$.

For the "only if" direction, assume $A^{\mathfrak{I}} = A$ but $A^{\mathcal{F}} \neq A$. By construction of $\mathcal{F}$, the latter can only be the case if A contains one element of each $C_i \in C$. Thus, the valuation $val : \{p_1, \ldots, p_\ell\} \to \{true, false\}$ with

$$val(p_i) = \begin{cases} true & \text{if } p_i \in A \\ false & \text{otherwise} \end{cases}$$

witnesses the satisfiability of C. □

Drawing from the above black-box case, a straightforward deterministic algorithm for testing $(\cdot)^{\mathfrak{I}} \leq (\cdot)^{\mathcal{F}}$ would need to subsequently generate all closed sets of $\mathfrak{I}$ (e.g. by Ganter's algorithm [6,7]) and check closedness w.r.t. $\mathcal{F}$. This algorithm would, however, require exponential time w.r.t. $|M|$ in the worst case.

4.2 Adding a Closed Set

We now consider the task of making a closure operator φ minimally "finer" by requiring that a given set A be a closed set.

Definition 8. *Given a closure operator φ on M and some $A \subseteq M$, the A-*refinement *of φ (written $\varphi{\downarrow}A$) is defined as the coarsest closure operator ψ with $\psi \leq \varphi$ and $\psi(A) = A$.*

It is straightforward to show that B is a $\varphi{\downarrow}A$-closed set exactly if it is φ-closed or the intersection of A and a φ-closed set. Clearly, if a closure operator is represented as formal context, refinements can be computed by simply adding a row, i.e. for any context $\mathcal{F}$ on M and set $A \subseteq M$ we have for $\mathcal{F}' := \mathcal{F} \cup \{A\}$ that $(\cdot)^{\mathcal{F}}{\downarrow}A \equiv (\cdot)^{\mathcal{F}'}$. Of course, $\mathcal{F}'$ will in general not be size-minimal even if $\mathcal{F}$ is.

Fact 12. *Given a context $\mathcal{F}$ on M and some $A \in M$, an $\mathcal{F}'$ with $(\cdot)^{\mathcal{F}'} \equiv (\cdot)^{\mathcal{F}}{\downarrow}A$ can be computed in $O(|M|)$ time and constant space. Moreover, we have $|\mathcal{F}'| \leq |\mathcal{F}| + 1$.*

If the closure operator is represented in terms of implications, a little more work is needed for this task.

Proposition 13. *Given an implication set $\mathfrak{I}$ on M and some $A \in M$, an $\mathfrak{I}'$ with $(\cdot)^{\mathfrak{I}'} \equiv (\cdot)^{\mathfrak{I}}{\downarrow}A$ can be computed in $O(|\mathfrak{I}| \cdot |M|^2)$ time. Moreover, we have $|\mathfrak{I}'| \leq |\mathfrak{I}| \cdot |M|$.*

Proof. Algorithm 5 ensures the claimed complexity behavior. We now show that it is correct by proving that a subset of M is $(\cdot)^{\mathfrak{I}'}$-closed iff it is $(\cdot)^{\mathfrak{I}}$-closed or the intersection of A and some $(\cdot)^{\mathfrak{I}}$-closed set.

For the "if" direction, note that all implications from $\mathfrak{I}'$ are entailed by $\mathfrak{I}$, therefore all $(\cdot)^{\mathfrak{I}}$-closed sets are $(\cdot)^{\mathfrak{I}'}$-closed. Further we note that A is obviously $(\cdot)^{\mathfrak{I}'}$-closed. As intersections preserve closedness the above implies that all intersections of A and an $\mathfrak{I}$-closed sets must be $(\cdot)^{\mathfrak{I}'}$-closed.

For the "only if" direction, let S be a $(\cdot)^{\mathfrak{I}'}$-closed set. If we assume $S \subseteq A$ we find that $S^{\mathfrak{I}'} = S^{\mathfrak{I}} \cap A$. If $S \nsubseteq A$, there exists an $m \in S \setminus A$. But then we find $S^{\mathfrak{I}'} = S^{\mathfrak{I}}$. □

4.3 Adding an Implication

The task dual to the one from the preceding section is to make a given closure operator coarser by requiring that all closed sets of the coarsened version respect a given implication. In other words, all closed sets not respecting the implication are removed.

Definition 9. *Given a closure operator φ on M and some implication $\mathfrak{i} = A \to B$ with $A, B \subseteq M$, the* $\mathfrak{i}$-coarsening *of φ (written $\varphi{\uparrow}\mathfrak{i}$) is defined as the finest closure operator ψ with $\varphi \leq \psi$ and $B \subseteq \psi(A)$.*

Clearly, if a closure operator is represented as implication set, coarsenings can be computed by simply adding the implication to the set. Note that $\mathfrak{I}' := \mathfrak{I} \cup \{\mathfrak{i}\}$ will in general not be size-minimal.

Fact 14. *Given an implication set $\mathfrak{I}$ on M and some implication $\mathfrak{i}$ on M, an $\mathfrak{I}'$ with $(\cdot)^{\mathfrak{I}'} \equiv (\cdot)^{\mathfrak{I}}{\uparrow}\mathfrak{i}$ can be computed in $O(|M|)$ time and constant space. Moreover, we have $|\mathcal{F}'| \leq |\mathcal{F}| + 1$.*

If the closure operator is represented by a context, a little more work is needed for this task.

Proposition 15. *Given a context $\mathcal{F}$ on M and some implication $\mathfrak{i}$ on M, an $\mathcal{F}'$ with $(\cdot)^{\mathcal{F}'} \equiv (\cdot)^{\mathcal{F}}{\uparrow}\mathfrak{i}$ can be computed in $O(|\mathcal{F}|^2 \cdot |M|)$ time. Moreover, we have $|\mathcal{F}'| \leq |\mathcal{F}|^2$.*

Proof. It is easy to check that Algorithm 6 satisfies the given complexity bounds. We show its correctness by verifying that a set is $(\cdot)^{\mathcal{F}'}$-closed exactly if it is $(\cdot)^{\mathcal{F}}$-closed and respects $A \to B$.

For the "if" direction, let S be an $(\cdot)^{\mathcal{F}}$-closed set that respects $A \to B$. This means that either $B \subseteq S$ or $A \nsubseteq S$. In the first case, note that every $F \in \mathcal{F}$ with $S \subseteq F$ respects

Algorithm 5. `addClosedSet`

Input: implication set $\mathfrak{I}$ on M, set $A \subseteq M$
Output: implication set $\mathfrak{I}'$ with $(\cdot)^{\mathfrak{I}'} \equiv (\cdot)^{\mathfrak{I}}\!\downarrow\! A$

$\mathfrak{I}' := \emptyset$
for each $B \to C \in \mathfrak{I}$ **do**
  if $B \to C$ is respected by A **then**
    $\mathfrak{I}' := \mathfrak{I}' \cup \{B \to C\}$
  else
    $\mathfrak{I}' := \mathfrak{I}' \cup \{B \to C \cap A\}$
    for each $m \in M \setminus A$ **do**
      $\mathfrak{I}' := \mathfrak{I}' \cup \{B \cup \{m\} \to C\}$
    end for
  end if
end for
output $\mathfrak{I}'$

Algorithm 6. `addImplication`

Input: context $\mathcal{F}$ on M,
implication $\mathfrak{i} = A \to B$ on M
Output: context $\mathcal{F}'$ with $(\cdot)^{\mathcal{F}'} \equiv (\cdot)^{\mathcal{F}}\!\uparrow\!\mathfrak{i}$

$\mathcal{F}' := \emptyset$
for each $C \in \mathcal{F}$ **do**
  if C respects $A \to B$ **then**
    $\mathcal{F}' := \mathcal{F}' \cup \{C\}$
  else
    for each $D \in \mathcal{F}$ with $A \nsubseteq D$ **do**
      $\mathcal{F}' := \mathcal{F}' \cup \{C \cap D\}$
    end for
  end if
end for
output $\mathcal{F}'$

$A \to B$ and thus each such F is contained in $\mathcal{F}'$ as well. Since S is the intersection of all these F it must itself be $(\cdot)^{\mathcal{F}'}$-closed. In the second case, there must be some $F \in \mathcal{F}$ with $S \subseteq F$ with $A \nsubseteq F$. Thus we obtain

$$\begin{aligned} S &= \textstyle\bigcap_{S\subseteq F'\in\mathcal{F}} F' \\ &= (\textstyle\bigcap_{S\subseteq F'\in\mathcal{F}, S \text{ respects } A\to B} F') \cap (\textstyle\bigcap_{S\subseteq F'\in\mathcal{F}, S \text{ violates } A\to B} F') \cap F \\ &= (\textstyle\bigcap_{S\subseteq F'\in\mathcal{F}, S \text{ respects } A\to B} F') \cap (\textstyle\bigcap_{S\subseteq F'\in\mathcal{F}, S \text{ violates } A\to B} F' \cap F) \end{aligned}$$

and see that S is an intersection of $(\cdot)^{\mathcal{F}'}$-closed sets and hence itself $(\cdot)^{\mathcal{F}'}$-closed.

For the "only if" direction, consider an arbitrary $(\cdot)^{\mathcal{F}'}$-closed set S. It can be easily checked that all $F \in \mathcal{F}'$ respect $A \to B$, hence also S does. Moreover, by definition, every $F \in \mathcal{F}'$ is an intersection of elements of $\mathcal{F}$ and thus $(\cdot)^{\mathcal{F}}$-closed. □

4.4 Conversion of Representations

We conclude this paper by considering the problem of extracting minimal implicational or contextual representations from black-box closure operators. While the problem of finding a minimal implication base has been considered extensively, the dual task has hardly been considered so far. In both cases, however, no output-polynomial algorithm could be established.

We start by considering the dual task: given a black-box closure operator φ, how can we compute $\mathcal{F}(\varphi)$? Algorithm 7 displays a semi-naïve approach which essentially computes the row-reduced version of the context containing all φ-closed sets, but size-minimizes the context on the way by progressing in reverse lectic order. This yields us with an algorithm requiring $O(2^{|M|} \cdot (t_\varphi + \#\mathcal{F}(\varphi))$ time but only $|M|$ space. If φ is represented by an implication set $\mathfrak{I}$, this amounts to a time complexity of $O(2^{|M|} \cdot |M| \cdot (|\mathfrak{I}| + |\mathcal{F}(\varphi)|)$

Algorithm 7. `extractContext`

Input: closure operator φ on set M
Output: context $\mathcal{F}(\varphi)$

$\mathcal{F} = \emptyset$
for each $F \subseteq M$, enumerated in inverse lectic order **do**
  if $F = \varphi F$ **then**
    if $F \neq F^{\mathcal{F}}$ **then**
      $\mathcal{F} = \mathcal{F} \cup \{F\}$
    end if
  end if
end for
output $\mathcal{F}$

Algorithm 8. `contextToImpSet`

Input: context $\mathcal{F}$ on set M
Output: implication set $\Im((\cdot)^{\mathcal{F}})$

compute $\Im_{\mathcal{F}}$
$\Im = \Im_{\mathcal{F}}$
for each $m_F \in M^+ \setminus M$ **do**
  $\Im' = \emptyset$
  for each $A \to B \in \Im$ **do**
    $\Im' := \Im' \cup \{A \to B \setminus \{m_F\}\}$
    if $m_F \in B \setminus A$ **then**
      for each $C \to D \in \Im$ with $m_F \in C$ **do**
        $\Im' := \Im' \cup \{A \cup C \setminus \{m_F\} \to D\}$
      end for
    end if
  end for
  $\Im :=$ `minimizeImpSet`$(\Im')$
end for
output $\Im$

Unfortunately, this still means that the algorithm is worst-case time-exponential w.r.t. $|M|$, even if $\#\mathcal{F}(\varphi)$ is "small" (i.e., polynomially bounded w.r.t. $|M|$). As a straightforward example, consider the closure operator φ_{id} with $\varphi_{\text{id}}(A) = A$ for all $A \subseteq M$, for which $\mathcal{F}(\varphi_{\text{id}}) = \{M \setminus \{m\} \mid m \in M\}$. In fact, the question whether a better, tractable behavior can be obtained at all has to be refuted: it follows rather directly from Thm 5.2 of [5], that no output-polynomial algorithm for this task can exist.[2]

Finally reviewing the more popular task of determining the stem base of a given formal context, the following can be shown by an inspection of Ganter's algorithm for enumerating all pseudo-closed sets of a closure operator [6,7].

Proposition 16 (essentially Ganter 1984). *Let φ be a closure operator on a set M for which computing of closures can be performed in time t_φ and space s_φ. Then $\Im(\varphi)$ can be computed in time $O(2^{|M|} \cdot (t_\varphi + \#\Im(\varphi))) = O(2^{|M|} \cdot (t_\varphi + |M| \cdot |\Im(\varphi)|))$ and space $O(s_\varphi)$.*

For the case of φ being explicitly represented by a context, this implies that converting a contextual representation into an implicational one can be done in time $O(2^{|M|} \cdot (\#\mathcal{F} + \#\Im((\cdot)^{\mathcal{F}}))) = O(2^{|M|} \cdot |M| \cdot (|\mathcal{F}| + |\Im((\cdot)^{\mathcal{F}})|))$.

The results from Section 3 give rise to a quite different approach of computing $\Im((\cdot)^{\mathcal{F}})$ from a given $\mathcal{F}$. Starting from the polynomial-size implicational representation $\Im^{\mathcal{F}}$ of a context $\mathcal{F}$, one can one-by-one remove the auxiliary attributes m_F by a resolution procedure, while minimizing the intermediate implicational representations via

[2] More precisely, the authors of [5] provide a representation of propositional Horn theories that admits for polynomial computation of the associated closure operator but does not allow for polynomial delay enumeration of "'characteristic models", that is intents of the corresponding reduced context.

`minimizeImpSet`. This method is formally specified in Algorithm 8. While the correctness of the algorithm is a rather immediate, establishing complexity results is the subject of ongoing work. Whether the algorithm turns out to be output-polynomial must, however, be doubted given that this would imply the existence of an output-polynomial algorithm for finding the transversal hypergraph of a given hypergraph (as first observed in [12] and put in FCA terms in [3]), which has been an open problem for over 20 years now (see [4] for a comprehensive overview). Moreover, it has been shown that no polynomial-delay algorithms for enumerating the stembase in lectic [2] or inverse lectic [18] order can exist unless $P = NP$.

5 Conclusion

We have investigated runtime and memory requirements for diverse tasks related to closure operators. The overview displayed in Table 1 reveals a certain duality between the two representations forms – context or implication set – and ascertains that none can be generally preferred to the other.

Table 1. Time complexities for different representations and tasks

	context $\mathcal{F}$	implication set $\mathfrak{I}$
closure	$O(\lvert\mathcal{F}\rvert \cdot \lvert M\rvert)$	$O(\lvert\mathfrak{I}\rvert \cdot \lvert M\rvert)$
turn to minimal $\mathcal{F}'$	$O(\lvert\mathcal{F}\rvert^2 \cdot \lvert M\rvert)$	$O(2^{\lvert M\rvert} \cdot \lvert M\rvert \cdot (\lvert\mathfrak{I}\rvert + \lvert\mathcal{F}((\cdot)^{\mathfrak{I}})\rvert))$
turn to minimal $\mathfrak{I}'$	$O(2^{\lvert M\rvert} \cdot \lvert M\rvert \cdot (\lvert\mathcal{F}\rvert + \lvert\mathfrak{I}((\cdot)^{\mathcal{F}})\rvert))$	$O(\lvert\mathfrak{I}\rvert^2 \cdot \lvert M\rvert)$
check if $\mathfrak{I}'$ finer	$O(\lvert\mathcal{F}\rvert + \lvert\mathfrak{I}'\rvert) \cdot \lvert M\rvert \cdot 2^{\lvert M\rvert}$	$O(\lvert\mathfrak{I}\rvert \cdot \lvert\mathfrak{I}'\rvert \cdot \lvert M\rvert)$
check if $\mathcal{F}'$ finer	$O(\lvert\mathcal{F}\rvert \cdot \lvert\mathcal{F}'\rvert \cdot \lvert M\rvert)$	$O(\lvert\mathcal{F}'\rvert \cdot \lvert\mathfrak{I}\rvert \cdot \lvert M\rvert)$
check if $\mathfrak{I}'$ coarser	$O(\lvert\mathcal{F}\rvert \cdot \lvert\mathfrak{I}'\rvert \cdot \lvert M\rvert)$	$O(\lvert\mathfrak{I}\rvert \cdot \lvert\mathfrak{I}'\rvert \cdot \lvert M\rvert)$
check if $\mathcal{F}'$ coarser	$O(\lvert\mathcal{F}\rvert \cdot \lvert\mathcal{F}'\rvert \cdot \lvert M\rvert)$	$O(\lvert\mathcal{F}'\rvert + \lvert\mathfrak{I}\rvert) \cdot \lvert M\rvert \cdot 2^{\lvert M\rvert}$
extract from φ	$O(\lvert M\rvert \cdot \lvert\mathcal{F}(\varphi)\rvert \cdot (t_\varphi + \lvert M\rvert \cdot \lvert\mathfrak{I}(\varphi)\rvert))$	$O(\lvert M\rvert \cdot \lvert\mathcal{F}(\varphi)\rvert \cdot (t_\varphi + \lvert M\rvert \cdot \lvert\mathfrak{I}(\varphi)\rvert))$
add implication	$O(\lvert\mathcal{F}\rvert^2 \cdot \lvert M\rvert)$	$O(M)$
add closed set	$O(M)$	$O(\lvert\mathfrak{I}\rvert \cdot \lvert M\rvert^2)$

There are many open questions left. On the theoretical side, central open problems are if there are algorithms transforming contextual into implicational representations and vice versa in output polynomial time. Note that a negative answer to this question would also disprove the existence of polynomial-delay algorithms.

On the practical side, coming back to our initial motivation, it should be experimentally investigated if variants of standard FCA algorithms can be improved by adding the option of working with alternative closure operator representations.

Moreover, the proposed alternative algorithm for computing the Duquenne-Guigues base should be evaluated against Ganter's algorithm on typical datasets from practical use cases, in order to assess its practical use.

Acknowledgements. The author is deeply indebted to the reviewers who gave extremely valuable hints to existing related work. This work has been supported by the project ExpresST funded by the German Research Foundation (DFG).

References

1. Day, A.: The lattice theory of functional dependencies and normal decompositions. International Journal of Algebra and Computation 2(4), 409–431 (1992)
2. Distel, F.: Hardness of Enumerating Pseudo-intents in the Lectic Order. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 124–137. Springer, Heidelberg (2010)
3. Distel, F., Sertkaya, B.: On the complexity of enumerating pseudo-intents. Discrete Applied Mathematics 159(6), 450–466 (2011)
4. Eiter, T., Gottlob, G.: Hypergraph Transversal Computation and Related Problems in Logic and AI. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 549–564. Springer, Heidelberg (2002)
5. Eiter, T., Ibaraki, T., Makino, K.: Computing intersections of Horn theories for reasoning with models. Tech. Rep. IFIG research report 9803, Universität Gießen (1998), http://bibd.uni-giessen.de/ghtm/1998/uni/r980014.html
6. Ganter, B.: Two basic algorithms in concept analysis. Tech. Rep. 831, FB4, TH Darmstadt (1984)
7. Ganter, B.: Two Basic Algorithms in Concept Analysis. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 312–340. Springer, Heidelberg (2010)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1997)
9. Gottlob, G., Libkin, L.: Investigations on armstrong relations, dependency inference, and excluded functional dependencies. Acta Cybernetica 9(4), 385–402 (1990)
10. Guigues, J.-L., Duquenne, V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. Math. Sci. Humaines 95, 5–18 (1986)
11. Karp, R.M.: Reducibility Among Combinatorial Problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
12. Kavvadias, D., Papadimitriou, C.H., Siedri, M.: On Horn Envelopes and Hypergraph Traversals (Extended abstract). In: Ng, K.W., Balasubramanian, N.V., Raghavan, P., Chin, F.Y.L. (eds.) ISAAC 1993. LNCS, vol. 762, pp. 399–405. Springer, Heidelberg (1993)
13. Kuznetsov, S.O.: On the intractability of computing the duquenne-guigues base. Journal of Universal Computer Science 10(8), 927–933 (2004)
14. Kuznetsov, S.O., Obiedkov, S.: Counting Pseudo-intents and #P-completeness. In: Missaoui, R., Schmidt, J. (eds.) ICFCA 2006. LNCS (LNAI), vol. 3874, pp. 306–308. Springer, Heidelberg (2006)
15. Kuznetsov, S.O., Obiedkov, S.A.: Some decision and counting problems of the duquenne-guigues basis of implications. Discrete Applied Mathematics 156(11), 1994–2003 (2008)
16. Maier, D.: The Theory of Relational Databases. Computer Science Press (1983)
17. Mannila, H., Räihä, K.J.: Design of Relational Databases. Addison-Wesley (1992)
18. Babin, M.A., Kuznetsov, K.O.: Recognizing pseudo-intents is conp-complete. In: Kryszkiewicz, M., Obiedkov, S. (eds.) Proc. 7th Int. Conf. on Concept Lattices and Their Applications (CLA 2010). CEUR Workshop Proceedings, vol. 672. CEUR-WS.org (2010)
19. Rudolph, S.: Some Notes on Pseudo-closed Sets. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007. LNCS (LNAI), vol. 4390, pp. 151–165. Springer, Heidelberg (2007)
20. Sertkaya, B.: Some Computational Problems Related to Pseudo-intents. In: Ferré, S., Rudolph, S. (eds.) ICFCA 2009. LNCS(LNAI), vol. 5548, pp. 130–145. Springer, Heidelberg (2009)
21. Sertkaya, B.: Towards the Complexity of Recognizing Pseudo-intents. In: Rudolph, S., Dau, F., Kuznetsov, S.O. (eds.) ICCS 2009. LNCS, vol. 5662, pp. 284–292. Springer, Heidelberg (2009)
22. Wild, M.: Implicational bases for finite closure systems. In: Lex, W. (ed.) Arbeitstagung Begriffsanalyse und Künstliche Intelligenz, pp. 147–169. Springer (1991)

Distributed Formal Concept Analysis Algorithms Based on an Iterative MapReduce Framework

Biao Xu, Ruairí de Fréin, Eric Robson, and Mícheál Ó Foghlú

Telecommunications Software & Systems Group,
Waterford Institute of Technology, Ireland
{bxu,rdefrein,erobson,mofoghlu}@tssg.org

Abstract. While many existing formal concept analysis algorithms are efficient, they are typically unsuitable for distributed implementation. Taking the MapReduce (MR) framework as our inspiration we introduce a distributed approach for performing formal concept mining. Our method has its novelty in that we use a light-weight MapReduce runtime called Twister which is better suited to iterative algorithms than recent distributed approaches. First, we describe the theoretical foundations underpinning our distributed formal concept analysis approach. Second, we provide a representative exemplar of how a classic centralized algorithm can be implemented in a distributed fashion using our methodology: we modify Ganter's classic algorithm by introducing a family of $MR^{\star}$ algorithms, namely MRGanter and MRGanter+ where the prefix denotes the algorithm's lineage. To evaluate the factors that impact distributed algorithm performance, we compare our MR^{*} algorithms with the state-of-the-art. Experiments conducted on real datasets demonstrate that MRGanter+ is efficient, scalable and an appealing algorithm for distributed problems.

Keywords: Formal Concept Analysis, Distributed Mining, MapReduce.

1 Introduction

Formal Concept Analysis (FCA), pioneered in the 80's by Wille [1], is a method for extracting formal concepts –natural clusters of objects and attributes– from binary object-attribute relational data. FCA has great appeal in the context of knowledge discovery [2], information retrieval [3] and social networking analysis applications [4] because arranging data as a concept lattice yields a powerful and intuitive representation of the dataset [1,5].

The main short-coming of FCA –which has curtailed a more widespread uptake of the approach– is that FCA becomes prohibitively time consuming as the dataset size increases. However, association rules mining tends to deal with large datasets. FCA relies on the fact that the set of concept intents is closed under intersection [6], namely, a closure system. Appealingly, using this property, new formal concepts may be extracted iteratively by extending an existing intent,

F. Domenach, D.I. Ignatov, and J. Poelmans (Eds.): ICFCA 2012, LNAI 7278, pp. 292–308, 2012.

in practice, by intersecting it with a new attribute and shrinking the extent in an iteration. While existing FCA algorithms perform this iterative procedure efficiently for small centralized datasets, the recent explosion in dataset sizes, privacy protection concerns, and the distributed nature of the systems that collect this data, suggests that efficient *distributed* FCA algorithms are required. In this paper we introduce a distributed FCA approach based on a light-weight MapReduce runtime called Twister [7], which is suited to iterative algorithms, scales well and reduces communication overhead.

1.1 Related Work

Some well-known algorithms for performing FCA include Ganter's algorithm [8], Lindig's algorithm [9] and CloseByOne [10,11] and their variants [12,13]. Ganter introduces *lectic* ordering so that all possible attribute subsets of the data do not have to be scanned when performing FCA. Ganter's algorithm computes concepts iteratively based on the previous concept without incurring exponential memory requirements. In contrast, CloseByOne produces many concepts in each iteration. Bordat's algorithm [14] runs in almost the same amount of time as Ganter's algorithm, however, it takes a local concept generation approach. Bordat's algorithm introduces a data structure to store previously found concepts, which results in considerable time savings. Berry proposes an efficient algorithm based on Bordat's approach which requires a data structure of exponential size in [15]. A comparison of theoretical and empirical complexity of many well-known FCA algorithms is given in [16]. In addition, some useful principles for evaluating algorithm performance for sparse and dense data are suggested by Kuznetsov and Obiedkov; We consider data density when evaluating our approach.

The main disadvantage of the batch algorithms discussed above is that they require that the entire lattice is reconstructed from scratch if the database changes. Incremental algorithms address this problem by updating the lattice structure when a new object is added to database. Incremental approaches have been made popular by Norris [17], Dowling [18], Godin et al. [19], Capineto and Romano [20], Valtchev et al. [21] and Yu et al. [22]. In recent years, to reduce concept enumeration time, some parallel and distributed algorithms have been proposed. Krajca et al., proposed a parallel version based on CloseByOne [13]. The first distributed algorithm [23] was developed by Krajca and Vychodil in 2009 using the MapReduce framework [24]. In order to encourage more widespread usage of FCA, beyond the traditional FCA audience, we propose the development and implementation of efficient, distributed FCA algorithms. Distributed FCA is appealing as distributed approaches that can take advantage of cloud infrastructures to reduce enumeration time, are attractive for practitioners.

1.2 Contributions

We utilize the MapReduce framework in this paper to execute distributed algorithms on different nodes. Several implementations of MapReduce have been developed by a number of companies and organizations, such as Hadoop MapReduce by Apache[1], and Twister Iterative MapReduce[2], since its introduction by Google in 2004. A crucial distinction between the present paper and the work of Krajca and Vychodil [23] is that we use a Twister implementation of MapReduce. Twister supports iterative algorithms [7]: we leverage this property to reduce the computation time of our distributed FCA algorithms. In contrast, Hadoop architecture is designed for performing single step MapReduce. We implement new distributed versions (MRGanter and MRGanter+) of Ganter's algorithm and empirically evaluate their performance. In order to provide an established and credible benchmark under equivalent experimental conditions, MRCbo, the distributed version of CloseByOne is implemented as well using Twister.

This paper is organized as follows. Section 2 reviews Formal Concept Analysis and Ganter's algorithm. The theoretical underpinnings for implementing FCA using distributed databases are described in Section 3 to support our approach. Our main contribution is a set of Twister-based distributed versions of Ganter's algorithm. Section 4 presents an implementation overview and comparison of MapReduce, Hadoop and Twister. Empirical evaluation of the algorithms proposed in this paper is performed using datasets from the UCI KDD machine learning repository; experimental results are discussed in Section 5. In summary, MRGanter+ performs favourably in comparison to centralized versions.

2 Formal Concept Analysis

We continue by introducing the notational conventions used in the sequel. Let O and P denote a finite set of objects and attributes respectively. The data ensemble, S, may be arranged in Boolean matrix form as follows: the objects and attributes are listed along the rows and columns of the matrix respectively; The symbol $\times$ is entered in a row-column position to denote an object has that attribute; An empty entry denotes that the object does not have that attribute. Formally, this matrix describes the binary relation between the sets O and P. The object X has attribute Y if $(X, Y) \in I$, $X \subseteq O$ and $Y \subseteq P$. The triple (O, P, I) is called a formal context. For example, in Table 1, $O = \{1, 2, 3, 4, 5, 6\}$ and $P = \{a, b, c, d, e, f, g\}$, thus object $\{2\}$ has attributes $\{a, c, e, g\}$. We define a derivation operator on X and Y where $X \subseteq O$ and $Y \subseteq P$ as:

$$X' = \{p \in P \mid \forall t \in O : (t, p) \in I\} \tag{1}$$

$$Y' = \{t \in O \mid \forall p \in P : (t, p) \in I\}. \tag{2}$$

The operation X' generates the set of attributes which are common to objects in X. Similarly, Y' generates the set of objects which are common to attributes in Y. A pair $\langle X, Y\rangle$ is called a formal concept of (O, P, I) if and only if $X \subseteq O$, $Y \subseteq P$, $X' = Y$, and $Y' = X$. Given a formal concept $\langle X, Y\rangle$, X and Y are its *extent* and *intent*. The crucial property here is that the mappings $X \mapsto X''$ and

[1] http://hadoop.apache.org/mapreduce/
[2] http://www.iterativemapreduce.org/

Table 1. The symbol × indicates that an object has the corresponding attribute

	a	b	c	d	e	f	g
1	×	×		×		×	
2	×		×		×		×
3		×	×	×		×	×
4		×		×	×		
5	×			×	×	×	
6		×	×			×	×

$Y \mapsto Y''$, commonly known as *closure operators*, hold. The closure operator can be used to calculate the extent and intent that form a formal concept.

Establishing some notion of concept ordering, that is engendering a sub/super-concept hierarchy, is crucial in what follows. Given X_1, $X_2 \subseteq O$ and Y_1, $Y_2 \subseteq P$ the concepts of a context are ordered as follows: $\langle X_1, Y_1 \rangle \leqslant \langle X_2, Y_2 \rangle :\Longleftrightarrow X_1 \subseteq X_2 \Longleftrightarrow Y_2 \subseteq Y_1$, an ordering which is interesting because it facilitates the iterative formation of a complete lattice which is called the concept lattice of the context [6]. In the following sections we describe algorithms for concept lattice formation, namely Ganter's algorithm (also known as NextClosure) and CloseByOne. We then introduce our distributed extensions of these approaches.

2.1 Ganter: Iterative Closure Mining Algorithm

The NextClosure algorithm describes a method for generating new closures which guarantees every closure is enumerated once. Closures are generated iteratively using a pre-defined order, namely lectic ordering. The set of all formal concepts is denoted by $\mathcal{F}$. Let us arrange the elements of $P = \{p_1, \cdots, p_i, \cdots, p_m\}$ in an arbitrary linear order $p_1 < p_2 < \cdots < p_i < \ldots < p_m$, where m is the cardinality of the attribute set, P. The decision to use lectic ordering dictates that any arbitrarily chosen subset of P is also ordered according to the *lectic* ordering which was defined *ab initio*. Given two subsets Y_1, $Y_2 \subseteq P$, Y_1 is lectically smaller than Y_2 if the smallest element in which Y_1 and Y_2 differ belongs to Y_2.

$$Y_1 \leq Y_2 :\Longleftrightarrow \exists_{p_i}(p_i \in Y_2, p_i \notin Y_1, \forall_{p_j < p_i}(p_j \in Y_1 \Longleftrightarrow p_j \in Y_2)). \quad (3)$$

NextClosure uses (Eqn. 3) as a feasibility condition for accepting new candidate formal concepts. Typically this difference in set membership is made more explicit by denoting the smallest element, p_i, in which the set Y_1 and Y_2 differ.

$$Y_1 \leq_{p_i} Y_2 :\Longleftrightarrow \exists_{p_i}(p_i \in Y_2, p_i \notin Y_1, \forall_{p_j < p_i}(p_j \in Y_1 \Longleftrightarrow p_j \in Y_2)). \quad (4)$$

To fix ideas, if the order of $P = \{a, b, c, d, e, f, g\}$ is defined as $a < b < c < d < e < f < g$, and two subsets of P, or *itemsets*, $Y_1 = \{a, c, e, g\}$ and $Y_2 = \{a, b, e, g\}$ are examined then $Y_1 \leq Y_2$ because the smallest element in which the two sets differ is b and this element belongs to Y_2.

Each subset $Y \subseteq P$ may yield a closure, $Y'' \subseteq P$; The NextClosure algorithm attempts to find all closures systematically by exploiting lectic ordering.

Table 2. Formal concepts mined from Table 1, including empty concepts

F_1: $\langle\{1,2,3,4,5,6\},\{\}\rangle$	F_8: $\langle\{1,3,4,6\},\{b\}\rangle$	F_{15}: $\langle\{1,2,5\},\{a\}\rangle$
F_2: $\langle\{1,3,5,6\},\{f\}\rangle$	F_9: $\langle\{1,3,6\},\{b,f\}\rangle$	F_{16}: $\langle\{2,5\},\{a,e\}\rangle$
F_3: $\langle\{2,4,5\},\{e\}\rangle$	F_{10}: $\langle\{1,3,4\},\{b,d\}\rangle$	F_{17}: $\langle\{1,5\},\{a,d,f\}\rangle$
F_4: $\langle\{1,3,4,5\},\{d\}\rangle$	F_{11}: $\langle\{1,3\},\{b,d,f\}\rangle$	F_{18}: $\langle\{5\},\{a,d,e,f\}\rangle$
F_5: $\langle\{1,3,5\},\{d,f\}\rangle$	F_{12}: $\langle\{4\},\{b,d,e\}\rangle$	F_{19}: $\langle\{2\},\{a,c,e,g\}\rangle$
F_6: $\langle\{4,5\},\{d,e\}\rangle$	F_{13}: $\langle\{3\ 6\},\{b,c,f,g\}\rangle$	F_{20}: $\langle\{1\},\{a,b,d,f\}\rangle$
F_7: $\langle\{2,3,6\},\{c,g\}\rangle$	F_{14}: $\langle\{3\},\{b,c,d,f,g\}\rangle$	F_{21}: $\langle\{\},\{a,b,c,d,e,f,g\}\rangle$

Let the ordering of P be $p_1 < p_2 < \ldots < p_i < \ldots < p_m$, and consider the subset $Y \subseteq P$. The generative operation is the $\oplus$-operation: a new intent is generated by applying $\oplus$ on an existing intent and an attribute, and is defined as

$$Y \oplus p_i := ((Y \cap \{p_1, \ldots, p_{i-1}\}) \cup \{p_i\})'', \quad \text{where } Y \subseteq P \text{ and } p_i \in P. \tag{5}$$

NextClosure then compares the new candidate formal concept with the previous concept. If the condition in (Eqn. 4) is satisfied the candidate concept produced by (Eqn. 5) is kept and added to the lattice.

The $\oplus$-operator in (Eqn. 5) consists of intersection, union and closure operations; Lectic ordering and the associated complexity of these operations explains why NextClosure's ordered approach incurs high computational expense. Consequently the largest dataset-size NextClosure can practically process is small.

Example 1. *Consider the formal context in Table 1. Assume we have a concept* $\langle\{1,5\},\{a,d,f\}\rangle$*. We take the attribute set,* $Y = \{a,d,f\}$*, and calculate,* $Y \oplus e$*. First, we compute,* $\{a,d,f\} \cap \{a,b,c,d\} = \{a,d\}$*, then we append* e *and generate* $\{a,d\} \cup \{e\} = \{a,d,e\}$*. Performing* $\{a,d,f\} \oplus e = \{a,d,e\}''$ *yields the set,* $\{a,d,e,f\}$*. To demonstrate the role of lectic ordering, we compute* $Y \oplus c = \{a,c,e\}$*. According to the feasibility condition in (Eqn. 4),* $\{a,d,e,f\} \leq_c \{a,c,e\}$*. Thus, the set,* $\{a,c,e\}$*, is added to the concept lattice,* $\mathcal{F}$*. By repeating this process, NextClosure determines that there are* 21 *formal concepts in the concept lattice representation of the formal context in Table 1. The set of concepts,* $\mathcal{F}$*, is listed in Table 2.*

Pseudo code for NextClosure is described in the Algorithm 1 and 2 as background to our distributed approach. Algorithm 1 applies the closure operator on the null attribute set and generates the first intent, Y, which is the base for all subsequent formal concepts. New concepts are generated in turn by calling Algorithm 2 and concatenating the resultant concepts to the set of formal concepts, $\mathcal{F}$. As each candidate intent is extended with new attributes, the intent for the last iteration of this loop consists of the complete set of attributes. This feature is used to terminate the loop (in Line 2 of the Algorithm 1). Algorithm 2 accepts the formal context triple, (O, P, I) and current intent, Y, as inputs. By convention, the attribute set P is sorted in descending order. The $\oplus$-operator described in (Eqn. 5) is applied to produce candidate formal concepts. The concept feasibility condition (Eqn. 4) is used to verify whether a

Algorithm 1. AllClosure

Input: $\emptyset$: null attribute set.
Output: $\mathcal{F}$: Formal concepts set.

$Y \leftarrow \emptyset''$;
while Y is not the last closure **do**
  $Y \leftarrow$ NextClosure();
  $\mathcal{F} \leftarrow \mathcal{F} \cup Y$;
end while
return $\mathcal{F}$

Algorithm 2. NextClosure

Input: O, P, I, Y: formal context & current intent.
Output: Y.

for p_i from p_m down to p_1 **do**
  if $p_i \notin Y$ **then**
    candidate $\leftarrow Y \oplus p_i$;
    if candidate $\leq_{p_i} Y$ **then**
      $Y \leftarrow$ candidate;
      break;
    end if
  end if
end for
return Y

new candidate should be added to the set of formal concepts, $\mathcal{F}$. The approach taken in the CloseByOne algorithm is similar in spirit to the approach taken by the NextClosure algorithm: CloseByOne generates new formal concepts based on concept(s) generated in the previous iteration and tests their feasibility using the operator, $\leq_{p_i}$. The crucial difference is that the CloseByOne algorithm generates many concepts in each iteration. CloseByOne terminates when there are no more concepts that satisfy (Eqn. 4). In short, NextClosure only finds the first feasible formal concept in each iteration whereas CloseByOne potentially generates many. As a consequence, CloseByOne requires far fewer iterations.

The appeal of NextClosure, and explanation for our desire to make it more efficient lies in its thoroughness; the guarantee of a complete lattice structure which is a consequence of the main theorem of Formal Concept Analysis [6]. This thoroughness is due to lectic ordering and the iterative approach deployed by NextClosure; however, thoroughness comes at the cost of high complexity. The advent of efficient mechanisms for dealing with iterative algorithms using MapReduce captured by Twister allow us to couple NextClosure's thoroughness with a practical distributed implementation in this paper.

3 Distributed Algorithms for Formal Concept Mining

We continue by describing two methods for performing distributed NextClosure, namely, MRGanter and MRGanter+. An introduction to Twister is deferred to Section 4. We start by describing the properties of a partitioned dataset compared to its unpartitioned form. In many cases these properties are simply restatements of the properties of the derivations operators.

Given a dataset S, we partition its objects into n subsets and distribute the subsets over n different nodes. Without loss of generality, it is convenient to limit $n = 2$ here. We denote the partitions by S_1 and S_2. Alternatively we can think in terms of formal contexts and write the formal context, (O, P, I), in terms of the partitioned formal contexts (O_{S_1}, P, I_{S_1}) and (O_{S_2}, P, I_{S_2}). To fix ideas, we use the dataset in Table 1 as an exemplar and generate the partitions in Table 3.

Table 3. Partitioned datasets derived from Table 1, S_1 and S_2

S_1 or (O_{S_1}, P, I_{S_1})								S_2 or (O_{S_2}, P, I_{S_2})							
	a	b	c	d	e	f	g		a	b	c	d	e	f	g
1	×	×		×		×		4		×		×	×		
2	×		×		×		×	5	×			×	×	×	
3		×	×	×		×	×	6		×	×			×	×

The partitions are non-overlapping: the intersection of the partitions is the null set, $S_1 \cap S_2 = \emptyset$ and their union gives the full dataset $S = S_1 \cup S_2$. It follows that the partitions, S_1, S_2, have the same attributes sets, P, as the entire dataset S, however, the set of objects is different in each partition, e.g. O_{S_1} and O_{S_2}. Let Y_S, Y_{S_1} and Y_{S_2} denote an arbitrary attribute set Y with respect to the entire dataset S, and each of its partitions S_1 and S_2 respectively. By construction they are equivalent: $Y_S \equiv Y_{S_1} \equiv Y_{S_2}$. Similarly, Y'_S, Y'_{S_1} and Y'_{S_2} are the sets of objects derived by the derivation operation in each of the partitions S_1, S_2 and the entire dataset S respectively.

Property 1. *Given the formal context, (O, P, I), the two partitions (O_{S_1}, P, I_{S_1}) and (O_{S_2}, P, I_{S_2}) and an arbitrary itemset, $Y \subseteq P$, the property $Y'_S = Y'_{S_1} \cup Y'_{S_2}$ holds: the union of the sets of objects generated by the derivation of the attribute set Y in each of the partitions is equivalent to the set of objects generated by the derivation of the attribute set over the entire dataset, S.*

Appealing to the definition of the derivation operator proposed by Wille in [1], the set, Y'_S, is a subset of O, $Y'_S \subseteq O$. Moreover, $Y'_{S_1} \subseteq O_{S_1}$ and $Y'_{S_2} \subseteq O_{S_2}$. Given $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$, it follows that, $O_{S_1} \cup O_{S_2} = O$ and $O_{S_1} \cap O_{S_2} = \emptyset$; Therefore, $Y'_{S_1} \subseteq Y'_S$ and $Y'_{S_2} \subseteq Y'_S$. Finally, $Y'_{S_1} \cup Y'_{S_2} \equiv Y'_S$. As a counterexample, an object t that exists in Y'_S, but not in Y'_{S_1} or Y'_{S_2}, cannot exist because $O_{S_1} \cup O_{S_2} = O$ and $O_{S_1} \cap O_{S_2} = \emptyset$ and $Y_S = Y_{S_1} = Y_{S_2}$. If t is in Y'_S it must appear in Y'_{S_1} or Y'_{S_2}. In short, Property 1 allows us to process all objects independently: the objects can be distributed and processed in an arbitrary order and this will not affect the result of Y'. Property 1 is trivially extended to the case of n partitions. Now we describe how formal concepts can be combined from different partitions.

Property 2. *Given the formal context, (O, P, I), the two partitions (O_{S_1}, P, I_{S_1}) and (O_{S_2}, P, I_{S_2}) and an arbitrary itemset, $Y \subseteq P$, the property $Y''_S = Y''_{S_1} \cap Y''_{S_2}$ holds: The intersection of the closures of the attribute set, Y, with respect to each of the partitions S_1 and S_2 is equivalent to the closure of the attribute set, Y, with respect to the entire dataset S.*

By the definition of the partition construction method above, $S_1 \cup S_2 = S$, which implies that, $S_1 \subset S$ and $S_2 \subset S$. Recall that, $Y'_{S_1} \subset Y'_S$ and $Y'_{S_2} \subset Y'_S$, and from Property 1 we have that $Y'_S = Y'_{S_1} \cup Y'_{S_2}$. Appealing to the properties of the derivation operators, in [1], we have, $Y''_{S_1} \supseteq Y''_S$ and $Y''_{S_2} \supseteq Y''_S$. It is clear that Y''_{S_1} and Y''_{S_2} need not equal Y''_S, but by the definition of a closure

$(Y'_{S_1} \cup Y'_{S_2})' = (Y'_S)' = Y_S$, thus, $(Y'_{S_1} \cup Y'_{S_2})' = Y''_{S_1} \cap Y''_{S_2}$ follows trivially from the definition of the derivations operators.

Example 2. *Consider the following example. Taking itemset* $Y = \{b, d\}$*. We derive* $Y''_{S_1} = \{b, d, f\}$ *from the first partition* S_1*, and* $Y''_{S_2} = \{b, d, e\}$ *from* S_2*. We derive* $Y''_S = \{b, d\}$ *for the entire dataset* S*. Therefore* $Y''_S = Y''_{S_1} \cap Y''_{S_2}$*.*

Theorem 1. *Given a set of attributes* Y, $Y \subset P$*. Let* $\mathcal{F}^Y_{S_1}$ *and* $\mathcal{F}^Y_{S_2}$ *be the sets of closures based on* Y *in relation to* S_1 *and* S_2 *respectively. Then the closure set of* Y *in relation to* S *can be calculated from:* $\mathcal{F}^Y_S = \mathcal{F}^Y_{S_1} \cap \mathcal{F}^Y_{S_2}$

This is simply a consequence of Property 2 as, $\mathcal{F}^Y_S = Y''_S = Y''_{S_1} \cap Y''_{S_2} = \mathcal{F}^Y_{S_1} \cap \mathcal{F}^Y_{S_2}$ and $Y_S \equiv Y_{S_1} \equiv Y_{S_2}$ by definition of the partition.

Example 3. *Consider again Example 2. Appealing to Theorem 1, the formal concept with respect to the entire data set is the intersection of the formal concepts from each partition* $F^Y_S = F^Y_{S_1} \cap F^Y_{S_2} = \{b,d,f\} \cap \{b,d,e\} = \{b,d\}$*.*

We denote the k-th partition as S_k and then propose:

Theorem 2. *Given the closures* $\mathcal{F}^Y_{S_1}, \ldots, \mathcal{F}^Y_{S_n}$ *from* n *disjoint partitions,* $\mathcal{F}^Y_S = \mathcal{F}^Y_{S_1} \cap \ldots \cap \mathcal{F}^Y_{S_n}$*.*

A trivial inductive argument establishes that Theorem 2 holds. Theorem 1 proves the $n = 2$ case. Theorem 2 follows by recognizing that the dataset S at the $(k-1)$-th step of the proof can be thought as of consisting of two partitions only, the partition $S_1 \cup \cdots \cup S_{k-1}$ and a second partition S_k.

Calling on nothing more complex than: 1) the properties of the derivation operators, and 2) construction of non-overlapping partitions, we leverage Theorem 2 in order to apply the MapReduce, specifically the Twister variant, to calculate closures from arbitrary number of distributed nodes sure in the knowledge that the thoroughness of NextClosure is preserved.

3.1 MRGanter

To address the dataset size limitations imposed on NextClosure –owing to the complexity of the ⊕-operation– we deploy FCA across multiple nodes to reduce the execution time. We demonstrate how decompose NextClosure so that each sub-task is executed in parallel. In Algorithm 2, there were two stages involved in computing NextClosure: 1) computing a new candidate closure, and 2) making a judgment on whether to add it to the evaluated formal concepts. In MapReduce parlance, computing a new candidate closure corresponds to the map stage, and validating its feasibility corresponds to the reduce phase. In this paper, we only calculate the intent of a formal concept. The variables and constants used by distributed algorithms are summarized in Table 4. The main operation in the merging function is the intersection operator, which is applied on the set of local closures L_k generated at each node. Algorithm 3 gives the pseudo code for the merging function based on Theorem 2. To describe the merging operation, we

Table 4. Variables and constants used in distributed FCA

Variables/Constants	Description
p_i	an attribute in P, where $i = 1, \cdots, m$
L_k	complete set of local closures in data partition k, $k = 1, \cdots, n$.
l_i	an intent in L_k which is derived from p_i
d	the intent produced in the previous iteration
f	the newly generated intent
G	a container for storing newly generated intents

Algorithm 3. Merging function

```
Input: p_i, L_k, f.
Output: f.
 1: l_i ← the local closure in L_k in terms of p_i;
 2: f ← Ψ(l_i, f);
 3: return f
```

Algorithm 4. Map: MRGanter

```
Input: d.
Output: (d, L_k).
 1: for p_i from p_m down to p_1 do
 2:     if p_i is not in d then
 3:         l_i ← d ⊕ p_i;
 4:         associate l_i with p_i;
 5:         L_k ← L_k ∪ l_i;
 6:     end if
 7:     return (d, L_k);
 8: end for
```

Algorithm 5. Reduce: MRGanter

```
Input: (d,L_k).
Output: f.
 1: for p_i in P do
 2:     f ← initialize new intent;
 3:     for i from 1 up to m do
 4:         f ← merging(p_i, L_k, f);
 5:     end for
 6:     if f ≤_{p_i} d then
 7:         break;
 8:     else
 9:         continue;
10:     end if
11: end for
12: return f
```

Algorithm 6. Reduce: MRGanter+

```
Input: (d, L_k).
Output: G.
 1: H ← initialize a two-level hash table;
 2: for p_i in P do
 3:     f ← initialize new intent;
 4:     for i from 1 up to m do
 5:         f ← merging(p_i, L_k, f);
 6:     end for
 7:     if f is not in H then
 8:         add f into H;
 9:         add f into G;
10:     end if
11: end for
12: return G
```

introduce the notation, Ψ(l_i, f) =l_i ∩ f, which acts on two intents. The merging function is deployed at the reduce phase and only processes local closures derived from the same attribute (Line 1).

The Map phase described in the Algorithm 4 produces all local closures. The output consists of the previous intent d and a set of local intents L_k. In order to be used in the merging function the attribute which was used to form local closures should be recorded and passed (Line 4). All pairs with the same key, d, are sent to the same reducer. All local intents are used to form global intents and then filtered by the closure validation condition (Line 6 in Algorithm 5). Algorithm 5 accepts (d,L_k) from the k-th *mapper*s (see Section 4), where $k = 1, \cdots, n$. Only pairs with the same key, d, are accepted by a Reducer. Line 4

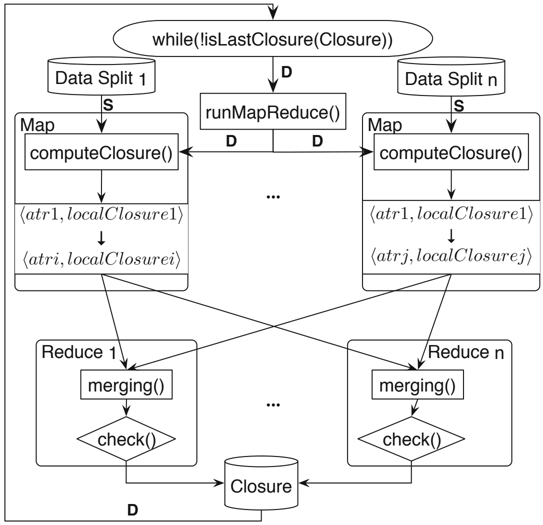


Fig. 1. MRGanter work flow: static data is loaded at the start of the procedure (labeled by S) and the dynamic data (Closures produced during each iteration) is passed and used in the next iteration (labeled by D)

generates an candidate closure f. This candidate is then validated. Successful candidates are outputted as a global closure f.

Fig. 1 depicts the iterative flow of control of MRGanter; the lines marked with "S" import static data from each partition, while the lines marked with "D" configure each map with the previous closure. Each new closure is tested to see if it is the last, e.g. it contains all attributes, P. If this condition is not met MRGanter continues. We present a worked example using the dataset in Table 3. Table 5, does not illustrate all results due to space limitations. MRGanter performs 20 iterations to determine all concepts.

3.2 MRGanter+

NextClosure calculates closures in lectic ordering to ensure every concept appears only once. This approach allows a single concept to be tested with the closure validation condition during each iteration. This is efficient when the algorithm runs on a single machine. For multi-machine computation, the extra computation and redundancy resulting from keeping only one concept after each iteration across many machines is costly. We modify NextClosure to reduce the number of iterations and name the corresponding distributed algorithm MRGanter+.

Rather than using redundancy checking, we keep as many closures as possible in each iteration; All closures are maintained and used to generate the next batch of closures. To this end, we modify Algorithm 5: the Map algorithm remains the same as in Algorithm 4. Algorithm 6 describes the ReduceTask for

Table 5. MRGanter: Only single a intent (bold) produced per iteration.

d	p_i	l_i from S_1	l_i from S_2	f
∅	g	{c,g}	{b,c,f,g}	{c,g}
	f	{b,d,f}	{f}	**{f}**
	e	{a,c,e,g}	{d,e}	{e}
	d	{b,d,f}	{d,e}	{d}
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}
{f}	g	{b,c,d,f,g}	{b,c,f,g}	{b,c,f,g}
	e	{a,c,e,g}	{d,e}	**{e}**
	d	{b,d,f}	{d,e}	{d}
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}
{e}	g	{a,c,e,g}	{a,. . . ,g}	{a,c,e,g}
	f	{a,. . . ,g}	{a,d,e,f}	{a,d,e,f}
	d	{b,d,f}	{d,e}	**{d}**
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}
{d}	g	{b,c,d,f,g}	{a,. . . ,g}	{b,c,d,f,g}
	f	{b,d,f}	{a,d,e,f}	**{d,f}**
	e	{a,. . . ,g}	{d,e}	{d,e}
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}

Table 6. MRGanter+: Many intents (bold) produced per iteration

d	p_i	l_i from S_1	l_i from S_2	f
∅	g	{c,g}	{b,c,f,g}	**{c,g}**
	f	{b,d,f}	{f}	**{f}**
	e	{a,c,e,g}	{d,e}	**{e}**
	d	{b,d,f}	{d,e	**{d}**
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	**{b}**
	a	{a}	{a,d,e,f}	**{a}**
{cg}	f	{b,c,d,f,g}	{b,c,f,g}	**{b,c,f,g}**
	e	{a,c,e,g}	{a,. . . ,g}	**{a,c,e,g}**
	d	{b,c,d,f,g}	{a,. . . ,g}	**{b,c,d,f,g}**
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}
{f}	g	{b,c,d,f,g}	{b,c,f,g}	{b,c,f,g}
	e	{a,c,e,g}	{d,e}	{e}
	d	{b,d,f}	{d,e}	{d}
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}
{e}	g	{a,c,e,g}	{a,. . . ,g}	{a,c,e,g}
	f	{a,. . . ,g}	{a,d,e,f}	**{a,d,e,f}**
	d	{b,d,f}	{d,e}	{d}
	c	{c,g}	{b,c,f,g}	{c,g}
	b	{b,d,f}	{b}	{b}
	a	{a}	{a,d,e,f}	{a}

MRGanter+. The Reduce in MRGanter+ merges local closures first in Line 5, and then recursively examines if they already exist in the set of global formal concepts H (Line 7). The set H is used to fast index and search a specified closure; it is designed as a two-level hash table to reduce its costs. The first level is indexed by the head attribute of the closure, while the second level is indexed by the length of the closure. New closures are stored in G. We present a running example based on the dataset in Table 3 for comparison. MRGanter+ produces many intents in each iteration. New intents are kept if they are not already in H. Notably, MRGanter+ requires 3 iterations to mine all concepts. Moreover, we implement CloseByOne proposed by Krajca and Vychodil in [23] based on the MapReduce framework and call it, MRCbo. Comparing MRGanter+ with MRCbo, we demonstrate that MRGanter+ typically generates more concepts in each iteration and uses fewer iterations. Detailed analysis is given in Section 5.2.

4 Twister MapReduce

The MapReduce framework adopts a divide-conquer strategy to deal with huge datasets and is applicable to many classes of problems [25]. A large number of computers, collectively referred to as a cluster, are used to run the algorithm.

MapReduce was inspired by the map and reduce functions commonly used in functional programming, for example Lisp. It was introduced by Google [24] and then implemented by many companies (Google, Yahoo!) and organizations (Apache). These implementations provide automatic parallelization and distribution, fault-tolerance, I/O scheduling, status and monitoring. The only demand made of the user is the formulation of the problem in terms of map and reduce functions. We use the terminology *mapper* and *reducer* when we refer to the map and reduce function respectively. The map function takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library provides the ability to acquire input pairs from files or databases which are stored in distributed way. Additionally, it can group all intermediate values associated with the same intermediate key I and pass them to the same reducer. The reduce function accepts an intermediate key I and a set of values associated with I. It merges these values to form a possibly smaller set of values.

Twister [7] was designed to enhance MapReduce's functionality by efficiently supporting iterative algorithms. Twister uses a public/subscribe messaging infrastructure for communication and data transfer, and introduces long running map/ reduce tasks which can be re-used in different iterations. These long running tasks, which last for the duration of the entire computation, ensures that Twister avoids reading static data in each execution of MapReduce; a considerable saving. For iterative algorithms, Twister categorizes data as being either static or dynamic. Static data is the distributed data in local machines. Dynamic data is typically the data produced by the previous iteration. Twister's *configure* phase allows the specification of where the mapper reads the static data. Calculation is performed cyclically based upon the dynamic and static data. All communication between the mappers and the reducers is handled by a broker network[3].

Unlike Twister, Hadoop focuses on single step MapReduce and lacks built-in support for iterative programs. For iterative algorithms, Hadoop MapReduce chains multiple jobs together. The output of a previous MapReduce task is used as the input for the next MapReduce task[4]. This approach is suboptimal; it incurs the additional cost of repetitively applying MapReduce –the disadvantage is that new map/reduce tasks are created repetitively for different iterations. This incurs considerable performance overhead costs.

5 Evaluation

We provide evidence of the effectiveness and scalability of our algorithm in this section. First we describe the experimental environment and the dataset characteristics for the datasets used. Then, we describe our experimental results.

5.1 Test Environment and Datasets

MRGanter and MRGanter+ are implemented in Java using the Twister runtime as the distributed environment. In addition, MRCbo, a distributed version of

[3] NaradaBrokering is used in this paper http://www.naradabrokering.org/

[4] http://hadooptutorial.wikispaces.com/Iterative+MapReduce+and+Counters

Table 7. UCI dataset characteristics: numbers of objects, attributes, and density

Dataset	mushroom	anon-web	census-income
objects	8124	32711	103950
attributes	125	294	133
density	17.36%	1.03%	6.7%

Table 8. Execution time: Distributed algorithms are the fastest (in seconds)

Dataset concepts	mushroom 219010	anon-web 129009	census-income 96531
NextClosure	618	14671	18230
CloseByOne	2543	656	7465
MRGanter	20269(5 nodes)	20110 (3 nodes)	9654 (11 nodes)
MRCbo	241 (11 nodes)	693 (11 nodes)	803 (11 nodes)
MRGanter+	198 (9 nodes)	496 (9 nodes)	358 (11 nodes)

CloseByOne proposed by Krajca and Vychodil [23] is implemented using the Twister model in order to provide a fair comparison with the algorithms proposed in the present paper. To illustrate the performance improvement of our distributed approach, we also evaluate NextClosure and CloseByOne.

The experiments were run on the Amazon EC2 cloud computing platform. We used High-CPU Medium Instances which had 1.7 GB of memory, 5 EC2 Compute Units (2 virtual cores with 2.5 EC2 Compute Units each), 350 GB of local instance storage, and a 32-bit platform. We selected 3 datasets from UCI KDD machine learning repository, mushroom, anon-web, and census-income for this evaluation[5]. These datasets have 8124, 32711, 103950 records and 125, 294, 133 attributes respectively. We used the percentage of 1s to measure the dataset density (see row 4 in Table 7). CPU time was used as the metric for comparing the performance of each of the algorithm. The number of iterations used by each algorithms was also recorded in Table 9.

5.2 Results and Analysis

In Table 8, we present the best test results for the centralized algorithms, NextClosure and CloseByOne, and the distributed algorithms, MRGanter, MRCbo and MRGanter+. In short, it is clear that MRGanter+ has the best overall performance for the mushroom, anon-web and census datasets when 9 nodes and 11 nodes are used respectively. In comparison with NextClosure, MRGanter+ demonstrates a 97.6% time saving improvement. MRGanter+ runs 102 times faster than MRGanter and 1.4 times faster than MRCbo. MRCbo runs much faster than CloseByOne when 11 nodes are used. It presents a 90.5% saving in time when dealing with the mushroom dataset compared to CloseByOne, but

[5] http://archive.ics.uci.edu/ml/index.html

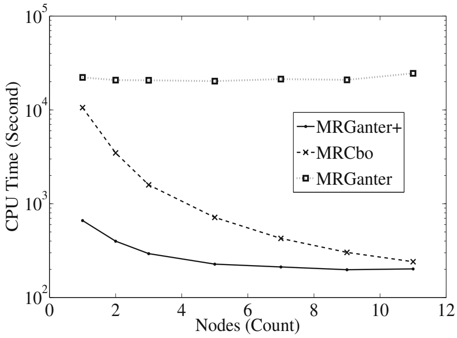


Fig. 2. Mushroom dataset: comparison of MRGanter+, MRCbo and MRGanter. MRGanter+ outperforms MRCbo and MRGanter when dense data is processed.

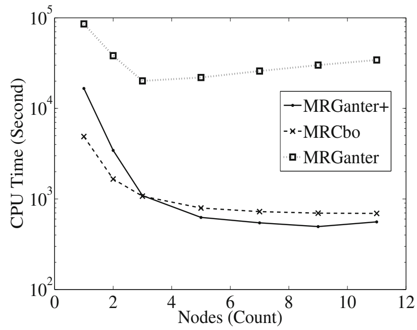


Fig. 3. Anon-web dataset: comparison of MRGanter+, MRCbo and MRGanter. MRGanter+ is faster when more than 3 nodes are used.

there is not much of difference when the anon-web dataset is processed. MRGanter takes the longest time to calculate the formal concepts for both the mushroom and anon-web datasets. It is much slower than even the centralized version, NextClosure. The census-income dataset is an exception because MRGanter saves up to half the time with 11 nodes. Among the MR* algorithms and centralized algorithms, MRGanter+ achieves the best performance.

Taking scalability into account, we tested MR* algorithms on a range of nodes to demonstrate the ability of the algorithms to decrease computation time by utilizing more computers. These results are presented in Fig. 2, 3 and 4 for each dataset.

In Fig. 2, MRCbo is slower than MRGanter+ although this curve decreases faster than MRGanter+ when we increase the number of nodes. The execution time of MRGanter+ is fast even on a single node and the execution time keeps decreasing up to the maximum number of nodes, 11. The performance of MRGanter is not beneficially affected by increasing the number of nodes. One explanation for this is the overhead incurred by distributing the computation, for example network communication overhead. This is markedly different from MRGanter+, because MRGanter+ produces substantially more intermediate data than MRGanter and MRCbo. Moreover, there is additional computation involved in the distributed algorithms in comparison with the centralized versions of these algorithms. Consider, for instance, the extra operation needed by the merging operation. The best number of nodes, in terms of performance speed, depends on the density characteristics of the dataset.

Fig. 3 demonstrates that MRGanter+ outperforms MRGanter for the anon-web dataset. One reason for this performance improvement is that both algorithms produce different numbers of concepts during each iteration. Table 9 indicates that MRGanter+ requires 12, 11 and 9 iterations for each of the datasets, whereas MRGanter requires 219010, 129009 and 96531 iterations to obtain all

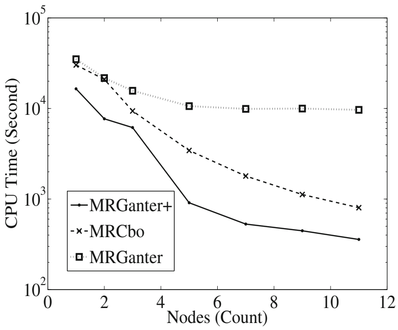


Fig. 4. Census dataset: comparison of MRGanter+, MRCbo and MRGanter. MRGanter+ is fastest when a large dataset is processed.

Table 9. Number of iterations required for each of the three datasets

Dataset	mushroom	anon-web	census-income
concepts	219010	129009	96531
NextClosure	219010	129009	96531
CloseByOne	14	11	11
MRGanter	219010	129009	96531
MRCbo	14	11	11
MRGanter+	12	11	9

concepts. These additional iterations incur higher network communication costs. Fig. 4 demonstrates that this is also the case for the census dataset. In addition, the curves in Fig. 4 are steeper than the curves in Fig. 2 and 3. These figures give evidence that the performance of the MR* algorithms is related to size and density of the data. Based on these results we posit that MR* algorithms scale well for large and sparse datasets. This evidence suggests that MR* algorithms may be a viable candidate tool for handling large datasets, particularly when it is impractical to use a traditional centralized technique.

Classical formal concept computing methods usually act on, and have local access to the entire database. Network communication is the primary concern when developing distributed FCA approaches: Frequent requests to remote databases incur significant time and resource costs. Performance improvements of the algorithms proposed in this paper may potentially arise from preprocessing the dataset so that the dataset is partitioned in a more efficient manner. One direction for improving these algorithms lies in making the partitions more even, in terms of density, so that the complexity is distributed more equably. In future work we we intend to explore the effect of data distribution between cluster nodes in more detail. We propose to extend this empirical study in a companion paper which examines algorithm performance on larger dataset sizes. We will also study the affects the data distribution has on the optimal number of nodes. In addition, we intend to extend these methods so that they reduce the size of intermediate data produced in each iteration. We posit that further improvement of the methods proposed here could motivate a more widespread adoption of FCA using the Map-Reduce framework.

6 Conclusion

In this paper we considered methods for extending the NextClosure FCA algorithm. A formal description of dealing with distributed datasets for the

NextClosure FCA was discussed. Two new distributed FCA algorithms, MRGanter and MRGanter+, were proposed based on this discussion. Various implementation aspects of these approaches were discussed based on empirical evaluation of the algorithms. These experiments demonstrated the advantages of our approach and the scalability in particular of MRGanter+. By comparing MRGanter+ with MRCbo and MRGanter, we found that the number of iterations significantly impacted the performance of distributed FCA, a promising result. In future work we hope to capitalize on this by improving the MR^* methodology by reducing the number of iterations of these approaches and to further reduce computation time.

References

1. Wille, R.: Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel (1982)
2. Lakhal, L., Stumme, G.: Efficient Mining of Association Rules Based on Formal Concept Analysis. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis. LNCS (LNAI), vol. 3626, pp. 180–195. Springer, Heidelberg (2005)
3. Polaillon, G., Aufaure, M.-A., Le Grand, B., Soto, M.: FCA for Contextual Semantic Navigation and Information Retrieval in Heterogeneous Information Systems. In: DEXA Workshops 2007, pp. 534–539 (2007)
4. Snásel, V., Horak, Z., Kocibova, J., Abraham, A.: Analyzing Social Networks Using FCA: Complexity Aspects. In: Web Intelligence/IAT Workshops 2009, pp. 38–41 (2009)
5. Caspard, N., Monjardet, B.: The Lattices of Closure Systems, Closure Operators, and Implicational Systems on a Finite Set: A Survey. Discrete Applied Mathematics, 241–269 (2003)
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
7. Ekanayake, J., Li, H., Zhang, B., Gunarathne, T., Bae, S.-H., Qiu, J., Fox, G.: Twister: a Runtime for Iterative MapReduce. In: Hariri, S., Keahey, K. (eds.) HPDC, pp. 810–818. ACM (2010)
8. Ganter, B.: Two Basic Algorithms in Concept Analysis. In: Kwuida, L., Sertkaya, B. (eds.) ICFCA 2010. LNCS, vol. 5986, pp. 312–340. Springer, Heidelberg (2010)
9. Lindig, C.: Fast Concept Analysis. In: Working with Conceptual Structures-Contributions to ICCS, pp. 235–248 (2000)
10. Kuznetsov, S.O.: A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-Lattice. Automatic Documentation and Mathematical Linguistics 27(5), 11–21 (1993)
11. Andrews, S.: In-Close, a Fast Algorithm for Computing Formal Concepts. In: The Seventeenth International Conference on Conceptual Structures (2009)
12. Vychodil, V.: A New Algorithm for Computing Formal Concepts. Cybernetics and Systems, 15–21 (2008)
13. Krajca, P., Outrata, J., Vychodil, V.: Parallel Recursive Algorithm for FCA. In: CLA 2008, vol. 433, pp. 71–82. CLA (2008)
14. Bordat, J.-P.: Calcul pratique du treillis de Galois d'une correspondance. Mathématiques et Sciences Humaines 96, 31–47 (1986)
15. Berry, A., Bordat, J.-P., Sigayret, A.: A Local Approach to Concept Generation. Ann. Math. Artif. Intell. 49(1), 117–136 (2006)

16. Kuznetsov, S.O., Obiedkov, S.A.: Comparing Performance of Algorithms for Generating Concept Lattices. J. Exp. Theor. Artif. Intell. 14, 189–216 (2002)
17. Norris, E.M.: An Algorithm for Computing the Maximal Rectangles in a Binary Relation. Rev. Roum. Math. Pures et Appl. 23(2), 243–250 (1978)
18. Dowling, C.E.: On the Irredundant Generation of Knowledge Spaces. J. Math. Psychol. 37, 49–62 (1993)
19. Godin, R., Missaoui, R., Alaoui, H.: Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. Computational Intelligence 11, 246–267 (1995)
20. Carpineto, C., Romano, G.: A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval. Machine Learning, 95–122 (1996)
21. Valtchev, P., Missaoui, R., Lebrun, P.: A Partition-based Approach Towards Constructing Galois (concept) Lattices. Discrete Mathematics, 801–829 (2002)
22. Yu, Y., Qian, X., Zhong, F., Li, X.-R.: An Improved Incremental Algorithm for Constructing Concept Lattices. In: Proceedings of the 2009 WRI World Congress on Software Engineering, WCSE 2009, vol. 04, pp. 401–405. IEEE Computer Society, Washington, DC (2009)
23. Krajca, P., Vychodil, V.: Distributed Algorithm for Computing Formal Concepts Using Map-Reduce Framework. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) IDA 2009. LNCS, vol. 5772, pp. 333–344. Springer, Heidelberg (2009)
24. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: OSDI, p. 13 (2004)
25. Chu, C.T., Kim, S.K., Lin, Y.A., Yu, Y., Bradski, G.R., Ng, A.Y., Olukotun, K.: Map-Reduce for Machine Learning on Multicore. In: Schölkopf, B., Platt, J.C., Hoffman, T. (eds.) NIPS, pp. 281–288. MIT Press (2006)

Author Index

Adam, Ruth 96
Allard, Pierre 112

Babin, Mikhail A. 7
Bartl, Eduard 16
Belohlavek, Radim 28
Berry, Anne 45

Chen, Hsinchun 1
Chollet, Stéphanie 61

De Raedt, Luc 2
Doerfel, Stephan 77

Elzinga, Paul 3
Endres, Dominik 96

Ferré, Sébastien 112
Fréin, Ruairí de 292

Gandrille, Etienne 61
Ganter, Bernhard 128
Giese, Martin A. 96
Glodeanu, Cynthia Vera 128
Goethals, Bart 4

Haraguchi, Makoto 244

Jäschke, Robert 77

Kaiser, Tim B. 140
Kerkhoff, Sebastian 148
Kirchberg, Markus 164
Ko, Ryan K.L. 164
Krupka, Michal 16, 180
Kuznetsov, Sergei O. 7

Lalanda, Philippe 61
Lastovicka, Jan 180
Lee, Bu Sung 164
Leonardi, Erwin 164
Lestideau, Vincent 61
Link, Sebastian 164

Macko, Juraj 195
Maurel, Yoann 61

Naidenova, Xenia A. 211
Noppeney, Uta 96

Obiedkov, Sergei 227
Ó Foghlú, Mícheál 292
Okubo, Yoshiaki 244

Pavlovic, Dusko 260
Peters, Ir. Edward 5

Raynaud, Olivier 61
Ridoux, Olivier 112
Robson, Eric 292
Rudolph, Sebastian 278

Schmidt, Stefan E. 140
Sigayret, Alain 45
Ślęzak, Dominik 6
Stumme, Gerd 77

Tan, Yu Shyang 164
Trnecka, Martin 28

Xu, Biao 292