Workshop Notes

12th International Workshop
"What can FCA do for Artificial Intelligence?"
FCA4AI 2024

27th European Conference on Artificial Intelligence
ECAI 2024

October 19 2024

Santiago de Compostela, Spain

Editors

Sergei O. Kuznetsov (HSE University Moscow)

Amedeo Napoli (LORIA Nancy)

Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2024/

ECAI 50TH ANNIVERSARY

Preface

The eleven preceding editions of the FCA4AI Workshop (see `http://www.fca4ai.hse.ru/`) showed that many researchers working in Artificial Intelligence are deeply interested in a well-founded method for classification and data mining such as Formal Concept Analysis (see `https://upriss.github.io/fca/fca.html`).

The FCA4AI Workshop Series started with ECAI 2012 (Montpellier) and the last edition was co-located with IJCAI 2023 (Macao, China). The FCA4AI workshop has now a long history and all proceedings are available as CEUR proceedings (see `http://ceur-ws.org/`, volumes 939, 1058, 1257, 1430, 1703, 2149, 2529, 2729, 2972, 3233, and 3489). This year, the workshop has again attracted researchers from different countries working on actual and important topics related to FCA, showing the diversity and the richness of the relations between FCA and AI.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies, i.e., implications and association rules, which can be used for many AI needs, e.g. knowledge discovery, machine learning, knowledge representation and reasoning, natural language and text processing. Recent years have been witnessing increased scientific activity around FCA. In particular an important line of work is aimed at extending the possibilities of FCA w.r.t. data and knowledge processing, and dealing with complex data. These extensions open new directions for AI practitioners. Accordingly, the workshop will investigate the following issues:

- How can FCA support AI activities such as knowledge discovery, knowledge representation and reasoning, machine learning, natural language processing, information retrieval...
- How can FCA be extended for helping AI researchers to solve new and complex problems, in particular how to combine FCA, neural classifiers, and LLMs, for allowing interpretability and producing valuable explanations...

First of all we would like to thank all the authors for their contributions and all the PC members for their reviews and their precious collaboration. The papers submitted to the workshop were carefully peer-reviewed by three members of the program committee, and the revised papers were prepared according to the reviews. We hope that these proceedings will be practical and useful for partipants to the FCA4AI 2024 Workshop and as well to all readers who are intersted in the close relations existing between FCA and AI.

The Workshop Chairs

Sergei O. Kuznetsov
National Research University Higher School of Economics, Moscow, Russia

Amedeo Napoli
Université de Lorraine, CNRS, LORIA, 54000 Nancy, France

Sebastian Rudolph
Technische Universität Dresden, Germany

Program Committee

Jaume Baixeries (UPC Barcelona, Catalunya)

Alexandre Bazin (LIRMM, Université de Montpellier, France)

Karell Bertet (L3I, Université de La Rochelle, France)

Peggy Cellier (IRISA, Université de Rennes, France)

Miguel Couceiro (LORIA, Université de Lorraine, Nancy France)

Diana Cristea (Babes-Bolyai University, Cluj-Napoca, Romania)

Florent Domenach (Akita International University, Japan)

Sébastien Ferré (IRISA, Université de Rennes, France)

Dmitry I. Ignatov (HSE University Moscow, Russia)

Mehdi Kaytoue (Infologic, Lyon, France)

Francesco Kriegel (Technische Universität Dresden, Germany)

Florence Le Ber (ENGEES/Université de Strasbourg, France)

Jesùs Medina (University of Càdiz, Spain)

Rokia Missaoui (UQO University Ottawa, Canada)

Sergei A. Obiedkov (Technische Universität Dresden, Germany)

Christian Sacarea (Babes-Bolyai University, Cluj-Napoca, Romania)

Baris Sertkaya (Frankfurt University of Applied Sciences, Germany)

Martin Trnecka (Palacky University Olomouc, Czech Republic)

Francisco José Valverde Albacete (Universidad Rey Juan Carlos de Madrid, Spain)

Renato Vimieiro (Universidade Federal de Minas Gerais, Belo Horizonte, Brazil)

Contents

Modelling Commonsense Knowledge about Concepts with Language Models

Zied Bouraoui[1]

[1]UFR des Sciences Jean Perrin, Rue Jean Souvraz SP 18, 62307 Lens Cedex, France

Abstract

Modeling concepts and their relationships is crucial for many knowledge-intensive tasks, such as few-shot and zero-shot learning and knowledge base completion. In this talk, I will explore strategies for learning effective concept and relation representations from language models and provide an overview of how these embeddings can enhance downstream applications, such as completing ontologies with plausible missing rules.

Keywords

concepts and relations, few-shot and zero-shot learning, language models, knowledge base completion

Published in Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph (Eds.): The 12th International Workshop "What can FCA do for Artificial Intelligence?", FCA4AI 2024, co-located with ECAI 2024, October 19 2024, Santiago de Compostela, Spain.

zied.bouraoui@cril.fr (Z. Bouraoui)
http://www.cril.univ-artois.fr/~bouraoui/ (Z. Bouraoui)
0000-0002-1662-4163 (Z. Bouraoui)

KLM-style Defeasible Reasoning on Concepts

Fei Liang[3], Krishna Manoorkar[1,*], Alessandra Palmigiano[1,2] and Apostolos Tzimoulis[1]

[3] *School of Philosophy and Social Development, Shandong University, China*
[1] *Vrije University, Amsterdam, the Netherlands*
[2] *Department of Mathematics and Applied Mathematics, University of Johannesburg, South Africa*

Abstract

In this paper, we introduce a KLM-style framework for defeasible reasoning about formal concepts. This framework can be used both for theoretical developments and in applications of non-monotonic reasoning about formal concepts.

Keywords

Formal Concept Analysis, Non-monotonic reasoning, KLM framework

1. Introduction

Non-monotonic logics are a class of logics which allow for inference relation to be non-monotonic, i.e. such that adding more knowledge or preferences can lead to some inferences to be retracted. These logical frameworks are intended to formally account for forms of reasoning which allow for exceptions and revision of conclusions. Non-monotonic logics play a crucial role in several fields of artificial intelligence, such as common-sense reasoning [1], ethical AI [2], and argumentation theory [3]. Various formal frameworks for non-monotonic reasoning have been developed, including Default Logic [4], AGM Belief Revision [5], Defeasible Entailment Reasoning [6], Conditional Logic [7], Circumscription [8], Autoepistemic logic [9].

Formal Concept Analysis (FCA) [10] is an established mathematical framework used in Knowledge Representation and Reasoning to study FCA hierarchies. The basic structures in FCA, namely formal contexts and their associated concept lattices, have been systematically linked with—and used as semantic environments of—a large family of lattice-based propositional logics, prominent examples of which are lattice-based modal logics, and their theory has been developed as a family of logics for reasoning about (formal) concepts in the context of data structures and information theory [11, 12, 13, 14]. Each logic in this family is defined in terms of a *monotone* consequence (or entailment) relation $C_1 \vdash C_2$ between concepts, which is semantically interpreted as 'C_1 is a subconcept of C_2', that is, 'all the objects in the extension of C_1 are in the extension of C_2', or equivalently, 'all the features in the intension of C_2 are in the intension of C_1'. On the basis of this framework, various more sophisticated logical frameworks have been proposed, including epistemic logic for categories and categorization endowed with a 'common knowledge' operator accounting for prototypicality [12], a basic environment for a Dempster-Shafer theory of concepts [15], a unifying environment for Rough Set Theory and FCA [16], many-valued logics accounting for vague categories [17], a specifically FCA-based description logic for FCA [18, 19], and various proof-theoretic frameworks laying the foundations of the computational theory of these logics [20, 21].

Deciding whether some concept inclusion is entailed by a given FCA knowledge base (e.g. a set of concept inclusions) is an important reasoning task which can be efficiently carried out by lattice-based propositional logics such as those mentioned above. However, in many applications, especially in

Workshop 'What can FCA do for AI'?, 2024
*Corresponding author.
f.liang@sdu.edu.cn (F. Liang); k.b.manoorkar@vu.nl (K. Manoorkar); alessandra.palmigiano@vu.nl (A. Palmigiano); a.tzimoulis@vu.nl (A. Tzimoulis)
0000-0002-4932-1329 (F. Liang); 0000-0003-3664-7757 (K. Manoorkar); 0000-0001-9656-7527 (A. Palmigiano); 0000-0002-6228-4198 (A. Tzimoulis)

context of large data, as well as in many real-life situations, a part of the available knowledge may be *defeasible* (i.e. presented in form of concept-inclusions which allow for exceptions). Studying defeasible entailment on concepts would allow us to infer knowledge from knowledge consisting of defeasible concept-inclusions, and to capture, and hence implement, common sense reasoning about concepts. For example, if a generic object (animal) a is in thecategory of 'mammals', then we can reasonably conclude that a is viviparous. However, if we receive additional information that the animal is a platypus, then we can conclude it is not viviparous.

Formally, we can define the following three defeasible counterparts of the monotone entailment relation $\vdash$ discussed above: (1) Relation $\mid\!\sim_A$ interpreted as 'all the objects in C_1, with some possible exceptions, are in C_2' or 'typical objects of C_1 are in C_2', (2) relation $\mid\!\sim_X$ interpreted as 'all the objects C_1 have all the features shared by C_2, with some possible exceptions' or 'all the objects of C_1 have typical features of C_2', and (3) relation $\mid\!\sim_{AX}$ interpreted as 'all the objects C_1, with some possible exceptions, have all the features of C_2 with some possible exceptions' or 'all the typical objects of C_1 have all the typical features of C_2'. For example, let C_1 and C_2 represent the concepts of 'mammals' and 'viviparous animals', respectively. Since mammals are typically viviparous, we have $C_1 \mid\!\sim_A C_2$. However, if we introduce C_3, representing the concept of 'echidnas', which are a kind of oviparous mammal, we find that $C_3 \vdash C_1$ (i.e., all echidnas are mammals), hence $C_3 \mid\!\sim_A C_1$, but $C_3 \not\mid\!\sim_A C_2$ (i.e., typically, echidnas are not viviparous).

In the present paper, we propose to extend the framework of Kraus, Lehmann, and Magidor (commonly referred to as the KLM framework) [6] to formalize defeasible entailment between concepts.

Since FCA does not have a natural notion of negation on concepts, the KLM framework cannot be directly applied to the FCA environment. Nonetheless, it can be suitably extended to FCA. Specifically, we define the FCA-counterparts of classical non-monotonic entailment relations such as the *cumulative entailment* $\mathbf{C}$, and the *cumulative entailment with loop* $\mathbf{CL}$. These counterparts are the three defeasible entailment relations $\mid\!\sim_A$, $\mid\!\sim_X$, and $\mid\!\sim_{AX}$ mentioned above. We do not include the preferential entailment system $\mathbf{P}$, as the counterpart of the rule OR in classical defeasible reasoning depends on the fact that the semantic counterpart of classical disjunction is the set-theoretic union, while in FCA disjunction is interpreted as the closure of the union. In fact, unlike what is the case in the classical setting, the FCA-counterpart of $\mathbf{C}$ is already complete w.r.t. the class of FCA preferential models (cf. Theorems 1, 2, 2). Moreover, as the language of FCA does not have a natural notion of negation for concepts, FCA-counterparts of axioms such as rational monotonicity are not available. It would be interesting for future research to explore whether some FCA counterparts of such rules can be defined.

Open directions on the front of semantic investigations concern the definition of the FCA-counterparts of cumulative models, cumulative ordered models, preferential models, and preferential ordered models and the proof of completeness theorems for different reasoning systems w.r.t. these classes of models. An interesting aspect of this research concerns exploring the similarities and differences between—as well as the relationships among—the defeasible consequence relations $\mid\!\sim_A$, $\mid\!\sim_X$, and $\mid\!\sim_{AX}$.

2. KLM framework for reasoning on concepts

In [22], the first steps were taken for developing the KLM framework in the setting of FCA, by introducing only the defeasible entailment relation $\mid\!\sim_A$ on formal concepts. Here, we start by recalling this framework and the results proved there.

To generalize the cumulative reasoning to the FCA setting, we modify the original framework [6] as follows: In [6], the language of underlying logic is assumed to be closed under all the classical connectives including negation and implication. However, lattice-based propositional logic does not have negation and implication in its language. Thus, we replace the formula $\phi \rightarrow \psi$ in the rules and axioms of $\mathbf{C}$ with the sequent $\phi \vdash \psi$, which encodes the entailment at a meta-logical level, rather than at the object language level. For any formal context $\mathbb{P} = (A, X, I)$, a *model* based on $\mathbb{P}$ is a tuple $\mathbb{M} = (\mathbb{P}, V)$ s.t. $V : \mathcal{L} \rightarrow \mathbb{P}^+$ is a homomorphism from the term algebra $\mathcal{L}$ of the propositional logic of lattices into the concept lattice $\mathbb{P}^+$ associated with $\mathbb{P}$. For any $\phi \in \mathcal{L}$, we let $[\![\phi]\!]_{\mathbb{M}}$ (resp. $(\![\phi]\!)_{\mathbb{M}}$)

denote the extension (resp. intension) of $V(\phi)$ (dropping the subscripts when the context is clear), and $\mathbb{M} \models \phi \vdash \psi$ iff $[\![\phi]\!]_{\mathbb{M}} \subseteq [\![\psi]\!]_{\mathbb{M}}$ iff $(\![\psi]\!)_{\mathbb{M}} \subseteq (\![\phi]\!)_{\mathbb{M}}$.

A *lattice-based cumulative logic* consists of an *entailment relation*, i.e. a set of $\mathcal{L}$-sequents $\phi \vdash \psi$ closed under all axioms and rules of lattice-based propositional logic, and a *cumulative entailment relation*, i.e. a set of $\mathcal{L}$-sequents $\phi \mid\!\sim_A \psi$ closed under the Reflexivity axiom $\phi \mid\!\sim_A \phi$ and the rules

Left Logical Equivalence (LLE) $\dfrac{\phi\vdash\psi \quad \psi\vdash\phi \quad \phi\mid\!\sim_A\chi}{\psi\mid\!\sim_A\chi}$ $\qquad$ $\dfrac{\phi\vdash\psi \quad \chi\mid\!\sim_A\phi}{\chi\mid\!\sim_A\psi}$ Right Weakening (RW)

Cautious Monotonicity (CM) $\dfrac{\phi\mid\!\sim_A\psi \quad \phi\mid\!\sim_A\chi}{\phi\wedge\psi\mid\!\sim_A\chi}$ $\qquad$ $\dfrac{\phi\wedge\psi\mid\!\sim_A\chi \quad \phi\mid\!\sim_A\psi}{\phi\mid\!\sim_A\chi}$ (Cut).

From (LLE) and (RW) it follows that logically equivalent formulas are $\mid\!\sim_A$-entailed by the same formulas. A cumulative entailment relation $\mid\!\sim_A$ is *loop-cumulative* if it satisfies the following rule.

$$\frac{\phi_0\mid\!\sim_A\phi_1 \quad \phi_1\mid\!\sim_A\phi_2 \quad \dots \quad \phi_{n-1}\mid\!\sim_A\phi_n \quad \phi_n\mid\!\sim_A\phi_0}{\phi_0\mid\!\sim_A\phi_n}\ \text{(Loop)}$$

Let us define the FCA-counterparts of the models of defeasible reasoning by suitably adapting the approach used in [23] to define KLM-style modal logics.

A *pointed model* is a tuple $\mathbb{M}_a = (\mathbb{P}, V, a)$, where $\mathbb{M}$ is a model, and $a \in A$. Let $\mathcal{M} = (S, l, \prec)$ be a tuple s.t. S is a non-empty set (of *states*), $l : S \to \mathcal{P}(\mathcal{U})$ maps each state to a set of pointed models, and $\prec$ is a binary relation on S. For any $\phi \in \mathcal{L}$ and $s \in S$, $s \models \phi$ iff $a \in [\![\phi]\!]_{\mathbb{M}}$ for all $\mathbb{M}_a \in l(s)$. $\mathcal{M}$ is a *cumulative model* if, for any $\phi \in \mathcal{L}$, the set $\widehat{\phi} := \{s \mid s \in S, s \models \phi\}$ is *smooth* (i.e. for any $t \in \widehat{\phi}$, either t is $\prec$-minimal in $\widehat{\phi}$, or $s \prec t$ for some $\prec$-minimal element $s \in \widehat{\phi}$). A cumulative model $\mathcal{M} = (S, l, \prec)$ is *strong* if $\prec$ is asymmetric (i.e. $s \prec t$ implies $t \not\prec s$ for all $s, t \in S$) and $\widehat{\phi}$ has a minimum for every $\phi \in \mathcal{L}$; is *ordered* if $\prec$ is a strict partial order; is *preferential* if l assigns a single pointed model to each state. Any cumulative model $\mathcal{M}$ defines a *cumulative entailment* $\mid\!\sim_{\mathcal{M}}$ by: $\phi_1 \mid\!\sim_{\mathcal{M}} \phi_2$ iff for any s, if s is minimal in $\widehat{\phi_1}$, then $s \in \widehat{\phi_2}$.

It is easy to check that $\mid\!\sim_{\mathcal{M}}$ is a cumulative entailment relation. Reflexivity follows from $\min(\widehat{\phi}) \subseteq \widehat{\phi}$. (LLE) holds since $\widehat{\phi} = \widehat{\psi}$ implies $\min(\widehat{\phi}) = \min(\widehat{\psi})$. (RW) holds since $\min(\widehat{\chi}) \subseteq \phi$ and $\widehat{\phi} \subseteq \widehat{\psi}$ imply that $\min(\widehat{\chi}) \subseteq \psi$. As to (CM), if $\min(\widehat{\phi}) \subseteq \widehat{\psi}$, $\min(\widehat{\phi}) \subseteq \widehat{\chi}$, and $s \in \min(\widehat{\phi \wedge \psi})$, then, if $s \notin \min(\widehat{\phi})$, by smoothness, $s' \prec s$ for some $s' \in \min(\widehat{\phi})$. Hence, as $\min(\widehat{\phi}) \subseteq \widehat{\psi}$, $s' \in \widehat{\phi \wedge \psi}$, contradicting the minimality of s. The soundness of (Cut) is shown similarly.

Theorem 1. *(cf. [22]) A consequence relation is cumulative (resp. loop-cumulative) iff it coincides with $\mid\!\sim_{\mathcal{M}}$ for some* strong *(resp.* ordered*) cumulative model $\mathcal{M}$, iff it coincides with $\mid\!\sim_{\mathcal{M}}$ for some* preferential *(resp.* preferential ordered*) cumulative model $\mathcal{M}$.*

The defeasible entailment $\mid\!\sim_X$ can be characterized by dualizing the rules for $\mid\!\sim_A$, using the well known fact that the order on concepts is defined by reverse inclusion on their intensions.

A *lattice-based dually cumulative logic* consists of the entailment relation $\vdash$ of a lattice-based propositional logic, and a *dually cumulative entailment relation*, i.e. a set of $\mathcal{L}$-sequents $\phi \mid\!\sim_X \psi$ closed under the Reflexivity axiom $\phi \mid\!\sim_X \phi$ and the rules

Right Logical Equivalence (RLE) $\dfrac{\phi\vdash\psi \quad \psi\vdash\phi \quad \chi\mid\!\sim_X\phi}{\chi\mid\!\sim_X\psi}$ $\qquad$ $\dfrac{\phi\vdash\psi \quad \psi\mid\!\sim_X\chi}{\phi\mid\!\sim_X\chi}$ Left Weakening (LW)

Dual Cautious Monotonicity (DCM) $\dfrac{\psi\mid\!\sim_X\phi \quad \chi\mid\!\sim_X\phi}{\chi\mid\!\sim_X\phi\vee\psi}$ $\qquad$ $\dfrac{\chi\mid\!\sim_X\phi\vee\psi \quad \psi\mid\!\sim_X\phi}{\chi\mid\!\sim_X\phi}$ Dual Cut (DCut).

The rules above are obtained from the rules for $\mid\!\sim_A$ by switching the order of the consequence relation and interchanging $\vee$ and $\wedge$. This corresponds to the idea that the lattice of set of concept intensions under set inclusion forms a complete lattice dual to the concept lattice. From (RLE) and (LW) it follows that logically equivalent formulas $\mid\!\sim_X$-entail the same formulas.

Note that the rule loop is invariant under dualizing. A dually cumulative entailment relation is *loop-cumulative* if it satisfies the rule Loop.

$$\frac{\phi_0\mid\!\sim_X\phi_1 \quad \phi_1\mid\!\sim_X\phi_2 \quad \dots \quad \phi_{n-1}\mid\!\sim_X\phi_n \quad \phi_n\mid\!\sim_X\phi_0}{\phi_0\mid\!\sim_X\phi_n}\ \text{(Loop)}$$

We can define models for the various types of dually cumulative relations (i.e. *dually cumulative models* and their *strong*, *ordered*, and *preferential* subclasses) by replacing pointed models with *dually pointed models*, i.e. tuples $\mathbb{M}_x := (\mathbb{M}, x)$ s.t. $\mathbb{M}$ is a model and $x \in X$. All other parts of the definitions remain unchanged, including the dually cumulative entailment $\mid\!\sim_{\mathcal{M}}$ associated with a dual cumulative model $\mathcal{M}$. We can show soundness of all the above rules w.r.t. these models in a manner analogous to soundness proof of $\mid\!\sim_A$ rules. The proof of the following completeness theorem is similar to the previous one.

Theorem 2. *A consequence relation is dually cumulative (resp. dually loop-cumulative) iff it coincides with $\mid\!\sim_{\mathcal{M}}$ for some* strong *(resp.* ordered*) dually cumulative model $\mathcal{M}$, iff it coincides with $\mid\!\sim_{\mathcal{M}}$ for some* preferential *(resp.* preferential ordered*) dually cumulative model $\mathcal{M}$.*

A lattice-based *bi-cumulative logic* consists of an entailment relation $\vdash$ for lattice-based propositional logic, a cumulative entailment relation $\mid\!\sim_A$ and a dually cumulative entailment relation $\mid\!\sim_X$. Such a logic is *loop-cumulative* when both $\mid\!\sim_A$ and $\mid\!\sim_X$ are. Semantic models for these logics can be defined as tuples $\mathcal{M}_{AX} = (\mathcal{M}_A, \mathcal{M}_X)$, s.t. $\mathcal{M}_A$ is a cumulative model and $\mathcal{M}_A$ is a dually cumulative model; the corresponding (*strong*, *ordered*, and *preferential*) subclasses are defined by imposing the corresponding conditions on $\mathcal{M}_A$ and $\mathcal{M}_X$, and the bi-cumulative logic associated with $\mathcal{M}_{AX}$ is specified by $\mid\!\sim_{\mathcal{M}_A}$ and $\mid\!\sim_{\mathcal{M}_X}$.[1] The following is a straightforward corollary of the previous completeness results.

Theorem 3. *A pair of entailment relations defines a (loop-cumulative) bi-cumulative logic iff it arises from some (ordered) strong bi-cumulative model, iff it arises from some preferential (resp. preferential ordered) bi-cumulative model.*

Finally, we consider *expanded bi-cumulative logics* as bi-cumulative logics endowed with a third type $\mid\!\sim_{AX}$ of defeasible entailment, closed under the following rules except (Loop); when satisfying also (Loop), such a logic is *loop-cumulative.*

$$\text{(LLE)}\ \frac{\phi\vdash\psi \quad \psi\vdash\phi \quad \phi\mid\!\sim_{AX}\chi}{\psi\mid\!\sim_{AX}\chi} \qquad \frac{\phi\vdash\psi \quad \psi\vdash\phi \quad \chi\mid\!\sim_{AX}\phi}{\chi\mid\!\sim_{AX}\psi}\ \text{(RLE)}$$

$$Comb_A\ \frac{\phi\mid\!\sim_A\psi}{\phi\mid\!\sim_{AX}\psi} \qquad \frac{\phi\mid\!\sim_X\psi}{\phi\mid\!\sim_{AX}\psi}\ Comb_X$$

$$(\text{CM}_A)\ \frac{\phi\mid\!\sim_A\psi \quad \phi\mid\!\sim_{AX}\chi}{\phi\wedge\psi\mid\!\sim_{AX}\chi} \qquad \frac{\psi\mid\!\sim_X\phi \quad \chi\mid\!\sim_{AX}\phi}{\chi\mid\!\sim_{AX}\phi\vee\psi}\ (\text{CM}_X)$$

$$(\text{Cut}_A)\ \frac{\phi\wedge\psi\mid\!\sim_{AX}\chi \quad \phi\mid\!\sim_A\psi}{\phi\mid\!\sim_{AX}\chi} \qquad \frac{\chi\mid\!\sim_{AX}\psi\vee\phi \quad \psi\mid\!\sim_X\phi}{\chi\mid\!\sim_{AX}\phi}\ (\text{Cut}_X)$$

$$\text{(Loop)}\ \frac{\phi_0\mid\!\sim_{AX}\phi_1 \quad \phi_1\mid\!\sim_{AX}\phi_2 \quad \ldots \quad \phi_{n-1}\mid\!\sim_{AX}\phi_n \quad \phi_n\mid\!\sim_{AX}\phi_0}{\phi_0\mid\!\sim_{AX}\phi_n}$$

The intuition behind these rules can be explained in the following manner.

- **(LLE) and (RLE)**: These rules simply say that $\mid\!\sim_{AX}$ respects logical equivalence. Note that $\mid\!\sim_{AX}$ is not assumed to be monotonic in either argument. This is consistent with the intended interpretation of $C_1 \mid\!\sim_{AX} C_2$ as 'typical objects of C_1 have typical features of C_2'. As typicality, which is a non-monotonic operator, is applied both to C_1 and C_2, it is natural to allow $\mid\!\sim_{AX}$ to be non-monotonic in both arguments.
- **Comb$_A$ and Comb$_X$**: These rules are sound under the intended interpretations of $\mid\!\sim_A$, $\mid\!\sim_X$, and $\mid\!\sim_{AX}$.
- **CM$_A$ and CM$_X$**: These rules state that the condition $\phi \mid\!\sim_A \psi$ (resp. $\psi \mid\!\sim_X \phi$) is enough to ensure the monotonicity of $\mid\!\sim_{AX}$ in the second (resp. first) argument.
- **Cut$_\mathbf{A}$ and Cut$_\mathbf{X}$**: We can perform a cut on the formula which is the second (resp. first) argument in a sequent containing $\mid\!\sim_{AX}$ using a sequent containing $\mid\!\sim_A$ and $\mid\!\sim_{AX}$.
- **Loop**: The loop rule behaves analogously to the loop rule for $\mid\!\sim_A$ or $\mid\!\sim_X$.

We believe that a further justification for these rules will be given by the completeness theorem for the expanded FCA bi-cumulative logic and FCA bi-cumulative ordered logic w.r.t. natural models for such systems conjectured below.
An entailment relation $\mid\!\sim_{\mathcal{M}_{AX}}$ can be associated with any bi-cumulative model $\mathcal{M}$ as follows: for any ϕ_1, ϕ_2, $\phi_1 \mid\!\sim_{\mathcal{M}_{AX}} \phi_2$ iff aIx for any $s_1 \in S_A$ and $s_2 \in S_X$, and all pointed models $\mathbb{M}_a \in l(s_1)$, $\mathbb{M}_x \in l(s_2)$ based on the same formal context $\mathbb{P} = (A, X, I)$ and valuation V on it. This corresponds to the idea that a typical object of ϕ_1 should have a typical feature of ϕ_2 when described in same (formal) context. We finish with the following conjecture.

[1]Note that we do not assume any relationship between the partial orders on $\mathcal{M}_A$ and $\mathcal{M}_X$. However, in many applications these two orders have some relationship which needs to be formalized. Studying logics with such relationships would be an interesting future direction for this project.

Conjecture 1. *A triple of of entailment relations defines an expanded (loop-cumulative) bi-cumulative logic iff there exists a (ordered) strong bi-cumulative model $\mathcal{M}$, iff there exists some preferential (resp. preferential ordered) bi-cumulative model $\mathcal{M}$, such that* $\mid\sim_A = \mid\sim_{\mathcal{M}_A}$, $\mid\sim_X = \mid\sim_{\mathcal{M}_X}$, *and* $\mid\sim_{AX} = \mid\sim_{\mathcal{M}_{AX}}$.

3. Conclusion and future directions

In this work, we take first steps in defining a KLM style framework for defeasible reasoning on concepts. This opens several directions for future research and applications:

Formally modelling scenarios involving defeasible concept inclusions: Several real-life scenarios involving reasoning about concepts include defeasible reasoning. Our framework can be used to formally model these scenarios. A toy example (consisting only of $\mid\sim_A$) is discussed in [22].

Reasoning from defeasible knowledge bases: As discussed in the introduction, one of the primary aim of this work is to develop a framework for reasoning from knowledge given in the form of conceptual inclusions. In this direction, it would be interesting to study the complexity of various reasoning systems described in the present paper. In the classical setting, it is known that the complexity of defeasible reasoning is same as the complexity of the underlying logic [24]. As reasoning about conceptual inclusions is known to be polynomial-time, showing a similar result in the FCA-setting would show that reasoning in these logics is computationally efficient.

Belief revision for conceptual knowledge: In several applications, we are interested in scenarios where the reasoner may need to incorporate new possibly inconsistent knowledge with existing beliefs of the agents. In the classical setting, non-monotonic logics have been used to define belief revision operators [25]. It would be interesting to define and study revision operators in the setting of FCA using the non-monotonic reasoning systems introduced in the present paper.

Acknowledgments

This paper is partially funded by the EU MSCA (grant No. 101007627). The research of Krishna Manoorkar is supported by the NWO grant KIVI.2019.001 awarded to Alessandra Palmigiano. The research of Fei Liang is supported by the Chinese Ministry of Education of Humanities and Social Science Project (23YJC72040003) and the Young Scholars Program of Shandong University (11090089964225).

References

[1] G. Brewka, Nonmonotonic reasoning: logical foundations of commonsense, volume 12, Cambridge University Press, 1991.

[2] J.-G. Ganascia, Ethical system formalization using non-monotonic logics, in: Proceedings of the annual meeting of the cognitive science society, volume 29, 2007.

[3] D. Walton, Argumentation theory: A very short introduction, in: Argumentation in artificial intelligence, Springer, 2009, pp. 1–22.

[4] R. Reiter, A logic for default reasoning, Artificial intelligence 13 (1980) 81–132.

[5] C. E. Alchourrón, P. Gärdenfors, D. Makinson, On the logic of theory change: Partial meet contraction and revision functions, The Journal of Symbolic Logic 50 (1985) 510–530. URL: http://www.jstor.org/stable/2274239.

[6] S. Kraus, D. Lehmann, M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, Artificial intelligence 44 (1990) 167–207.

[7] B. F. Chellas, Basic conditional logic, Journal of philosophical logic (1975) 133–153.

[8] J. McCarthy, Circumscription—a form of non-monotonic reasoning, Artificial intelligence 13 (1980) 27–39.

[9] W. Marek, M. Truszczyński, Autoepistemic logic, Journal of the ACM (JACM) 38 (1991) 587–618.

[10] B. Ganter, R. Wille, Formal concept analysis: mathematical foundations, Springer Science & Business Media, 2012.

[11] R. Limarga, M. Pagnucco, Y. Song, A. Nayak, Non-monotonic reasoning for machine ethics with situation calculus, in: M. Gallagher, N. Moustafa, E. Lakshika (Eds.), AI 2020: Advances in Artificial Intelligence, Springer International Publishing, Cham, 2020, pp. 203–215.
[12] W. Conradie, S. Frittella, A. Palmigiano, M. Piazzai, A. Tzimoulis, N. M. Wijnberg, Toward an epistemic-logical theory of categorization, Electronic Proceedings in Theoretical Computer Science 251 (2017) 167–186. doi:10.4204/eptcs.251.12.
[13] W. Conradie, S. Frittella, A. Palmigiano, M. Piazzai, A. Tzimoulis, N. M. Wijnberg, Categories: how i learned to stop worrying and love two sorts, in: Logic, Language, Information, and Computation: 23rd International Workshop, WoLLIC 2016, Puebla, Mexico, August 16-19th, 2016. Proceedings 23, Springer, 2016, pp. 145–164.
[14] W. Conradie, A. Palmigiano, C. Robinson, N. Wijnberg, Non-distributive logics: from semantics to meaning, in: A. Rezus (Ed.), Contemporary Logic and Computing, volume 1 of *Landscapes in Logic*, College Publications, 2020, pp. 38–86. URL: arXivpreprintarXiv:2002.04257.
[15] S. Frittella, K. Manoorkar, A. Palmigiano, A. Tzimoulis, N. Wijnberg, Toward a dempster-shafer theory of concepts, International Journal of Approximate Reasoning 125 (2020) 14–25.
[16] W. Conradie, S. Frittella, K. Manoorkar, S. Nazari, A. Palmigiano, A. Tzimoulis, N. M. Wijnberg, Rough concepts, Information Sciences 561 (2021) 371–413.
[17] W. Conradie, A. De Domenico, K. Manoorkar, A. Palmigiano, M. Panettiere, D. P. Prieto, A. Tzimoulis, Modal reduction principles across relational semantics, Fuzzy Sets and Systems 481 (2024) 108892.
[18] I. van der Berg, A. De Domenico, G. Greco, K. B. Manoorkar, A. Palmigiano, M. Panettiere, Non-distributive description logic, arXiv preprint arXiv:2307.09561 (2023).
[19] K. B. Manoorkar, A. De Domenico, A. Palmigiano, Description logic for rough concepts, in: International Joint Conference on Rough Sets, Springer, 2024, pp. 67–89.
[20] I. van der Berg, A. De Domenico, G. Greco, K. B. Manoorkar, A. Palmigiano, M. Panettiere, Labelled calculi for lattice-based modal logics, in: Indian Conference on Logic and Its Applications, Springer, 2023, pp. 23–47.
[21] I. van der Berg, A. De Domenico, G. Greco, K. B. Manoorkar, A. Palmigiano, M. Panettiere, Labelled calculi for the logics of rough concepts, in: Indian Conference on Logic and Its Applications, Springer, 2023, pp. 172–188.
[22] Y. Ding, K. Manoorkar, N. W. Switrayni, R. Wang, Defeasible reasoning on concepts, arXiv preprint submission id : submit/5840042 (2024).
[23] K. Britz, T. Meyer, I. Varzinczak, Preferential reasoning for modal logics, Electronic Notes in Theoretical Computer Science 278 (2011) 55–69. URL: https://www.sciencedirect.com/science/article/pii/S1571066111001344. doi:https://doi.org/10.1016/j.entcs.2011.10.006, proceedings of the 7th Workshop on Methods for Modalities (M4M'2011) and the 4th Workshop on Logical Aspects of Multi-Agent Systems (LAMAS'2011).
[24] D. Lehmann, M. Magidor, What does a conditional knowledge base entail?, Artificial intelligence 55 (1992) 1–60.
[25] P. Gärdenfors, D. Makinson, Nonmonotonic inference based on expectations, Artificial Intelligence 65 (1994) 197–245.

Conjunctive Concept Algebras

Named Perspective

Jens Kötters, Stefan E. Schmidt

Technische Universität Dresden, Germany

Abstract

Concept lattices of relational structures establish a database-theoretic variant of Formal Concept Analysis (FCA). As shown in recent work, these concept lattices naturally extend to concept algebras, by means of a semigroup action. Extensionally, these concept algebras form subalgebras of (a variant of) SPJR table algebras (the conjunctive query fragment of Codd's relational algebra). By that means, an axiomatic characterization of the concept $\wedge$-subalgebras (up to isomorphism, u.t.i.) has been obtained. However, the axioms are difficult to memorize, and in some respects, the semigroup action proved cumbersome to work with. In this paper, we reformulate the axioms, using the signature of Tarski's cylindric algebras (an algebraization of first-order predicate logic). The axioms compare surprisingly well to the cylindric algebra axioms, and the concept $\wedge$-subalgebras correspond to cylindric set algebras. We also obtain an axiomatic characterization of the concept $\bigwedge$-subalgebras (u.t.i.).

Keywords

Concept Algebras, Cylindric Algebra, Conjunctive Queries, Database Theory, Algebraic Logic

1. Motivation

Formal Concept Analysis (FCA) [1] is a mathematical theory of concepts. The central notion in FCA is the *concept lattice*, a complete lattice which describes a hierarchy of concepts. As the *Basic Theorem of FCA* states [1, p. 20], every complete lattice can be represented as a concept lattice. So in this sense, FCA is the theory of complete lattices, from a different perspective.

In the first publication on FCA [2], Rudolf Wille explains what this different perspective was meant to achieve. He was inspired by von Hentig [3], who warned that, as an effect of growing specialization, sciences were becoming disconnected from their surroundings and original motivations, and needed to be *restructured* to re-enable such connections. Wille writes:

> "*Restructuring lattice theory* is an attempt to reinvigorate connections with our general culture by interpreting the theory as concretely as possible, and in this way to promote better communication between lattice theorists and potential users of lattice theory." [2, emphasis added]
>
> "For this purpose we go back to the origin of the lattice concept in nineteenth-century attempts to formalize logic, where the study of hierarchies of concepts played a central rôle [...]." [2]

More than a decade later, when FCA was already established and had been successfully applied, Wille announced a second project [4], called *restructuring mathematical logic*.

> "The connections of logic to reality have been narrowed since Frege's turn to predicate logic, the leading paradigm of mathematical logic today. Thus, restructuring has to establish a broader understanding of mathematical logic, in particular, by elaborating the pragmatic dimension.
>
> For activating real communication and argumentation, it seems to be most important to build enough bridges from the logic-mathematical theory to reality. One way to do this is to revitalize the traditional paradigm of logic given by 'the three essential main functions of thinking - *concepts, judgments and conclusions*' ([5, p. 6])." [4, in-text citation adapted]

FCA4AI (Twelfth Edition), co-located with ECAI 2024, October 19, 2024, Santiago de Compostela, Spain

A formalization of concepts had already been achieved by FCA. In a follow-up paper [6], Wille observed that another well-known theory of concepts, *Conceptual Graphs* [7] by John Sowa, already offered a formalization of judgments and conclusions. More concretely, Wille points to a mathematization of conceptual graphs by Chein and Mugnier [8]. We gather that *S-graphs* (with "*S* for Sowa, for simple, [...]" [8]), which formalize the most basic type of conceptual graph, represent judgments, and that conclusions can be characterized by graph homomorphism [8, Thm. 1]. The question was then how the two theories can be unified, and Wille proceeds with a proposal. First, he introduces *abstract concept graphs* as slightly modified S-graphs. Then he introduces *power context families*, which represent relational data, and also support the usual notion of concepts for FCA. Finally, he introduces *concept graphs*, which combine abstract concept graphs with concepts from a power context family, thereby obtaining a formalization of judgments that builds on FCA concepts. A summary of concept graphs is presented in [9, Sect. 6.5].

Wille's inspirational paper [6] marked the beginning of a new era of FCA, where relations entered the stage. *Relational Concept Analysis* (RCA) [10][11] provides a deeper integration of concepts with relations, adapts to relational databases through *conceptual scaling* [1], supports different kinds of logical quantifiers, and is being applied in practice. Baader et al. [12] combine FCA with *Description Logics*, a more human-centered branch of logic (cf. Wille's criticism w.r.t. predicate logic above). Kötters [13] introduces a database-theoretic FCA variant; a detailed and refined presentation is given in [14, Sects. 3–5], and the paper at hand continues the theoretical developments.

Conjunctive queries [15, Ch. 4] are a natural and fundamental class of database queries. They were introduced by Chandra and Merlin [16], and have their origin in mathematical logic. Unifying the logical and database-theoretic viewpoints, we identify conjunctive queries with *primitive-positive formulas* (i.e. first-order formulas built from atoms using $\{\exists, \wedge\}$), evaluated in *relational structures*, where

$$\mathrm{res}_{\mathfrak{G}}(\varphi) := \{t \in G^X \mid \mathfrak{G} \models \varphi[t]\} \tag{1}$$

defines the *result table* of a formula φ (with set $X := \mathrm{free}(\varphi)$ of free variables) in a relational structure $\mathfrak{G}$ (with universe G).[1] A *relational database* is a finite relational structure [16, p. 77]. For any relational database $\mathfrak{G}$, the result operation $\mathrm{res}_{\mathfrak{G}}$ is part of a Galois connection, which means that we obtain a concept lattice $\mathfrak{B}(\mathfrak{G})$. This establishes a fundamental connection between FCA and database theory.

Tableau queries [15, p. 43] are structural representations of conjunctive queries. Accordingly, we represent a formula φ with $X := \mathrm{free}(\varphi)$ by a pair $(\mathfrak{N}, \nu)$, consisting of a relational structure $\mathfrak{N}$ and a *window* $\nu : X \to N$, elsewhere called the *summary* [15, p. 43], and obtain the result table as a set

$$\mathrm{res}_{\mathfrak{G}}(\mathfrak{N}, \nu) = \{f \circ \nu \mid f : \mathfrak{N} \to \mathfrak{G}\} \tag{2}$$

of homomorphisms as "seen through the window". From a graph-theoretical perspective, a relational structure is a graph [19]. Likewise, a tableau query can be considered a graph (cf. [14, Sect. 3.1] for our drawing conventions). Under this perspective, $(\mathfrak{N}, \nu)$ is a query graph, and $\mathrm{res}_{\mathfrak{G}}(\mathfrak{N}, \nu)$ contains the pattern matches in the data graph $\mathfrak{G}$. Tableau queries offer a natural way to express infinite conjunctive queries, and indeed, we have not required that $(\mathfrak{N}, \nu)$ must be finite. In order to maintain the logical perspective, a *graph logic* [14, Sect. 3.4] can be formulated. Homomorphisms $f : (\mathfrak{N}_1, \nu_1) \to (\mathfrak{N}_2, \nu_2)$ of tableau queries are defined in the obvious way, and correspond to logical implication in the graph logic.

The following reasons suggest that the database-theoretic FCA variant matches Wille's intention with the restructuring project:

- Wille indicates [6, pp. 291f.,300] that suitable notions of judgments and conclusions are offered by S-graphs and their homomorphisms. Since S-graphs represent primitive-positive formulas [8,

[1] Details of the unification: Note that queries can be represented in *prenex normal form* [17, Sect. 8.4]; constants are not allowed, but unary relations can play the role of constants [17, Sect. 8.1]; equality is allowed, e.g. the query $x{=}x$ requests a list of all objects in the database, even though such a query is not natively supported in Codd's data model [18]; equality does not enhance expressivity greatly, because in many instances, equality can be eliminated by substitution [15, p. 47f.] or would be expressed by variable repetition in a *conjunctive calculus query* [15, p. 45].

Sect. 9.1], we might as well consider tableau queries and their homomorphisms; the difference being that S-graphs represent closed formulas (i.e. sentences), whereas tableau queries may, and generally do, represent open formulas (having one or more free variables). By allowing free variables, we obtain concept extensions (cf. eq. (1)).[2]

- Conceptual graphs were initially motivated as a human-centered query language for relational databases [20].[3]
- The widespread use of relational databases suggests practical relevance and good availability of data.
- The result operation corresponds to the activity of querying a database, which suggests a pragmatic dimension.
- The implementation of the classical flight example [21] is not based on concept graphs, but on abstract concept graphs, interpreted as conjunctive queries.

We provide some logical background in Sect. 2, and give a short account of cylindric algebra in Sect. 3. In Sect. 4 we summarize recent results on table algebras [22][23], and also extend a result in Sect.4.4. In Sect. 5, we introduce conjunctive concept algebras and present our main results (Props. 10 and 11).

2. Preliminaries

We assume it is generally known what is meant by a *first-order formula*, and what it means that a first-order formula φ *holds* in a *structure* $\mathfrak{A}$ under a *variable assignment* α, written $\mathfrak{A} \models \varphi[\alpha]$ or $(\mathfrak{A}, \alpha) \models \varphi$, cf. [17]. A *signature* is generally a set M of function symbols, constants, and relation symbols. The set of first-order formulas over the signature M is denoted by $\mathrm{FO}(M)$. If M contains only relation symbols, it is called a *relational signature*, and a structure $\mathfrak{A}$ over M is called a *relational structure*. Because of our take on database theory, we always assume that M is a relational signature; this does not limit expressivity in general [17, Sect. 8.1]. Each symbol $m \in M$ has an *arity* $|m| \geq 1$. For technical convenience, we identify the countably infinite set of variables with the ordinal $\omega = \{0, 1, 2, \dots\}$. An *atomic formula* in $\mathrm{FO}(M)$ is either a *relational atom* $Rx_1 \dots x_n$, an *equality atom* $x{=}y$, or one of the special atoms true (the *tautology*) or false (the *contradiction*), for arbitrary $x_1, \dots, x_n, x, y \in \omega$.

Logical implication between formulas $\varphi, \psi \in \mathrm{FO}(M)$ in the *standard semantics* is introduced as in [17]. We say that φ *logically implies* ψ, and denote this by $\varphi \models \psi$, if $(\mathfrak{G}, \alpha) \models \varphi$ implies $(\mathfrak{G}, \alpha) \models \psi$ for all structures $\mathfrak{G}$ of signature M and all variable assignments $\alpha \in G^\omega$. From (5) we obtain that $\varphi \models \psi$ holds if and only if $\varphi^{\mathfrak{G}} \subseteq \psi^{\mathfrak{G}}$ for all structures $\mathfrak{G}$ (of signature M), i.e. logical implication is conveniently expressed via the solution sets. Accordingly, φ and ψ are *logically equivalent*, denoted by $\varphi \mathrel{=\!\mid\!\models} \psi$, if $\varphi^G = \psi^G$ for all $\mathfrak{G}$.

Then clearly, in the *table semantics*, φ and ψ should be *logically equivalent* if $\mathrm{res}_{\mathfrak{G}}(\varphi) = \mathrm{res}_{\mathfrak{G}}(\psi)$ for all $\mathfrak{G}$. The formulas $x{=}x$ and $y{=}y$ are then not equivalent, because the result tables have schemas $\{x\}$ and $\{y\}$, respectively, at least for nonempty $\mathfrak{G}$ and different x, y. So while the special atom true is a tautology, the equality atoms $x{=}x$ and $y{=}y$ are not. A logic with undefined variables provides a formal underpinning: the modified result operation

$$\mathrm{res}^*_{\mathfrak{G}}(\varphi) := \{t \in \mathrm{Tup}(G) \mid (\mathfrak{G}, t) \models \varphi\} \tag{3}$$

uses the finite tuples in $\mathrm{Tup}(G) := \bigcup\{G^X \mid X \in \mathcal{P}_{\mathrm{fin}}(\omega)\}$ as variable assignments, and if an assignment t is not defined on all variables in $\mathrm{free}(\varphi)$, then $(\mathfrak{G}, t) \not\models \varphi$. We refer to this as the *tuple set*

[2]Beyond the formal analogy, the distinction between concepts and judgments in the conjunctive query approach needs to be clarified.
[3]Interestingly, Sowa's article predates Chandra and Merlin's [16] by a year.

semantics. The function

$$h : \begin{cases} \mathrm{Tab}(G) \to \mathcal{P}_{\mathrm{fin}}(\mathrm{Tup}(G)) \\ T \mapsto \{t \in \mathrm{Tup}(G) \mid t|_{\mathrm{schema}(T)} \in T\} \end{cases} \tag{4}$$

satisfies $\mathrm{res}^*_{\mathfrak{G}} = h \circ \mathrm{res}_{\mathfrak{G}}$, so it relates table semantics and tuple set semantics. It forms an embedding $h : (\mathrm{Tab}(G), \bowtie) \to (\mathcal{P}_{\mathrm{fin}}(\mathrm{Tup}(G)), \cap)$ of meet-semilattices, i.e. an injective homomorphism; and as such, it also forms an order embedding $h : (\mathrm{Tab}(G), \leq) \to (\mathcal{P}_{\mathrm{fin}}(\mathrm{Tup}(G)), \subseteq)$; thereby providing a set interpretation of the tables and their order, see also [14, Sect. 3.5][24]. In particular, $\varphi \lesssim \psi :\Leftrightarrow \forall \mathfrak{G} : \mathrm{res}_{\mathfrak{G}}(\varphi) \leq \mathrm{res}_{\mathfrak{G}}(\psi) \Leftrightarrow \forall \mathfrak{G} : \mathrm{res}^*_{\mathfrak{G}}(\varphi) \subseteq \mathrm{res}^*_{\mathfrak{G}}(\psi)$ denotes *logical implication* in both the table semantics and the tuple set semantics. So both semantics are equivalent; we can use either of them, depending on the purpose. Finally, we write $\varphi \simeq \psi$ if and only if $\varphi \lesssim \psi$ and $\psi \lesssim \varphi$, which coincides with our initially postulated notion of equivalence.

The following proposition relates table semantics with standard semantics.

Proposition 1. *Let* $\varphi, \psi \in \mathrm{PP}(M)$. *Then* $\varphi \lesssim \psi$ *if and only if* $\varphi \models \psi$ *and* $\mathrm{free}(\psi) \subseteq \mathrm{free}(\varphi)$.

Proof. The case $\varphi = \mathsf{false}$ is trivial. Now let $\varphi \neq \mathsf{false}$ and $\psi = \mathsf{false}$. Because φ is primitive-positive (and not the contradiction), it is satisfiable (cf. [17, Ex. 3.4]), i.e. there exists $\mathfrak{G}$ such that $\varphi^{\mathfrak{G}} \not\subseteq \emptyset = \psi^{\mathfrak{G}}$, so $\varphi \not\models \psi$; and likewise, we obtain $\varphi \not\lesssim \psi$. The case $\varphi, \psi \neq \mathsf{false}$ is covered in [14, Prop. 3]. □ □

3. Cylindric Algebra

The two most fundamental disciplines of logic, as of today, are *propositional logic* and *predicate logic*; and by predicate logic, we usually mean *first-order logic. Boolean algebras* are well-known algebraizations of propositional logic. Likewise, *cylindric algebras* by Alfred Tarski are algebraizations of first-order logic. The classical monographs on cylindric algebras are the works of Henkin, Monk and Tarski [25][26], and for an introduction, we refer to the papers of Németi [27] and Monk [28]. We first present *cylindric set algebras* (Sect. 3.1), then turn to *cylindric algebras* in general (Sect. 3.2).

3.1. Cylindric Set Algebras

Every relational structure $\mathfrak{G}$ with signature M induces a *solution operation* $(\cdot)^{\mathfrak{G}} : \mathrm{FO}(M) \to \mathcal{P}(G^\omega)$ that maps each first-order formula φ to its *solution set*

$$\varphi^{\mathfrak{G}} := \{\alpha \in G^\omega \mid (\mathfrak{G}, \alpha) \models \varphi\} \ . \tag{5}$$

The algebra $\mathbf{FO}(M) = (\mathrm{FO}(M), \vee, \wedge, \neg, \mathsf{false}, \mathsf{true}, \exists_x, x{=}y)_{x,y \in \omega}$ interprets $\vee$, $\wedge$, $\neg$ and $\exists_x$ (for all $x \in \omega$) as syntactic operations, e.g. $\vee(\varphi, \psi) := (\varphi \vee \psi)$ and $\exists_x(\varphi) := (\exists x \varphi)$. Moreover, it contains false, true, and all equality atoms $x{=}y$ as distinguished elements. The solution operation forms a homomorphism $(\cdot)^{\mathfrak{G}} : (\mathrm{FO}(M), \vee, \wedge, \neg, \mathsf{false}, \mathsf{true}) \to (\mathcal{P}(G^\omega), \cup, \cap, (\cdot)^{\complement}, \emptyset, G^\omega)$. In this sense, the logical operations are represented by set operations. Likewise, existential quantification over x is represented by the *cylindrification* $C_x : \mathcal{P}(G) \to \mathcal{P}(G)$, defined by

$$C_x(A) := \{\alpha \in G^\omega \mid \exists g \in G : \alpha\tfrac{g}{x} \in A\} \ , \tag{6}$$

where $\alpha\frac{g}{x} \in G^\omega$ is the *modification* of α that satisfies $\alpha\frac{g}{x}(x) = g$ and $\alpha\frac{g}{x}(y) = \alpha(y)$ for all $y \in \omega \setminus \{x\}$. Finally, the equality atoms $x{=}y$ are represented by the *diagonals*

$$D_{xy} := \{\alpha \in G^\omega \mid \alpha(x) = \alpha(y)\} \ . \tag{7}$$

This motivates $\mathbf{Cs}(G) := (\mathcal{P}(G^\omega), \cup, \cap, (\cdot)^{\complement}, \emptyset, G^\omega, C_x, D_{xy})_{x,y \in \omega}$ as a set-theoretic counterpart of $\mathbf{FO}(M)$; but note that in principle, G and M are independent. In summary, the relational structure $\mathfrak{G}$ induces the *solution homomorphism* $(\cdot)^{\mathfrak{G}} : \mathbf{FO}(M) \to \mathbf{Cs}(G)$.

The homomorphic image $\mathbf{Cs}(\mathfrak{G}) := [\mathbf{FO}(M)]^{\mathfrak{G}}$ is the subalgebra of $\mathbf{Cs}(G)$ that consists of the solution sets. More generally, a subalgebra of $\mathbf{Cs}(G)$ is called a *cylindric set algebra* with base G and dimension ω. We now pose two questions, and state the answers below, as found in Monk [28]:

a) How are the solution set algebras $\mathbf{Cs}(\mathfrak{G})$ characterized from among all cylindric set algebras of dimension ω?

b) Is there an axiomatic characterization for the cylindric set algebras of dimension ω?

a) The algebras $\mathbf{Cs}(\mathfrak{G})$ are precisely the *locally finite-dimensional* and *regular* cylindric set algebras of dimension ω (cf. [28, Thm. 12.2]), b) The cylindric set algebras of dimension ω are not first-order axiomatizable (cf. [28, p. 279]).

3.2. Cylindric Algebras

Because of negative results with regard to first-order axiomatization of cylindric set algebras and other concrete notions, *cylindric algebra* were introduced. They are defined by a finite schema of equations, to provide for a good theory, and are meant to circumscribe the interesting classes of concrete algebras sufficiently well. In that sense, the notion of cylindric algebra is arbitrary, cf. [27, Sect. 4].

Definition 2. *A* cylindric algebra *is an algebra* $(V, \vee, \wedge, \neg, 0, 1, c_x, d_{xy})_{x,y\in\omega}$ *consisting of a binary* supremum $\vee$*, a binary* infimum $\wedge$*, a unary* complement $\neg$*, a* zero *element* 0*, a* one *element* 1*, a unary* cylindrification c_x *for each* $x \in \omega$*, and a* diagonal *element* d_{xy} *for each* $(x, y) \in \omega \times \omega$*, which satisfies*

(CA0) $(V, \vee, \wedge, \neg, 0, 1)$ *is a Boolean algebra*

(CA1) $c_x(0) = 0$

(CA2) $u \le c_x(u)$

(CA3) $c_x(u \wedge c_x(v)) = c_x(u) \wedge c_x(v)$

(CA4) $c_x(c_y(u)) = c_y(c_x(u))$

(CA5) $d_{xx} = 1$

(CA6) $x \neq y, z \Rightarrow d_{yz} = c_x(d_{yx} \wedge d_{xz})$

(CA7) $x \neq y \Rightarrow c_x(d_{xy} \wedge u) \wedge c_x(d_{xy} \wedge \neg u) = 0$

for all $u, v \in V$ *and all* $x, y, z \in \omega$*.*

4. Table Algebras

From an extensional point of view, concept lattices of relational structures are table algebras. In quest for a Basic Theorem, this motivates the study of table algebras.

4.1. DPJR Algebras

The *SPJR algebra* [15, Sect. 4.4] allows to specify conjunctive queries using algebraic operations; these are the table operations of *selection*, *projection*, *(natural) join* and *renaming*, indicated by the letters. It is also called the *named conjunctive algebra*, because it operates on tables with named columns (as opposed to tables with ordered columns). While Abiteboul et al. [15] refer to SPJR algebra as a query language, it better suits our extensional viewpoint to think of it as an algebra of tables, with concrete operations.

We define a *table* as a set $T \subseteq G^X$, where $X \subseteq \omega$ is a finite set of *column names* (not column numbers), an element $t \in T$ is a *row*, $t(x)$ is the *entry* in row t and column x, and G is an arbitrary set. Hence,

$$\mathrm{Tab}(G) = \bigcup\{\mathcal{P}(G^X) \mid X \subseteq \omega \text{ finite}\} \tag{8}$$

contains all tables with entries in G. Note that while X must be finite, a table can have an infinite number of rows if G is infinite. Naturally, the empty set $\emptyset$ represents the *empty table*. The *schema* of a table $T \in \mathrm{Tab}(G)$ is uniquely defined by

$$\mathrm{schema}(T) := \begin{cases} X & \text{if } T \in G^X \text{ and } T \neq \emptyset \\ \omega & \text{if } T = \emptyset \end{cases} . \tag{9}$$

Note that $G^\emptyset$ has a single element $\emptyset$, called the *empty tuple*, and $\{\emptyset\} \in \mathcal{P}(G^\emptyset)$ is the unique table with schema $\emptyset$.

For finite $X \subseteq \omega$, the set $\mathrm{Tab}(G)[X] := \mathcal{P}(G^X)$ is the X-*slice* of $\mathrm{Tab}(G)$. The *natural join* of tables $S \in \mathrm{Tab}(G)[X]$ and $T \in \mathrm{Tab}(G)[Y]$ is a table $T \in \mathrm{Tab}(G)[X \cup Y]$, defined by

$$S \bowtie T := \{t \in G^{X \cup Y} \mid t|_X \in S \text{ and } t|_Y \in T\} \ . \tag{10}$$

Moreover, for all $x, y \in \omega$, we define the *diagonal*

$$E_{xy} := \{t \in G^{\{x,y\}} \mid t(x) = t(y)\} \ . \tag{11}$$

The natural join is associative and commutative [15, p. 58], and trivially idempotent, i.e. $(\mathrm{Tab}(G), \bowtie)$ is a meet-semilattice, with the implied table order $T_1 \leq T_2 :\Leftrightarrow T_1 = T_1 \bowtie T_2$. The tables $\emptyset$ and $\{\emptyset\}$ are the absorbing element and neutral element, respectively, w.r.t. to the join. This means that they are also the smallest and greatest elements in the lattice order.

A *finite partial transformation* of ω is a partial function $\lambda : \omega \rightharpoonup \omega$, defined on a finite set $\mathrm{def}(\lambda) = X \subseteq \omega$, and we set $\mathrm{rng}(\lambda) = \{\lambda(x) \mid x \in \mathrm{def}(\lambda)\}$. We use $\mathcal{T}_{\mathrm{fp}}(\omega)$ to denote the set of finite partial transformations on ω. The pair $(\mathcal{T}_{\mathrm{fp}}, \circ)$ is a semigroup, with $\circ$ as composition of partial functions, which naturally acts on the tables through the *right multiplication*

$$\cdot \begin{cases} \mathrm{Tab}(G) \times \mathcal{T}_{\mathrm{fp}}(\omega) \to \mathrm{Tab}(G) \\ (T, \lambda) \mapsto T \cdot \lambda := \{t \circ \lambda \mid t \in T\} \end{cases} . \tag{12}$$

The right multiplicaton encodes three different table operations: projection, renaming and column duplication. For the *partial identity* $\pi_X : \omega \rightharpoonup \omega$, which can be written $\{(x, x) \mid x \in X\}$ as a relation, $T \cdot \pi_X$ is the *projection* of T on the column set X. Note that right multiplication is totally defined, so generally $\mathrm{schema}(T \cdot \pi_X) = \mathrm{schema}(T) \cap X$. A *partial bijection* is an injective function $\xi : \omega \rightharpoonup \omega$, and it acts as a *renaming* on $\mathrm{Tab}(G)$. Moreover, a *folding* is a partial function $\delta : \omega \rightharpoonup \omega$ with $\delta \circ \delta = \delta$, and for each $x \in \mathrm{def}(\delta)$, the table $T \cdot \delta$ has a column x which is a copy of $\delta(x)$; the column $\delta(x)$ is fixed because of $\delta \circ \delta = \delta$. This completely describes right multiplication, since every $\lambda \in \mathcal{T}_{\mathrm{fp}}(\omega)$ acts as a sequence of these operations [22, Lemma 1]; more concretely, there is a decomposition $\lambda = \pi_X \circ \xi \circ \delta$, and furthermore $T \cdot (\pi_X \circ \xi \circ \delta) = ((T \cdot \pi_X) \cdot \xi) \cdot \delta$. For the above reason, we call

$$\mathrm{DPJR}(G) = (\mathrm{Tab}(G), \bowtie, \emptyset, \{\emptyset\}, \cdot, E_{xy}, \mathrm{schema})_{x,y \in \omega} \tag{13}$$

the *full DPJR algebra* with *base* G. A *DPJR algebra* with *base* G is a subalgebra of $\mathrm{DPJR}(G)$. Before we proceed, the relation with SPJR algebras shall be explained.

Abiteboul et al. [15, p. 57] refer to two kinds of selection, denoted by $\sigma_{A=a}$ and $\sigma_{A=B}$, where A and B are column names, and a denotes an object in the universe. The reference to a reflects a database-theoretic convention, whereby objects in the universe are exposed as constants. Note however, that in our formalization of conjunctive queries, which unifies the database-theoretic and logical viewpoints (cf. the footnote in Sect. 1), we strictly allow relation symbols only. So the corresponding variant of SPJR algebra would only use the second kind of selection (i.e. $\sigma_{A=B}$, which deletes all rows having different entries in the A and B columns). It is a moderately easy exercise to show that DPJR algebra (without diagonals) is equivalent to this variant of SPJR algebra. The diagonals are not part of SPJR algebra; their inclusion in the DPJR algebra also caters to the unified viewpoint.

4.2. Conjunctive Table Algebras

We motivate conjunctive table algebras in the same way we have motivated cylindric set algebras in Sect. 3.1. A first-order formula is *primitive-positive* if it is built from atoms using $\{\wedge, \exists\}$. The set of primitive-positive formulas over the relational signature M is denoted by $\mathrm{PP}(M)$. The algebra $\mathbf{PP}(M) := (\mathrm{PP}(M), \wedge, \mathsf{false}, \mathsf{true}, \exists_x, x{=}y, \mathrm{free})_{x,y \in \omega}$ extends $\mathrm{PP}(M)$ with the respective syntactic operations and constants (cf. the algebra $\mathbf{FO}(M)$ in Sect. 3.1), and it also includes the function free : $\mathrm{PP}(M) \to \mathcal{P}(\omega)$, which maps each formula to its set of free variables; for the special atoms, we define $\mathrm{free}(\mathsf{true}) = \emptyset$ and $\mathrm{free}(\mathsf{false}) = \omega$.

Every relational structure $\mathfrak{G}$, with universe G and signature M, induces a *result operation* $\mathrm{res}_{\mathfrak{G}} : \mathrm{PP}(M) \to \mathrm{Tab}(G)$ that maps each formula φ to its *result table*, given by

$$\mathrm{res}_{\mathfrak{G}}(\varphi) := \{t \in G^{\mathrm{free}(\varphi)} \mid (\mathfrak{G}, t) \models \varphi\} \ . \tag{14}$$

In particular, we have $\mathrm{res}_{\mathfrak{G}}(\mathsf{false}) = \emptyset$ and $\mathrm{res}_{\mathfrak{G}}(\mathsf{true}) = \{\emptyset\}$. Note that each variable in $\mathrm{free}(\varphi)$ corresponds to a column in the result table $\mathrm{res}_{\mathfrak{G}}(\varphi)$.

Next, we identify the table operations which correspond to the logical operations. Existential quantification is matched by column deletion; we define the *deletion operation* $\mathrm{del}_x : \mathrm{Tab}(G) \to \mathrm{Tab}(G)$ by

$$\mathrm{del}_x(S) := \{t|_{X \setminus \{x\}} \mid t \in S\} \ . \tag{15}$$

Note that $\mathrm{del}_x(S) = S$ if $x \notin X$. The other required operations have already been defined in Sect. 4.1. As expected, we have

$$\begin{aligned} \mathrm{res}_{\mathfrak{G}}(\varphi \wedge \psi) &= \mathrm{res}_{\mathfrak{G}}(\varphi) \bowtie \mathrm{res}_{\mathfrak{G}}(\psi) \\ \mathrm{res}_{\mathfrak{G}}(\mathsf{false}) &= \emptyset \\ \mathrm{res}_{\mathfrak{G}}(\mathsf{true}) &= \{\emptyset\} \\ \mathrm{res}_{\mathfrak{G}}(\exists_x \varphi) &= \mathrm{del}_x(\mathrm{res}_{\mathfrak{G}}(\varphi)) \\ \mathrm{res}_{\mathfrak{G}}(x = y) &= E_{xy} \ , \end{aligned}$$

and if $\mathrm{res}_{\mathfrak{G}}(\varphi) \neq \emptyset$, then also $\mathrm{schema}(\mathrm{res}_{\mathfrak{G}}(\varphi)) = \mathrm{free}(\varphi)$. This motivates to define $\mathbf{Tab}(G) := (\mathrm{Tab}(G), \bowtie, \emptyset, \{\emptyset\}, \mathrm{del}_x, E_{xy}, \mathrm{schema})_{x,y \in \omega}$ as the *full conjunctive table algebra* with *base* G.

As indicated, in the case $\mathrm{res}_{\mathfrak{G}}(\varphi) = \emptyset$, the free variables of φ can not be recovered from the result table, and in this sense they are not preserved. Consequently, we do not consider $\mathrm{res}_{\mathfrak{G}} : \mathbf{PP}(M) \to \mathbf{Tab}(G)$ to be a proper homomorphism, and refer to it as a *zero-tolerant homomorphism*, a slightly weaker kind of homomorphism. But it does preserve all logical operations and constants, so the homomorphic image $\mathbf{Tab}(\mathfrak{G}) := \mathrm{res}_{\mathfrak{G}}[\mathbf{PP}(M)]$ is a subalgebra of $\mathbf{Tab}(G)$. This motivates our main definition.

Definition 3 (Conjunctive Table Algebra)**.** *A* conjunctive table algebra *with* base G *is a subalgebra* $\mathfrak{A}$ *of* $\mathbf{Tab}(G)$.

The X*-slice* of $\mathfrak{A}$, for each $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$, is the set $\mathfrak{A}[X] := \{T \in A \mid T \in G^X\}$. For convenience, we define $\mathfrak{A}^*[X] := \{T \in A \mid \mathrm{schema}(T) = X\} = \mathfrak{A}[X] \setminus \{\emptyset\}$. Note that $n = \{0, \ldots, n-1\}$, so $\mathfrak{A}[n] = \mathfrak{A}[\{0, \ldots, n-1\}]$ and $\mathfrak{A}^*[n] = \mathfrak{A}[\{0, \ldots, n-1\}]$.

In Sect. 3.1, we have presented two questions (and their answers) on cylindric set algebras. We formulate their counterparts in our database-theoretic setting:

a) How are the algebras $\mathbf{Tab}(\mathfrak{G})$ characterized from among all conjunctive table algebras?

b) Is there an axiomatic characterization for the conjunctive table algebras?

Proposition 4. *Conjunctive table algebras and DPJR algebras are equivalent:*

i) Every conjunctive table algebra is closed under right multiplication.

ii) Every DPJR algebra is closed under deletions.

Proof. i) Let $\mathfrak{A}$ be a conjunctive table algebra. We show $T \cdot \lambda \in A$ for all $T \in \mathfrak{A}[Y]$, $Y \in \mathcal{P}_{\mathrm{fin}}(\omega)$, and $\lambda \in \mathcal{T}_{\mathrm{fp}}(\omega)$. Since $T \cdot \lambda = T \cdot \lambda|_{\lambda^{-1}(Y)}$, we may assume w.l.o.g. that $\mathrm{rng}(\lambda) \subseteq Y$, i.e. $\lambda : X \to Y$ for some $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$. If $X \cap Y = \emptyset$, then $T \cdot \lambda = \mathrm{del}_Y(T \bowtie E_\lambda) \in \mathfrak{A}[X]$. Otherwise, let $\xi : Y \to Z$ be a bijection onto some $Z \in \mathcal{P}_{\mathrm{fin}}(\omega)$ with $Z \cap X = \emptyset$ and $Z \cap Y = \emptyset$. By reduction to the previous case, we first obtain $T \cdot \xi^{-1} \in \mathfrak{A}[Z]$, and then $T \cdot \lambda = (T \cdot \xi^{-1}) \cdot (\xi \circ \lambda) \in \mathfrak{A}[X]$.

ii) Let $\mathfrak{A}$ be a DPJR algebra. For all $T \in \mathfrak{A}[X]$, $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$ and $x \in \omega$, we have $\mathrm{del}_x(T) = T \cdot \pi_{X \setminus \{x\}} \in \mathfrak{A}[X \setminus \{x\}]$. □

Proposition 5. *The conjunctive table algebras are precisely the result table algebras* $\mathbf{Tab}(\mathfrak{G})$ *of relational structures* $\mathfrak{G}$*.*

Proof. By definition, every algebra $\mathbf{Tab}(\mathfrak{G})$ is a conjunctive table algebra. Now let $\mathfrak{A}$ be a conjunctive table algebra with base G. Let $M_{\mathfrak{A}} = \bigcup_{n\geq 1} \mathfrak{A}^*[n]$ be the relational signature which uses $\mathfrak{A}^*[n]$ as its set of n-ary relation symbols. Each $T \in \mathfrak{A}^*[n]$ is also a set of n-tuples, i.e. an n-ary relation. Let $\mathfrak{G}_{\mathfrak{A}}$ be the relational structure with universe G and signature $M_{\mathfrak{A}}$, given by the map $I : M_{\mathfrak{A}} \to M_{\mathfrak{A}}$ that maps each $T \in \mathfrak{A}^*[n]$ (as a symbol) to $T \in \mathfrak{A}^*[n]$ (as a relation), i.e. $I = \mathrm{id}_{M_{\mathfrak{A}}}$. It remains to show $\mathrm{res}_{\mathfrak{G}_{\mathbf{A}}}[\mathbf{PP}(M_{\mathfrak{A}})] = \mathfrak{A}$.

"$\subseteq$:" By definition of $\mathfrak{G}_{\mathfrak{A}}$, we have $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(T(0,\ldots,n-1)) = T \in M_{\mathfrak{A}} \subseteq A$ for all relational atoms $T(0,\ldots,n-1)$. Let $\sigma : n \to X$ be a substitution of variables, such that $n \cap X = \emptyset$. Then $T(\sigma(0),\ldots,\sigma(n-1))$ is equivalent to $\varphi_{T,\sigma} := \exists 0 \ldots \exists n-1 : (T(0,\ldots,n-1) \wedge 0{=}\sigma(0) \wedge \cdots \wedge n-1{=}\sigma(n-1))$. So $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(T(\sigma(0),\ldots,\sigma(n-1))) = \mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(\varphi_{T,\sigma}) = \mathrm{del}_0 \ldots \mathrm{del}_{n-1}(T \bowtie E_{0\sigma(0)} \bowtie \cdots \bowtie E_{n-1,\sigma(n-1)}) \in A$. Every relational atom $T(x_1,\ldots,x_n)$ is obtained from $T(0,\ldots,n-1)$ by two such substitutions, i.e. $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(T(x_1,\ldots,x_n)) \in A$ for all relational atoms. By induction, $\mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}(\varphi) \in A$ for all $\varphi \in \mathrm{PP}(M)$.

"$\supseteq$:" Let $T \in \mathfrak{A}^*[X]$ for some $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$ with cardinality $n := \#X$. We choose an arbitrary bijection $\xi : n \to X$, and obtain $T \cdot \xi \in \mathfrak{A}^*[n] \subseteq \mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}[\mathbf{PP}(M_{\mathfrak{A}})]$. By Prop. 4, the homomorphic image is closed under right multiplication, so we also have $T = (T \cdot \xi) \cdot \xi^{-1} \in \mathrm{res}_{\mathfrak{G}_{\mathfrak{A}}}[\mathbf{PP}(M_{\mathfrak{A}})]$. □

Proposition 5 provides a simple answer to our question a) above: The algebras $\mathbf{Tab}(\mathfrak{G})$ are precisely the conjunctive table algebras. The primary question is how the algebras $\mathbf{Tab}(\mathfrak{G})$ can be axiomatized. As we have seen now, the formal framework of cylindric set algebras fits the question perfectly (which was not the case for cylindric set algebra, cf. question a) in Sect. 3.1). An answer to our question b) is given in Sect. 4.3.

4.3. Projectional Semilattices

The main result of [23] is the axiomatic characterization of conjunctive table algebras by projective semilattices. The given axiomatization is not a first-order axiomatization, but a comparison with cylindric algebra axioms, given below, should convince the reader of their value.

Definition 6 ([23, Def. 2])**.** *A* projectional semilattice *is an algebra* $(V, \wedge, 0, 1, c_x, d_{xy}, \mathrm{dom})_{x,y\in\omega}$ *consisting of a binary infimum* $\wedge$*, a* zero *element* 0*, a* one *element* 1*, a unary* cylindrification c_x *for each* $x \in \omega$*, a* diagonal *element* d_{xy} *for each* $(x, y) \in \omega \times \omega$*, and a* domain function $\mathrm{dom} : V \to \mathcal{P}(\omega)$*, which satisfies*

- **(PS0)** $(V, \wedge, 0, 1)$ *is a bounded semilattice*
- **(PS1)** $c_x(0) = 0$
- **(PS2)** $u \leq c_x(u)$
- **(PS3)** $c_x(u \wedge c_x(v)) = c_x(u) \wedge c_x(v)$
- **(PS4)** $c_x(c_y(u)) = c_y(c_x(u))$
- **(PS5)** $u \neq 0 \Rightarrow (u \neq c_x(u) \Leftrightarrow u \leq d_{xx})$
- **(PS6)** $x \neq y, z \Rightarrow d_{yz} = c_x(d_{yx} \wedge d_{xz})$
- **(PS7)** $x \neq y \Rightarrow d_{xy} \wedge c_x(d_{xy} \wedge u) \leq u$
- **(PS8)** $u \neq 0 \Rightarrow \mathrm{dom}(u)$ *finite*
- **(PS9)** $\mathrm{dom}(u) = \{x \in \omega \mid u \leq d_{xx}\}$
- **(PS10)** $\mathrm{dom}(u) = \emptyset \Rightarrow u = 1$
- **(PS11)** $d_{xx} \neq 0$
- **(PS12)** $d_{xy} = d_{yx}$

for all $u, v \in V$ *and* $x, y, z \in \omega$*.*

Proposition 7 ([23, Thms. 1,3])**.** *The conjunctive table algebras over non-empty universes are precisely (up to isomorphism) the projectional semilattices.*

The axioms (**PS0**), ..., (**PS7**) correspond to the axioms (**CA0**),...,(**CA7**) for cylindric algebras. Axiom (**CA0**) asserts a Boolean algebra; since we do not consider disjunction and negation, axiom (**PS0**) only asserts a bounded semilattice. The Axioms (**CA1**), (**CA2**), (**CA3**), (**CA4**) and (**CA6**) are identical to (**PS1**), (**PS2**), (**PS3**), (**PS4**) and (**PS6**), respectively. Cylindric algebra axiom (**CA5**) states $d_{xx} = 1$, reflecting that $x{=}x$ is a tautology; however, the *table semantics* in eq. (1) corresponds to a logic with undefined variables, where $x{=}x$ is not a tautology! We consider (**PS5**) to be a suitable replacement: Under the definition axiom (**PS9**), axiom (**CA5**) asserts $\mathrm{dom}(u) = \omega$ for all $u \neq 0$; whereas axiom (**PS5**) asserts $\mathrm{dom}(u) = \{x \in \omega \mid c_x(u) \neq u\}$ for all $u \neq 0$; the latter set is known as the *dimension set* $\Delta(u)$ in the terminology of cylindric algebras. Axiom (**PS7**) is the historical axiom (**CA7**); the contemporary axiom (**CA7**) is equivalent but involves negation! Historically, there was also an axiom (**CA8**), stating that $\Delta(u)$ is finite for all $u \in V$. Since $\mathrm{dom}(u) = \Delta(u)$ for $u \neq 0$, we can identify (**CA8**) with (**PS8**), disregarding the case $u = 0$.

4.4. Complete Projectional Semilattices

The table algebras $\mathrm{Tab}(G)$ are complete lattices [14, Sect. 3.5]. The join $\bowtie_{i\in I} T_i$ of a family $(T_i)_{i\in I}$ is the empty table if $\bigcup_{i\in I} \mathrm{schema}(T_i)$ is infinite (because no other tables with infinite schema are contained in $\mathrm{Tab}(G)$), and is otherwise defined in the natural way.

A conjunctive table algebra $\mathfrak{A}$ is *complete* if $\bowtie_{i\in I} T_i \in A$ for all families $(T_i)_{i\in I}$ in A. In this section, we provide an axiomatic characterization of complete conjunctive table algebras. Likewise, we say that a projectional semilattice $(V, \wedge, 0, 1, c_x, d_{xy}, \mathrm{dom})_{x,y\in\omega}$ is *complete* if $(V, \leq)$ is a complete lattice.

Proposition 8. *The complete conjunctive table algebras over non-empty universes are precisely (up to isomorphism) the complete projectional semilattices.*

Proof. Trivially, every complete conjunctive table algebra is a complete projectional semilattice. Now let $\mathfrak{A}$ be a complete projectional semilattice. In the proof of [23], an embedding $\mathrm{ext}_\alpha : \mathfrak{A} \to \mathrm{Tab}(G)$ into a full table algebra with non-empty base G is obtained, where $\alpha : \bigcup_{X\in\mathcal{P}_{\mathrm{fin}}(\omega)} G^X \to A$ is a tuple labeling of $\mathfrak{A}$ (cf. [22, Def. 4]), in particular it satisfies $\mathrm{schema}(\alpha(t)) = \mathrm{def}(t)$ and

$$\alpha(t) \cdot \lambda = \alpha(t \circ \lambda) \tag{16}$$

for all $\lambda \in \mathcal{T}_{\mathrm{fp}}(\omega)$. The embedding ext_α is defined by

$$\mathrm{ext}_\alpha(u) := \{t \in G^X \mid \alpha(t) \leq u\} \tag{17}$$

for all $u \in \mathfrak{A}[X]$ and $X \in \mathcal{P}_{\mathrm{fin}}(\omega)$. Our proof amounts to an adaptation of the infimum case in the proof of [22, Thm. 2]. From that paper, we also obtain [22, Prop. 3x)]

$$\alpha(t) \leq u_i \Leftrightarrow \alpha(t) \cdot \pi_{X_i} \leq u_i \ . \tag{18}$$

Now let $(u_i)_{i\in I}$ be a family of elements in $\mathfrak{A}$. We have to show $\mathrm{ext}_\alpha(\bigwedge_{i\in I} u_i) = \bowtie_{i\in I} \mathrm{ext}_\alpha(u_i)$. If $\bigcup_{i\in I} \mathrm{dom}(u_i)$ is infinite, we obtain $\mathrm{ext}_\alpha(\bigwedge_{i\in I} u_i) = \mathrm{ext}_\alpha(0) = \emptyset = \bowtie_{i\in I} \mathrm{ext}_\alpha(u_i)$. Otherwise,

$$\begin{aligned} t \in \mathrm{ext}_\alpha(\bigwedge_{i\in I} u_i) &\underset{(17)}{\Leftrightarrow} \forall i \in I : \alpha(t) \leq u_i \underset{(18)}{\Leftrightarrow} \forall i \in I : \alpha(t) \cdot \pi_{X_i} \leq u_i \\ &\underset{(16)}{\Leftrightarrow} \forall i \in I : \alpha(t|_{X_i}) \leq u_i \underset{(17)}{\Leftrightarrow} \forall i \in I : t|_{X_i} \in \mathrm{ext}_\alpha(u_i) \Leftrightarrow t \in \bowtie_{i\in I} \mathrm{ext}_\alpha(u_i) \ . \end{aligned}$$

□

5. Conjunctive Concept Algebras

For every relational structure $\mathfrak{G}$, the result operation $\mathrm{res}_\mathfrak{G}$ of eq. (2) is part of a Galois connection, from which a concept lattice is obtained in the usual way, cf. [14, Sect. 5][13]. The pair of maps can be stated as

$$\mathrm{res}_\mathfrak{G}(\mathfrak{N}, \nu) := \{t \in G^{\mathrm{def}(\nu)} \mid (\mathfrak{N}, \nu) \lesssim (\mathfrak{G}, t)\} \tag{19}$$

$$\mathrm{info}_{\mathfrak{G}}(T) := \prod_{t \in T} (\mathfrak{G}, t) \tag{20}$$

where $(\mathfrak{N}, \nu) \lesssim (\mathfrak{G}, t) :\Leftrightarrow \exists f\, f : (\mathfrak{N}, \nu) \to (\mathfrak{G}, t)$ denotes the existence of a tableau query homomorphism, and $\prod_{t \in T}(\mathfrak{G}, t)$ is the direct product of tableau queries. A *concept* of $\mathfrak{G}$ is a pair $(T, (\mathfrak{N}, \nu))$ such that $T = \mathrm{res}_{\mathfrak{G}}(\mathfrak{N}, \nu)$ and $(\mathfrak{N}, \nu) = \mathrm{info}_{\mathfrak{G}}(T)$. The table $\mathrm{ext}(T, (\mathfrak{N}, \nu)) := T$ is the concept's *extent*, and the tableau query $\mathrm{int}(T, (\mathfrak{N}, \nu)) := (\mathfrak{N}, \nu)$ is the concept's *intent*. For practical purposes, the intents can be simplified by reduction to connected components and query minimization, cf. [13, Figs. 5,2]. Complexity of intents can be further reduced by pattern projections [14, Sect. 6.2][29], but this amounts to considering an $\bigwedge$-sublattice of $\mathfrak{B}(\mathfrak{G})$. For theoretical purposes, we use eqs. (19) and (20) as they are. The concept lattice of $\mathfrak{G}$ is denoted by $\mathfrak{B}(\mathfrak{G})$. It is a complete lattice; we denote the *infimum* by $\wedge$, the *supremum* by $\vee$, the *top concept* by $\top$ and the *bottom concept* by $\bot$. Every concept of $\mathfrak{B}(\mathfrak{G})$ has a *domain* $\mathrm{dom}(T, (\mathfrak{N}, \nu)) := \mathrm{schema}(T) \subseteq \omega$, and the X-*slice* of $\mathfrak{B}(\mathfrak{G})$ is the set $\mathfrak{B}(\mathfrak{G})[X] := \{C \in \mathfrak{B}(\mathfrak{G}) \mid \mathrm{dom}(C) = X\} \cup \{\bot\}$.

The operations of the DPJR algebra can be lifted to concepts, which results in *orbital concept lattices* [30]. The right multiplication on concepts is defined by $(T, [(\mathfrak{N}, \nu)]) \cdot \lambda := (T \cdot \lambda, [(\mathfrak{N}, \nu \circ \lambda)]) \in \mathfrak{B}(\mathfrak{G})$, where intents are classes of equivalent tableau queries, or their representives (for technical details see [14, Sect. 4.3]). Note that if $C \in \mathfrak{B}(\mathfrak{G})[Y]$ and $\lambda : X \to Y$, then $C \cdot \lambda \in \mathfrak{B}(\mathfrak{G})[X]$. Also in [30], we have introduced *equality concepts* $\mathfrak{E}_{xy}$ for each $(x, y) \in \omega \times \omega$. We now introduce a *deletion operation* del_x on $\mathfrak{B}(\mathfrak{G})$ for every $x \in \omega$, given by $\mathrm{del}_x(C) := C \cdot \pi_{X \setminus \{x\}}$ for $C \in \mathfrak{B}(\mathfrak{G})[X]$. The following definition is inspired by the definition of cylindric set algebras.

Definition 9. *The algebra* $\mathfrak{C}(\mathfrak{G}) := (\mathfrak{B}(\mathfrak{G}), \wedge, \bot, \top, \mathrm{del}_x, \mathfrak{E}_{xy}, \mathrm{dom})_{x,y \in \omega}$ *is the* full conjunctive concept algebra *with base* $\mathfrak{G}$*. A* conjunctive concept algebra *with base* $\mathfrak{G}$ *is a subalgebra of* $\mathfrak{C}(\mathfrak{G})$.

We infer from Prop. 4 that right multiplication is a derived operation on $\mathfrak{C}(\mathfrak{G})$; i.e. the conjunctive concept algebras coincide with the subalgebras of orbital concept lattices. Note that the primitive-positive formulas correspond to the finite tableau queries [14, Sect. 3.2][15]. The subalgebra $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G}) := \{C \in \mathfrak{B}(\mathfrak{G}) \mid \mathrm{ext}(C) \in \mathrm{res}_{\mathfrak{G}}(\mathrm{PP}(M))\}$ consists of the *primitive-positive definable* concepts of $\mathfrak{G}$. The concept algebra $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G})$ is essentially a concept algebra in the sense of Andreká and Németi [31], applied there specifically to cylindric set algebras, and used with other kinds of logic in [32].

By Prop. 5, for each set G, there exists a relational structure $\mathfrak{G}$ such that $\mathbf{Tab}(G) = \mathrm{res}_{\mathfrak{G}}[\mathrm{PP}(M)]$. In other words, $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G})$ is isomorphic to $\mathbf{Tab}(G)$. Then necessarily, we have $\mathfrak{C}_{\mathrm{fin}}(\mathfrak{G}) = \mathfrak{C}(\mathfrak{G})$. So in conclusion, for every set G, there exists a conjunctive concept algebra $\mathfrak{C}(\mathfrak{G})$ that is isomorphic to the table algebra $\mathbf{Tab}(G)$. This means that Props. 7 and 8 translate to concepts:

Proposition 10. *The subalgebras of conjunctive concept algebras (up to isomorphism) are precisely the projectional semilattices.*

Proposition 11. *The complete subalgebras of conjunctive concept algebras (up to isomorphism) are precisely the complete projectional semilattices.*

While we have not arrived at a Basic Theorem, a substantial connection to algebraic logic has been made. The remaining question is whether every complete subalgebra of a concept lattice $\mathfrak{B}(\mathfrak{G})$ is itself isomorphic to a concept lattice. We conjecture that this is the case.

Conjecture 12. *The complete concept algebras (up to isomorphism) are precisely the complete projectional semilattices.*

6. Related Work

Imieliński and Lipski [24] have described a mapping from a relational algebra into a cylindric set algebra, which acts as an embedding under certain assumptions. As Düntsch and Mikulás[33] have pointed out, the table schema is not preserved by this mapping, so this mapping can not be truly considered an embedding. In order to preserve the table schema, they include a new element in the cylindric set algebra, which does not occur in tables. This new element amounts to a value of "undefined", so that the sets in the cylindric set algebra become sets of partial functions.

In this paper, we suggest to take a different route, and adapt the axioms of cylindric algebra to the database-theoretic setting. In his survey paper, Németi [27] presents variants of cylindric algebras, and also discusses the merits of such an approach [27, Sect. 7(4)], citing Howard [34] and Craig [35] as protagonists. However, they work with a different signature, which includes negation/complements, and which supports the unnamed perspective, while we present an axiomatization in the named perspective (cf. [15] for perspectives), which is closer to the original axioms. Variants of cylindric algebras, which are based on other first-order fragments (cylindrification only, cylindrification with union, cylindrification with union and intersection) are presented by Hansen [36].

7. Conclusion

We have characterized conjunctive concept $\wedge$-subalgebras by axioms in the style of cylindric algebras, and have more specifically likened them to cylindric set algebras. This establishes a connection between FCA and algebraic logic in the database-theoretic setting. In addition, we have obtained an axiomatic characterization of conjunctive concept $\bigwedge$-subalgebras. Since $\bigwedge$-sublattices correspond to pattern projections [29], we have thus axiomatized conjunctive pattern concept algebras (to be defined in a suitable way). Moreover, we have conjectured that the conjunctive concept $\bigwedge$-subalgebras are precisely the conjunctive concept algebras (u.t.i.). The results raise the question how concept $\vee$-subalgebras and concept $\bigvee$-subalgebras can be axiomatically characterized. Moreover, while conjunctive concept $\wedge$-subalgebras correspond to cylindric set algebras, is there also a well-motivated counterpart of cylindric algebras in this setting? Finally, we suggest to use *relational concept algebra* as a generic notion, and consider conjunctive concept algebras, as well as their orbital counterpart [30], as special kinds.

Acknowledgments

The authors wish to thank István Németi, who described a related notion of concept algebra in his work, for the feedback, and for pointing us to the article of Ben Hansen [36].

References

[1] B. Ganter, R. Wille, Formal concept analysis: mathematical foundations, Springer, Berlin, 1999.
[2] R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts, in: I. Rival (Ed.), Ordered Sets, Reidel, Dordrecht–Boston, 1982, pp. 445–470.
[3] H. von Hentig, Magier oder Magister? Über die Einheit der Wissenschaft im Verständigungsprozeß, Suhrkamp, Frankfurt, 1974.
[4] R. Wille, Restructuring mathematical logic: an approach based on Peirce's pragmatism, in: A. Ursini, P. Aglianò (Eds.), Logic and Algebra, Marcel Dekker, New York, 1996, pp. 267–281.
[5] I. Kant, Logic, Dover, New York, 1988.
[6] R. Wille, Conceptual graphs and formal concept analysis, in: D. Lukose, H. S. Delugach, M. Keeler, L. Searle, J. F. Sowa (Eds.), Proceedings of ICCS 1997, 5th International Conference on Conceptual Structures, volume 1257 of *LNCS*, Springer, Heidelberg, 1997, pp. 290–303.
[7] J. F. Sowa, Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, MA, 1984.
[8] M. Chein, M.-L. Mugnier, Conceptual graphs: Fundamental notions, Revue d'Intelligence Artificielle 6 (1992) 365–406.
[9] R. Wille, Formal concept analysis and contextual logic, in: P. Hitzler, H. Schärfe (Eds.), Conceptual Structures in Practice, Chapman & Hall/CRC, 2009, pp. 137–173.

[10] M. Huchard, C. Roume, P. Valtchev, When concepts point at other concepts: the case of UML diagram reconstruction, in: Proceedings of the 2nd Workshop on Advances in Formal Concept Analysis for Knowledge Discovery in Databases (FCAKDD 2002), 2002, pp. 32–43.
[11] M. Huchard, M. R. Hacene, C. Roume, P. Valtchev, Relational concept discovery in structured datasets, Annals of Mathematics and Artificial Intelligence 49 (2007) 39–76.
[12] F. Baader, R. Molitor, Building and structuring description logic knowledge bases using least common subsumers and concept analysis, in: B. Ganter, G. W. Mineau (Eds.), Proceedings of ICCS 2000, volume 1867 of *LNCS*, Springer, Berlin, Heidelberg, 2000, pp. 292–305.
[13] J. Kötters, Concept lattices of a relational structure, in: H. D. Pfeiffer, D. I. Ignatov, J. Poelmans, N. Gadiraju (Eds.), Proceedings of ICCS 2013, volume 7735 of *LNCS*, Springer, 2013, pp. 301–310.
[14] J. Kötters, P. W. Eklund, Conjunctive query pattern structures: A relational database model for formal concept analysis, Discrete Applied Mathematics 273 (2020) 144–171.
[15] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[16] A. K. Chandra, P. M. Merlin, Optimal implementation of conjunctive queries in relational databases, in: Proceedings of the ninth annual ACM symposium on Theory of computing, STOC ’77, ACM, New York, NY, USA, 1977, pp. 77–90.
[17] H.-D. Ebbinghaus, J. Flum, W. Thomas, Mathematical Logic, second ed., Springer, New York, 1994.
[18] E. F. Codd, A relational model of data for large shared data banks, Commun. ACM 13 (1970) 377–387.
[19] P. Hell, J. Nešetřil, Graphs and Homomorphisms, Oxford University Press, Oxford,New York, 2004.
[20] J. F. Sowa, Conceptual graphs for a data base interface, IBM Journal of Research and Development 20 (1976) 336–357.
[21] B. Groh, A Contextual-Logic Framework based on Relational Power Context Families, Ph.D. thesis, Griffith University, 2002.
[22] J. Kötters, S. E. Schmidt, Orbital semilattices, 2022. URL: https://arxiv.org/abs/2206.07790. doi:10.48550/ARXIV.2206.07790.
[23] J. Kötters, S. E. Schmidt, Conjunctive table algebras, 2024. URL: https://arxiv.org/abs/2404.01354.
[24] T. Imieliński, W. Lipski, The relational model of data and cylindric algebras, J. Comput. Syst. Sci. 28 (1984) 80–102.
[25] L. Henkin, J. D. Monk, A. Tarski, Cylindric Algebras, Part 1, North-Holland, 1971.
[26] L. Henkin, J. D. Monk, A. Tarski, Cylindric Algebras, Part 2, North-Holland, 1985.
[27] I. Németi, Algebraizations of quantifier logics, an introductory overview, https://old.renyi.hu/pub/algebraic-logic/survey.html, 1997. 12.1th Version.
[28] J. D. Monk, Lectures on cylindric set algebras, in: C. Rauszer (Ed.), Algebraic Methods in Logic and Computer Science, volume 28 of *Banach Center Publications*, Institute of Mathematics, Polish Academy of Sciences, Warsaw, 1993, pp. 253–290.
[29] B. Ganter, S. O. Kuznetsov, Pattern structures and their projections, in: H. S. Delugach, G. Stumme (Eds.), Proceedings of ICCS 2001, volume 2120 of *LNCS*, Springer, 2001, pp. 129–142.
[30] J. Kötters, S. E. Schmidt, Orbital concept lattices, in: T. Braun, D. Cristea, R. Jäschke (Eds.), Proceedings of ICCS 2022, volume 13403 of *LNAI*, Springer, 2022, pp. 66–81.
[31] H. Andréka, I. Németi, How many varieties of cylindric algebras are there, Transactions of the American Mathematical Society 369 (2017) 8903–8937. Http://dx.doi.org/10.1090/tran/7083.
[32] H. Andréka, Z. Gyenis, I. Németi, I. Sain, Universal Algebraic Logic. Dedicated to the Unity of Science., Birkhäuser, 2022.
[33] I. Düntsch, S. Mikulás, Cylindric structures and dependencies in relational databases, Theor. Comput. Sci. 269 (2001) 451–468.
[34] C. M. Howard, An Approach to Algebraic Logic, Ph.D. thesis, University of California, Berkeley, 1965.
[35] W. Craig, Logic in Algebraic Form, North-Holland, Amsterdam, 1974.
[36] B. Hansen, Finitizability questions for some reducts of cylindric algebras, in: Logic Colloquium ’92, CSLI Publications and FOLLI, 1995, pp. 115–134.

When contranominal scales give a solution to the Zarankiewicz problem?

Dmitry I. Ignatov

National Research University Higher School of Economics, Moscow, Russia
dignatov@hse.ru

Abstract. The paper formulates Zarankiewicz problem in terms of formal contexts as follows: What is $z(m, n; s, t)$, the largest size of the incidence relation of a formal context with m objects and n attributes, for which there is no a formal concept with the given extent s and t intent sizes and larger? Exact formulas for the case $n = m$, and $s+t = n+1+k$ with valid ranges of s, t, and k using the contranominal scales of sizes $n-k$ and maximal symmetric contexts are obtained. Moreover symmetric versions of $z_{\lceil n/2 \rceil}(n)$ function are studied and expected ansatz-based solutions as second degree polynomials for $z_{\lfloor n/2 \rfloor}(n)$ are disproven with Formal Concept Analysis assisted tools and concrete lower bounds obtained for $z_5(11), z_6(13), z_7(15), z_8(17)$, and $z_9(19)$.

Keywords: Zarankiewicz problem, maximal biclique, formal concepts, contranominal scale, extremal combinatorics

1 Introduction

The Zarankiewicz problem dates back to 1950s and asks for the maximal number of edges in a bipartite graph of fixed size free of bilcliques with given sizes of its parts [21]. This is an analogue of a famous problem studied by Turan on the maximal size of a graph free of p-clique. The corresponding function $z(m, n; s, t)$ counting the number of edges in a bipartite graph with parts of sizes m and n and no biclique with sizes of components s and t respectively is called the Zarankiewicz function or number and is the subject of ongoing research, while the problem still open in general.

It is interesting that the original problem was published first in terms of grids in French [21]. We take its translation from [20] except the term grid not lattice to avoid confusion with French "trellis" normally used for lattice; moreover in the original Zarankiewicz formulation "un réseau plan formé" was used):

> "Let R_n where $n > 3$ be an $n \times n$ square grid. Find the smallest natural number $k_2(n)$ for which every subset of R_n of size $k_2(n)$ contains 4 points that are all the intersections of 2 rows and 2 columns. More generally, find the smallest natural number $k_j(n)$ for which every subset of R_n of size $k_j(n)$ contains j^2 points that are all the intersections of j rows and j columns."

Note that $k_j(n)$ is $z(n, n; j, j) + 1$.

Such incidence structrues like grids are naturally represented by binary relations, Boolean matrices and formal contexts, while the latter serve for object-attribute incidence representation if Formal Concept Analysis.

Formal Concept Analysis (FCA) is a branch of modern lattice theory and it studies (formal) concepts and their hierarchies [8]. The adjective "formal" indicates a strict mathematical definition of a pair of sets (of objects and attributes, respectively), called, the extent and the intent and named the formal concept as a whole. This formalisation is possible because of the use of the algebraic lattice theory and Galois connections.

So, our goal here is to consider the formualtion of Zarankiewicz problem in terms of FCA and see what this approach and the existing tools can add to the state of the art. Thus, bipartite graphs can be considered as formal contexts, and its maximal bicliques as formal concepts of the context.

Moreover, recent results on extremal lattice theory and Boolean matrix factorisation with FCA show that formal contexts called contranominal scales are of high importance. For example, the work of Albano and Chornomaz [2] answers the question how large is the size of concept lattices when contranominal scales of a certain size are not contained in the input context of a fixed size, while our previous work shows that the state-of-the-art Boolean matrix factorisation algorithms are suboptimal on contranominal scales. This is also a basic fact that a contranominal scale of size $n \times n$ has the largest possible number of formal concepts, 2^n, for the given n, while the concept extent sizes run through all $\{1, 2, \ldots, n\} = [n]$.

The paper is organised as follows. Section 2 gives basics of FCA theory. Section 3, formulates the studied problem for $z(n, m; s, t)$ in FCA terms. Section 4 presents obtained theoretical results including fully symmetric case for the Zarankiewicz function like $z(n, n; \lfloor n \rfloor)$. Section 5 briefly overviews the most relevant works. Section 6 concludes the paper.

2 FCA Basics

We mainly follow notation from [8].

Definition 1. Formal context $\mathbb{K}$ *is a triple* (G, M, I) *where* G *is a set of* objects*,* M *is a set of* attributes*, and* $I \subseteq G \times M$ *is an incidence binary relation.*

The binary relation I is interpreted as follows: for $g \in G$, $m \in M$ we write gIm if the object g has the attribute m.

For a formal context $\mathbb{K} = (G, M, I)$ and any $A \subseteq G$ and $B \subseteq M$ a pair of mappings is defined:

$$A^{\uparrow} = \{m \in M \mid gIm \text{ for all } g \in A\}, \quad B^{\downarrow} = \{g \in G \mid gIm \text{ for all } m \in B\},$$

these mappings define Galois connection between partially ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$ on disjunctive union of G and M. The set A is called *closed set*, if $A^{\uparrow\downarrow} = A$ [5].

Definition 2. *A* formal concept *of the formal context* $\mathbb{K} = (G, M, I)$ *is a pair* (A, B)*, where* $A \subseteq G$*,* $B \subseteq M$*,* $A^{\uparrow} = B$ *and* $B^{\downarrow} = A$*. The set* A *is called the* extent*, and* B *is the* intent *of the formal concept* (A, B)*.*

It is evident that the extent and intent of any formal concept are closed sets.

The set of all formal concepts of a context $\mathbb{K}$ is denoted by $\mathfrak{B}(G, M, I)$. This set forms an algebraic lattice call concept lattice where the concepts are ordered via set inclusion of their extents (dually intents).

Note that $(.)^{\uparrow}$ and $(.)^{\downarrow}$ derivation operators are usually unified by a single symbol like prime $(.)'$ or $(.)^{I}$, when formal contexts with different incidence relations, say I and J are used simultaneously.

For every set S the *contranominal scale* is defined as $\mathbb{N}^c_S = (S, S, \neq)$. In what follows, we consider $\mathbb{N}^c_n$ with $S = [n] = \{1, \ldots, n\}$ without loss of generality.

The surveys on advances in FCA theory and its applications can be found in [17, 18].

3 Problem Statement

We propose the following most general formulation of the Zarankiewiecz problem.

Problem 1. What is $z(m, n; s, t)$, the largest size of the incidence relation I of a formal context $\mathbb{K} = (G, M, I)$ with $|G| = m$ and $|M| = n$, for which there is no a formal concept (A, B) with $|A| \geq s$ and $|B| \geq t$?

We need these inequalities in the formulation, $|A| \geq s$ and $|B| \geq t$, since there might be a concept of size $(s + 1) \times t = |A||B|$ but not of $s \times t$ containing the subcontext of sizes $s \times t$ due to maximality of concepts in terms of the number of objects and attributes (cf. maximal bicliques).

4 Results

4.1 Contranominal Scales and Maximal Symmetric Contexts

Lemma 1. *([10, 20]) Let* $\mathbb{K} = (G, M, I \subseteq G \times M)$ *with* $G = \{g_1, \ldots, g_n\}$ *and* $M = \{m_1, \ldots, m_n\}$*, then this context does not contain a concept with extent and intent sizes* p *and* q *or larger, respectively, if*

$$\sum_{i=1}^{n} \binom{|m_i'|}{p} \leq (q-1)\binom{n}{p}. \tag{1}$$

Lemma 1 is the instantiation of the pingeonhole principle [1], where the pigeons are subsets of attributes' extents of size p, while the holes are subsets of objects of the same size.

Lemma 2. *([8]) For each concept* (A, B) *of the contranominal scale of size* n*,* $([n], [n], \neq)$*,* $|A| + |B| = n$*.*

Property 1. $z(n; n+1, q) = n^2$ for any $q > 0$.

Form Lemma 2 we infer Property 2.

Property 2. If a formal context $\mathbb{K} = (G, M, I)$ contains as its subcontext a contranominal scale of size k, $\mathbb{N}^c_k$, then $\mathbb{K}$ should contain a formal concept (A, B) with $|A| \geq p$ and $|B| \geq q$ where $p + q = k$.

Theorem 1. *A contranominal scale of size n, $([n], [n], \neq)$, gives a solution to Zarankiewicz problem with $m = n$, $p + q = n$ for $p, q > 0$, and $z(n; p+1, q) = z(n; p, q+1) = n(n-1)$.*

Proof. 1) Admissibility. By Lemma 1 we should have

$$\sum_{i=1}^{n} \binom{n-1}{p+1} \leq (q-1)\binom{n}{p+1}.$$

Or

$$(n-1-p)\binom{n}{p+1} \leq (q-1)\binom{n}{p+1},$$

$$n - 1 - p \leq q - 1.$$

We substitute $n - p = q$ by the condition and get the identity $q - 1 \leq q - 1$.

One can also show that our context ifs free of any concept (A, B) with $|A| = p+1$, $|B| = q$ and $|A| = p$, $|B| = q+1$. By Lemma 2 $|A|+|B| = n$, which implies $p + q + 1 = n$, the contradiction.

2) Maximality. Then let us also show that the contranominal scale is the maximal context in terms of its number of incident object-attribute pairs.

Assume that we can add one more object-attribute pair, say (g_n, m_n) to the contranominal scale. Then our context will contain one full row, full column and a contranominal scale of size $n - 1$ as subscontext disjoint from these full row and column. Since full rows and columns are both reducible, then the resulting context give rise the same number of concepts that the contranominal scale of size $n - 1$. For each concept (A, B) of the $\mathbb{N}^c_n$ with $|A| = p$ and $|B| = q$, either $g_n \in A$ ($m_n \notin B$) or $m_n \in B$ ($g_n \notin A$), so the concept of the new context I, (A^{II}, A^I) or (B^I, B^{II}), will have extent and intent sizes $p, q+1$ or $p+1, q$, respectively. Due to the context symmetry, cases $p, q+1$ and $p+1, q$ are both realised for each p, q pair.

Similarly, for Lemma 1, its inequality becomes false.

□

Can we also use contranominal scales for other types of solutions? The answer is yes. Thus one can place new object-attribute pair on the main diagonal of a contranominal scale.

Theorem 2. *A context $\mathbb{K}$ obtained from a contranominal scale of size n as $\mathbb{K} = ([n], [n], \neq \cup (i, i))$ for $i \in [n]$ gives a solution to Zarankiewicz problem with $m = n$, $p + q = n - 1$ for $p, q > 0$, and $z(n; p+2, q+1) = z(n; p+1, q+2) = n(n-1) + 1$.*

Proof. 1) Admissibility. By Lemma 1 the following inequality should hold

$$\binom{n}{p+2} + \sum_{i=1}^{n-1} \binom{n-1}{p+2} \leq q\binom{n}{p+2}.$$

After simplification we get the inequality $\frac{n-1}{n}(q-1) \leq q-1$.

2) Let us check maximality by adding a new pair, say (g_{n-1}, m_{n-1}). First, we should note that our context contains contranominal scale isomorphic to $([n-1],[n-1],\neq)$. So, the sum of sizes of its concept's extent and intent is $n-1 = p+q$, but since we have extra full row and column, each concept of the considered context will have extent and intent size $p+1$ and $q+1$. It implies that there is no any concept with sizes of extent $p+2$ and intent $q+1$ (same for $p+2$ and $q+1$).

□

Can we generalise this solution up to k added pairs to the contranominal scale of size n? Again, the answer is yes.

Theorem 3. *A context $\mathbb{K}$ obtained from a contranominal scale of size n as*

$$\mathbb{K} = ([n],[n],\neq \cup \bigcup_{i\in S}(i,i))$$

for $S \in \binom{[n]}{k}$ gives a solution to Zarankiewicz problem with $m = n$, $p+q = n-k$, $0 < k \leq n$ for
1) $p, q > 0$ (or $p = q = 0$), and

$$z(n; p+1+k, q+k) = z(n; p+k, q+1+k) = n(n-1)+k,$$

2) $q = 0, p > 0$,

$$z(n; p+1+k, q+k) = n^2$$

and

$$z(n; p+k, q+1+k) = n(n-1)+k,$$

3) $p = 0, q > 0$,

$$z(n; p+1+k, q+k) = n(n-1)+k$$

and

$$z(n; p+k, q+1+k) = n^2.$$

Proof. The proof of Admissibility is similar.

$$k\binom{n}{p+k+1} + (n-k)\binom{n-1}{p+k+1} \leq (q+k-1)\binom{n}{p+k+1}.$$

We get inequality $\frac{n-k}{n}(q-1) \leq q-1$, which is true for $q > 0$. For $q = 0$ the inequality is false.

The solution for 2) $q = 0$ falls into two basic cases (up to the symmetry p and q). When $p = 0$, $n = k$ and by Property 1 $z(n; n+1, n) = z(n; n, n+1) = n^2$, which coincides with $n(n-1) + k$. When $p > 0$, $p + k = n$, and

$$z(n; n+1, k) = n^2$$

(by Property 1) and

$$z(n; n, 1+k) = n(n-1) + k$$

(has to be proven).

The last subcase admissible by noting that any new object-attribute pair on the diagonal will result in the full column and we get the concept of size $n \times (1+k)$. And the contranominal scale of size $n-k$ cannot be replaced by any other subcontext within the context region of size $n - k \times n$ with the same number of incident pairs since placing at least two missing pairs in one row gives rise to a full column but placing all $n-k$ into distinct rows results in presence of the contranominal scale of the same size. Case 3) is similar.

The maximality condition for 1) is proven similarly to Theorem 2.

□

Corollary 1. *A context* $([n], [n], = \setminus \bigcup\limits_{i \in S}(i,i))$ *with* $S \in \binom{[n]}{k}$ *has the maximal number of incident pairs being free from the concept with extent and intent sizes* n *and* $k+1$ *(symmetrically,* $k+1$ *and* n*), respectively.*

Remark 1. Note that $z(n; p+1+k, q+k)$ and $z(n; p+k, q+1+k)$ can be recast as $z(n; p+1+k, n-p)$ and $z(n; p+k, n-p-1)$.

4.2 Single variable Zarankiewicz function

In earlier works, Zarankiewicz, Guy [10] and others paid a lot attention to the function $z_a(n) = z(n, n; a, a)$.

An interesting question would be what the obtained results can do for this case.

For an odd n, i.e. $n = 2t - 1$ for $t > 0$, we get

$$z(2t-1; t, t) = 2(2t-1)(t-1),$$

but we cannot tackle the even case since $p+1$ and q have different parity when $p + q = 2t$ and cannot be equal. However, by Theorem 1.3 from [4] we have

$$z(2t; t, t) = 4t^2 - 3t - 1 = 2(2t - 1/2)(t-1).$$

We combine these two results into a single formula as follows:

$$z(n; \lceil n/2 \rceil, \lceil n/2 \rceil) = 2\Big(2t - \frac{1}{4} + (-1)^n \frac{3}{4}\Big)\Big(t - 1\Big),$$

here $t = \lceil n/2 \rceil$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\lceil n/2 \rceil$	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10
$z_{\lceil \frac{n}{2} \rceil}(n)$	0	0	6	9	20	26	42	51	72	84	110	125	156	174	210	231	272	296	342	369

Or

$$z_{\lceil \frac{n}{2} \rceil}(n) = 4\lceil n/2 \rceil^2 - \frac{9}{2}\lceil n/2 \rceil + \frac{1}{2} + (-1)^n\Big(\frac{3}{2}\lceil n/2 \rceil - \frac{3}{2}\Big).$$

This upper symmetrisation for odd $n = 2t-1$ is possible with contranominal scales, however they do not work in $z(n; \lfloor n \rfloor, \lfloor n \rfloor) = z(2t-1; t-1, t-1)$, since $\lfloor n \rfloor = t-1$ and $p = t-2$ and $q = t-1$ violates $p + q = n$. Similarly, for $n = 2t+1$ we deal with $z(2t+1; t, t)$ and have $p+q = t-1+t = 2t-1 \neq 2t+1$, the violation. This case is also beyond of reach for Theorem 3 (since $p + q = (t-1-k)+(t-k) \neq n-k$), Theorems 1.2 ($n = 2t+1$ implies the contradiction $2t + 1 \leq 2t - 1$)) and 1.3 (valid for even cases) from [4].

We know from the literature [20] and OEIS that $z_2(5) = 12$, $z_3(7) = 33$, and $z_4(9) = 61$. This is enough to find coefficients of $z(2t + 1; t, t)$ as a quadratic polynomial $at^2 + bt + c$. Case $n = 3$ is omitted resulting in zero pairs by the definition.

The system

$$\begin{cases} 4a + 2b + c = 12 \\ 9a + 3b + c = 33 \\ 16a + 4b + c = 61. \end{cases}$$

results in $P_{2t+1}(t) = \frac{7}{2}t^2 + \frac{7}{2}t - 9 = \frac{7}{2}t(t+1) - 9$ as a candidate for $z(2t+1; t, t)$.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\lfloor n/2 \rfloor$	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10
$z_{\lfloor \frac{n}{2} \rfloor}(n)$	0	0	9	12	26	33	51	61	84	?	125	?	174	?	231	?	296	?	369

If our ansatz based on the fact that $z(n, n; p, q)$ is totally bounded by n^2 and all possible variables enters linearly or quadratically is correct, we should obtain the next value for $z_5(11)$ as 96.

In reality, the following contexts for $n = 11$ in Figure 1 and 2 were obtained ad hoc based on the usage of contranominal scales as building blocks and validated with our implementations of CbO [15] and NextClosure [8] and cross-checked with concept generation algorithm In-Close [3], by adding extra crosses (pairs) to the context and checking absence of concepts larger than or equal to 5×5 and 6×6 full subcontexts, respectively, in terms of extent times intent sizes.

So, the knowledge base on the behaviour of $z_{2t+1}(t)$ is updated. At least, it is not that regular to be described by the same polynomial of degree 2, $P_{2t+1}(t)$, for the range $t \geq 2$.

But what if we still have doubts, especially, since $z(2t-1, t, t) = 2(2t-1)(t-1)$ and its even $n = 2t$ counterpart has roots at $t = 1$. We can consider another

$\mathbb{K}_{z_5(11)\geq 97}$	1	2	3	4	5	6	7	8	9	10	11
1	×	×				×	×	×	×	×	×
2		×			×	×	×	×	×	×	×
3		×	×	×	×	×	×	×		×	×
4	×		×	×	×	×	×	×	×	×	×
5	×	×		×	×	×		×	×	×	×
6	×	×	×		×		×	×	×	×	×
7	×	×	×	×		×	×	×	×	×	
8	×	×	×	×	×		×	×	×		
9	×	×	×	×	×	×		×	×		
10	×	×	×	×	×	×	×		×	×	×
11	×	×	×	×	×	×	×	×			×

Fig. 1. A formal context for the obtained lower bound $z_5(11) \geq 97$

quadratic polynomial $Q_{2t+1}(t) = \frac{9}{2}t^2 - \frac{3}{2}t - 3 = \frac{9}{2}(t + \frac{2}{3})(t - 1)$. Then for $n = 9$, $t = 4$ we have $Q_{2t+1}(4) = 63$ but it contradicts previous knowledge $z_9(4) = 61$ ([20], OEIS sequence for $k_n(4) = z_n(4)+1$ is A006616[1]). Q starts to overestimate z at t=4, while P underestimates z first at $t = 11$, the lowest upper bound for $z_4(9)$ from the best known ones is by Roman [19] (the bound by Nikiforov [16] give higher values) is 64, while for $z_5(11)$ it is 102, and 148 for $z_6(13)$ (smaller than $Q_{2t+1}(13)$).

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\lfloor n/2 \rfloor$	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10
$Q_{2t+1}(t)$		0		12		33		**63**		102		150		207		273		348	
$z_{\lfloor \frac{n}{2} \rfloor}(n)$	0	0	9	12	26	33	51	61	84	$\geq$97	125	$\geq$142	174	$\geq$ 192	231	$\geq$253	296	$\geq$320	369
$P_{2t+1}(t)$				12		33		61		**96**		138		187		243		306	

All the contexts and codes are placed in Dropbox for reviewing purposes[2].

4.3 Reality Checks

Let us have a look at the function behaviour for some small n, for example, 5. The rows and columns of Table 1 with $s = 1$ or $t = 1$ are filled by the conditions that every object (or attribute) should have (be shared by) $t-1$ attributes $s-1$ (objects).

Assembling Theorems 1, 2, 3 altogether, we have $p + q = n - k$ and $p + 1 + k + q + k = s + t$. For $t = s$ in the region $s + t = n + k + 1$, $0 \leq k \leq n - 1$, if we substitute k in $n(n-1)+k$, we get $z(n; t, t) = n^2 - 2n + 2t - 1$ for $n+1 \leq 2t \leq 2n$.

[1] https://oeis.org/a006616

[2] https://www.dropbox.com/scl/fo/z7k82pyu3xwmnu4cun884/AOLV3EAueoUfy7jM-D1Gm0c?rlkey=vs6hbn6i2vn2e6ncnim8qg0b2&dl=0

$\mathbb{K}_{z_6(13)\geq 142}$	1	2	3	4	5	6	7	8	9	10	11	12	13
1	×	×				×	×	×	×	×	×	×	×
2		×			×	×	×	×	×	×	×	×	×
3		×	×	×	×	×	×	×	×	×		×	×
4	×		×	×	×	×	×	×	×	×	×	×	×
5	×	×		×	×	×	×	×		×	×	×	×
6	×	×	×		×	×	×		×	×	×	×	×
7	×	×	×	×		×		×	×	×	×	×	×
8	×	×	×	×	×		×	×	×	×	×	×	×
9	×	×	×	×	×	×		×	×	×	×	×	
10	×	×	×	×	×	×	×		×	×	×		
11	×	×	×	×	×	×	×	×		×	×		
12	×	×	×	×	×	×	×	×	×		×	×	×
13	×	×	×	×	×	×	×	×	×	×			×

Fig. 2. A formal context for the obtained lower bound $z_6(13) \geq 142$

Actually, within that region (on and below the backward diagonal), in axes n, z when t is fixed, we deal with the parabola, while on the level $n = const$ we have the family of disjoint lines.

Table 1. $z(5, 5, s, t)$; the numbers given in the literature are italic, while obtained by our formulas are below the stepwise line on the diagonal (and also in bold if present in the referenced literature, e.g., in [11])

s, t	1	2	3	4	5
1	0	5	10	15	20
2	5	*12*	*16*	***20***	21
3	10	*16*	***20***	***21***	22
4	15	***20***	***21***	***22***	23
5	20	21	22	23	24

s, t	1	2	3	4	5	6
1	0	6	12	18	24	30
2	6	*16*	*22*	*25*	30	31
3	12	*22*	*26*	***30***	31	32
4	18	*25*	***30***	***31***	32	33
5	24	30	31	32	33	34
6	30	31	32	33	34	35

What if we would to see a solution for a certain non-trivial value of z from those tables above the stepwise line? We can check a suitable context of a given size with $|I| = z(n; s, t)$.

For example, take the value $z(6; 2, 4) = z(6; 4, 2) = 25$ (see OEIS sequence A006614[3]).

Thus, starting with a contranominal scale of size $5 = 4 - 1 + 2$, we have found the non-extensible (by adding new crosses) context $\mathbb{K}_{z(6;4,2)}$ shown in Figure 3. Its concept lattice diagram shows that there are no concepts of sizes 4×2 or (2×4).

[3] `https://oeis.org/006614`

$\mathbb{K}_{z(6;4,2)}$	a	b	c	d	e	f
1			×	×	×	×
2		×		×	×	×
3	×		×		×	×
4	×	×		×		×
5	×	×	×			×
6	×	×	×	×	×	

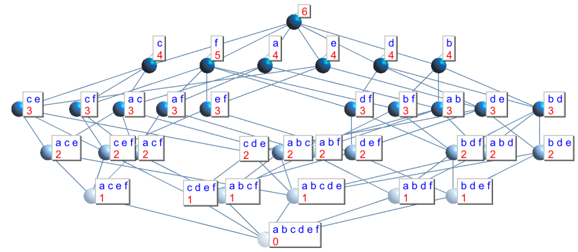


Fig. 3. A formal context for $z(6; 4, 2) = 25$ and its concept lattice diagram with labeling by extent size and full intent

Actually, if we are given s and t and an examined context, we should care that contranominal scales of size $s + t$ are not contained in the given context, while scales of size $s + t - 1$ can be used as biulding blocks, allocated and modified.

5 Related Work

The most relevant for our studies are the works by Balbuena et al. [4] and Tan [20]. The work of Tan [20] demonstrates how to obtain not only values but also possible solutions to the first several dozen values of n for $z_a(n)$ and $a \in \{2, 3, 4\}$ with SAT solvers. In [4] more general cases for $z(m, n; s, t)$ are considered under $max\{m, n\} \leq s + t - 1$ and $z(m, n; t, t)$ if $2t \leq n \leq 3t - 1$; the exact formulas obtained. The authors used matchings to subtract them from the considered bipartite graphs and obtain the solutions and claimed formulas. Our theorems are in accordance with their results where the scopes of the theorems overlap for $m = n$. They also rely on [9], where the so called half-half case was considered with $z(2s, 2t; s, t)$, which is not applicable for cases with odd n.

We partially reproduce Theorem 1.2 and Theorem 1.3 from [4] since we rely on them in Subsection 4.2.

Theorem 4. *(A part of Theorem 1.2 [4]) Let m, n, s, t be integers with $2 \leq s < m$, $2 \leq t < n$ and such that $\max\{m, n\} \leq s + t - 1$. Then*

$$z(m, n; s, t) = mn - (m + n - s - t + 1).$$

Theorem 5. *(A part of Theorem 1.3 [4]) Theorem 1.3. Let m, t be integers such that $2 \leq t \leq m \leq 2t$. Then*

$$z(m, 2t; t, t) = m \cdot 2t - (2m - t + 1).$$

There is also Theorem 1.4 but it forbids $t = n/2$.

A large fraction of past and recent works devoted to various inequalities [19] and asymptotic studies [16, 6] whose estimates are usually overly high for rather small n like 11 or not enough general by considering special cases for $z_a(n)$ with small a like 2, 3 or with rather complex a being a polynomial, or cases when ratio of m and n in $z(m, n; s, t)$ is rather high up to some binomial coefficient including m or n to sample out.

6 Conclusion

One can see that FCA as a theory and as an analytical tool can help to study combinatorial mathematical problems, which is in line with works of B. Ganter [7] on integer partiton lattices, work C. Jakel on the ninth Dedekind number [14] and on our previous works on (maximal) antichains enumeration in Boolean and partitions lattices [12] and symmetric contexts and maximal independent sets for the cover graph of a Boolean cube [13]. We hope it can also do both in other cases when we deal with Boolean matrices or ordered structures, to provide theoretical keys to enumeration and counting problems and help to compute missing numbers, which may lead to interesting conjectures and theorems. Last but not least it contributes to the inventory of Experimental Mathematics as AI-assisted tool.

Acknowledgments I would like to thank my close relatives, Anastasia, Galina, Igor, Maria, Nadezda, Vera and Zamira for their patience and care. I am also thankful to OEIS editors and maintainers for the timely available resource with relevant sequences and references.

References

1. Martin Aigner and Günter M. Ziegler. *Pigeon-hole and double counting*, pages 195–205. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.
2. Alexandre Albano and Bogdan Chornomaz. Why concept lattices are large: extremal theory for generators, concepts, and VC-dimension. *Int. J. Gen. Syst.*, 46(5):440–457, 2017.
3. Simon Andrews. In-close2, a high performance formal concept miner. In *Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK, July 25-29, 2011. Proceedings*, pages 50–62, 2011.
4. C. Balbuena, P. García-Vázquez, X. Marcote, and J.C. Valenzuela. New results on the Zarankiewicz problem. *Discrete Mathematics*, 307(17):2322–2327, 2007.
5. Garrett Birkhoff. *Lattice Theory.* Harvard University, Cambridge, MA, eleventh printing edition, 2011.
6. Guangzhou Chen, Daniel Horsley, and Adam Mammoliti. Zarankiewicz numbers near the triple system threshold. *Journal of Combinatorial Designs*, 32(9):556–576, 2024.
7. Bernhard Ganter. Notes on integer partitions. *International Journal of Approximate Reasoning*, 142:31–40, 2022.
8. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer, Berlin/Heidelberg, 1999.

9. Jerrold R. Griggs and Chih-Chang Ho. On the half–half case of the Zarankiewicz problem. *Discrete Mathematics*, 249(1):95–104, 2002. Combinatorics, Graph Theory, and Computing.
10. Richard K. Guy. A problem of Zarankiewicz. Number 12. The University of Calgary, Department of Mathematics, 1967.
11. Richard K. Guy. A many-facetted problem of zarankiewicz. In G. Chartrand and S. F. Kapoor, editors, *The Many Facets of Graph Theory*, pages 129–148, Berlin, Heidelberg, 1969. Springer Berlin Heidelberg.
12. Dmitry I. Ignatov. A note on the number of (maximal) antichains in the lattice of set partitions. In Manuel Ojeda-Aciego, Kai Sauerwald, and Robert Jäschke, editors, *Graph-Based Representation and Reasoning*, pages 56–69, Cham, 2023. Springer Nature Switzerland.
13. Dmitry I. Ignatov. On the maximal independence polynomial of the covering graph of the hypercube up to n=6. In Dominik Dürrschnabel and Domingo López Rodríguez, editors, *Formal Concept Analysis*, pages 152–165, Cham, 2023. Springer Nature Switzerland.
14. Christian Jäkel. A computation of the ninth Dedekind Number, 2023.
15. Sergei O Kuznetsov. Mathematical aspects of concept analysis. *Journal of Mathematical Sciences*, 80(2):1654–1698, 1996.
16. Vladimir Nikiforov. A contribution to the zarankiewicz problem. *Linear Algebra and its Applications*, 432(6):1405–1411, 2010.
17. Jonas Poelmans, Dmitry I. Ignatov, Sergei O. Kuznetsov, and Guido Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert Syst. Appl.*, 40(16):6538–6560, 2013.
18. Jonas Poelmans, Sergei O. Kuznetsov, Dmitry I. Ignatov, and Guido Dedene. Formal concept analysis in knowledge processing: A survey on models and techniques. *Expert Syst. Appl.*, 40(16):6601–6623, 2013.
19. Steven Roman. A problem of Zarankiewicz. *Journal of Combinatorial Theory, Series A*, 18(2):187–198, 1975.
20. Jeremy Tan. An attack on Zarankiewicz's problem through SAT solving. *arXiv e-prints*, pages arXiv–2203, 2022.
21. K. Zarankiewicz. Problem p 101. In *Colloq. Math.*, number 2, page 301, 1951.

Training Neural Networks Based on Formal Concepts

Sergei O. Kuznetsov[1,*,†], Mariia Zueva[2,†]

[1]HSE University, 20 Myasnitskaya St, Moscow, 101000, Russian Federation

Abstract

This paper presents a modernization of the neural network architecture based on concept lattices, *FCA-CLNet*, utilizing pre-clustering of data based on groups of attributes, unified by a shared interpretable meaning. This approach aims to create a compact model for data classification, with the added benefit of enabling subsequent interpretation of results in scenarios involving a large number of data features.

Keywords

Neural Networks, Clustering, Formal Concept Analysis

1. Introduction

Interpretability in the context of neural networks is an important aspect of research, as it allows us to understand how and why the model makes certain decisions. In recent years, interpretable neural networks have been actively researched and developed in order to overcome the problem of the "black box" and ensure the clarity and explainability of the decision-making process. This is especially important in areas where the decisions made by the model have a significant impact on people's lives and well-being, such as medicine, finance and justice. Finding a balance between the high performance of the model and its interpretability is a key factor for creating reliable and transparent systems capable of interacting with people in confidence.

With the growing demand for AI explainability, many papers addressed the problem of explaining «black box» systems and simultaneously tried to formulate the criteria and measures for evaluating explainability of the model design. In [1] the authors suggested using three core criteria for evaluating machine learning models, namely, interpretability, transparency and explainability. In [2] it was proposed to use expert opinions combined with statistical methods to measure the effectiveness of machine learning models. A first attempt in making a theory of interpretable neural networks (INNs) seems to be made in [3]. The authors managed to align the sparse coding method with existing neural network's architecture, so that the system had the interpretability of the model-based method and the efficiency of the learning-based one.

A series of works have intended to review and classify all existing interpretable methods. In [4] the authors have classified existing interpretable approaches by problem addressed, black-box type and explanation provided, with the purpose to help researchers solve the needed tasks. In [5] the authors suggested to divide interpretable neural network approaches into two types, model decomposition neural networks and semantic interpretable neural networks (INNs). The first one unites methods which inherit domain theoretical knowledge and implement it in the neural network architecture. The decomposition alternative INN starts by taking a complicated mathematical or physical model and breaking it down into smaller, manageable modules. After that it maps the computing of the obtained modules in accordance with the prior knowledge with hyper-parameters of neural network or its hidden layers, thus enhancing their interpretability [6, 7]. The idea can be described as using controllable artificial parameters and structures of neural network instead of the weights without mathematical and physical meaning. This approach requires a theoretical model of the domain. An illustration of this concept is the utilization of

FCA4AI 2024, Santiago de Compostela, Spain

*Corresponding author.

†These authors contributed equally.

skuznetsov@hse.ru (S. O. Kuznetsov); m.zueva@hse.ru (M. Zueva)

0000-0003-3284-9001 (S. O. Kuznetsov); 0009-0000-6332-9936 (M. Zueva)

convex or non-convex optimization algorithms to address mathematical modeling challenges, providing a framework for shaping the objective function. This method is applicable to such tasks as solving partial differential equations (PDE) [8], image deblurring, super-resolution, and other problems [9, 10, 11]. The second approach is semantic INNs [12], which is meant to explain the model's decision afterwards, with the process close to human semantic interpretation. The authors highlighted three different branches in this approach, namely, convolution neural network (CNN) visualization [6], decision tree regularization [13], and semantic knowledge graph [14]. In [15] the taxonomy of interpretable methods was proposed. This paper categorized existing architecture designs by three criteria, namely, the type of engagement (passive or active), the type of explanation and the focus, varying from local to global interpretability and provided the way how to order them in subcategories. The first architecture of neural networks based on the formal concept analysis (FCA) approach was proposed in [16]. In this work the authors propose building neural networks based on concept lattices and on lattices coming from monotone Galois connections. Later, in 2022 in [17] the authors integrated conceptual information into the message passing through graph neural networks (GNNs). The authors of [18] proposed an approach using BERT, which can learn more information from the maximal bi-cliques, which correspond to formal concepts, and use them to make link prediction.

This paper explores the potential of incorporating clustering methods into a compact neural network architecture. Specifically, it introduces a modernization of the neural network framework, *FCA-CLNet*, which leverages concept lattices in conjunction with pre-clustering of data based on semantic attribute groups. The proposed approach is particularly suited for scenarios involving a large number of data features and aims to improve the interpretability of the model's performance.

2. Clustering

Clustering is a widely used useful tool for working with big data and data mining. A large number of clustering algorithms have been developed, each of which has its own area of application. A number of works have been devoted to creating a taxonomy of clustering methods. In [19] authors proposed a categorization framework to classify existing clustering algorithms into groups. They divided all algorithms into partitioning-based, hierarchical-based, density-based, grid-based and model-based:

Partitioning-based algorithms [20] first set the clusters from initial data and then redistribute the data points towards better group organization. Widely used K-means algorithm belongs to this category.

Hierarchical-based methods are intended to organize data hierarchically, based on the medium of proximity. Using these methods datasets can be represented by dendrograms, where each leaf node corresponds to an individual data point. Hierarchical-based methods are divided into agglomerative (bottom-up) and divisive (top-down) approaches [21]. In the former the process starts with clusters containing one object, and then they are united together towards more suited. In the latter the whole dataset is one cluster at the beginning and then it is recursively split into smaller ones till reaching the stopping criterion.

Grid-based methods are based on splitting the data space on grids and accumulating grid-data. The advantages of this approach are fast processing time and independence of the number of data objects.

In model-based methods [22] it is supposed that there is a mixture of probability distributions that generate the given data, so these approaches try to accommodate the data to the predefined mathematical model. These approaches are divided into statistical and neural network approaches.

In this paper, we chose four well-known clustering algorithms for preclustering the data features: K-means, Mean-Shift, DBSCAN and HDBSCAN.

3. Formal Concept Analysis

In FCA-CLNet architecture we operate with the terms related to formal concepts analysis (FCA). Let us recall some basic definitions of FCA [23]. The basic FCA structure is a binary datatable, called formal context, where rows stay for the set of objects, denoted by G, the columns stay for the set of attributes, denoted M and binary relation $I \subseteq G \times M$ is defined in the way so that $(g, m) \in I$ if the object g possesses the attribute m. The triple $K = (G, M, I)$ is called a *formal context*. Derivation operators $(\cdot)'$ for $A \subseteq G$, $B \subseteq M$ are defined as follows:

$$A' = \{m \in M \mid gIm \; for \; all \; g \in A\}, \tag{1}$$

$$B' = \{g \in G \mid gIm \; for \; all \; m \in B\}, \tag{2}$$

These derivation operators form *(antimonotone) Galois connection* on the ordered powersets $(2^G, \subseteq)$ and $(2^M, \subseteq)$.

we define a *classical formal concept of a formal context* K as a pair (A, B) such that $A \in G$, $B \in M$, $A' = B$, $B' = A$. Here A is called an *extent* and B is called an *intent* of the formal concept (A, B).Classical formal concepts are ordered by the relation $\geq$:

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2, \tag{3}$$

which defines a complete (algebraic) lattice on the set of concepts called *concept lattice* $L = (G, M, I)$. The covering relation corresponding to the partial order $\leq$, (if it exists) is defined as $\prec$:

$$(A_1, B_1) \prec (A_2, B_2) \iff (A_1, B_1) \leq (A_2, B_2) \tag{4}$$

and there is no concept (A_3, B_3) such that $(A_1, B_1) < (A_3, B_3) < (A_2, B_2)$.

Classical formal concepts are also called antimonotone formal concepts or formal concepts based on antimonotone Galois connection.

In our study we use another type of formal concepts called formal concepts based on *monotone Galois connection* or monotone formal concepts [24]. They are defined as pairs (A, B), which satisfy monotone Galois connection, that is

$$A^{\vee} = \{b \mid \nexists a \in G \backslash A \text{ such that } aIb\}, \tag{5}$$

$$B^{\wedge} = \{a \mid \exists b \in B \text{ such that } aIb\}, \tag{6}$$

where $A \subseteq G, B \subseteq M$ and $A = B^{\vee}, B = A^{\wedge}$.

In other words, for each set of objects A, we match all the attributes belonging only to objects from A'. On the other hand, the set of attributes B corresponds to the set of all objects B' satisfying at least one attribute from B. A and B are also called an *extent* and an *intent* of the formal concept.

A partial order on the set of all monotone formal concepts is defined as:

$$(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subset A_2 \leftrightarrow B_1 \subset B_2. \tag{7}$$

.

We also can define *monotone concept lattice* based on this partial order.

All monotone formal concepts can be obtained from the given formal context $K = (G, M, I)$ by finding its complement context $\bar{K} = (G, M, \bar{I})$ and then finding all its classical formal concepts.

4. FCA-CLNet

The proposed method utilizes a neural network architecture based on concept lattices. The idea of this neural network was proposed in [16]. This article extends the approach by incorporating an additional step, namely data pre-clustering, to derive novel features.

The method description is as follows:

Suppose $K = (G, M, I)$ is a *formal context*, where G is the set of objects, M is a set of attributes and I is a binary relation.

Figure 1: Dataset preprocessing using clustering

1. From the set of attributes M choose disjoint sets of attributes $M_1, M_2, \dots, M_k$, such that $M_1 \cup M_2 \cup \dots \cup M_k = M$ and elements of each set can be unified by a shared interpretable meaning. For example, for the formal context related to banking data, such attributes as “gender”, “marital status”, “number of dependents” can be unified as “client personal information”, and “education”, “self-employment”, “income”, “co-applicant’s income” as “the client’s ability to repay the loan”.
2. Separately apply a chosen clustering method to the attribute sets $M_1, M_2, \dots, M_k$ and obtain clustering results as sets of clusters $C_1 = \{c_{11}, ..., c_{1t}\},\ C_2 = \{c_{21}, ..., c_{2t}\}, \dots, C_k = \{c_{k1}, ..., c_{kt}\}$.
3. Create a new formal context $K_{cl} = (G, M_{cl}, I_{cl})$, where G is the initial set of objects, $M_{cl} = \{C_1 \cup C_2 \cup \dots \cup C_k\}$ is a new attribute set, where each attribute stands for a cluster, I_{cl} - a binary membership relation to a given cluster. The example of dataset transformation is shown at Figure 1.
4. Find the most stable concepts based on monotone Galois connection [24] according to $\Delta - stability$ index [25]. Algorithm Sofia [26] can be used for this purpose.
5. Choose the “most interesting” concepts based on interestingness indices [27] to reduce the size of concept lattice (F1-score, accuracy, etc.) The example of concept lattice size reduction is shown at Figure 2.
6. Build neural network based on the reduced concept lattice. The architecture of the neural network is given as follows (Figure 3):
    - ***Input layer*** is created by the obtained attributes from dataset pre-clustering. Each attribute represents one of the clusters.
    - ***Hidden layers*** consisting of neurons corresponding to the resulting clusters.
    - ***Last hidden layer*** is connected to an ***output layer*** in which the number of neurons corresponds to the number of classes.

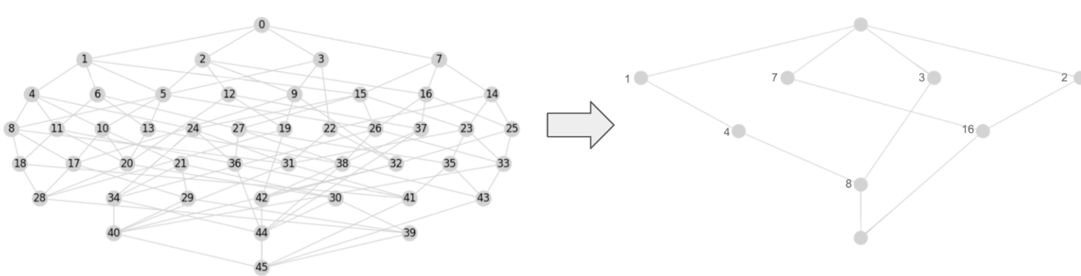


Figure 2: Concept lattice size reduction using "most interesting" concepts

In the current study, two approaches for choosing “the best” concepts were tested: based on F1-score and based on the accuracy metrics. For a single concept (A, B), the metrics was calculated with the following method:

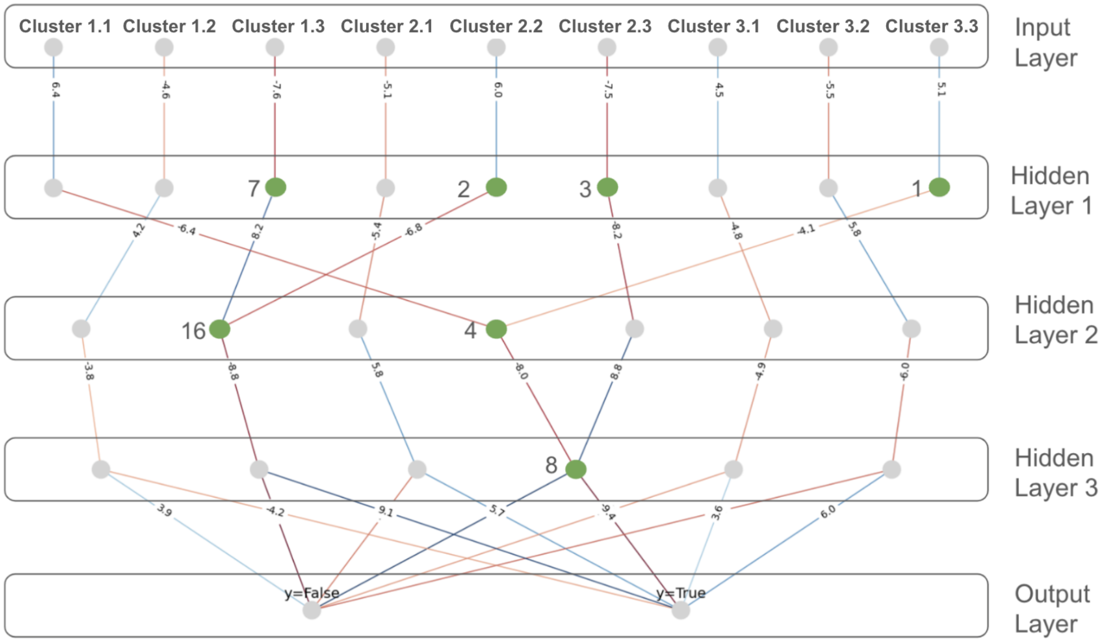


Figure 3: Neural network architecture based on concept lattice

- Assume that:
  $y_{pred}[g_i] = True,\ if\ g_i \in A$,
  $y_{pred}[g_i] = False,\ if\ g_i \notin A$; - an object is predicted True if it is in the extent of the concept and False otherwise;
- F1-score
  F1-score = F1-score(y, y_{preds}), where y_{preds} - predicted target values, y - real target values;
- accuracy = accuracy(y, y_{preds});
- Sort the concepts by the metrics value and choose 10 top concepts for building the neural network.

5. Experimental Part

To automatically find concepts for the *FCA-CLNet* architecture, build and train a neural network, this study uses the *FCApy* library (https://pypi.org/project/fcapy). This library provides the necessary tools for working with formal concepts and allows to automate the process of building and training a neural network based on these concepts.

Also, for a general understanding of the neural network, it is worth noting that sigmoid activation function is used for hidden layers. The value of softmax function is used for the output layer. When learning, binary cross-entropy is used as a loss function, and the Adam algorithm with the learning rate = 0.01 is used as an optimizer.

In this study, the performance of the model was compared with the following basic methods: *KNeighborsClassifier*, *LogisticRegression*, *RandomForestClassifier*, *CatBoostClassifier*, *XGBClassifier* and *TabNetClassifier*. Each of these methods was tested both on the initial dataset and on the dataset after clustering.

6. Data Description

For the purpose of our study we have chosen three datasets for binary classification from UCI Machine Learning Repository (https://archive.ics.uci.edu/) (Table 1):

Table 1
Dataset characteristics

Dataset	Number of objects	Number of attributes	Number of classes in target attribute
Credit Approval	690	15	2
Wine Quality	4898	11	2
Mammographic Mass	961	5	2

Each dataset represents a separate task and has its own unique characteristics, such as feature types, data size, class distribution, and noise presence. This approach allows one to consider different scenarios and evaluate the performance of models on different types of data.

7. Results

For the experimental evaluation, four clustering methods were applied for feature pre-clustering: K-means, Mean-Shift, DBSCAN, and HDBSCAN. 10 "most interesting" concepts were selected as neurons for the neural network architecture using two distinct concept selection methods. The results obtained for these two methods are presented in Table 2.

Table 2
FCA-CLNet concept selection method results (10 concepts). Weighted F1-score.

Clustering method	Best concepts selection	Loan Approval	Wine Quality	Mammographic
K-means	F1-score	0.79	**0.76**	**0.79**
	Accuracy	**0.83**	0.7	0.76
Mean-Shift	F1-score	0.84	0.72	0.74
	Accuracy	0.84	0.74	**0.79**
DBScan	F1-score	0.79	0.7	0.75
	Accuracy	0.81	0.69	0.76
HDBScan	F1-score	0.79	0.71	**0.77**
	Accuracy	**0.86**	0.7	0.73

The table shows that there is no significant difference in performance among the concept selection methods across all three datasets. For the K-means clustering approach, the F1-score-based concept selection method demonstrates better results in two out of the three datasets. The higher performance observed in the Loan Approval dataset may be attributed to the fact that the grouped features for clustering are more semantically similar than in the other datasets. Conversely, the method performs worst on the Wine Quality dataset, potentially indicating that this method is more effective when features can be easily divided into interpretable groups.

Subsequently, the performance of the proposed model was compared with that of classical machine learning methods on the same datasets. The performance of the FCA-CLNet model is very close to that of classical machine learning models, see Table 3.

8. Conclusion

In this paper, we investigated the application of feature pre-clustering for computing neural network architecture based on concept lattices, The proposed *FCA-CLNet* method demonstrated performance comparable to that of classical machine learning models, suggesting the potential for successfully integrating clustering methods into FCA-based approaches. While the results are promising, further development of the model is necessary to enhance its performance.

Table 3
Model Comparison. Weighted F1-score.

ML method	Clustering Method	Loan Approval	Wine quality	Mammographic
K-Neighbors	Without clustering	0.64	0.71	0.80
	K-Means	0.70	0.69	0.82
	Mean-Shift	0.71	0.66	0.81
	DBScan	0.70	0.67	0.81
	HBDScan	0.69	0.66	0.80
Logistic Regression	Without clustering	0.72	0.74	0.81
	K-Means	0.72	0.71	0.84
	Mean-Shift	0.72	0.68	0.84
	DBScan	0.74	0.69	0.81
	HBDScan	0.72	0.69	0.81
Naive Bayes	Without clustering	0.19	0.73	0.81
	K-Means	0.72	0.68	0.80
	Mean-Shift	0.22	0.38	0.84
	DBScan	0.68	0.54	0.64
	HBDScan	0.35	0.60	0.62
Random Forest	Without clustering	0.72	0.80	0.79
	K-Means	0.70	0.72	0.82
	Mean-Shift	0.73	0.68	0.84
	DBScan	0.72	0.69	0.81
	HBDScan	0.72	0.74	0.82
XGBoost	Without clustering	0.66	**0.81**	0.80
	K-Means	0.72	0.73	0.82
	Mean-Shift	0.73	0.68	0.84
	DBScan	0.72	0.70	0.82
	HBDScan	0.72	0.68	0.82
FCA - CLNet	K-Means	0.79	0.76	0.79
	Mean-Shift	**0.84**	0.72	0.74
	DBScan	0.79	0.7	0.75
	HBDScan	0.79	0.71	0.77

9. Acknowledgement

The work by S.O. Kuznetsov and M.M. Zueva in preparing this article was supported by grant 22-11-00323 of the Russian Science Foundation and carried out at the National Research University Higher School of Economics, Moscow.

References

[1] R. Roscher, B. Bohn, M. F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, Ieee Access 8 (2020) 42200–42216.

[2] O. Loyola-Gonzalez, Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view, IEEE access 7 (2019) 154096–154113.

[3] K. Gregor, Y. LeCun, Learning fast approximations of sparse coding, in: Proceedings of the 27th international conference on international conference on machine learning, 2010, pp. 399–406.

[4] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM computing surveys (CSUR) 51 (2018) 1–42.

[5] Z. Liu, F. Xu, Interpretable neural networks: principles and applications, Frontiers in Artificial Intelligence 6 (2023).

[6] Q. Zhang, Y. N. Wu, S.-C. Zhu, Interpretable convolutional neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8827–8836.

[7] N. Shlezinger, J. Whang, Y. C. Eldar, A. G. Dimakis, Model-based deep learning, Proceedings of the IEEE (2023).
[8] S. H. Rudy, S. L. Brunton, J. L. Proctor, J. N. Kutz, Data-driven discovery of partial differential equations, Science advances 3 (2017) e1602614.
[9] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 57 (2004) 1413–1457.
[10] Z. Wang, D. Liu, J. Yang, W. Han, T. Huang, Deep networks for image super-resolution with sparse prior, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 370–378.
[11] Y. Li, M. Tofighi, J. Geng, V. Monga, Y. C. Eldar, Efficient and interpretable deep blind image deblurring via algorithm unrolling, IEEE Transactions on Computational Imaging 6 (2020) 666–681.
[12] F.-L. Fan, J. Xiong, M. Li, G. Wang, On interpretability of artificial neural networks: A survey, IEEE Transactions on Radiation and Plasma Medical Sciences 5 (2021) 741–760.
[13] N. Frosst, G. Hinton, Distilling a neural network into a soft decision tree, arXiv preprint arXiv:1711.09784 (2017).
[14] Q. Zhang, R. Cao, S. Zhang, M. Redmonds, Y. N. Wu, S.-C. Zhu, Interactively transferring cnn patterns for part localization, arXiv preprint arXiv:1708.01783 (2017).
[15] Y. Zhang, P. Tiňo, A. Leonardis, K. Tang, A survey on neural network interpretability, IEEE Transactions on Emerging Topics in Computational Intelligence 5 (2021) 726–742.
[16] S. O. Kuznetsov, N. Makhazhanov, M. Ushakov, On neural network architecture based on concept lattices, in: Foundations of Intelligent Systems: 23rd International Symposium, ISMIS 2017, Warsaw, Poland, June 26-29, 2017, Proceedings 23, Springer, 2017, pp. 653–663.
[17] M. Shao, Z. Hu, W. Wu, H. Liu, Graph neural networks induced by concept lattices for classification, International Journal of Approximate Reasoning 154 (2023) 262–276.
[18] S. Peng, H. Yang, A. Yamamoto, Bert4fca: A method for bipartite link prediction using formal concept analysis and bert, arXiv preprint arXiv:2402.08236 (2024).
[19] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: Taxonomy and empirical analysis, IEEE transactions on emerging topics in computing 2 (2014) 267–279.
[20] A. K. Jain, R. C. Dubes, Algorithms for clustering data, Prentice-Hall, Inc., 1988.
[21] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, C.-T. Lin, A review of clustering techniques and developments, Neurocomputing 267 (2017) 664–681.
[22] C. Fraley, A. E. Raftery, How many clusters? which clustering method? answers via model-based cluster analysis, The computer journal 41 (1998) 578–588.
[23] B. Ganter, R. Wille, Formal concept analysis: mathematical foundations, Springer Science & Business Media, 2012.
[24] L. Zhao, M. J. Zaki, N. Ramakrishnan, Blosom: a framework for mining arbitrary boolean expressions, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 827–832.
[25] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Scalable estimates of concept stability, in: Formal Concept Analysis: 12th International Conference, ICFCA 2014, Cluj-Napoca, Romania, June 10-13, 2014. Proceedings 12, Springer, 2014, pp. 157–172.
[26] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Fast generation of best interval patterns for nonmonotonic constraints, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 157–172.
[27] S. O. Kuznetsov, T. Makhalova, On interestingness measures of formal concepts, Information Sciences 442 (2018) 202–219.

Clustering with Stable Pattern Concepts

Egor Dudyrev[1,2,*,†], Mariia Zueva[2,†], Sergei O. Kuznetsov[2,†] and Amedeo Napoli[1,†]

[1] *Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France*
[2] *HSE University, 20 Myasnitskaya St, Moscow, 101000, Russian Federation*

Abstract
Clustering aims at finding disjoint groups of similar objects in data and is one major task in Machine Learning. Yet, it is gaining more attention in Formal Concept Analysis community in these last years. This paper proposes an original approach to the clustering of complex data based on Formal Concept Analysis (FCA) and Pattern Structures. Stable concepts are considered as cluster candidates and the SOFIA algorithm is used to discover the set of stable concepts in linear time. Then an algorithm inspired by a rare itemset mining algorithm is designed to build a clustering with good properties, i.e., high internal cohesion within a cluster and high external separation between the clusters. Some interestingness measures allowing us to choose the best clustering are discussed. Finally the present approach is compared to some other well-known algorithms such as KMeans, DBScan, and Optic.

Keywords
Formal Concept Analysis, Pattern Structures, Clustering, Rare Itemset Mining

1. Introduction

Clustering aims at finding disjoint groups of similar objects in data and is one major task in Machine Learning [1, 2]. Although the relations between clustering and Formal Concept Analysis (FCA) are known and studied since a long time [3], clustering started gaining a new interest in the FCA community in the last years [4, 5]. Besides that, it should be noticed that Conceptual Clustering [6] and Biclustering [7, 8] have always attracted attention in FCA community.

FCA can be considered as a powerful mathematical framework in data analysis and classification [9]. Thus relations between FCA and clustering are worth to study. However, FCA faces three main problems when applied to clustering. Firstly, plain FCA only considers so called Formal Contexts based on binary datasets while most of the data are either numerical or of more complex nature. Secondly, without additional constraints, concept lattices can be exponential in the size of data (formal contexts) which makes plain FCA algorithms not applicable to big data. Thirdly, FCA concepts are organized in a concept lattice and are overlapping, while clustering is based on a partition into non-overlapping clusters. In this paper we propose an original approach to overcome these three problems: (1) we use Pattern Structures to extend FCA to deal with (almost) any kind of complex descriptions, (2) we use the SOFIA algorithm to discover a limited set of cluster candidates in linear time, and (3) we propose an algorithm to select non-overlapping clusters from the set of given cluster candidates based on Rare Itemset Mining.

Pattern Structures are used for clustering in [4], where authors are considering Pattern Structures adapted to numerical and sequential data. The present paper studies clustering of tabular data of any type, where every column is represented by an arbitrary pattern structure, making the present approach more versatile and more universal.

The idea of concept stability in FCA was first introduced in [10] and then refined in [11] giving rise to Δ-stability. Roughly speaking, the Δ-stability of a concept shows how many objects the concept will

FCA4AI 2024, Santiago de Compostela, Spain
*Corresponding author.
† These authors contributed equally.
egor.dudyrev@loria.fr (E. Dudyrev); m.zueva@hse.ru (M. Zueva); skuznetsov@hse.ru (S. O. Kuznetsov); amedeo.napoli@loria.fr (A. Napoli)
https://egordudyrev.github.io/ (E. Dudyrev)
0000-0002-2144-3308 (E. Dudyrev); 0009-0000-6332-9936 (M. Zueva); 0000-0003-3284-9001 (S. O. Kuznetsov); 0000-0001-5236-9561 (A. Napoli)

lose when making its description more precise (recall that a concept is composed of a set of objects and a set of attributes materializing their common description). The use of concept stability for selecting concepts adapted to clustering is studied in [12]. However, these authors are using concept stability to select interesting concepts from the whole set of concepts which may be of exponential size. By contrast, in our approach we make use of the Sofia algorithm [13, 14], to select the stable concepts in linear time, without requiring to construct the whole set of concepts.

The problem of avoiding overlapping clusters when covering the whole data is addressed in various ways. For example, the authors of [6] are trying to discover similar sublattices of concepts w.r.t. a predefined similarity measure. Thus the latter approach is closer to "Conceptual Clustering" rather than to clustering of objects. The authors of [15] are solving a biclustering problem, which is more specific than clustering, and, firstly they are discovering a set of non-overlapping concepts that covers most of the data, and then they are adding missing objects to the discovered biclusters until the whole data is covered. By contrast, the authors of [4] are considering overlapping concepts as clusters, and then delete objects lying in the overlap from all but one cluster to which they "mostly" belong.

From our side, we think that the overlap between clusters is a natural phenomenon as there are many things in our world which cannot be strictly attached to only one single concept[1]. Thus, by contrast, we propose to build clusters with the smallest possible overlap, and then to draw the attention of the analyst to these overlapping objects.

2. Concepts as Clusters

2.1. A Bit of Formal Concept Analysis Terminology

Formal Concept Analysis [9] is a mathematical formalism based on lattice theory and aimed at data analysis and classification. In FCA, data are represented thanks to a **formal context** (G, M, I) where G is the set of objects, M is the subset of attributes, and $I \subseteq G \times M$ is the binary relation between objects and attributes. A formal context or more simply context can be represented as a binary table where rows stand for objects, columns for attributes, and a cross is lying in a cell when the corresponding object has the corresponding attribute.

Given a formal context (G, M, I), we define two **derivation operations** denoted as **'** ("prime"): given a set of objects $A \subseteq G$, the first operation returns $A' \subseteq M$, i.e., the set of attributes common to all objects in A, while, given a set of attributes $B \subseteq M$ the second operation returns $B' \subseteq$, i.e., the set of all objects having all attributes in B. More formally:

$A' = \{m \in M \mid \forall g \in A, (g, m) \in I\}, \forall A \subseteq G$, and

$B' = \{g \in G \mid \forall m \in B, (g, m) \in I\}, \forall B \subseteq M$.

For the sake of simplicity, we denote the description on a single object $g \in G$ as g' rather than $\{g\}'$, while we denote the objects described by a single attribute $m \in M$ as m', rather than $\{m\}'$.

A **formal concept** (A, B) is a pair where the set of objects A and the set of attributes B verify $A' = B$ and $B' = A$. In concept (A, B), the set of objects A is called the **extent** and the set of attributes B is called the **intent**. Moreover, objects can be organized into a concept lattice thanks to the subsumption relation –a partial ordering– where a concept (A_1, B_1) is subsumed by a concept (A_2, B_2) iff $A_1 \subseteq A_2$ or dually $B_2 \subseteq B_1$

A subset of attributes $D \subseteq M$ is called a **minimal generator** of concept (A, B) when it is a minimal subset of attributes, whose extent is A. In other words, removing any attribute $m \in D$ from description D will change its extent, i.e., $\forall m \in D, (D \setminus \{m\})' \neq D'$.

Finally, the **support** of any description $D \subseteq M$ is given by the cardinality of the set of objects having D as description, i.e., $supp(D) = |D'|$.

[1] An interesting example is given by "Pheasant Island", that belongs either to France or to Spain depending on the time of the year!

2.2. Formal Concepts as Clusters

Clustering is generally defined as the problem of discovering a set of disjoint clusters that cover all the data, such that objects belonging to the same cluster are more similar than objects belonging to different clusters. The choice of a similarity measure depends on a the type of data and the task at hand. For example, considering numerical data, in clustering based on K-means (see for example [16]) every object is described by a vector of real numbers and the similarity between objects is in the inverse proportion to the Euclidean distance between the object descriptions. In clustering based on DBScan (see again [16]) the similarity between objects is based on the amount of close common neighbours in the Euclidean space.

In our framework, when two objects $g_1, g_2 \in G$ are described by the corresponding sets of attributes –aka itemsets– $g'_1, g'_2 \subseteq M$, a natural way to define the similarity between two objects is provided by the Jaccard similarity coefficient [16] between the descriptions:

$$\texttt{sim}(g_1, g_2) \coloneqq J(g'_1, g'_2) = \frac{|g'_1 \cap g'_2|}{|g'_1 \cup g'_2|}. \tag{1}$$

Using equation 1 the clustering task can be defined as follows, where $\wp(G)$ is the powerset of the set of objects G:

$$\begin{aligned} &\text{Discover a set of clusters } \mathcal{C} \subseteq \wp(G), \text{ such that:} \\ &\bigcup_{C_i \in \mathcal{C}} C_i = G \\ &\forall\, C_i,\, C_j \in \mathcal{C}, C_i \cap C_j = \emptyset \\ &\forall\, g,\, g_i \in C_i, g_j \in C_j : \texttt{sim}(g, g_i) \gg \texttt{sim}(g, g_j) \end{aligned} \tag{2}$$

Now let us consider a formal concept (A, B) and the similarity between two objects $g_1, g_2 \in A$:

Proposition 2.1. *Given a formal concept (A, B), the similarity between any pair of objects from extent A is lower-bounded by the length of the concept intent $|B|$:*

$$\texttt{sim}(g_1, g_2) \geq \frac{|B|}{|M|} \tag{3}$$

Proof. Since g_1 and g_2 belong to concept (A, B), the concept's intent B is included in their common description $g'_1 \cap g'_2$. Meanwhile the union of the descriptions $g'_1 \cup g'_2$ cannot be larger than the maximal description M, and then $g'_1 \cup g'_2 \subseteq M$. Thus, the following formulas hold true:

$$\texttt{sim}(g_1, g_2) = \frac{|g'_1 \cap g'_2|}{|g'_1 \cup g'_2|} \geq \frac{|B|}{|g'_1 \cup g'_2|} \geq \frac{|B|}{|M|}.$$

□

Therefore, a formal concept (A, B) can be considered as a cluster of objects A that are at least $|B|/|M|$ similar. Following this reformulation, the objective of clustering is to discover a set of concepts with large intents but tiny or no overlapping between the extents of concepts.

3. Clustering pipeline

The clustering pipeline proposed in this paper is shown in Figure 1. Below we discuss the different FCA techniques allowing us to efficiently build an optimal clustering.

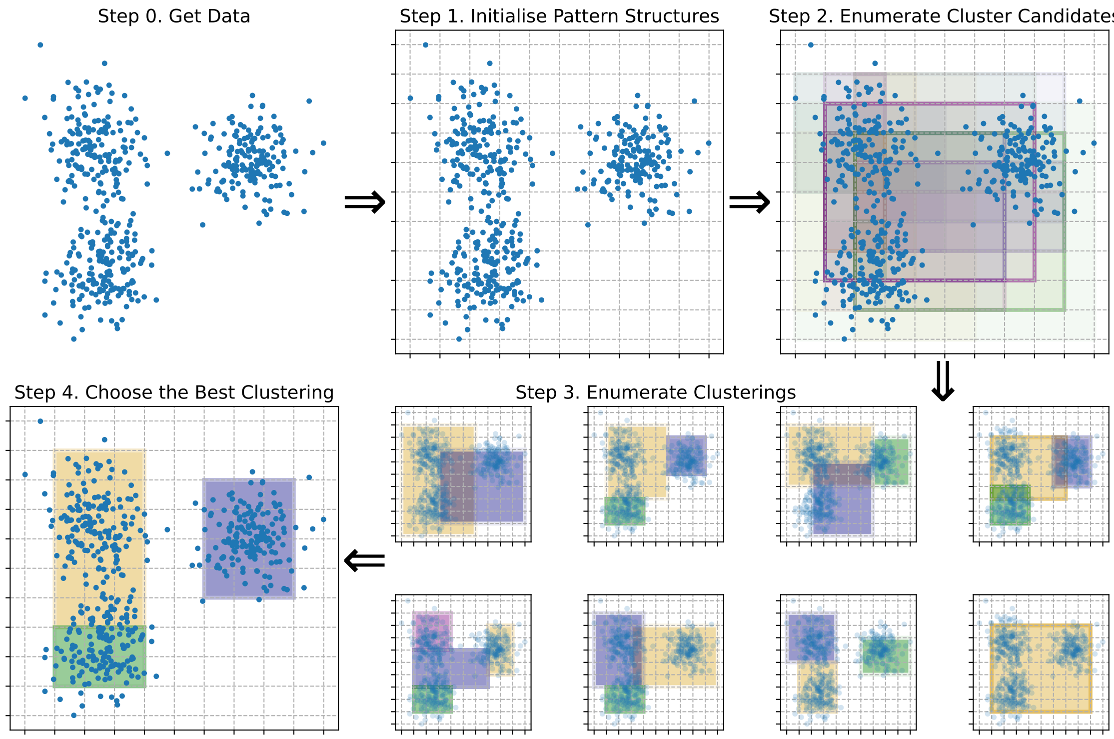


Figure 1: The pipeline proposed for building a clustering pipeline based on FCA techniques.

3.1. Step 1. Initializing the Pattern Structure

While plain FCA works with contexts representing binary datasets, we rely in the present framework on Pattern Structures, an extension of FCA allowing us to deal with many types of complex data. In the following, we focus on Interval and Cartesian Pattern Structures to take into account multidimensional numerical data. Below we recall the definitions and techniques of Pattern Structures.

Recall that a Formal Context is a triple (G, M, I) where G is a set of objects, M is a set of attributes, and $I \subseteq G \times M$ is a set of pairs (g, m) indicating that object g is described by attribute m. In such a formal context, the space of object descriptions $\underline{\mathbb{D}} = (\mathbb{D}, \subseteq)$ is the powerset of attributes $\mathbb{D} = \wp(M)$, ordered by inclusion $\subseteq$. It should be noticed that such description space $\underline{\mathbb{D}} = (\mathbb{D}, \subseteq)$ forms a lattice, i.e., for every pair of descriptions $D_1, D_2 \in \mathbb{D}$ there is exactly one meet and one join: $\exists! D_\wedge \in \mathbb{D}, D_1 \cap D_2 = D_\wedge$ and $\exists! D_\vee \in \mathbb{D}, D_1 \vee D_2 = D_\vee$.

The Pattern Structure formalism [17, 18, 19] generalizes plain FCA and in particular the description space $\underline{\mathbb{D}}$. The latter consists of a description set $\mathbb{D}$ equipped with operation $\sqcap$ that defines a complete meet semilattice on $\mathbb{D}$, i.e., for any pair of descriptions $D_1, D_2 \in \mathbb{D}$ there exists meet (infimum): $\exists! D_\wedge \in \mathbb{D}, D_1 \sqcap D_2 = D_\wedge$. The operation $\sqcap$ defines natural order $\sqsubseteq$: $X \sqsubseteq Y \iff X \sqcap Y = X$.

Then the description space $\underline{\mathbb{D}}$ combined with the set of objects G and a mapping $\delta : G \to \mathbb{D}$ forms a **pattern structure** $(G, \underline{\mathbb{D}}, \delta)$, that can be considered as an analogue and a generalization of a formal context (G, M, I) as follows. For any pattern description $D \in \mathbb{D}$, one can define the pattern extent:

$D^\diamond = \{g \in G \mid D \sqsubseteq \delta(g)\}$,

and for any subset of objects $A \subseteq G$, one can define the pattern intent:

$A^\diamond = \sqcap\{\delta(g) \mid g \in A\}$.

A pair of corresponding pattern extent A and pattern intent D forms a **pattern concept**: (A, D), where $A^\diamond = D, D^\diamond = A$.

This paper focuses on Interval and Cartesian Pattern Structures for modelling multidimensional numerical data. The **Interval Pattern Structure** works with the description space $\underline{\mathbb{D}}_{\text{int}}$ of intervals bounded by real numbers $\mathbb{D}_{\text{int}}$ ordered by interval subsumption $\sqsubseteq_{\text{int}}$:

$$\mathbb{D}_{\text{int}} = \{[l, r] \mid l, r \in \mathbb{R}, l \leq r\}, \text{ and } \forall [l_1, r_1], [l_2, r_2] \in \mathbb{D}_{\text{int}}, [l_1, r_1] \sqsubseteq_{\text{int}} [l_2, r_2] \iff [l_1, r_1] \supseteq [l_2, r_2].$$

The **Cartesian Pattern Structure** allows to combine various description spaces $\underline{\mathbb{D}}_1, \underline{\mathbb{D}}_2, \ldots, \underline{\mathbb{D}}_n$ in a single description space $\underline{\mathbb{D}}_\times = (\mathbb{D}_\times, \sqsubseteq_\times)$ such that:

$$\mathbb{D}_\times = \bigtimes_{1 \leq i \leq n} \mathbb{D}_i \text{ and } \forall D, E \in \mathbb{D}_\times, D \sqsubseteq_\times E \iff \bigwedge_{1 \leq i \leq n} D_i \sqsubseteq_i E_i.$$

In this paper, every column in a numerical dataset is processed thanks to a distinct Interval Pattern Structure $(G, \underline{\mathbb{D}}_i, \delta_i)$, and the whole dataset is processed thanks to a Cartesian Pattern Structure $(G, \underline{\mathbb{D}}_\times, \delta_\times)$ built on top of base Interval Pattern Structures.

We also provide the "support function" $\texttt{supp} : \mathbb{D} \to \mathbb{N}$ to compute either the number of objects described by a binary description $B \subseteq M$, or the number of objects described by a pattern description $D \in \mathbb{D}$:

$$\texttt{supp}(B) = |B'|, \forall B \subseteq M, \quad \texttt{supp}(D) = |D^\diamond|, \forall D \in \mathbb{D} \tag{4}$$

This allows us to highlight the similarities when using either FCA or Pattern Structures for the clustering task.

Although Interval Pattern Structures may work with an infinite space of intervals $\mathbb{D}$, we restrict each pattern structure to only deal with 11 evenly-spaced interval borders, i.e., $V \subseteq \mathbb{R}, |V| = 11$, $\mathbb{D} = \{[l, r] \mid l, r \in V, l \leq r\}$, assuming that every object description lies inside the largest interval $[\min(V), \max(V)]$. This restriction allows us to reduce the computational time and to improve the stability of descriptions, as this is discussed in the next section.

3.2. Step 2. Enumerating Cluster Candidates

Previously, we have defined the Jaccard similarity measure between two objects in a formal context. However, it is much less straightforward to define a similarity measure in a Pattern Structure. For example, considering the two descriptions of cities: "population from 10k to 100k people, in East Asia" and "population from 100k to 1M people, in France". Discovering which cities are the more similar depends on the arbitrary similarity function defined for each pattern dimension, e.g., population and geographical location, and on the arbitrary way to aggregate these similarity functions into a single similarity measure. Below we describe how to use stable concepts to mimic the similarity for any type of descriptions.

Concept stability is defined in [10] as the percentage of subsets in a concept extent having a common description B:

$$\texttt{stab}(A, B) \coloneqq \frac{|\{A_2 \subseteq A \mid A_2' = B\}|}{2^{|A|}}. \tag{5}$$

However, due to its exponential nature, stability is hard to compute in practice. This is why Δ-stability was introduced in [11] as a linear-time upper bound of concept stability:

$$\Delta\texttt{stab}(A, B) \coloneqq |A| - \max_{\substack{B_2 \subseteq M \\ \text{s.t. } B \subset B_2}} \texttt{supp}(B_2). \tag{6}$$

Delta-stability can also be adapted to a pattern concept (A, D) whose description belongs to a description space $\mathbb{D}$:

$$\Delta\texttt{stab}(A, D) \coloneqq |A| - \max_{\substack{D_2 \in \mathbb{D} \\ \text{s.t. } D \sqsubset D_2}} \texttt{supp}(D_2). \tag{7}$$

In general terms, the value of Δ-stability can be interpreted as "how many objects from A one will lose when making description D just a bit more precise". Then, given a concept (A, D) with a high

Δ-stability, although the exact similarity between objects in A cannot be measured, any more precise concept describe fewer objects in a bounded way.

Another useful characteristic of stable concepts is that there exist efficient algorithms such as SOFIA [13] and gSOFIA [14] that can be used to directly mine only stable concepts without computing the excessively large amount of non-stable concepts.

3.3. Step 3. Enumerating Clustering Candidates

In our setting, there are some characteristics for determining a "good clustering", e.g., a set of clusters that are also well separated concepts. However, there are two main properties that are necessary for achieving a good clustering: a clustering should cover most of the objects in the data and clusters should not overlap too much. Below we present an algorithm enumerating clustering candidates satisfying these two properties.

More formally, let us first consider a set of clusters candidates $\mathbb{C} \subseteq \wp(G)$, where every cluster $C \in \mathbb{C}$ is a closed set of objects, i.e., $C'' = C$. Then a **clustering** is any subset of clusters $\mathcal{C} \subseteq \mathbb{C}$. A clustering $\mathcal{C}$ is called **broad** if it covers more than θ_{cov} objects: $\texttt{cov}(\mathcal{C}) = |\bigcup_{C_i \in \mathcal{C}} C_i| > \theta_{cov}$. A clustering $\mathcal{C}$ is called **minimal broad** clustering if it is a broad clustering, and all its proper subsets are not broad clusterings: $\texttt{cov}(\mathcal{C}) > \theta_{cov}$ and $\forall \mathcal{C}_2 \subset \mathcal{C}, \texttt{cov}(\mathcal{C}_2) \leq \theta_{cov}$. A clustering $\mathcal{C}$ is called θ_{ol}-**non-overlapping** if every pair of clusters overlaps for at most θ_{ol} objects: $|C_i \cap C_j| \leq \theta_{ol}, \forall C_i,\, C_j \in \mathcal{C}$. Then, our task consists in enumerating minimal broad non-overlapping clusterings built from the set of clusters $\mathbb{C}$.

The latter problem of discovering minimal broad non-overlapping clusterings is far from being simple, as it can even be related to the famous Set Covering Optimisation Problem. However, a satisfactory solution can be found when the problem is related to the "Rare Itemset Mining" problem [20], which was formerly addressed in the pattern mining and FCA communities. Rare Itemset Mining focuses on discovering **minimal rare itemsets**, that are minimal subsets of attributes $D \subseteq M$ of a formal context (G, M, I) whose support is below a given threshold $\theta_{\min}$: $D \subseteq M$ s.t. $\texttt{supp}(D) = |\bigcap\{m' \mid m \in D\}| < \theta_{\min}$ and $\forall m \in D, \texttt{supp}(D \setminus \{m\}) \geq \theta_{\min}$.

It can be noticed that discovering **minimal broad clusterings** –possibly overlapping– can be reduced to discovering minimal rare itemsets, as minimal broad clusterings are the minimal subsets of clusters $\mathcal{C} \subseteq \mathbb{C}$ whose coverage is above a given threshold θ_{cov}, i.e., $\mathcal{C} \subseteq \mathbb{C}$ s.t. $\texttt{cov}(\mathcal{C}) = |\bigcup C_i| > \theta_{cov}$ and $\forall C_i \in \mathcal{C}, cov(\mathcal{C} \setminus \{C_i\}) \leq \theta_{cov}$.

The relationship between discovering minimal broad clusterings and minimal rare itemsets allows us to use Rare Itemset Mining algorithms for finding minimal broad clusterings. To do so, one should search for minimal rare itemsets in the inverted clusters context $(G, \mathbb{C}, \notin)$ with minimal support threshold $\theta_{\min} = |G| - \theta_{cov}$.

Proposition 3.1. *Let us consider the "inverted cluster context"* $K_{\overline{\mathbb{C}}} = (G, \mathbb{C}, \notin)$*, where* G *is a set of objects,* $\mathbb{C}$ *a set of clusters, and* $\notin$ *the incidence relation such that* $\notin = \{(g, c) \in G \times \mathbb{C} | g \notin \mathbb{C}\}$*.*

Then a subset of clusters $\mathcal{C} \subseteq \mathbb{C}$ *is a minimal broad clustering in* $K_{\overline{\mathbb{C}}} = (G, \mathbb{C}, \notin)$ *w.r.t. the coverage threshold* θ_{cov} *if and only if it is a minimal rare itemset in context* $K_{\overline{\mathbb{C}}}$ *w.r.t. the minimal support threshold* $\theta_{\min} = |G| - \theta_{cov}$*.*

Proof. Consider the logical statement over two literals a and b: $\overline{a} \wedge \overline{b} = \overline{a \vee b}$. Now, let a, b be attributes of an arbitrary formal context (G, M, I). An analogous property of attribute extents can be inferred: $(G \setminus a') \cap (G \setminus b') = G \setminus (a' \cup b')$.

In the inverted cluster context $K_{\overline{\mathbb{C}}} = \{G, \mathbb{C}, \notin\}$, every attribute is a cluster $C \in \mathbb{C}$, and the context is designed in such a way –thanks to the the $\notin$ relation– that the extent of every cluster-as-attribute is the complement of the cluster itself: $C' = G \setminus C$. Here the expression "cluster-as-attribute" stands for an attribute representing a cluster in the inverted context.

Thus, the extent of any subset of clusters-as-attributes $D \subseteq \mathcal{C}$ in this context contains objects described by none of the clusters: $\bigcap\{C' \mid C \in D\} = G \setminus \bigcup\{C \mid C \in D\}$. Given that $\texttt{supp}(D) = |\bigcap\{C' \mid C \in D\}|$ and $\texttt{cov}(D) = |\bigcup\{C \mid C \in D\}|$, it comes that $\texttt{supp}(D) = |G| - \texttt{cov}(D)$. This equality, in turn, gives rise to the proposition, i.e., $\texttt{supp}(D) < \theta_{\min} \iff \texttt{cov}(D) > |G| - \theta_{\min} = \theta_{cov}$. □

To the best of our knowledge, paper [21] was the first to propose the idea of representing the union of attributes extents via the extents of the same attributes in the inverted context. There, the authors considered the unions of attributes as the intents of monotone Galois connections and used formal concepts to mimic the behaviour of linear regressions and neural networks.

As we have shown, one can reuse algorithms about Rare Itemset Mining to enumerate all broad clusterings. For example, in this work we have re-implemented the algorithm MRG-Exp (also known as Carpathia-G-Rare) proposed in [20]. However, there are two particularities of a clustering task that are not really considered in Rare Itemset Mining: (1) minimal rare itemsets may contain an arbitrary amount of attributes, while a clustering often contains only a few clusters, and (2) attributes in a minimal rare itemset may have highly overlapping extents, while clusters in a clustering are supposed to be disjoint. To satisfy these two requirements, we add two parameters in our implementation of MRG-Exp algorithm. Firstly, we introduce **maximal size parameter** η_{size} to only consider clusterings $\mathcal{C}$ containing at most η_{size} clusters: $|\mathcal{C}| \leq \eta_{\text{size}}$. And secondly, we add **minimal added coverage parameter** η_{cov} that defines the minimal amount of objects a cluster should add to a clustering: $\forall C \in \mathcal{C}, \texttt{cov}(\mathcal{C}) - \texttt{cov}(\mathcal{C} \setminus \{C\}) \geq \eta_{\text{cov}}$. It can be noticed that, when η_{cov} is set to 1, the condition on minimal added coverage becomes the condition on the minimality of a clustering: $\forall C \in \mathcal{C}, (\texttt{cov}(\mathcal{C}) - \texttt{cov}(\mathcal{C} \setminus \{C\}) \geq 1) \iff (\texttt{cov}(\mathcal{C}) \neq \texttt{cov}(\mathcal{C} \setminus \{C\})$.

To summarize this section, we state that we solve the problem of enumerating minimal broad non-overlapping clusterings by relating it to the problem Rare Itemset Mining with an additional non-overlapping requirement. That is, we re-implement the MRG-Exp algorithm, while replacing all intersections of extents in the algorithm with their unions, and replacing all tests of the form "support < $\theta_{\min}$" by dual tests of the form "coverage > θ_{cov}". Finally, we reduce the search space of clusterings by specifying the restriction on the maximal size of a clustering, and by specifying the minimal added support threshold for every concept in a clustering.

3.4. Step 4. Selecting the Best Clustering

Now we know how to enumerate minimal broad non-overlapping clustering candidates. However, one can obtain multiple –sometimes, thousands of– minimal broad non-overlapping clustering candidates. Below we propose some measures for guiding the choice of the best clustering out of the possibly very large set of broad minimal non-overlapping candidates.

The main criterion for interestingness –or goodness– of a clustering $\mathcal{C} \subseteq \wp(G)$ is the **coverage** of the clustering, i.e., the number of objects covered by the clustering $\mathcal{C}$: $\texttt{cov}(\mathcal{C}) = |\bigcup C_i|$.

The second most important criterion for goodness of a clustering $\mathcal{C} \subseteq \wp(G)$ is the **overlap**, i.e., the size of the pairwise intersections of clusters in $\mathcal{C}$: $\texttt{ovlap}(\mathcal{C}) = \sum_{C_i, C_j \in \mathcal{C}} |C_i \cap C_j|$. We do not normalise the size of the overlaps by the number of pairs of concepts and the normalisation procedure is explained at the end of this section.

Another criterion for differentiating two clustering candidates is to measure their **sizes**: $\texttt{size}(\mathcal{C}) = |\mathcal{C}|$. Depending on the task and the data, the analyst running the clustering might prefer clustering candidates with a specific number of clusters.

Moreover, in some cases, an analyst may prefer or penalize imbalanced clustering candidates where the sizes of the clusters in the clustering $\mathcal{C}$ are highly varying. The **imbalance** of a clustering $\mathcal{C}$ is measured as the standard deviation of the cardinalities of its clusters: $\texttt{imb}(\mathcal{C}) = \texttt{std}(\langle |C_1|, |C_2|, \ldots, |C_{|\mathcal{C}|}| \rangle)$.

An analyst may also prefer clustering candidates consisting of mostly stable concepts. Then the **stability** of a clustering $\mathcal{C}$ is measured as the average delta-stability of its concepts: $\texttt{stab}(\mathcal{C}) = \sum\limits_{C_i \in \mathcal{C}} \Delta\texttt{stab}(C_i)/|\mathcal{C}|$.

Finally, since we study multidimensional numerical data, we will give priority to dense clusters. More precisely, in n-dimensional data, the clusters have the form of hyperrectangles, i.e., $D = \langle [l_1, r_1], [l_2, r_2], \ldots, [l_n, r_n] \rangle$. The **density** of a clustering $\mathcal{C}$ is defined as the average density of its clusters-concepts $(A, D) \in \mathcal{C}$: $\texttt{density}(\mathcal{C}) = \sum\limits_{(A,D) \in \mathcal{C}} \texttt{density}\big((A, D)\big)$, where $\texttt{density}\big((A, D)\big) = |A|/\Pi_{j=1}^{n}(r_j - l_j)$.

In order to aggregate all measures related to a clustering in a single measure, every clustering is associated with a **reward function**, which is a weighted sum of the above measures. In addition, to improve the interpretability of weights in the reward function, we normalize the values of each basic measure in such a way that the lowest possible value of any basic measure is 0, and the maximal possible value of any basic measure is 1, i.e., we apply MinMax scaling to the values of the computed basic measures.

4. Experiments and Discussion

This paper presents our first attempt in building a clustering problem based on FCA and Pattern Structures. For testing these first ideas, we have run tests over artificial and accessible datasets provided by SciKit Learn to compare the present results with various State-of-the-Art clustering algorithms. The results of the original algorithm comparison is presented on the web page https://scikit-learn.org/stable/modules/clustering.html.

We chose to compare the experiment results returned by or *FCA-based* algorithm that follows the pipeline presented above, with three well-known clustering methods. We considered (1) *K-Means* which is one of the most popular and the most simple clustering methods, (2) *DBScan* which is one of the most popular density-based clustering method, and (3) *Optics* which is one of the most versatile −while also the less time efficient− algorithm presented in SciKit Learn.

The plots on Figure 2 present the clustering obtained by 4 algorithms on 6 datasets. It can be seen that no clustering method is perfect: for example, K-Means does not work well on circular data (the top row #1), DBscan and Optics do not find all three clusters on the "blobs" data (row #5), while FCA-based algorithm works nicely on "blobs"-based data (rows #3 and #5) but fails on the other datasets.

It should be noticed that all these different clustering methods are based on different principles and processes. K-Means clustering operates over centroids of clusters in multi-dimensional data. Thus, it naturally tends to discover “blobs”-like clusters (rows #3, #5). DBScan and Optics are density-based approaches. Therefore, they tend to discover nonlinear continuous clusters (e.g. rows #1, #2, #4) but fail when the objects of two clusters are placed too close to each other. Finally, the FCA-based algorithm searches for clusters that having more the form of a hyperrectangle. Thus, the latter tends to discover “blobs”-like clusters as K-Means does.

The main disadvantage of the current FCA-based algorithm is the running time. As the results in Figure 2 show, the FCA-based approach may work up to 1780 times slower than the the slowest competitor which is Optics. Table 1 presents the running times and the sizes of the output computed at each step of the proposed pipeline. It can be seen that most of the time is spent in Step 3 of the pipeline, corresponding to the computing of minimal broad non-overlapping clustering candidates. Actually, during this step hundreds of thousands of clustering candidates are produced leading to a very high redundancy, while only a few best candidates are interesting. The minimization of the number of clustering candidates discovered during Step 3 will also reduce the time required in Step 4 of the pipeline, whose objective is the evaluation of the returned clustering candidates.

Thus, an important direction in future work is to develop a new algorithm for finding only hundreds of best broad non-overlapping clustering candidates. Meanwhile, it should be noticed that in most of the cases the total running time in Table 1 are already lying within “reasonable time slots” of tens of seconds.

One could argue that an FCA-based algorithm can also find nonlinear clusters, as in rows #1, #2, and #4, when using a polygon-based pattern structure (see [22, 23]) instead of the combination of Interval and Cartesian pattern structures. Indeed, this is also one main future work.

The results for these experiments were obtained on a MacBook Pro with Apple M2 chip and 16 GB of RAM. The source code for the experiments can be found in the Git repository https://github.com/EgorDudyrev/Paper_StablePatternClustering.

dataset	Step 2 # stable concepts	 stable concepts time (s)	Step 3 # clusterings	 clusterings time (s)	Step 4 statistics time (s)	total time (s)
noisy_circles	1 150	0.06	129 629	84.73	4.28	89.07
noisy_moons	636	0.04	99 082	15.86	3.08	18.98
varied	564	0.04	71 696	8.77	2.26	11.07
aniso	342	0.03	21 353	1.55	0.96	2.54
blobs	554	0.04	51 796	7.17	2.37	9.57
no_structure	1 139	0.05	96 914	84.18	3.19	87.42

Table 1
The time and the size of the output for every step of the proposed clustering pipeline.

5. Conclusions

In this paper we have presented an original pipeline for clustering numerical data using Formal Concept Analysis and Pattern Structures. The pipeline consists of four steps: (1) we encode the data via Interval and Cartesian Pattern Structures, (2) we find the set of stable cluster candidates thanks to the gSofia algorithm, (3) we enumerate the set of minimal broad non-overlapping clustering candidates, and (4) we select the best clustering candidates based on a set of interestingness measures. We also show that this approach outputs some reasonable clusterings when applied to artificial datasets from the SciKit Learn package, while running in a matter of seconds.

As future work we are planning to mainly improve the third step of the pipeline, by reducing the space of the clustering candidates. We will also run experiments over real-world complex datasets with numerical, categorical, and textual elements. Finally, our research raises the question of the type of clusters that can be found when using an FCA framework, i.e., how to define a pattern structure able to describe dense continuous clusters, or rotated hyperrectangles, or any polygons in multidimensional space.

References

[1] L. Kaufman, P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley, 1990.
[2] B. G. Mirkin, Core Data Analysis: Summarization, Correlation, and Visualization, Second Edition, Undergraduate Topics in Computer Science, Springer, 2019.
[3] C. Carpineto, G. Romano, A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval, Machine Learning 24 (1996) 95–122.
[4] S. E. Boukhetta, M. Trabelsi, Formal Concept Analysis for Trace Clustering in Process Mining, in: M. Ojeda-Aciego, K. Sauerwald, R. Jäschke (Eds.), Graph-Based Representation and Reasoning, Springer Nature Switzerland, Cham, 2023, pp. 73–88.
[5] M. Alwersh, L. Kovács, K-Means Extensions for Clustering Categorical Data on Concept Lattice, International Journal of Advanced Computer Science and Applications (IJACSA) 14 (2023).
[6] T. T. Quan, S. C. Hui, T. H. Cao, A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data, in: CLA 2004, volume 2004, VS˘B – Technical University of Ostrava, Dept. of Computer Science, 2004, pp. 1–12.
[7] N. Juniarta, M. Couceiro, A. Napoli, A Unified Approach to Biclustering Based on Formal Concept Analysis and Interval Pattern Structure, in: P. K. Novak, T. Smuc, S. Dzeroski (Eds.), Proceedings of 22nd International Conference on Discovery Science (DS), Lecture Notes in Computer Science 11828, Springer, 2019, pp. 51–60.
[8] D. I. Ignatov, B. W. Watson, Towards a Unified Taxonomy of Biclustering Methods, 2017. doi:10.48550/arXiv.1702.05376. arXiv:1702.05376.
[9] B. Ganter, R. Wille, Formal Concept Analysis, Springer, Berlin, 1999.

[10] S. O. Kuznetsov, On stability of a formal concept, Annals of Mathematics and Artificial Intelligence 49 (2007) 101–115.
[11] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Scalable Estimates of Concept Stability, in: C. V. Glodeanu, M. Kaytoue, C. Sacarea (Eds.), Formal Concept Analysis, Lecture Notes in Computer Science 8478, Springer International Publishing, Cham, 2014, pp. 157–172.
[12] J. Cigarrán, Á. Castellanos, A. García-Serrano, A step forward for Topic Detection in Twitter: An FCA-based approach, Expert Systems with Applications 57 (2016) 21–36.
[13] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Fast Generation of Best Interval Patterns for Nonmonotonic Constraints, in: A. Appice, P. P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge, C. Soares (Eds.), Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), Lecture Notes in Computer Science 9285, Springer, Cham, 2015, pp. 157–172.
[14] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Efficient Mining of Subsample-Stable Graph Patterns, in: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, New Orleans, LA, 2017, pp. 757–762.
[15] R. Belohlavek, N. Kulkarni, V. Vychodil, A Novel Approach to Cell Formation, in: S. Ferré, S. Rudolph (Eds.), Formal Concept Analysis, Springer, Berlin, Heidelberg, 2009, pp. 210–223.
[16] M. J. Zaki, W. M. Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014.
[17] B. Ganter, S. O. Kuznetsov, Pattern Structures and Their Projections, in: H. S. Delugach, G. Stumme (Eds.), Proceedings of the 9th International Conference on Conceptual Structures, Lecture Notes in Computer Science 210, Springer, 2001, pp. 129–142.
[18] M. Kaytoue, S. O. Kuznetsov, A. Napoli, S. Duplessis, Mining gene expression data with pattern structures in formal concept analysis, Information Sciences 181 (2011) 1989–2001.
[19] A. Belfodil, S. O. Kuznetsov, M. Kaytoue, On Pattern Setups and Pattern Multistructures, International Journal of General Systems 49 (2020) 785–818.
[20] L. Szathmary, A. Napoli, P. Valtchev, Towards Rare Itemset Mining, in: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2007, pp. 305–312.
[21] S. O. Kuznetsov, N. Makhazhanov, M. Ushakov, On Neural Network Architecture Based on Concept Lattices, in: M. Kryszkiewicz, A. Appice, D. Ślęzak, H. Rybinski, A. Skowron, Z. W. Raś (Eds.), Foundations of Intelligent Systems, Springer International Publishing, Cham, 2017, pp. 653–663.
[22] A. Belfodil, S. O. Kuznetsov, C. Robardet, M. Kaytoue, Mining Convex Polygon Patterns with Formal Concept Analysis, in: C. Sierra (Ed.), Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI, ijcai.org, 2017, pp. 1425–1432.
[23] C. Demko, K. Bertet, J. Viaud, C. Faucher, D. Mondou, Description lattices of generalised convex hulls, International Journal of Approximate Reasoning 174 (2024) 109269.

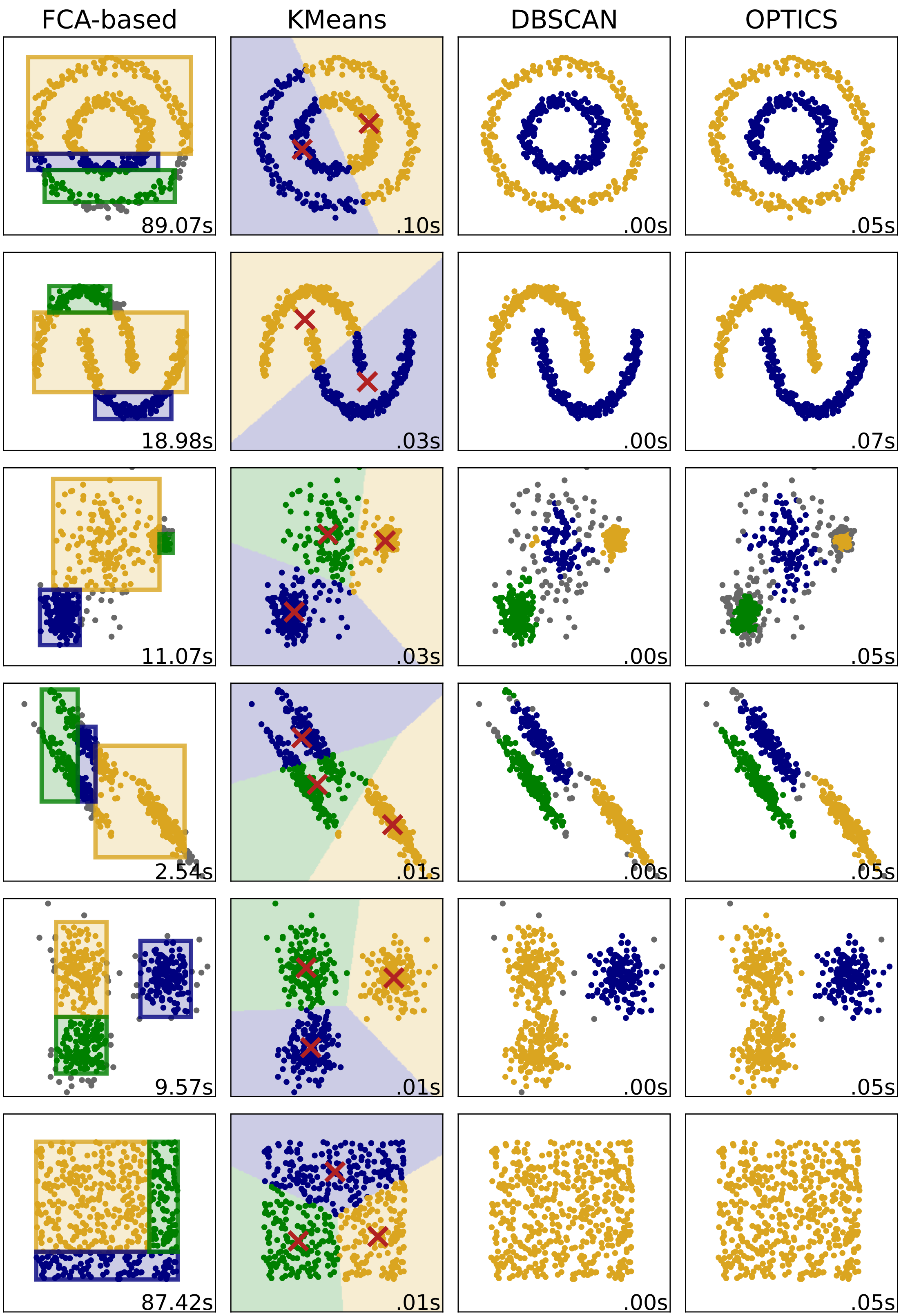


Figure 2: This visual comparison of the clusters produced by the different clustering approaches is inspired by the figure from Sci-Kit learn https://scikit-learn.org/stable/modules/clustering.html. The sets of dots having the same color correspond to clusters while sets of grey dots if any represent objects which are not belonging to any cluster.

Clustering with Axialities

Sergei O. Kuznetsov[1][0000−0003−3284−9001]

HSE University, Pokrovsky bd.11, Moscow 109028, Russia skuznetsov@hse.ru

Abstract. Formal concepts can be considered as rigid biclusters where all objects from the bicluster (formal extent) share all attributes from the intent. Relaxed versions of concept-based bicluster, e.g. OA-biclusters, are also well-known. In this note we show that *axial* (aka monotone, disjunctuve) concepts arising from axialities (adjunctions on powersets of objects and attributes) can help to perform clustering of tricky data like those where clusters are not separable by hyperplanes or present complex dynamical objects, where standard formal concepts and interval patterns would hardly help to catch the required patterns.

1 Introduction

It is well known that Formal Concept Analysis (FCA) presents natural tools for clustering [1]. A formal concept can be considered as a rigid (bi)cluster where all objects of the (bi)cluster (formal extent) share all attributes of the intent, which embodies the similarity of the objects from the extent. Relaxed versions of concept-based bicluster, e.g. OA-biclusters [5, 6] are also well-known. Another well-studied FCA-based clustering model is the one based on interval pattern structures [7]. In this note we show that axial (disjunctuve [9]) concepts arizing from axialities (adjunctions aka residuated mappings or monotone Galois connections on powersets of objects and attributes) [1] can help to naturally cluster tricky data like dynamic streaming data or data of the form 1.1, 2.1, 4.1 in Fig.1, where clusters are dense sets of points with clear connectivity property, so that standard formal concepts and interval pattern concepts would hardly help to catch the required patterns.

2 Definitions and Main Idea

First, let us recall the definitions of (interval) pattern structure and pattern concept [4, 7].

A *pattern structure* [4] is a triple $(G, \underline{\mathbb{D}}, \delta)$, which is a generalization of a formal context (G, M, I) so that G is a set of objects, $\mathbb{D} = (D, \sqcap)$ is a complete semilattice on descriptions from set D with meet (infimum) $\sqcap$, and $\delta : G \to \mathbb{D}$ takes an object from G to its description in D. For any pattern description $d \in \mathbb{D}$ one can define its *pattern extent* $d^\diamond = \{g \in G \mid D \sqsubseteq \delta(g)\}$ and for any subset of objects $A \subseteq G$ one can define its *pattern intent* $A^\diamond = \sqcap\{\delta(g) \mid g \in A\}$. A pair of corresponding pattern extent A and pattern intent d forms a *pattern concept*: (A, d), where $A^\diamond = d, d^\diamond = A$.

The description semilattice of an *interval pattern structure* [7] $\mathbb{D}_{\text{int}}$ consists of tuples of real-numbered intervals $\mathbb{D}_{\text{int}}$, where intervals are ordered by interval subsumption $\sqsubseteq_{\text{int}}$:

$\mathbb{D}_{\text{int}} = \{[l, r] \mid l, r \in \mathbb{R}, l \leq r\}$ and $\forall [l_1, r_1], [l_2, r_2] \in \mathbb{D}_{\text{int}}, [l_1, r_1] \sqcap_{\text{int}} [l_2, r_2] = [min\{l_1, l_2\}, max\{r_1, r_2\}]$ so that $[l_1, r_1] \sqsubseteq_{\text{int}} [l_2, r_2] \iff [l_1, r_1] \supseteq [l_2, r_2]$.

Interval pattern concepts propose a natural way of clustering numerical data as proposed in [7]. The experiments show that interval pattern concepts, whose intents make hyperrectangles with axis-aligned edges and faces can be successfully used for clustering data like 5.1, 6.1 (Fig.1), can be used with less success in clustering data like 3.1 and perform much worse for data of the form 1.1, 2.1, 4.1.

So, in this note we propose another FCA-based tool - called axial (aka disjunctive, monotone [9]) concepts - which can help in clustering data that are hard to cluster using formal or interval pattern concepts.

Let $K = (G, M, I)$ be a formal context, then *axialities* (aka adjunctions or residuated mappings on powersets) [2] are defined for K as

$$^{\leftarrow}A = \{b \in M \mid aIb \text{ for no } a \in G \setminus A\}, \tag{1}$$

$$^{\rightarrow}B = \{a \in G \mid aIb \text{ for some } b \in B\}. \tag{2}$$

where $A \subseteq G$ is a subset of objects and $B \subseteq M$ is a subset of attributes.

An *axial* (or disjunctive [9]) *concept* based on axialities is defined in a similar way as the standard formal concept [3], i.e. as a pair (A, B), where $A \subseteq G, B \subseteq M$ and $A =^{\rightarrow} B, B =^{\leftarrow} A$. Unlike formal concepts, the extents and intents of axial concepts are isotonic, i.e. for two axial concepts (X_1, Y_1) and (X_2, Y_2) one has $X_1 \subseteq X_2$ iff $Y_1 \subseteq Y_2$

While for some clusterization tasks in the left column of Figure 1, like 3,5,6, the generalization of formal concepts to interval pattern concepts fits quite well, the clustering tasks 1,2,4 are hardly well-solvable by means of interval patterns, since they make only axis-aligned hyperrectangles and are insensitive to density and continuity properties of data.

Here we propose to apply axial concepts by first making a transformation of original data, which is well-known in Machine Learning as the "kernel trick"[10].

First, we introduce data model which will be studied further. Let G be a set of data points in a metric space with metric d. Let $A_1, \ldots, A_n$ be disjoint subsets of data points: $A_i \subseteq G$, $A_i \cap A_j = \emptyset$. We call the family of sets $A_1, \ldots, A_n$ (ϵ, k)-dataset if $d(a_i, a_j) > \epsilon$ for every $a_i \in A_i$ and $a_j \in A_j$ where $i \neq j$.

Let us define the following formal context, which we call *ϵ-kernel context*: (G, G, I_ϵ), where $I_\epsilon \subseteq G \times G$ is defined as $(g, h) \in I_\epsilon$ iff $d(g, h) \leq \epsilon$.

Proposition For each cluster A_i there is an axial concept (A_i, A_i) of the context (G, G, I_ϵ).

Proof. By the construction of the context (G, G, I_ϵ) every subset $A \subseteq A_i$ makes the monotone concept (A, A).

Example 1. Consider a simplified example of a dataset of type 2.1 in Fig.1 where the set of data points is $G = \{g_1, \ldots, g_{12}\}$ as in Fig. 2, with $A_1 =$

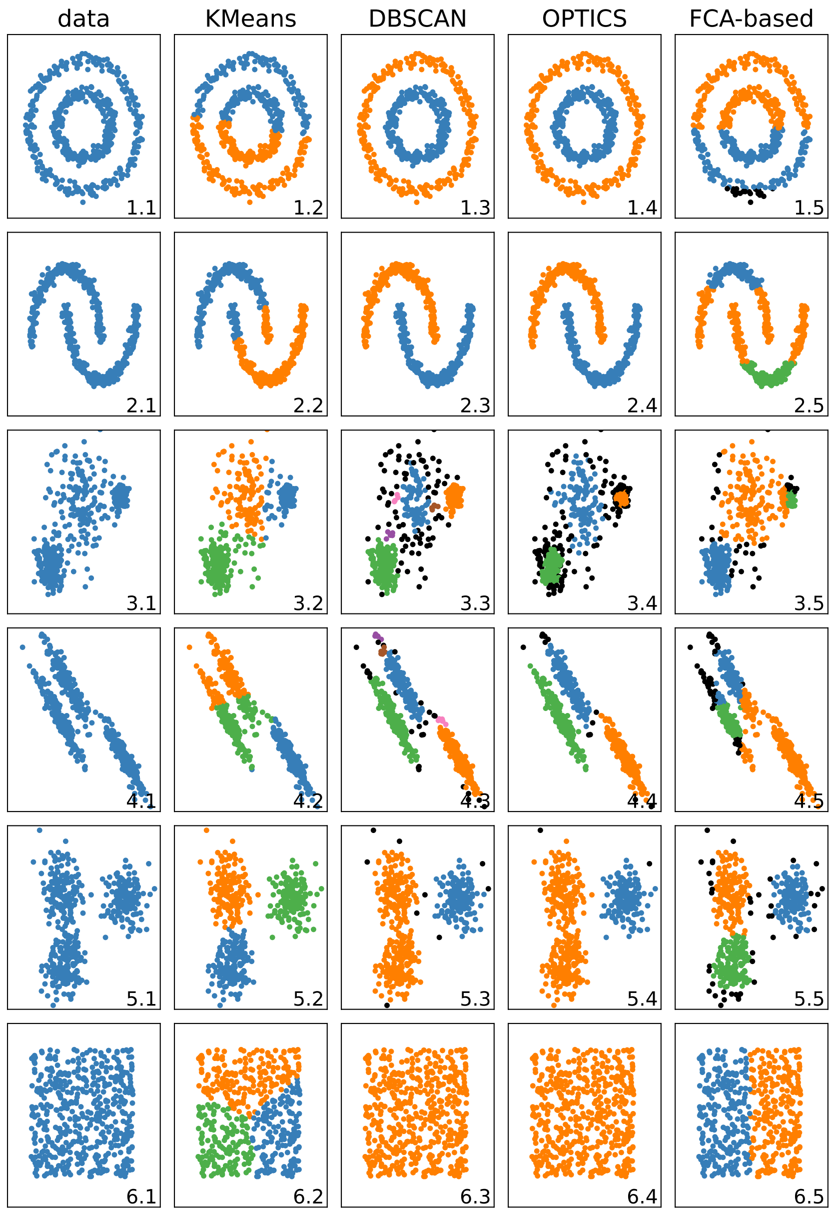


Fig. 1. The left-most column presents clustering data from Sci-Kit learn https://scikit-learn.org/stable/modules/clustering.html. The other columns stay give visual comparison of clusterings based on various approaches: KMeans, DBSCAN, OPTICS and FCA-based. Dots colours correspond to clusters, black dots represent non-clustered objects (outliers).

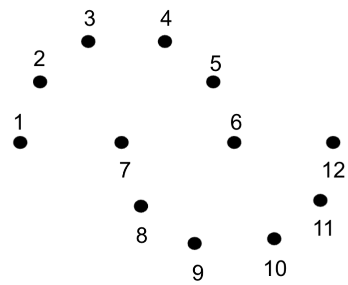


Fig. 2. Data Visualization

$\{g_1, \ldots, g_6\}$, $A_2 = \{g_7, \ldots, g_{12}\}$ and $d(g_1, g_2), d(g_2, g_3), d(g_3, g_4), d(g_4, g_5), d(g_5, g_6) < \epsilon$ and $d(g_7, g_8), d(g_8, g_9), d(g_9, g_{10}), d(g_{10}, g_{11}), d(g_{11}, g_{12}) < \varepsilon$ and for any $g_i \in A_1$ and $g_j \in A_2$ one has $d(g_i, g_j) > \varepsilon$. Then the cross-table of (G, G, I_ϵ) is given in Table 1.

Consider now that ε takes values $\varepsilon_1 < \varepsilon_2 < \varepsilon_3$. For $\varepsilon = \varepsilon_1$ close to zero, the resulting clusters would contain only single points. Increasing ε to $\varepsilon = \varepsilon_2$ we obtain two clusters staying for sets A_1 and A_2. If we increase ε further to $\varepsilon = \varepsilon_3$, the clusters will merge in one.

Similar effects will be observed for data of the types 1.1, 4.1 in Fig.1. As for data of the types 3.1 and 5.1 where there are "bridges" between clusters, let us consider the following example.

	1	2	3	4	5	6	7	8	9	10	11	12
1	x	x										
2	x	x	x									
3		x	x	x								
4			x	x	x							
5				x	x	x						
6					x	x						
7							x	x				
8							x	x	x			
9								x	x	x		
10									x	x	x	
11										x	x	x
12											x	x

Table 1. Context (G, G, I_ε) for ε_2

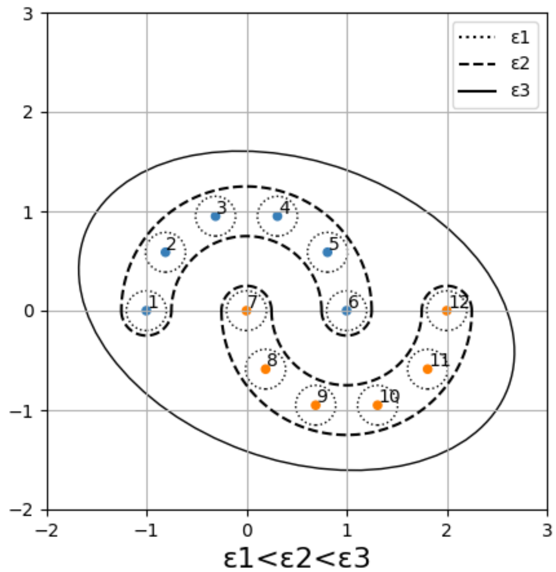


Fig. 3. The dotted lines stay for ε_1, the dashed lines stay for ε_2, and the solid line stays for ε_3.

Example 2. Consider another example in Fig. 4. Here two clusters A_1 and A_2 are not totally disjoint, but have a "bridge" element g_5 shared by both clusters.

	1	2	3	4	5	6	7	8	9
1	x	x	x	x					
2	x	x	x	x					
3	x	x	x	x					
4	x	x	x	x	x	x			
5				x	x	x			
6				x	x	x	x	x	x
7						x	x	x	x
8						x	x	x	x
9						x	x	x	x

Table 2. Context (G, G, I_ε) for ε_2

In Fig.5 we see the diagram of the axial concept lattice for the context in Table 2. Notation $\overline{a, b}$ with $a < b$ denotes the set of elements (both objects and

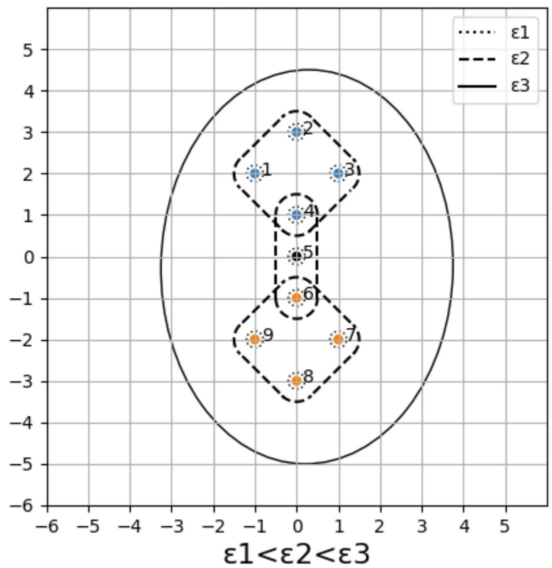


Fig. 4. Clusters A_1 and A_2 share common element g_5

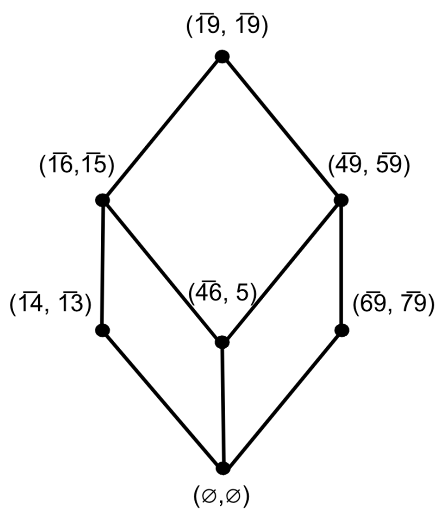


Fig. 5. Diagram of the lattice of axial concepts for the context in Table 2

attributes) $\{a, a+1, \ldots, b\}$. Every concept propose a cluster and every antichain gives a clusterization, where clusters may intersect.

Note that clustering in this case can also be easily performed by using formal concepts $(\overline{1,4},\overline{1,4})$, $(\overline{4,6},\overline{4,6})$, $(\overline{6,9},\overline{6,9})$, $(4,\overline{1,6})$, $(6,\overline{4,9})$, with objects 4 and 6 playing the role of outliers in their clusters.

3 Computing clusters as axial concepts

It is well-known [5] that (A,B) is a axial (disjunctuve) concept of context (G,M,I) iff $(G \setminus A,(G \setminus A)')$ is a formal concept of $(G,M,\bar{I})$. So, to compute axial concepts of (G,G,I_ϵ), one can use standard FCA algorithms like CbO [10]. '1 For example, to compute maximal (both by extent and intent) axial concepts, one can compute minimal extents of $(G,G,\bar{I_\epsilon})$, which can be done in $O(k \times |G|^2)$ time.

Although clusters correspond to axial extents of ε-kernel context, not every extent makes a "good" cluster. For Example 1 with the context in Table 1 every subset $\overline{1,k}$ for $k \in \overline{1,12}$, except for $k=7$, makes an axial extent, however the desired cluster among them is only $\overline{1,6}$, which corresponds to the axial concept $(\overline{1,6},\overline{1,6})$. Consider a CbO-like object-wise strategy of computing axial concepts by adding object $k+1$ to the current axial extent $\overline{1,k}$. Till $k=6$ it runs in a uniform way by adding new row and new column. However, when one tries to add object 7 (or any of the objects 8,9,10,11,12) to the extent $\overline{1,6}$ of the concept $(\overline{1,6},\overline{1,6})$, one again, performing $^{\leftarrow}$ and $^{\rightarrow}$ operations, obtains axial concept $(\overline{1,6},\overline{1,6})$. This actually signifies that objects 7,8,9,10,11,12 have no similarity to objects $\overline{1,6}$ and the construction of the cluster should be terminated, making it $\overline{1,6}$. This observation can be formalized as a general rule as follows: if for a current axial extent A adding any new object and performing operations $\vee$ and $\wedge$ results in the old extent A, then one should output A as a cluster. One can design other similar rules as the "termination criterion" depending on the data and problem setting.

For example, consider data in Fig. 4 with ε_2 and respective context in Table 3. Since 5 has only two neighbors, the respective column and row have only three entries. All other elements have at least three neighbors. So, the algorithm computing axial concepts here may have a termination condition such that if the algorithm gets a row (column) with less than 4 entries, thus outputting two clusters A_1 and A_2 as required.

It is also worth noting that transforming initial data to the ε-kernel context given by a table results in quadratic increase of the data size. The kernel context is a convenient tool for mathematical modeling, but computation of axialities for clustering can be made more efficient if the algorithms are adapted to the initial data representation. Then, instead of traversing rows and columns of the kernel cross-table, one can operate with circles of ε-neighborhoods of the points in original representation.

4 Conclusion

We have proposed an idea of a clustering framework based on ε-kernel trick, axialities, and respective axial concepts, so that clusters correspond to special types of axial concepts of the ε-kernel context related to the original dataset. Axialities propose a natural way to express continuity in clusters, where not all points of a cluster are close to each other, however, as in dynamical data like streaming data, there is a continuous path joining any two points of the cluster. The proposed formalization allows a natural way of computing clusters by means of standard FCA-algorithms. However, not all axial concepts correspond to good clusters, so the main challenge for a particular clustering setting remains to find easily computable conditions that would allow efficiently selecting exactly those axial concepts that correspond to best clustering. The further work would also require extensive experiments for choosing optimal values of parameter ϵ.

5 Acknowledgment

The work by S.O. Kuznetsov in preparing this article was supported by grant 22-11-00323 of the Russian Science Foundation and carried out at the National Research University Higher School of Economics, Moscow.

References

1. C. Carpineto and G. Romano, A lattice conceptual clustering system and its application to browsing retrieval. Machine learning (1996), 24, 95-122.
2. M. Erné, Adjuunctions and Galois Connections. Origins, History and Development, in K. Denecke, M. Erné, S.L.Wismath (Eds.), Galois Connections and Applications, Springer, 2004.
3. B. Ganter and R. Wille, Formal Concept Analysis, Mathematical Foundations (Second Edition), Springer (2024).
4. B. Ganter, S.O. Kuznetsov: Pattern Structures and Their Projections. Proc. ICCS 2001: 129-142.
5. D.I. Ignatov, S.O. Kuznetsov, J. Poelmans, L.E. Zhukov: Can triconcepts become triclusters? Int. J. Gen. Syst. 42(6): 572-593 (2013).
6. D.I. Ignatov, D.V. Gnatyshak, S.O. Kuznetsov, B.G. Mirkin: Triadic Formal Concept Analysis and triclustering: searching for optimal patterns. Mach. Learn. 101(1-3): 271-302 (2015).
7. M. Kaytoue et al. "Mining gene expression data with pattern structures in formal concept analysis." Information Sciences 181.10 (2011): 1989-2001.
8. S.O. Kuznetsov, A fast algorithm for computing all intersections of objects from an arbitrary semilattice. Nauchno-Tekhnicheskaya Informatsiya Ser. 2 - Informatsionnye protsessy i sistemy, No. 1, pp.17-20, 1993.
9. S.O. Kuznetsov, N. Makhazhanov, M. Ushakov: On Neural Network Architecture Based on Concept Lattices. Proc. ISMIS 2017: 653-663
10. B. Schölkopf, A.J. Smola, F. Bach, Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2018.

A system for different concepts generation and application

Xenia Naidenova[1,*,†], Victor Shagalov[2,†] and Tatiana Martirova[3,†]

[1,3] *Medical Academy, St.-Petersburg, 194044 Academic Lebedev Street, Russia*
[2] *MatrixDnA Ltd, Israel*

Abstract
In this paper, programming library and algorithms for solving formal concept related tasks in real world domains are presented. The main goal of the proposed system is the searching of all closed itemsets (concepts). Constructing Galois lattice of concepts allows to additionally generate good classification tests and functional dependences for given classifications on a given data set. In general, these tasks are based on ordinal procedure for shallow or deep machine learning for classifications. We show that formal concept analysis is closely related to modeling plausible classification reasoning

Keywords
formal concepts, good classification tests, functional dependencies, plausible reasoning

1. Introduction

Modern accent in Machine Learning (ML) is shifted to the numerical solutions as opposed to plausible reasoning. Of course, the linear additive model or kernel model allows great data compression but at the same time the source of information is lost. On the other hand, the Formal Concept Analysis (FCA) has a native ability to model plausible reasoning. When there is no explanation based on a model that is difficult for understanding and sometimes conflicting with human sense, then the obtained results are not reliable. Obviously, integration of plausible reasoning with the FCA as one of the instruments of ML is crucial in the context of AI.

The problem of finding all closed sets (concept lattice) has been solved by many researchers: B. Ganter, D. Borchmann, M. Zaki, S. Kuznetsov, and many others. The source for many of these works was the algorithm of B. Ganter [1]. The Nex-Closure algorithm was proposed in [2] as an improvement of previous versions of this algorithm. One of the most efficient algorithms, Charm, has been proposed by M. Zaki in [3]. The algorithm presented in this paper is based on a previously developed algorithm for extracting only good classification tests (GCTs) [4] from a given context. The algorithm uses the original decomposition of the source context into the attributive and object sub-contexts described in [5].

In the paper [4], it was shown that good classification tests (GCTs) are formal concepts and therefore they are contained in the Galois lattice built over a given context with additional attribute(s) that specify the partitioning of context's objects into non-overlapping classes. However, all the algorithms developed for deriving GCTs as formal concepts did not aim to build and did not build the complete Galois lattice over a given context, on the contrary, these algorithms generate only those elements of the lattice that correspond to all good classification (diagnostic) tests (redundant and non-redundant, i.e. test generators).

* Corresponding author.
† These authors contributed equally.
ksennaidd@gmail.com (X. Naidenova); shagalovv@gmail.com (V. Shagalov); martta462@yandex.ru (T. Martirova)
0000-003-2377-7093 (X. Naidenova); 0009-0001-7131-9081 (V. Shagalov); 0000-0003-0000-6608 (T. Martirova)

The first algorithm for generating good maximally redundant classification tests (GMRTs) was implemented in system SISIF [6], but it had a very small memory. In addition to GMRTs, the SISIF also has generated functional dependencies (FDs) as the best approximation of a given classification of given objects. The system also implemented an algorithm for extracting all generators from a given GMRT, equivalent to it. An overview of the main algorithms developed for building GCTs can be found in [5].

This paper presents a new system for extracting the different types of itemsets (concepts, dependencies, logical rules, classification tests) based on constructing the lattice of all closed frequent concepts in a given context. This system has the following features:

1. Work with large datasets;
2. Work with multivalued attributes of objects;
3. Well-structured and simple for usage;
4. Applicable for multiple FCA task;

Further, the work is organized as follows. Section 2 gives basic definitions related to the FCA, GCTs, and plausible reasoning rules. Section 3 describes Diagnostic Test Machine (DTM) as a software library for finding different concepts and logical rules in data sets. Section 4 briefly describes the experiments. Section 5 deals with the plausible reasoning rules application, and Section 6 offers some concluding remarks and describes some future investigation.

2. Basic definition

Let $S = \{1, 2,\ldots, N\}$ be the set of objects' indices (objects, for short) and $T = \{A_1, A_2, \ldots, A_j, \ldots A_m\}$ be the set of attributes' values (values, for short). Each object is described by a collection of values from T. Let $s \subseteq S$, $t \subseteq T$. Denote by t_i, $t_i \subseteq T$, $i = 1,\ldots, N$ the description of object with index i.

The definition of good test is based on two mapping $2^S \rightarrow 2^T$ and $2^T \rightarrow 2^S$ determined as follows:

$t = \text{val}(s) = \{$intersection of all t_i: $t_i \subseteq T, i \in s\}$ and

$s = \text{obj}(t) = \{i: i \in S, t \subseteq t_i\}$.

Of course, we have obj(t) = {intersection of all $s(A)$: $s(A) \subseteq S$, $A \in t$}. Operations val(s), obj(t) are reasoning operations related to discovering the general feature of objects the indices of which belong to s and to discovering the indices of all objects possessing the feature t.

The basic operator of plausible reasoning [3] connecting it with the FCA, is the generalization rule (GR) defined as follows:

generalization_of(t) = t' = val(obj(t)): generalization_of(s) = s' = obj(val(s)).

Galois Lattice consists of closed pairs (s, t) called concepts and defined by the generalizing rule: val(obj(t)) = t, obj(val(s)) = s.

In general, the concept has maximal coverage of examples of some dataset by a given itemset that cannot be extended by any other attribute to get the same sample coverage.

2.1. Classification (diagnostic) tests

In classification problems, each object has a class label, which is not part of the domain description. Labeling is a kind of partitioning of a data set or ontology.

Let $S(+)$ and $S(-) = S\backslash S(+)$ be the sets of positive and negative class of objects, respectively.

A **diagnostic (classification) test for *S*(+)** is a pair (s, t) such that $t \subseteq T$ ($s = \text{obj}(t) \neq \emptyset$), $s \subseteq S(+)$ and $t \not\subset t'$ $\forall t'$, $t' \in S(-)$.

A diagnostic test (s, t), $t \subseteq T$ ($s = \text{obj}(t) \neq \emptyset$) is **good** for $S(+)$ if and only if any extension $s^* = s \cup i$, $i \notin s$, $i \in S(+)$ implies that (s^*, val(s^*)) is not a test for $S(+)$.

It means that if (s, t) is a good test for $S(+)$, then s of it is non-extendable, i. e. adding to s any i from $S(+)$ not belonging to s implies that for val($s \cup i$) there exists such a $t' \in S(-)$ that val($s \cup i$) $\subseteq t'$.

A good test (s, t), $t \subseteq T$ (s = obj(t) ≠ ∅) for $S(+)$ is **irredundant** (GIRT) if any narrowing $t^* = t \backslash A$, $A \in t$ implies that (obj(t^*), t^*)) is not a test for $S(+)$.

A good test (s, t) for $S(+)$ is **maximally redundant** (GMRT) if any extension of $t^* = t \cup A$, $A \notin t$, $A \in T$ implies that (obj(t^*), t^*) is a test for $S(+)$, but not a good one.

To align the above original definitions with FCA terminology, they can be redefined as:

Maximal test (MT) for S+ is closure itemset (CI) (s, t) with confidence = |s+|/|s|.

Irredundant test (IT) for S+ is CI (s, t) if any narrowing $t^* = t \backslash A$, $A \in t$ implies that obj(t^*) ≠ s.

The goodness of a diagnostic test turns into its confidence.

2.2. Functional dependencies

Functional dependency $X \rightarrow C$ is a relation between the collection $X \subseteq T$ of attributes and the given classification C of objects into classes $C1$, ..., Ck. Denote by P(X) = {$p1$, $p2$, …., pm) the partition of S generated by the values of the collection of attributes X, where pj, j = 1, 2, …, m, $m \geq k$, are classes of P(X), each of which is associated with one and only one collection of values of attributes of X. The definition of functional dependency between attributes is based on the definition of the relation of partial order over the set of partitions generated by the set of considered attributes. This relation is introduced as follows: P(X) ≤ P(Y) iff P(X) ⊆ P(Y), X, $Y \subseteq T$.

A pair P(X), P(Y) are said to be in the inclusion relation iff every block of P(X) is contained in one and only one block of P(Y).

If P(X) = P(C), then X is the ideal approximation of classification C or ideal test based on a functional dependency. If this condition is not satisfied, then X, $X \subseteq T$, X corresponds to a good approximation of C, if P(X) is the closest to P(C) element of Partition Lattice over a given context, i. e., for all P(Y), $Y \subseteq T$ condition (P(X) ⊂ P(Y) ⊆ P(C)) implies P(X) = P(Y). In this case, we said that $X \rightarrow C$ is a FD in T. FD in the form $X \rightarrow Y$ is known as conditional FD.

In [4], A method is given to transform initial contexts into the contexts for searching for FDs by any algorithm of discovering GMRTs.

2.3. Implicative dependencies as plausible rules of the first type

In this paper, we focus on conceptual knowledge the main elements of which are objects, properties (attribute values), and classifications (attributes). Taking into account that implications express the links between concepts (object ↔ class, object ↔ property, property ↔ class) we deem classification reasoning to be based on using and searching for only one type of logical dependencies, namely, implicative dependencies.

Implicative dependences are the result of GCTs inferring. Consider, for example, a GMRT as a pair (obj(t), t). In this pair, t is a collection of attribute values, $t \subseteq T$, and |obj(t)| is the support of t, and obj(t) $\subseteq S(+)$. Thus, we can form an implicative rule $t \rightarrow S(+)$. This assertion is transformed in a reasoning rule. The left part of this rule is t (a set of values from T) and $S(+)$ can be the name of a class in the classification of S.

Implicative assertions are considered as plausible rules (PR) of the first type. Generally, we have the following rules of the first type (the left part of rules can contain any number of different values from a given context): **Implication**: a, b, c, …→ d. **Interdiction or forbidden rule**: a, b, c, … → *false* (*never*). This rule can be transformed into several implications such as a, b, … → not c; a, c, … → not b; b, c, … → not a. **Compatibility (associations)**: a, b, c, … → VA, where VA is the frequency of rule's occurrence (related to the confidence of the left part of this rule). Generally, the compatibility rule represents a most frequently observed combination of values. **Diagnostic rule**: x, $d \to a$; x, $b \to$ not a; d, $b \to false$. For example, d and b can be two values of the same attribute. This rule works when the truth of 'x' has been proven and it is necessary to determine whether 'a' is true or not. If 'x & d' is true, then 'a' is true, but if 'x & b' is true, then 'a' is false. **Rule of alternatives**: a or $b \to true$ (*always*); a, $b \to false$. This rule says that 'a' and 'b' cannot be simultaneously true, either 'a' or 'b' can be true but not both. In the rules, $a, b, c, d, \in T$, $x \subset T$.

The plausible reasoning rules of the first type are formed from GCTs (maximally redundant and non-redundant). Let X_1, X_2 and Y_1, $Y_2 \subseteq T$ be good maximally redundant and good maximally non-redundant classification tests. Let $X_1 \to q_1$, $X_2 \to q_2$, $Y_1 \to q_1$, $Y_2 \to q_2$ be implications, where q_1, $q_2 \in GOAL$, are two different classes of objects. We can form the following forbidden rules: $X_1 \to$ not q_2, and X_2, $\to$ not q_1.

The rules of alternative: "a or $b \to true$; a, $b \to false$" is indeed the case when a and b are values of the same attribute.

The diagnostic rule can be obtained from two good maximally non-redundant test. For example, compute $int = Y_1 \cap Y_2$. Then we have diagnostic rule: 'if int is *true*, then $int \cup (Y_1 \backslash int) \to q_1$; $int \cup (Y_2 \backslash int) \to q_2$.

Compatibilities rules can be obtained from frequent associations.

3. Diagnostic Test Machine

Diagnostic Test Machine (DTM) is a software library for finding, implicative and functional dependencies in data sets. All dependencies generated by the system are redundant and frequent, until otherwise explicitly declared. In particular. The DTM finds all value-based (like in the Charm algorithm [3]) and attribute-based frequent formal concepts that are independent of the final task. Once the lattice of concepts is found, it can generate all good (confident) maximally redundant diagnostic (classification) tests (GMRT) and good approximating FDs for a given classification (partition of objects). This step is task-dependent (Figure 1). It also has the ability to generate all non-redundant implications from redundant dependencies and associative rules.

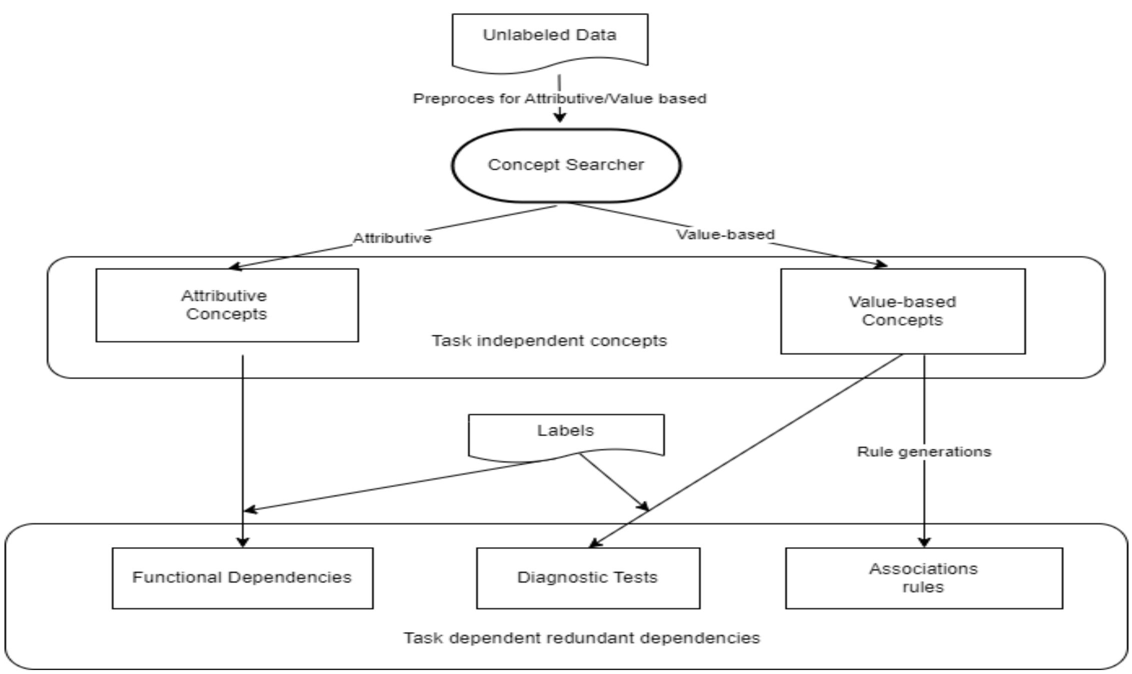


Figure 1: The tasks flow diagram in the DTM

The library is applicable for a number of scenarios and purposes, but mainly:
- to construct FDs and simultaneously the dimensionality reduction in initial data;
- to construct GCTs for classification task.

Below some details for main steps and details of the library implementation are listed.

3.1. Row data preprocessing.

The library supports categorical (ordinal/nominal) and numerical (discrete/continuous) domains for attributes. One of the main problems in this area is working with numeric attributes. In addition to the trivial simple partitioning into equal wide ranges, the library includes additional methods to solve this problem, namely: Minimum Description Length (MDL) [8] and the Kolmogorov-Smirnov algorithm. These methods must be provided with a target partition. There are two obvious options for this: to use a forward-defined classification or, in the case of categorical attributes, to use its composite partitioning. Despite this, there are still questions on this issue.

We transform the row data to dual horizontal - vertical bit vector representation. This allows to work effectively with dense datasets (like "Mushrooms") due to having the equal width records and sparse ones like any store of transactions db.

3.2. Concepts searching algorithm

This algorithm is described in [6]. It is based on the procedure of decomposing the main task into attributive and object subtasks (projections) most fully described in [7].

The root of search tree or initial task (Alg.1, init_task) is built on a given preprocessed training set. It can be attribute-based or sample (object)-based (transposed) task, which initiates the search from the join (lower bound) or meet (upper bound) of the lattice [14]. The choice of attribute or object together with the lattice traversal strategy provides a powerful basis for implementing various algorithms for the FCA problems. Only a row coverage vector is used for concept representation, which correspondents to the 'rows' vector in the Task structure (see Main structures). The algorithm recursively decomposes the current task into depth-first search subtasks, selecting attribute/value according to selected strategy.

The search tree generates only closed elements of concept lattice (closed itemsets) (Alg. 1, find_concepts) and does not produce any redundant subtasks. The traversal strategy (attribute/sample selection) may vary depending on the task. So, if the task is to find all frequent concepts, the optimal strategy will be to select the attribute with the minimum support, but when the task is diagnostic, the strategy with the maximum support will be much more reliable. Once the attribute is selected, the subtask corresponding to some concept is formed using the generalization rule (Alg. 1, sub_task). Of course, the search tree could achieve the same task in several ways. The logic of cutting off a dead-end or solved subtasks and stopping the search is also encapsulated (Alg. 1, add_concept).

Some of the main structures and operations on them are defined below. The "." operator provides access to the structure fields.

Main structures
BitVector : Operations: & - bitwise and operation ∨ - bitwise or operation ¬ - bitwise not operation weight() - sum of all bit values

```
DualBitMatrix – structure effectively supports dataset (DS) representation (horizontal and
    vertical)

    Fields:
      rows : [BitVector] // set of BitVector corresponding to each example in the DS
      cols :   [BitVector] // transposed rows BitVector set  for each attribute in the DS
      height: int // number of rows
      width:  int // number of cols

    Operations:

      BitVector & (BitVector vector)   // returns intersection of given  rows/cols
      BitVector ∨ (BitVector vector)   // returns union of given rows/cols
```

```
Task – subset of DS (DualBitMatrix) in both dimension. It is corresponds to concept and defined
    by the generalizing rule: val(obj(t)) = t

    Fields:
        cols   : BitVector      // cols subset of the DS
        rows  : BitVector    // rows subset of the DS
        cross : BitVector     // the task rows intersection db.&( rows )
```

```
Lattice –  structure consists of founded concepts and responsible for  search tree pruning

    Fields:
        concepts :   { BitVector } //set of concept
        minsup : int   // minimal support threshold
```

Algorithm 1. Frequent concepts search procedure

```
Input:  db : DualBitMatrix, minsup: int // training set, minimal support
Output: L: Lattice

T = init_task(db)
L = Lattice (Ø, minsup)

find_concepts(T, L) begin   // traversal of the task lattice
    if add_concept(L, T.rows) then
        while (sub_T = select_subtask(T, strategi)) is not null do
            find_concepts(sub_T, L)
            T.cols = T.cols & ¬sub_T.cross // removes subtask
        end while
    end if
end find_concepts

init_task(db) begin
    rows  = ¬BitVector(db.height)
    cols  = ¬BitVector(db.width)
    cross = db.&(rows)
    return Task(rows ,  cols, cross)
end init_task
```

```
select_subtask(Task t, strategy) begin
     a =  find_best_sub_task(t, strategy) // return best attribute according the strategy
    if a >=0 then
        return sub_task(t, a)
    else
        return null
    end if
end select_subtask

sub_task(Task t, int a) begin // get sub task/concept by given attribute
    rows  := t.rows  &  db.cols[a]        // t = obj(a)
    cross := db.&(rows)                   // s = val(t)
    cols  := t.cols  & ¬cross
    return Task(rows ,  cols, cross)
end sub_task

add_concept(L, c) begin
    support = c.weight()
    if support < L.minsup then
        return false
    else if c ∈ L.concepts then // all subtask were solved
        return false
    end if
    L.concepts= L.concepts ∪{c}
    return true
end add_ concept
```

3.3. Maximal tests generator for given classification

Once we have all frequent concepts, obtaining all MTs (frequent implications) is as trivial as intersecting of the goal vector (bit vector with ones for the target class objects) with the concept and thresholding the result by the minimum confidence parameter (Alg. 2).

```
Algorithm 2. Concept to maximal test procedure
Input:
goal : BitVector,  concept : BitVector
    minconf: float [0:1]  // minimal confidence
Output:
    implication : (concept, confidence)-> goal

concept_to_implication(goal, concept , minconf) begin
    goal weight = (concept & goal).weight()
    concept weight = concept. weight()
    float confidence = goal weight /concept weight
    if( confidence >= minconf) then
        return (concept, confidence)-> goal
    else then
        return null
    end if
end concept_to_implication
```

3.4. Diagnostic task

The diagnostic or classification task is to assign an unlabeled example to a certain class for which tests were obtained in the previous step. One problem here is that the tests are generally redundant. But the task to generate all non-redundant tests has the exponential complexity. Therefore, the DTM bypasses the problem with a simple check below (Alg.3).

As mentioned earlier, the concept has a dual representation of objects/attributes, and the algorithms described above use only the first one. Of course, the diagnostic task requires the second representation, the creation of which is trivial for the given training dataset and has been omitted here. Therefore, the test structure used below has both representations (rows and columns).

Algorithm 3. Procedure for checking the equivalence of coverings

```
Input: sample: BitVector, Test test, BitMatrix db
Output: Boolean

test_sample(sample, test,  db) begin
        BitVector u  = test.cols & sample;
        return test.rows = db.&(u);
end  test_sample
```

The project code and some other datasets can be found at https://gitlab.com/shagalovv/dtm

3.5. Example

To illustrate the process, we use a small dataset from [3] (Table 1). The original data is transformed into an internal dense representation with an additional column, which is the external classification. The classification column will be masked during the concept discovery stage. Now, the Examples are presented in Tables 2-6.

Table 1: Raw dataset

Object Index	Itemset
1	A C T W 0
2	C D W 0
3	A C T W 0
4	A C D W 1
5	A C D T W 1
6	C D T 1

Value-based dependencies are in Table 2.

Table 2: Closed frequent itemsets (min confidence = 1)

N	Support	Objects	Items
1	1	5	ACDTW
2	3	1 3 5	ACTW
3	2	4 5	ACDW
4	3	2 4 5	CDW
5	4	1 3 4 5	ACW

6	5	1 2 3 4 5	CW
7	2	5 6	CDT
8	4	1 3 5 6	CT
9	4	2 4 5 6	CD
10	6	1 2 3 4 5 6	C

Table 3: Frequent tests in the case (min confidence = 1)

N	Support	Confidence	Goal	Tests
1	1	1	GOAL[1]	A C D T W
2	2	1	GOAL[1]	A C D W
3	2	1	GOAL[1]	C D T

Functional dependencies: dense source data is transformed to the data for functional dependencies search (with no duplicates for brevity). As in value-based task, the classification column will be masked on concepts discovery stage.

Table 4: Transformed raw data for attributive task

N	A	C	D	T	W	GOAL
1	0	1	1	0	0	0
2	0	1	0	1	0	0
3	1	1	0	0	1	0
4	0	1	1	0	1	0
5	1	1	0	1	1	0
6	0	1	1	0	0	1
7	0	1	1	1	0	1
8	0	1	0	0	1	1
9	1	1	1	0	1	1
10	1	1	0	1	1	1

Table 5:Intents of concepts containing frequent functional dependencies (min support = 1)

N	Support	Objects	Intents of concepts
1	1	7	CDT
2	1	9	ACDW
3	2	4 9	CDW
4	5	1 4 6 7 9	CD
5	2	5 10	ACTW
6	4	2 5 7 10	CT
7	4	3 5 9 10	ACW
8	6	3 4 5 8 9 10	CW
9	10	1 2 3 4 5 6 7 8 9 10	C

Table 6: Frequent functional dependencies for the given classification (min confidence = 1)

N	Support	Confidence	Goal	Left part of dependency
1	1	1	GOAL[0]	C D T
2	1	1	GOAL[0]	A C D W

4. Experiments

For the DTM performance testing experiments, the well-known Mushroom dataset and the lesser-known Adult dataset were used, see Table 7. Both were shuffled and split in a ratio of 80% training set to 20% testing set

Table 7: Datasets description

Data sets	Type	Attributes per Types	Records №	Issues
mushrooms	dense	22- categorical + label	8124	missing values
adults	dense	8 - categorical 6 - numerical + label	32561	missing values, class imbalance, repeated samples

The search processes are controlled by a search strategy for selecting subtasks by attributes. Namely, the strategies are: "support (max)" (the choice of attributes with max support, "the unordered or left-to-right choice of attributes" (uno), and "maximum support (min)" (the choice of attributes with min support).

Table 8 shows the results of the search for value-based concepts, and Table 9 shows the results of searching for the diagnostic tests. The number of solved subtask/time in the Table 8 is determined for concept task only. The '-' means the absence of data.

Table 8: Value-based concepts result for min support 1

Data sets	Task dimensions	Concepts №	Coverage	Number of Solved subtasks/time(ms): min/uno/max
mushrooms	6499 x 116	212959	6499	301718/ 2189 793889/ 5533 1447275/ 12161
adults	26048 x 146	2037104	26048	2456837/ 52693 7522715/ - 20394837/ -

Table 9: Test search results for min confidence 1

Data sets	Class/members №	Tests №	Coverage
Mushrooms	p /3161 e/ 3338	76855 78867	3161 3338

Adults	≤ 50K/ 19729	683836	17257
	> 50K/ 6319	55822	4260

In Tables 8 and 9, "Coverage" means the number of objects in the given context belonging to at list one of obtained tests.

Table 10 shows the results of the search for both functional dependencies and conditional ones.

Table 10: Functional dependencies search results for min support 1 and confidence 1

Data sets	Task dimensions	Attributive Concepts №	FDs №	Solved tasks № /time(ms): min/uno/max
mushrooms	16901x 22	202150	27254	225445/ 3497 332896/ 5391 438333/7161
adults	23879x 14	12288	0	12288/ 248 12288/234 16384/303

5. Plausible rule application

The FCA is certainly one of the most powerful tools for analyzing data and building knowledge models based on the lattice of formal concepts extracted from a learning context. Remarkable introduction to the FCA and its applications in information retrieval and related fields is contained in [9].

However, the FCA has a number of drawbacks, one of which should be recognized as the impossibility of directly using formal concepts in the tasks of classifying objects. Computer knowledge structures are traditionally declarative, mechanisms of their using are separated from them and, as a rule, these mechanisms are often fixed.

Currently, various methods for building classifiers are proposed based on concepts extracted from training contexts. These methods use several ideas: 1) forming formal concepts as classifiers and recognizing classes of new objects by navigating through the levels of the conceptual lattice [10, 11]; 2) transition from classifiers constructed by methods other than the FCA to a lattice of formal concepts containing only concepts associated with the decision rules of these classifiers [12].

The first method is quite cumbersome. Essentially, it's about extracting concepts whose extents contain objects of only one class. To do this, the authors in [11] move from the two-digit to the nominal (multivalued) description of objects and introduce the labeling of objects into a context. Now, a nominal (multi-valued) context is a quadruple $\langle I_{nom}, A_{nom}, ς, R_{nom}\rangle$, where I_{nom} is the set of n_{nom} instances, A_{nom} is the set of m_{nom} attributes, ς is the set of values, R_{nom} is a relation defined between I_{nom}, A_{nom} and ς. R_{nom} is a set of triples.

A similar idea, but more easily implemented, is given in [13]. In [12], the decision tree is considered as a set of classification rules and a method for transforming the constructed decision tree over a given context into an isomorphic lattice of concepts is proposed.

The extraction of GCTs is the basis for obtaining rules for classification plausible reasoning.

Consider plausible reasoning rules of the second type and a model of plausible inference.

Let *x* be a pattern (a set of true values of some attributes observed simultaneously). Our goal is to define the target value, i.e. the label of a possible class of objects to which this pattern can be associated. Deductive steps of reasoning consist of inferring consequences from some observed values with the use of the rules of the first kind (i.e., knowledge).

Using implication: Let r be an implication, left(r) and right(r) be the left and right part of r, respectively. If left(r) $\subseteq$ *x*, then *x* can be extended by right(r): $x \leftarrow x \cup$ right(r). **Using interdiction**: Let r be an implication $x \rightarrow$ not *k*. If left(r) $\subseteq x$, then *k* is the forbidden value for all extensions of *x*. **Using compatibility**: Let r = '*a*, *b*, *c, …* $\rightarrow$ *k*, VA (confidence of the rule)'. If left(r) $\subseteq x$, then *k* can be used to extend *x* along with the calculated value VA for this extension. **Using diagnostic rules**: Let r be a diagnostic rule such as 'x, *d* $\rightarrow$ *a*; *x*, *b* $\rightarrow$ not *a*', where '*x*' is *true*, and '*a*', 'not *a*' are hypotheses or possible values of some attribute. Using diagnostic rule implies to infer whether '*a*' or 'not a' is true.

The rules listed above are the rules of "forward inference". Another way to include the first-type rules in natural reasoning can be called "backward inference". **Generating hypothesis** or abduction rule. Let r be an implication y $\rightarrow$ *k*. Then the following hypothesis is generated "if *k* is *true*, then *y* may be *true*".

When applied, the above rules generate the reasoning, which is not demonstrative. The purpose of reasoning is to infer all possible hypotheses on the value of some target attribute. It is essential that hypotheses do not contradict with knowledge (the first type rules) and the observable real situation under which the reasoning takes place. Inference is reduced to obtain all intrinsically consistent extensions of *x*, in which the number of involved attributes is maximum possible and there are no prohibited pairs of values in such extensions. All hypotheses have different admissibility, which is determined by the quantity and "quality" of rules of compatibility involved in inferring each of them.

As a result of learning, we can form the following knowledge bases (KB): the Attribute Base (AtB), containing the relations between problem domain concepts (Ontology), and the Assertion Base (AsB), containing the assertions, formulated in terms of the concepts, and the rules of the first type obtained from learning context. Let a request to the KB be: SEARCHING VALUE OF class of object IF (an observable pattern of object's values = *x*).

Step 1. Take out all the assertions *as* $\in$ AsB containing at least one value from the request *x*. **Step 2**. Delete from the set of selected assertions all of these that contradict with the request. Assertion contradicts with the request if it contains the value of an attribute which is different from the value of this attribute in the request. **Step 3**. Take out the values of attributes appearing in remaining assertions. If we have several hypotheses (several names of target classes), an attempt is made to refute one of the hypotheses. For this goal, it is necessary to find a forbidden rule containing one of the hypotheses, some subset of values from the request and does not contain any other value. **Step 4**. If we have not a hypothesis or we cannot refute the existing hypotheses, then an attempt is made to find a value of some attribute that is not in the request (in order to extend the request). For this goal, it is necessary to find an assertion (implication) that contains a subset of values from the request and one and only one value of some new attribute which are not in the request. For extending request, the compatibilities rules can also be used. The extending obtained must not contain any forbidden set of values. **Step 5**. Forming the extended request. Steps 1, 2, 3, 4 are repeated.

The process of pattern recognition can require inferring new rules of the first type from data when i) the result of reasoning contains several hypotheses and it is impossible to choose one and only one of them (uncertainty), and ii) there does not exist any hypothesis.

6. Conclusion

In this paper, a system for solving formal concept related tasks in real world domains is presented. The main goal of the system is the searching for all closed itemsets (concepts). Constructing Galois lattice of concepts allows to additionally generate GCTs and approximating

FDs for given classifications on a given data set. In general, these tasks are based on ordinal procedure for shallow or deep machine learning for classifications. We show that the FCA is closely related to modeling plausible classification reasoning.

In future work, we plan to implement a fully scalable incremental version of the algorithm for distributed computing to cope with truly “big data” problems. We plan also to improve the lattice navigation to reduce some dead ends in the context of probabilistic reasoning.

Another urgent task is to create a system for generating plausible reasoning rules and models of plausible reasoning based on constructing and browsing a lattice of concepts.

References

[1] B. Ganter and K. Reuter. Finding all closed sets: A general approach. Oder, Vol. 8, N 3, pp. 389-290, 1991.

[2] D. Borchmann. Next-Closure Algorithm – enumerating semilattice elements for a generating set. CLA 2012, pp. 9-20, 2012.

[3] Zaki, Mohammed & Hsiao, Ching-Jiu. CHARM: An efficient algorithm for Closed Itemset Mining, in Proceeding of the Second SIAM International Conference on Data Mining, 2002, pp. 457-473. doi: 10.1137/1.9781611972726.27.

[4] Naidenova, X., Good Classification Tests as Formal Concepts, in: F. Domenach, D.I. Ignatov, and Poelmans (Eds.): ICFCA 2012, LNAI 7278, Springer-Verlag Berlin Heidelberg 2012, 226-226. doi: 10.1007/978-3-642-29892-9.

[5] Naidenova, X, An Incremental learning algorithm for inferring logical rules from examples in the framework of the common reasoning process, in: Triantaphyllow E., and Felici, G. (Eds.), Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques, Massive Computing Series, Springer, Heidelberg, Germany, 2006, pp. 89-147.

[6] Naidenova, X. A., Plaksin, M.V., and Shagalov, V. L., Inductive inferring all good classification tests, in: Valkman J. (Ed.), “Knowledge-Dialog-Solution”, Proceedings of the International Conference, volume 1, Kiev, Institute of Applied Informatics Press, 1995, pp. 79–84.

[7] Naidenova, X., Parkhomenko V. Contribution to attributive or object sub-contexts in inferring good maximally redundant tests. Discrete Applied Mathematics, Vol. 273, pp. 217-231, 2020.

[8] Fayyad, Usama M., and Keki B. Irani, Multi-interval discretization of continuous valued attributes for classification learning, IJCAI, Vol. 93, N 2, pp. 1022-1027, 1993.

[9] Dmitry I. Ignatov, Introduction to Formal Concept Analysis and its Applications in Information Retrieval and Related Fields. arXiv: 1703.02819v1[csIR]8Mar2017.

[10] Hayfa AZIBI, Nida Meddouri and Mondher Maddouri, Survey on formal concept analysis based supervised classification techniques, in A.I. Tallón-Ballesteros and C. Chen (Eds.), Mchine learning and Artificial Intelligence, the author and IOS Press, 2020, pp. 21-29. doi:10.3233/FAIA200762.

[11] Nida Meddouri and Mondher Maddouri, efficient Closure Operators for FCA Based Classification, International Journal of Artificial Intelligence and Machine Learning, vol. 10, no. 2, pp 79-98, 2020. doi: 10.4018/IJAIML.2020070105.

[12] Egor Dudyrev, Sergey O. Kuznetsov, Decision Concept Lattice vs. Decision Trees and Random Forests, arXiv: 2106.00387v1[csLG]1jun2021 (pp.1-8).

[13] T. Makhalova, S.O. Kuznetsov, and Amedeo Napoli, Closure structure: a deeper insight, in S.O. Kuznetsov, A. Napoli, and S. Rudolph (Eds.), Proceedings of Fights International Workshop “What can FCA do for Artificial Intelligence?” 2020, pp. 45-55.

[14] Davey, B. A. Introduction to Lattices and Order. Cambridge University Press, 2002.

Improvements to lattice drawing with fca.sty

Tobias Schlemmer

Dresden, Germany

Abstract

For documenting theoretical and empirical results with of Formal Concept Analysis Bernhard Ganter provided a LaTeX package that allows to typeset Formal Contexts and Line diagrams of Lattices and ordered sets. This package has been heavily reworked during the last years. Here a short status of the achievements and open challenges shall be given.

Keywords

LaTeX package, typesetting FCA, typesetting, formal context, lattice diagrams

1. Introduction

Bernhard Ganter's `fca.sty` is a LaTeX [1] package for typesetting Formal Concept Analysis [2]. This includes special symbols, Formal Contexts, and Lattice Diagrams. This package has been overhauled by the author in the last two years [3]. In the result, the package has improved support for formal contexts and drawing line diagrams.

The intention behind this effort was no less than to improve the typographical quality of papers about Formal Concept Analysis and related subjects. Furthermore, the package should provide a maximum amount of compatibility to existing LaTeX code based on former versions of `fca.sty`.

The main changes are:

- remove limits to the number of columns, rows, concepts, etc. of formal contexts and concept lattices,
- add a parser for Burmeister Context files,
- add an interface to allow a arbitrary LaTeX code for symbols in context tables – this also includes new symbols and colouring of crosses,
- use PGF [4] as backend for simple lattice diagrams and Ti*k*Z [4] for more sophisticated documentations
- expose the improvements from these packages to the users.

In the following sections these changes are shortly introduced one by one. These improvements offer tools that can help to improve the quality of publications about Formal Concept Analysis, however they do not reach this goal. Several hurdles lie on its way. Some of these shall be discussed at the end of this paper.

2. Keeping things separated

The new `fca.sty` consequently uses prefixes to macros and environments in order to avoid interference with other packages. In most cases this should be transparent to the users. However, this cannot be fully avoided:

- Global configuration macros that originally didn't start with `\fca` or `cxt` must be adapted to the new system. Inside the `cxt` and `diagram` environments these macros are provided without prefix in order to avoid unnecessary errors.

FCA4AI: What can FCA do for Artificial Intelligence? October 19–11, 2024, Santiago de Compostella
Tobias.Schlemmer@web.de (T. Schlemmer)
https://schlemmersoft.de/ (T. Schlemmer)
0000-0003-4350-3110 (T. Schlemmer)

- All new diagram styles (see below) must be called using their full path starting with `/fca/` when accessed from outside of FCA macros. If possible they are mapped to the corresponding `/pgf/` or `/tikz/` styles. Details are given in the documentation of `fca.sty`.

3. Improvements to formal contexts

Formula 1	1.	2.	disqualified
Verstappen	×		
Hamilton		×	
Leclerc		×	×

```
\begin{cxt}
  \cxtinput{formula1.cxt}
\end{cxt}
```

Figure 1: A formal context loaded from a Burmeister file. Left: Context, Right: source code.

Support for typesetting formal contexts had been added to `fca.sty` long ago. This support had its limitations. Some of them have been lifted. The way how object and attribute names are stored has been reworked as well as the table header generation. So the number of possible columns is not limited by the package, anymore. The general TEXnical limits should be large enough, even for unusual useage of the package: the number of possible macros, the maximum counter value, and the available memory.

New macros have been introduced that allow the definition of new symbols, and redefine existing ones. The characters denoting the symbols are stored as macros. So they can be defined to consume arguments. Additionally, digits have been predefined so that simple many-valued contexts can be typeset without additional setup. In cases where it is really necessary, it is a new macro allows to inject arbitrary code in the `tabular` environment of the context.

An optional positional argument has been added to the `cxt` environment, so that it is easier to place them in multi column environments. These improvements are demonstrated in Fig. 2.

Last but not least, a parser for contexts in the Burmeister format has been integrated into the package, as shown in Fig. 1. This parser maps the different parts of a `.cxt` file to the corresponding macros of a `cxt` environment. So markup or special signs can be typeset in the same way as in the corresponding macros of the LATEX environment `cxt`.

4. Lattice diagrams

Lots of work has been invested into the rejuvenation of the diagram code. The syntax has been carefully adapted such that existing diagrams can be directly integrated in new documents, or they need only minor adjustments. An example is given in Fig. 3. Each vertex has a name (traditionally a number) and coordinates. The edges and labels are anchored using these names. Additionally, labels have a printed description and can be shifted relatively to their position.

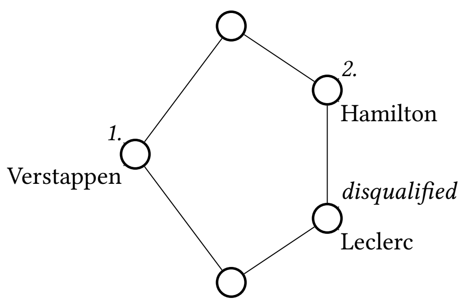
In order to allow a consistent appearance in sophisticated diagrams, the different elements of a diagram are organised in layers.

The original `fca.sty` package used the `\emlines` macro from the emTEX distribution in combination with standard LATEXs `picture` environment. Unfortunately, support for the emTEX specials has been removed from current TEX distributions, so the lines disappear from the diagrams. The package `tsemlines` [5] depends on Ti*k*Z. So it is more a quick hack than a lightweight solution to this problem. Another goal was to bridge the gap between the very simple `picture` environment and modern graphics drawing tools like Ti*k*Z. Thus the `diagram` environment is now based on PGF, a new environment `tikzdiagram` has been introduced as a Ti*k*Z version of `diagram`, and the important macros have been enhanced to use the syntax of Ti*k*Z.

A demo context	1.	2.	disqualified
Verstappen	1		∨
Hamilton	∧	×	⤢
Leclerc	i	23	×
nothing			

```
\begin{cxt}[c]
  \renewcommand{\fcaCxtArrowStyle}{\footnotesize\color{red
    }}%
  \fcaNewContextChar{v}{\cxtrlap{$\vee$}}
  \fcaNewContextChar{n}[1]{\cxtrlap{#1}}
  \fcaProvideContextChar{\wedge}{\cxtrlap{$\wedge$}}
  \fcaProvideContextChar{d}{ -- ignored -- }
  \fcaRenewContextChar{d}{\cxtrlap{$i$}}
  \cxtName{A demo context}
  \att{1.}
  \att{2.}
  \atr{disqualified}
  \obj{1.v}{Verstappen}
  \obj{\wedge xb}{Hamilton}
  \obj{dn{23}x}{Leclerc}
  \freeobj{\multicolumn{3}{c|}{}}{nothing}
\end{cxt}
```

Figure 2: A formal context typeset with the new features of the FCA packages. Left: Context, Right: corresponding source code.




```
\setlength{\unitlength}{0.9mm}
\begin{diagram}
  \Node(1)(20,10)
  \Node(2)(35,20)
  \Node(3)(5,30)
  \Node(4)(35,40)
  \Node(5)(20,50)
  %
  \Edge(1)(2)
  \Edge(1)(3)
  \Edge(2)(4)
  \Edge(3)(5)
  \Edge(4)(5)
  %
  \leftAttbox(3){1.}
  \rightAttbox(2){disqualified}
  \rightAttbox(4){2.}
  \leftObjbox(3){Verstappen}
  \rightObjbox(2){Leclerc}
  \rightObjbox(4){Hamilton}
\end{diagram}
```

Figure 3: A diagram example with 5 Vertices and 5 edges, attribute and object labels.

Between full Ti*k*Z support and the `picture` like environment there are several intermediate steps. At first the package can be loaded using `\usepackage{fca}`. This uses only the graphics layer of PGF and omits the syntax layer of Ti*k*Z. At this stage a limited support for Ti*k*Z like styles and attributes has been implemented. Naturally this is linked to the styles that have been implemented in `fca.sty`.

On the other end it is possible to load the package using the Ti*k*Z macro `\usetikzlibrary{fca}`. This enables to use of `diagram` inside a `tikzpicture` environment and the `tikzdiagram` environment which combines both in one environment. Using this approach all drawing macros are mapped to the corresponding Ti*k*Z macros which enables full Ti*k*Z support.

It is also possible to use `\usepackage{fca}` after loading Ti*k*Z. Both approaches enable additional styles to be used in a `diagram` environment, as `fca.sty` uses similar internals to Ti*k*Z. However, future versions of Ti*k*Z may cause errors and the set of supported styles may change depending on the Ti*k*Z version. So the latter approach is not recommended.

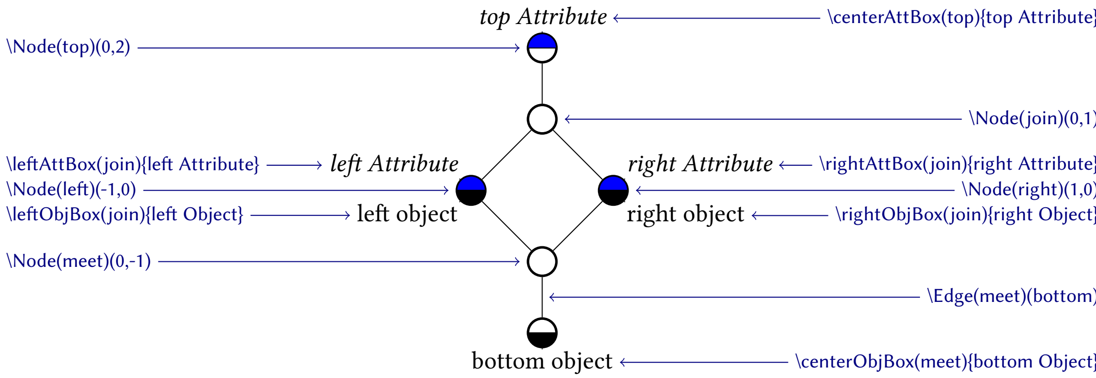
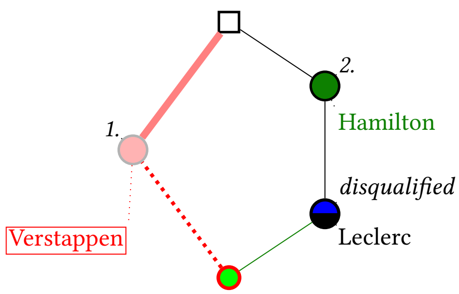


Figure 4: Elements of a `diagram` environment. The diagram shows a concept lattice. Arrows indicate which code is used to draw certain elements. The diagram is drawn using a `tikzdiagram` environment.




```
\setlength{\unitlength}{0.9mm}
\definecolor{darkgreen}{rgb
    }{0.05,0.5,0.}
\begin{diagram}
  \Node[draw=red, fill=green,
        line width=.5mm,
        radius=1.5mm](1)(20,10)
  \Node(2)(35,20)
  \Node[opacity=0.3, fill=red](3)(5,30)
  \Node[fill=darkgreen](4)(35,40)
  \Node[/tikz/rectangle](5)(20,50)
  %
  {\color{darkgreen}\Edge(1)(2)}
  \Edge[draw=red, dotted,
        line width=1.5pt](1)(3)
  \Edge(2)(4)
  \Edge[draw=red!50,
        line width=1mm](3)(5)
  \Edge(4)(5)
  %
  \rightAttbox[conexp style](2){
      disqualified}
  \leftAttbox(3){1.}
  \rightAttbox(4){2.}
  \leftObjbox[draw=red, inner sep=1pt](3)
      (1,-10){Verstappen}
  \rightObjbox[conexp style](2){Leclerc}
  \rightObjbox[text=darkgreen](4)(0,-2)
      {Hamilton}
\end{diagram}
```

Figure 5: An overly styled concept lattice demonstrating different kinds of markup including the use of Ti*k*Z styles.

Unfortunately, Ti*k*Z and the `picture` environment have different base units. While `picture` only allows to define one unit length, PGF and Ti*k*Z apply at least two affine transformations from the input to the output file. Though the behaviour of a `picture` environment can be emulated in PGF/Ti*k*Z, this behaviour is unstable and counter-intuitive to new users. The compatibility issue is solved using the following compromise:

- the `digaram` environment uses the old coordinate system if it is located outside of any graphics environment or inside a `pgfpicture` environment,
- the `tikzdiagram` environment and `diagram` inside `tikzpicture` use the tikz coordinate system.

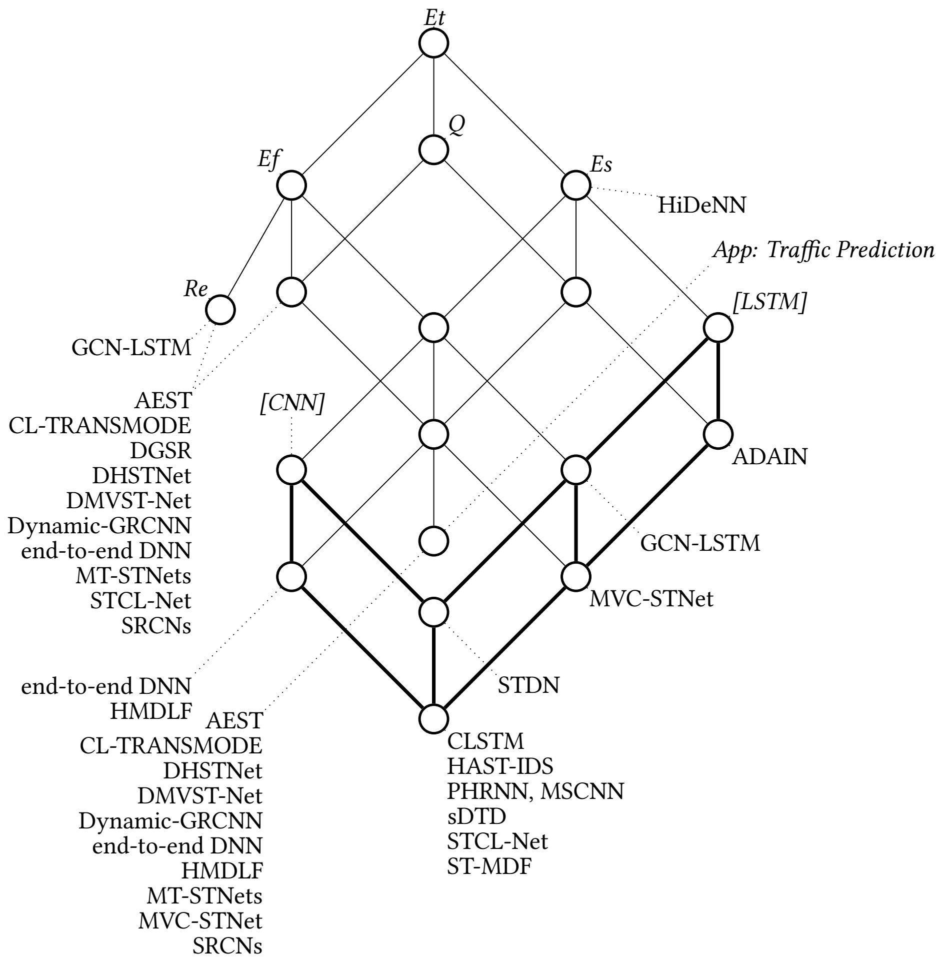


Figure 6: An iceberg lattice [6, Fig. 8], reworked for readability according to the standards defined in [2] with emphasis on the lower part.

- translation of old diagrams into Ti*k*Z diagrams can be easily done by opening the diagram with `\begin{tikzdiagram}[x=\unitlength,y=\unitlengh,…]`.
- All configuration macros in `diagram` environments are either available directly in `tikzdiagram` or can be expressed using style options.

As a real-world example, in Fig. 6 an iceberg lattice [6, Fig. 8] has been reworked using a chain decomposition layout exploiting the `calc` library of Ti*k*Z and the style used in [2]. As usual in Formal Concept Analysis, attributes are named only on their highest occurrences and apply to all nodes below them that can be connected with only rising lines to the corresponding label. This is in principle also true for object labels, which are valid for every node that can be connected going strictly upwards following the lines. However, many of the objects are attached to nodes that are not visible in the diagram. Their names are repeated at the lowest visible node, which often leads to multiple occurrences of he same name. Where appropriate, labels are reused for multiple nodes.

The lower part of the diagram is drawn with bolder lines. This part is referenced in an emphasised discussion in the text of the given article.

5. Open issues

It is planned to publish the package on CTAN, so that it can be integrated in the standard LaTeX distributions and Docker images. And despite the fact that the package is perfectly usable, some issues arise. However, it currently does not fulfil its primary goal: to improve the typographical quality of published diagrams in formal concept analysis. As it can be seen, in this paper both high-quality as well as low-quality typesetting is possible with this package. So how can it be modified to achieve this goal?

One approach would be to allow only good diagrams or make it at least hard to draw bad diagrams. One could argue that LaTeX also makes it hard to change dangerous parameters. This is impossible on the technical level as no algorithm exists that can check whether a diagram is good or bad. This decision depends on the writer's intention. Formalising this intention is nearly impossible. The levers we can pull (or not) are basically features and documentation. If we remove all possibly dangerous features from the package or its documentation, the package would be very inflexible. And it would be nearly impossible to interact with other graphical content.

On the other hand, documenting the package as a reference that simply lists all features, and encouraging people to play around with these features, also leads to over-styled diagrams that are hard to understand (cf. Fig. 2 and Fig. 5). Recall: Typography does not consider the taste of the author but studies how to format things such that they can be easily understood by the readers.

Also here, LaTeX can be used as a reference. Since the first version of LaTeX more and more features got configurable. So at first it simplified the use of TeX and then, it simplified the change of the layout.

This is the main issue that blocks publication on CTAN. Currently it is unclear, how to solve it. It seems as if a compromise could be to proper organise the information for the documentation. Other packages like Ti*k*Z and the `beamer` start with tutorials on their subjects. The difficulty of this approach lies in the fact, that such a tutorial should satisfy all stakeholders.

Other issues contain small inconsistencies between the PGF and the Ti*k*Z implementation of the diagram drawing code. These will be ironed out with the growth of the reference section in the documentation. However also the development of the reference is influenced by the above issue.

6. Conclusion

Despite the issues the new version of `fca.sty` is a powerful and usable package that provides:

- Certain symbols for Formal Concept Analysis,
- Context tables drawn from LaTeX code and Burmeister context files,
- Lattice Diagrams that can be drawn and enhanced with annotations in PGF and Ti*k*Z environments.

Happy TeXing!

References

[1] L. Lamport, LaTeX (2nd ed.): a document preparation system: user's guide and reference manual, Addison-Wesley Longman Publishing Co., Inc., USA, 1994.
[2] B. Ganter, R. Wille, Formal concept analysis: mathematical foundations, Springer, Berlin, 1999.
[3] B. Ganter, T. Schlemmer, fca.sty, Online Ressource: https://github.com/keinstein/latex-fca, 2024. Download via https://github.com/keinstein/latex-fca/archive/refs/heads/master.zip.
[4] T. Tantau, The TikZ and PGF Packages, 2024. URL: https://github.com/pgf-tikz/pgf/.
[5] T. Schlemmer, tsemlines, 2024. URL: https://ctan.org/pkg/tsemlines.
[6] T. Man, V. Y. Osipov, N. Zhukova, A. Subbotin, D. I. Ignatov, Neural networks for intelligent multilevel control of artificial and natural objects based on data fusion: A survey, Information Fusion 110 (2024) 102427. doi:10.1016/j.inffus.2024.102427.