Pierre Marquis
Odile Papini
Henri Prade
Editors

A Guided Tour of Artificial Intelligence Research

2

AI Algorithms

Springer

Pierre Marquis · Odile Papini · Henri Prade
Editors

A Guided Tour of Artificial Intelligence Research

Volume II: AI Algorithms

Springer

Editors
Pierre Marquis
CRIL-CNRS, Université d'Artois
and Institut Universitaire de France
Lens, France

Odile Papini
Aix Marseille Université, Université de
Toulon, CNRS, LIS
Marseille, France

Henri Prade
IRIT
CNRS and Université Paul Sabatier
Toulouse, France

ISBN 978-3-030-06166-1 ISBN 978-3-030-06167-8 (eBook)
https://doi.org/10.1007/978-3-030-06167-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

General Presentation of the Guided Tour of Artificial Intelligence Research

Artificial Intelligence (AI) is more than sixty years old. It has a singular position in the vast fields of computer science and engineering. Though AI is nowadays largely acknowledged for various developments and a number of impressive applications, its scientific methods, contributions, and tools remain unknown to a large extent, even in the computer science community. Notwithstanding introductory monographs, there do not exist treatises offering a detailed, up-to-date, yet organized overview of the whole range of AI researches. This is why it was important to review the achievements and take stock of the recent AI works at the international level. This is the main goal of this *A Guided Tour of Artificial Intelligence Research*.

This set of books is a fully revised and substantially expanded version, of a panorama of AI research previously published in French (by Cépaduès, Toulouse, France, in 2014), with a number of entirely new or renewed chapters. For such a huge enterprise, we have largely benefited the support and expertise of the French AI research community, as well as of colleagues from other countries. We heartily thank all the contributors for their commitments and works, without which this quite special venture would not have come to an end.

Each chapter is written by one or several specialist(s) of the area considered. This treatise is organized into three volumes: The first volume gathers twenty-three chapters dealing with the foundations of knowledge representation and reasoning formalization including decision and learning; the second volume offers an algorithm-oriented view of AI, in fourteen chapters; the third volume, in sixteen chapters, proposes overviews of a large number of research fields that are in relation to AI at the methodological or at the applicative levels.

Although each chapter can be read independently from the others, many cross-references between chapters together with a global index facilitate a nonlinear reading of the volumes. In any case, we hope that readers will enjoy browsing the proposed surveys, and that some chapters will tease their curiosity and stimulate their creativity.

July 2018

Pierre Marquis
Odile Papini
Henri Prade

Formal Concept Analysis: From Knowledge Discovery to Knowledge Processing

Sébastien Ferré, Marianne Huchard, Mehdi Kaytoue, Sergei O. Kuznetsov and Amedeo Napoli

Abstract In this chapter, we introduce Formal Concept Analysis (FCA) and some of its extensions. FCA is a formalism based on lattice theory aimed at data analysis and knowledge processing. FCA allows the design of so-called concept lattices from binary and complex data. These concept lattices provide a realistic basis for knowledge engineering and the design of knowledge-based systems. Indeed, FCA is closely related to knowledge discovery in databases, knowledge representation and reasoning. Accordingly, FCA supports a wide range of complex and intelligent tasks among which classification, information retrieval, recommendation, network analysis, software engineering and data management. Finally, FCA is used in many applications demonstrating its growing importance in data and knowledge sciences.

1 Introduction

Formal Concept Analysis (FCA) is a branch of applied lattice theory that appeared in 1980's (Ganter and Wille 1999). Roots of the application of lattices obtained through the definition of a Galois connection between two arbitrary partially ordered

S. Ferré
Univ Rennes, CNRS, IRISA, Rennes, France
e-mail: sebastien.ferre@irisa.fr

M. Huchard
LIRMM, Université de Montpellier, CNRS, Montpellier, France
e-mail: marianne.huchard@lirmm.fr

M. Kaytoue
INSA-LIRIS, Infologic, Lyon, France
e-mail: mehdi.kaytoue@insa-lyon.fr

S. O. Kuznetsov
National Research University Higher School of Economics (HSE), Moscow, Russia
e-mail: skuznetsov@hse.ru

A. Napoli (✉)
Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France
e-mail: Amedeo.Napoli@loria.fr

P. Marquis et al. (eds.), *A Guided Tour of Artificial Intelligence Research*,
https://doi.org/10.1007/978-3-030-06167-8_13

sets can also be found in the 1970's (Barbut and Monjardet 1970). Starting from a binary relation between a set of objects and a set of attributes, formal concepts are built as maximal sets of objects in relation with maximal sets of common attributes, by means of derivation operators forming a Galois connection. Concepts form a partially ordered set that represents the initial data as a hierarchy, called the concept lattice. This conceptual structure has proved to be useful in many fields in artificial intelligence and pattern recognition, e.g. knowledge management, data mining, machine learning, mathematical morphology, etc. In particular, several results and algorithms from itemset and association rule mining and rule-based classifiers were characterized in terms of FCA (Kuznetsov and Poelmans 2013). For example, the set of frequent closed itemsets is an order ideal of a concept lattice; association rules and functional dependencies can be characterized with the derivation operators; jumping patterns are defined as hypotheses, etc., not to mention efficient polynomial-delay algorithms for building all closed itemsets such as *NextClosure* (Ganter 2010) or *CloseByOne* (Kuznetsov 1999), which was rediscovered under different names, e.g., LCM algorithm (Uno et al. 2004).

In this chapter, we present the basics of FCA and several extensions allowing to work with complex data. FCA was generalized several times to partially ordered data descriptions to efficiently and elegantly deal with non binary, complex, heterogeneous and structured data, such as e.g. graphs (Liquiere and Sallantin 1998; Kuznetsov 1999). Different extensions include Fuzzy FCA (Belohlávek and Vychodil 2005; Belohlávek 2011), Generalized Formal Concept Analysis (Chaudron and Maille 2000), Logical Concept Analysis (Ferré and Ridoux 2000), and Pattern Structures (Ganter and Kuznetsov 2001). FCA has also been extended to relational data where concepts do not only depend on the description of objects, but also on the relationships between objects. These generalizations provide new ways of solving problems in several applications and include in particular Relational Concept Analysis (Rouane-Hacene et al. 2013), relational structures (Kötters 2013), Graph-FCA (Ferré 2015) and Triadic Concept Analysis (Lehmann and Wille 1995).

FCA is a data analysis and classification formalism with good mathematical foundations. Besides that, FCA is also related in several ways to Knowledge Discovery in Databases and to Knowledge Representation and Reasoning. Indeed, many links exist between concept lattices, itemsets and association rules, see e.g. (Bastide et al. 2000; Zaki 2005; Szathmary et al. 2014). In addition, FCA was used in many various tasks, such as the discovery of definitions for ontologies, text mining, information retrieval, biclustering and recommendation, bioinformatics, chemoinformatics, medicine (healthcare), Natural Language Processing, Social Network Analysis…(the last section about applications provide details).

Visualization plays a very important role in FCA and the display of the concept lattice is a very good support to exploration and interpretation of the data under study. Many tools were designed for this visualization purpose. One of the first tools to be created was Toscana (Eklund et al. 2000; Becker 2004), and the most well-known tool is probably Conexp.[1] Then a series of tools was developed for building

[1] http://conexp.sourceforge.net/.

friendly user interfaces for various applications such as email, pictures, or museum collections (Eklund et al. 2004; Ducrou et al. 2006; Wray and Eklund 2011). More recently, LatViz[2] was introduced for the drawing of concept lattices based on plain FCA and on pattern structures as well (Alam et al. 2016).

In the following, we present the basics of FCA and then we review a number of main extensions, namely pattern structures, relational concept analysis, Graph-FCA, and Triadic Concept Analysis. The seminal book (Ganter and Wille 1999) remains the reference for mathematical foundations of FCA, but let us mention some other very useful books, on the practical and various aspects of FCA (Carpineto and Romano 2004; Ganter et al. 2005) and on conceptual exploration (Ganter and Obiedkov 2016). Finally, the Formal Concept Analysis Homepage[3] also includes a lot of useful information about FCA, FCA Conferences, and Software.

2 The Basics of Formal Concept Analysis

2.1 Context, Concepts and the Concept Lattice

The framework of FCA is fully detailed in Ganter and Wille (1999) and we adopt the so-called "German notation" in this chapter. FCA starts with a formal context (G, M, I), where G denotes a set of objects, M a set of attributes, and $I \subseteq G \times M$ a binary relation between G and M. The statement $(g, m) \in I$ (also denoted by gIm) is interpreted as "object g has attribute m".

Two operators $(\cdot)'$ define a Galois connection between the powersets $(2^G, \subseteq)$ and $(2^M, \subseteq)$, with $A \subseteq G$ and $B \subseteq M$:

$A' = \{m \in M \mid \text{ for all } g \in A : gIm\}$ and $B' = \{g \in G \mid \text{ for all } m \in B : gIm\}$.

For $A \subseteq G$, $B \subseteq M$, a pair (A, B), such that $A' = B$ and $B' = A$, is called a "formal concept", where set A is called the "extent" and set B is called the "intent" of the concept (A, B). The dual aspect of a concept in knowledge representation is naturally materialized in a formal concept, where the intent corresponds to the description and the extent to the set of instances of the concept.

Concepts are partially ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ $(\Leftrightarrow B_2 \subseteq B_1)$. With respect to this partial order, the set of all formal concepts forms a complete lattice called the "concept lattice" of (G, M, I).

For illustration we reuse here a famous example introduced in Davey and Priestley (1990) where planets correspond to objects and their characteristics to attributes. The binary context is given below and the associated concept lattice is shown in Fig. 1.

[2] https://latviz.loria.fr/.

[3] http://www.upriss.org.uk/fca/fca.html.

Planets	Size			Distance to Sun		Moon(s)	
	small	medium	large	near	far	yes	no
Jupiter			x		x	x	
Mars	x			x		x	
Mercury	x			x			x
Neptune		x			x	x	
Pluto	x				x	x	
Saturn			x		x	x	
Earth	x			x		x	
Uranus		x			x	x	
Venus	x			x			x

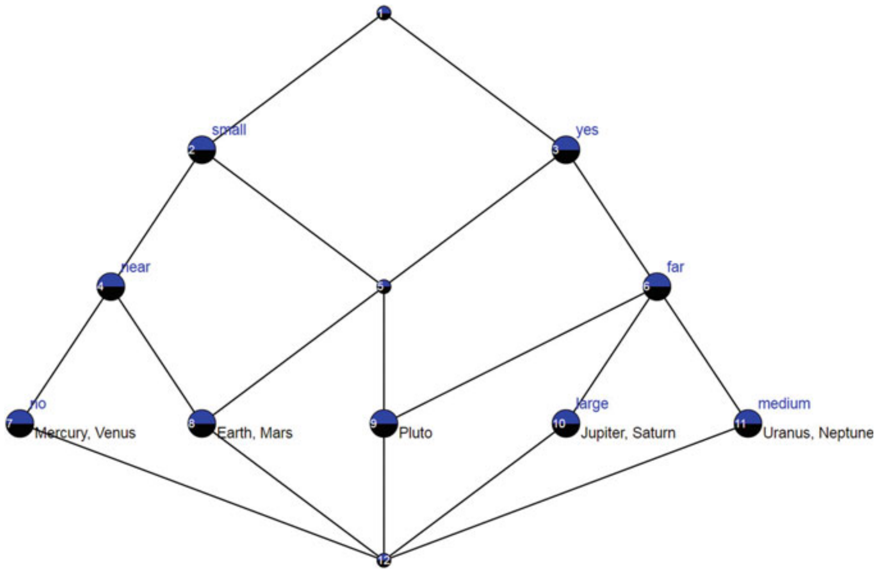


Fig. 1 The concept lattice of planets in reduced notation

The concept lattice, provided that it is not too large, can be visualized, navigated and interpreted in various ways.

- Visualization of the concept lattice: The labels of concepts are written in "reduced notation", meaning that the intent of a concept –in blue– is made of the union of all greater concept intents while the extent –in black– is made of the union of all lower concept extents.
  For example, concept #10 has the extent *(Jupiter, Saturn)* and the intent *(large, far, yes)*, while concept #4 has the extent *(Mercury, Venus, Earth, Mars)* and the intent *(near, small)*.
- Navigation and Information Retrieval: The concept lattice can be navigated for searching specific information, e.g. small planets which are far from the sun (concept #9) or large planets with moons (concept #10).

- Interpretation of implications and association rules: The concept lattice can be used for knowledge discovery, by interpreting the concepts and the associated rules. Indeed, considering the intents of the concepts, it is possible to discover implications and association rules as made precise here after.

When one considers non-binary contexts, e.g. numerical or interval data, conceptual scaling is often used for binarizing data and for obtaining a binary formal context (Ganter and Wille 1999). Then, a numerical dataset for example can be described by a many-valued context (G, M, W, I) where G is a set of objects, M a set of numerical attributes, W a set of values (e.g. numbers), and I a ternary relation defined on the Cartesian product $G \times M \times W$. The fact $(g, m, w) \in I$ or simply $m(g) = w$ means that object g takes value w for attribute m.

Then, standard FCA algorithms can be applied for computing concept lattices from scaled contexts (Kuznetsov and Obiedkov 2002). However, adaptations of these algorithms should be applied to more complex data such as intervals, sequences, trees or graphs, as detailed in the section about pattern structures.

2.2 Rules and Implications

Besides formal concepts and their ordering, which model the natural taxonomy of object classes described in terms of common attributes, another very important aspect of FCA is related to implicative dependencies, both exact and approximate. Dependencies of the first type, called "implications", are closely related to functional dependencies, whereas approximate implication dependencies became known as association rules in data mining.

Let us give definitions and examples. For $A, B \subseteq M$ the "attribute implication" $\mathtt{A} \Rightarrow \mathtt{B}$ holds if $\mathtt{A}' \subseteq \mathtt{B}'$ and the "association rule" $\mathtt{A} \rightarrow \mathtt{B}$ is valid when its "support" $s = \frac{|(A \cup B)'|}{|G|} = \frac{|A' \cap B'|}{|G|} \geq \sigma_s$ and its "confidence" $c = \frac{|(A \cup B)'|}{|A'|} = \frac{|A' \cap B'|}{|A'|} \geq \sigma_c$, where σ_s and σ_c are user-defined thresholds for support and confidence respectively. In this way, an attribute implication is an association rule with $c = 1$, since the latter is possible only when $A' = (A \cup B)' = A' \cap B'$, hence, when $A' \subseteq B'$.

Then, $no \Longrightarrow near$ and $near \Longrightarrow small$ are implications with a confidence of 1, while $far \longrightarrow medium$ and $small \longrightarrow near$ are association rules with a confidence of $2/5$ and $4/5$ respectively.

Implications obey Armstrong rules:

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

which are known as properties of functional dependencies in database theory. Indeed, implications and functional dependencies can be reduced to each other. A functional dependency $X \to Y$ is valid in a complete many-valued context (actually, a relational data table) (G, M, W, I) if the following holds for every pair of objects $g, h \in G$:

$(\forall m \in X, m(g) = m(h))$ then $(\forall n \in Y, n(g) = n(h))$. In Ganter and Wille (1999) it was shown that having a complete many-valued context $K = (G, M, W, I)$, one can define the context $K_N := (P_2(G), M, I_N)$, where $P_2(G)$ is the set of all pairs of different objects from G and I_N is given by $\{g, h\} I N_m \Leftrightarrow m(g) = m(h)$. Then, a set $Y \subseteq M$ is functionally dependent on the set $X \subseteq M$ in K iff the implication $X \rightarrow Y$ holds in the context K_N. In Baixeries et al. (2014), the relations between implications and functional dependencies are discussed in depth.

The inverse reduction (Kuznetsov 2001) is given as follows: For a context $K = (G, M, I)$ one can construct a many-valued context K_W such that an implication $X \Rightarrow Y$ holds iff Y is functionally dependent on X in K_W. For a context K the corresponding many-valued context is $K_W = (G, M, G \cup \{\times\}, I_M)$, where for any $m \in M, g \in G$ one has $m(g) = g$ if gIm does not hold and $m(g) = \times$ if gIm.

Having Armstrong rules, it is natural to ask for an implication base, i.e., for a minimal subset of implications from which all other implications of a context can be derived. The authors of Guigues and Duquenne (1986) gave an algebraic characterization of premises of a cardinality-minimum implication base, in terms of what is now called "pseudo-intents". Another important implication base, called "proper premise base" (Ganter and Wille 1999) or "direct canonical base" (Bertet and Monjardet 2010), is minimal w.r.t. application of first two Armstrong rules (the third rule is not applied in the derivation). Here we will not elaborate in this direction which is out of the scope of this chapter.

2.3 Algorithms for Computing Concepts

Many algorithms were proposed for computing the set of concepts and concept lattices. The survey (Kuznetsov and Obiedkov 2002) provides a state-of-the art by 2002. Most efficient practical algorithms have "polynomial delay". Recently, several new algorithms were proposed (Kourie et al. 2009; Outrata and Vychodil 2012). These algorithms, while having the same worst-case complexity, perform better in practice. As for implication bases, no total-polynomial-time algorithms for computing the minimum base are known, and due to intractability results (Kuznetsov 2004; Distel and Sertkaya 2011; Babin and Kuznetsov 2013), it looks like that such algorithms are not feasible. In applications, classical "NextClosure" algorithm (Ganter 2010) and a faster algorithm from (Obiedkov and Duquenne 2007) are the most used ones.

2.4 The Stability Measure

Stability (Kuznetsov 2007) was proposed as a measure of independence of a concept intent, i.e., the concept "meaning", on randomness in data composition. Every object could appear in data (context) at random, so a stable intent, i.e., intent with large stability, would stay for a relative large number of subsets of the concept extent having the same intent. Dually, one can speak of stability of an extent, by considering subsets of a concept intent. Here attributes can be noisy. Stability is a partial case of robustness (Tatti et al. 2014). A drawback of stability is its intractability ($\#P$-completeness). In Kuznetsov (2007), Babin and Kuznetsov (2012), Buzmakov et al. (2014) several tractable approximations of stability were proposed, including Δ-stability, which proved to behave quite similar to stability in practice. Stability was used in numerous applications, like detecting expert communities (Kuznetsov et al. 2007), groups of French verbs (Falk and Gardent 2014), and chemical alerts (Métivier et al. 2015). The tractable approximation of stability, called Δ-stability, was proven to be an antimonotonic measure in terms of "projection chains" (Buzmakov et al. 2017). Besides stability and support, well-known in data mining, many other interestingness measures of concepts were proposed, see recent survey (Kuznetsov and Makhalova 2018).

3 Pattern Structures

3.1 Introduction

When data are complex, i.e. numbers, sequences or graphs, instead of applying data transformation, such as discretization of numerical data, leading to space and time computational hardness, one may directly work on original data. A pattern structure is defined as a generalization of a formal context describing complex data (Ganter and Kuznetsov 2001; Kuznetsov 2009), and opens the range of applicability of plain FCA.

Besides pattern structures, Fuzzy FCA and Logical Concept Analysis (LCA) are two other extensions of FCA. Fuzzy FCA aims at extending FCA to data with graded or fuzzy attributes. There are two main ways to deal with fuzzy attributes in FCA, one using conceptual scaling and the second extending FCA into a fuzzy setting enabling to directly work with fuzzy attributes (see for example Belohlávek 2004; Belohlávek and Vychodil 2005; Cabrera et al. 2017; Belohlávek 2011; García-Pardo et al. 2013). One of the main approaches in Fuzzy FCA is based on the use of a residuated implication, and a comprehensive overview is proposed in Belohlávek (2008). LCA (Ferré and Ridoux 2000, 2004) is a generalization of concept analysis that shares with pattern structures the objective to work directly with complex data. It defines the same Galois connection, only using different notations, and it was mostly developed for information retrieval purposes. Here after, we focus on pattern structures while some results involving LCA are mentioned in the next sections.

In classical FCA, object descriptions are sets of attributes, which are partially ordered by set inclusion, w.r.t. set intersection: let $P, Q \subseteq M$ be two attributes sets, then $P \subseteq Q \Leftrightarrow P \cap Q = P$, and $(M, \subseteq)$, also written $(M, \cap)$, is a partially ordered set of object descriptions. Set intersection $\cap$ is a meet operator, i.e., it is idempotent, commutative, and associative. A Galois connection can then be defined between the powerset of objects $(2^G, \subseteq)$ and a meet-semi-lattice of descriptions denoted by $(D, \sqcap)$ ((standing for $(M, \cap)$). This idea is used to define pattern structures in the framework of FCA as follows.

Formally, let G be a set of objects, let $(D, \sqcap)$ be a meet-semi-lattice of potential object descriptions and let $\delta : G \longrightarrow D$ be a mapping that takes each object to its description. Then $(G, (D, \sqcap), \delta)$ is a "pattern structure". Elements of D are patterns and are ordered by a subsumption relation $\sqsubseteq$: $\forall c, d \in D, c \sqsubseteq d \iff c \sqcap d = c$.

A pattern structure $(G, (D, \sqcap), \delta)$ gives rise to two derivation operators $(\cdot)^{\square}$:

$$A^{\square} = \bigsqcap_{g \in A} \delta(g), \quad \text{for } A \in 2^G \quad \text{and} \quad d^{\square} = \{g \in G \mid d \sqsubseteq \delta(g)\}, \quad \text{for } d \in D.$$

These operators form a Galois connection between $(2^G, \subseteq)$ and $(D, \sqcap)$. Pattern concepts of $(G, (D, \sqcap), \delta)$ are pairs of the form (A, d), $A \subseteq G$, $d \in (D, \sqcap)$, such that $A^{\square} = d$ and $A = d^{\square}$. For a pattern concept (A, d), d is called "pattern intent" and it is the common description of all objects from A, called "pattern extent". When partially ordered by $(A_1, d_1) \leq (A_2, d_2) \Leftrightarrow A_1 \subseteq A_2 \quad (\Leftrightarrow d_2 \sqsubseteq d_1)$, the set of all pattern concepts forms a complete lattice called "pattern concept lattice".

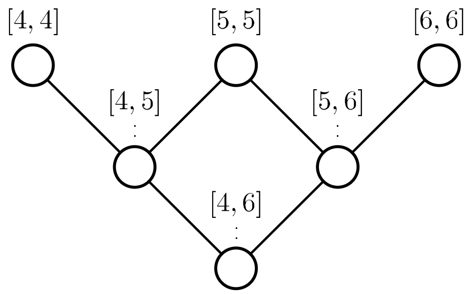
More importantly, the operator $(.)^{\square\square}$ is a closure operator and pattern intents are closed patterns. The existing FCA algorithms (detailed in Kuznetsov and Obiedkov 2002) can be used with slight modifications to compute pattern structures, in order to extract and classify concepts. Details can be found in Ganter and Kuznetsov (2001), Kaytoue et al. (2011c), Kuznetsov (2009).

Pattern structures are very useful for building concept lattices where the extents of concepts are composed of "similar objects" with respect to a similarity measure associated to the subsumption relation $\sqsubseteq$ in $(D, \sqcap)$ (Kaytoue et al. 2010).

Pattern structures offer a concise way to define closed patterns. They also allow for efficient algorithms with polynomial-delay (modulo complexity of computing $\sqsubseteq$ and $\sqcap$) (Kuznetsov 1999). In the presence of large datasets they offer natural approximation tools (*projections*, detailed below) and allow for lazy classification (Kuznetsov 2013).

When D is the power set of a set of items I, $\sqcap$ and $\sqsubseteq$ are the set intersection and inclusion, respectively: pattern intents are closed itemsets and we fall back to standard FCA settings. Originally, pattern structures were introduced to handle objects described by labeled graphs (Kuznetsov 1999; Kuznetsov and Samokhin 2005). A general approach for handling various types of descriptions was developed, for numbers and intervals (Kaytoue et al. 2011c), convex polygons (Belfodil et al. 2017), partitions (Baixeries et al. 2014), sequences (Buzmakov et al. 2016), trees (Leeuwenberg et al. 2015), and RDF triples in the web of data (Alam et al. 2015).

Fig. 2 Meet-semi-lattice $(D, \sqcap)$ with $D = \{[4,4], [5,5], [6,6], [4,5], [5,6], [4,6]\}$



3.2 Interval Pattern Structures

For illustration, we analyze object descriptions as tuples of numeric intervals. Pattern structures allow to directly extract concepts from data whose object descriptions are partially ordered (Kaytoue et al. 2011b, c).

A numerical dataset with objects G and attributes M can be represented by an interval pattern structure. Let G be a set of objects, $(D, \sqcap)$ a meet-semi-lattice of interval patterns ($|M|$-dimensional interval vectors), and δ a mapping associating with any object $g \in G$ an interval pattern $\delta(g) \in (D, \sqcap)$. The triple $(G, (D, \sqcap), \delta)$ is an "interval pattern structure".

The meet operator $\sqcap$ on interval patterns can be defined as follows. Given two interval patterns $c = \langle [a_i, b_i] \rangle_{i \in \{1,\ldots,|M|\}}$ and $d = \langle [e_i, f_i] \rangle_{i \in \{1,\ldots,|M|\}}$, then:

$$c \sqcap d = \langle [minimum(a_i, e_i), maximum(b_i, f_i)] \rangle_{i \in \{1,\ldots,|M|\}}$$

meaning that the convex hull of intervals on each vector dimension is taken. The meet operator induces the following subsumption relation $\sqsubseteq$ on interval patterns: $\langle [a_i, b_i] \rangle \sqsubseteq \langle [c_i, d_i] \rangle \Leftrightarrow [a_i, b_i] \supseteq [c_i, d_i], \; \forall i \in \{1, \ldots, |M|\}$ where larger intervals are subsumed by smaller intervals.

For example, with $D = \{[4,4], [5,5], [6,6], [4,5], [5,6], [4,6]\}$, the meet-semi-lattice $(D, \sqcap)$ is given in Fig. 2. The interval labeling a node is the meet of all intervals labeling its ascending nodes, e.g. $[4,5] = [4,4] \sqcap [5,5]$, and is also subsumed by these intervals, e.g. $[4,5] \sqsubseteq [5,5]$ and $[4,5] \sqsubseteq [4,4]$. An example of interval pattern subsumption is given by: $\langle [2,4], [2,6] \rangle \sqsubseteq \langle [4,4], [3,4] \rangle$ as $[2,4] \sqsubseteq [4,4]$ and $[2,6] \sqsubseteq [3,4]$.

	m_1	m_2	m_3
g_1	5	7	6
g_2	6	8	4
g_3	4	8	5
g_4	4	9	8
g_5	5	8	5

Let us consider the above numerical context where there are 5 objects and 3 attributes. The description of an object is a vector of intervals, e.g. $g_1 = \langle 5, 7, 6 \rangle$, where 5 stands for the closed interval [5, 5]. Then the meet operator $\sqcap$ captures the "similarity" between two objects descriptions (i.e. two vectors of intervals) as the convex hull of the intervals w.r.t. the order of the components of the vector. For example, the similarity of $g_1 = \langle 5, 7, 6 \rangle$ and $g_2 = \langle 6, 8, 4 \rangle$ is computed as

$$\begin{aligned} \{g_1, g_2\}^{\square} &= \sqcap_{g \in \{g_1, g_2\}} \delta(g) \\ &= \langle 5, 7, 6 \rangle \sqcap \langle 6, 8, 4 \rangle \\ &= \langle [5, 6], [7, 8], [4, 6] \rangle \end{aligned}$$

Conversely, we can compute the image of an interval vector following the definition of the Galois connection for interval pattern structures as follows:

$$\begin{aligned} \langle [5, 6], [7, 8], [4, 6] \rangle^{\square} &= \{g \in G | \langle [5, 6], [7, 8], [4, 6] \rangle \sqsubseteq \delta(g)\} \\ &= \{g_1, g_2, g_5\} \end{aligned}$$

And finally $(\{g_1, g_2, g_5\}, \langle [5, 6], [7, 8], [4, 6] \rangle)$ is a pattern concept.

The whole pattern concept lattice for the numerical context is given in Fig. 3. Even for small numerical contexts, the pattern concept lattice is usually large and close to a Boolean lattice. This shows that it can be hard to work with the whole pattern concept lattice and that operations for simplifying the lattice should be provided, such as projections which are detailed below.

Interval pattern structures offer a way to enumerate all hyper-rectangles of a numerical dataset (a tensor or numerical matrix), without redundancy thanks to the closure operator: all pattern intents correspond to a unique set of points in the Euclidean space and the intent gives the minimal bounding box containing them all. This is particularly interesting for mining numerical data without having to discretize it, either in a preprocessing phase, or on the fly, and thus suffering of imprecision. For example, it was used by Kaytoue et al. for enumerating contexts which induce exceptional models of some dataset (Kaytoue et al. 2017). As it is costly in the general case to enumerate all closed intervals, they proposed then a best-first search of interval patterns with a Monte Carlo Tree Search driven by a pattern quality measure which discriminates a class label (Bosc et al. 2017). Farther, the use of a pattern structure for the task of biclustering numerical data is also described.

3.3 Projections and Representation Context for Pattern Structures

Pattern structure projections simplify computation and reduce the number of concepts (Ganter and Kuznetsov 2001). For example, a set of labeled graphs can be projected to a set of k-chains (Kuznetsov and Samokhin 2005), while intervals

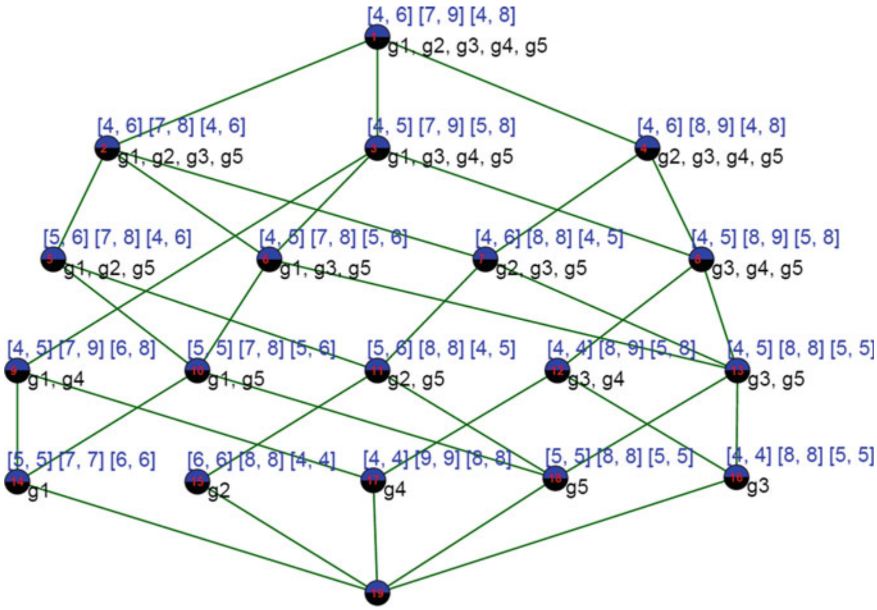


Fig. 3 An example of pattern concept lattice

can be enlarged (Kaytoue et al. 2010). In general, a projection ψ takes any pattern to a *more general* pattern covering more objects. A projection is $\sqcap$-preserving: $\forall c, d \in D, \psi(c \sqcap d) = \psi(c) \sqcap \psi(d)$. Numerical data can be projected when a similarity or a tolerance relation (i.e. symmetric, reflexive, but not necessarily transitive relation) between numbers is used, then a projection can be performed as a preprocessing task. A wider class of projections was introduced in Buzmakov et al. (2015): *o-projections* can modify not only object descriptions, but the semi-lattice of descriptions.

For any pattern structure, a *representation* context can be built, which is a binary relation encoding the pattern structure. Concepts in both data representations are in 1-1-correspondence. This aspect was studied for several types of patterns, designing the transformation procedures and evaluating in which conditions one data representation prevails (Kaytoue et al. 2011b; Baixeries et al. 2014). It should be noticed that the bijection does not hold in general for minimal generators, qualified as *free* or *key* in pattern mining (Kaytoue et al. 2011b). The impact of projections on representation contexts are investigated with the new class of *o*-projections in Buzmakov et al. (2015).

4 Relational Models in FCA

4.1 Relational Concept Analysis

Relational datasets arise in a wide range of situations, e.g. Semantic Web applications (Staab and Studer 2009), relational learning and data mining (Dzeroski and Lavrac 2001), refactoring of UML class models (Dao et al. 2004) and model-driven development (Stahl et al. 2006). Relational datasets are composed of a set of binary tables (`objects` × `attributes`) and inter-objects relations (`objects` × `objects`). The binary status of inter-objects relations is not really a limitation, as n-ary relations can always be transformed as a composition of binary relations, or reified.

Relational Concept Analysis (RCA) extends FCA to the processing of relational datasets in a way allowing inter-objects links to be materialized and incorporated into formal concept intents (Rouane-Hacene et al. 2013). Each object-attribute relation corresponds to an object category, while each object-object relation corresponds to inter-category links. The objects of one category are classified in the usual way into a concept lattice depending on the binary and relational attributes that they share. Inter-object links are scaled to become "relational attributes" connecting first objects to concepts and then concepts to concepts, in a similar way as role restrictions in Description Logics (DLs) (Baader et al. 2003). The relational attributes reflect the relational aspects within a formal concept. They also conform to the same classical rules for concept formation mechanisms from FCA which means that the relational concept intents can be produced by standard FCA algorithms. Due to the strong analogy between role restrictions in DLs and relational attributes in RCA, formal concepts can be almost readily converted into a DL-based formalism (Rouane-Hacene et al. 2007), e.g. for ontology engineering purposes as in Rouane-Hacene et al. (2010), Bendaoud et al. (2008).

RCA is introduced and detailed in Huchard et al. (2007). The data structure is described by a relational context family (RCF), which is a $(\mathbf{K}, \mathbf{R})$ pair where: $\mathbf{K} = \{K_i\}_{i=1,\ldots,n}$ is a set of contexts $K_i = (G_i, M_i, I_i)$, and $\mathbf{R} = \{r_j\}_{j=1,\ldots,p}$ is a set of r_j relations where $r_j \subseteq G_k \times G_\ell$ for some $k, \ell \in \{1, \ldots, n\}$. For a given relation r_j, the domain is denoted by $dom(r_j)$ and the range by $ran(r_j)$.

Figure 4 shows a simple example where three object categories, namely dishes, cereals and countries, are described by attributes, namely *Europe/Asia* for countries and *rice/wheat* for cereals. Three relations respectively connect dishes to cereals (*hasMainCereal (hmc)*), cereals to countries (*isProducedIn (ipi)*) and countries to dishes (*eatLotOf (elo)*). Relational Concept Analysis considers such data under the tabular form (RCF) presented in Table 1.

RCA is based on a "relational scaling" mechanism that transforms a relation $r_j \subseteq G_k \times G_\ell$ into a set of relational attributes that are added to the context describing the object set $ran(r_j)$. To that end, relational scaling adapts the DL semantics of role restrictions. For each relation $r_j \subseteq G_k \times G_\ell$, there is an initial lattice for each object set, i.e. L_k for G_k and L_ℓ for G_ℓ. A relational attribute is associated to an object $o \in G_k$ whenever $r_j(o)$ satisfies a given constraint, where $r_j(o)$ denotes the set of

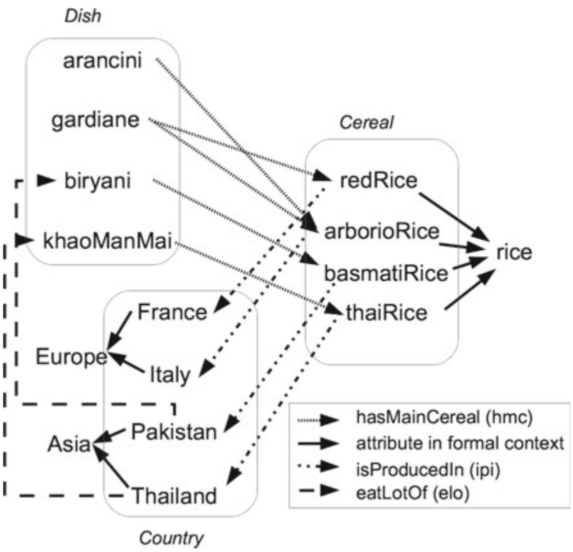


Fig. 4 Graph view on objects (dishes, cereals, and countries) with their attributes and relations

Table 1 Relational Context Family: Formal (object-attribute) contexts dishes, cereal, countries, and object-object relations hasMainCereal, isProducedIn, eatLotOf

`dish`	**arancini**	**gardiane**	**khaoManKai**	**biryani**
arancini	×			
gardiane		×		
khaoManKai			×	
biryani				×

`cereal`	**redRice**	**arborioRice**	**basmatiRice**	**thaiRice**	**rice**	**wheat**
redRice	×				×	
arborioRice		×			×	
basmatiRice			×		×	
thaiRice				×	×	

`country`	**Italy**	**France**	**Thailand**	**Pakistan**	**Europe**	**Asia**
Italy	×				×	
France		×			×	
Thailand			×			×
Pakistan				×		×

`hasMainCereal`	**redRice**	**arborioRice**	**basmatiRice**	**thaiRice**
arancini		×		
gardiane	×	×		
khaoManKai				×
biryani			×	

`isProducedIn`	**France**	**Italy**	**Pakistan**	**Thailand**
redRice	×			
arborioRice		×		
basmatiRice			×	
thaiRice				×

`eatLotOf`	**arancini**	**gardiane**	**khaoManKai**	**biryani**
Italy				
France				
Thailand			×	
Pakistan				×

objects in G_ℓ in relation with o through r_j. A relational attribute is composed of a scaling quantifier q, the name of the relation r_j, and the target concept C, and is denoted by "$q\ r_j(C)$".

There is a variety of scaling quantifiers. $\exists\forall r_j(C)$ (existential+universal scaling) is associated with o, when o has *at least one* r_j link and every such r_j link is *only* directed towards objects in the extent of C, i.e. $r_j(o) \subseteq extent(C)$ and $r_j(o) \neq \emptyset$. $\exists r_j(C)$ (existential scaling) is associated with o, when o has *at least one* r_j link directed to one object of the extent of C, i.e. $r_j(o) \cap extent(C) \neq \emptyset$. $\exists\supseteq r_j(C)$ (contains scaling) is associated with o, when o has r_j links directed *to all* objects of the extent of C, i.e. $extent(C) \subseteq r_j(o)$ and $extent(C) \neq \emptyset$.

Some other relational scaling operators exist in RCA, e.g. with percentages to relax $\exists\supseteq$ or $\exists\forall$ constraints, and cardinalities following the classical role restriction semantics in DLs.

In the general case, the data model can be cyclic at the schema level and at the individual level, resulting in an iterative process which converges after a number of steps depending on the dataset. Accordingly, the RCAexplore[4] system allows a variety of analyses: changing at each step the scaling quantifiers associated with the object-object contexts, the selected formal contexts and relations, and the conceptual structure which is built (concept lattice, AOC-poset (Berry et al. 2014) or iceberg lattice (Stumme et al. 2002)).

In the current example, applying the existential quantifier to all object-object relations, the iterative RCA process outputs three lattices, respectively for dishes, cereals and countries. The concept lattices are shown in Fig. 5 and the building process terminates after three steps. Here after, we detail the main steps of the concept construction:

- Step 0: *redRice*, *arborioRice*, *thaiRice* and *basmatiRice* are grouped together due to the common attribute *rice* (concept $cereal5$). Then *Pakistan* and *Thailand* are grouped together due to the common attribute *Asia* (concept $country5$).
- Step 1: relational attributes of the form $\exists hmc(C)$ (respectively $\exists ipi(C)$ and $\exists elo(C)$) where C is a concept built at step 0 are added to the dishes descriptions (resp. cereals and countries descriptions). All dishes have a cereal in $cereal5$ extent, thus they have the relational attribute $\exists hmc(cereal5)$. Then *basmatiRice* and *thaiRice* are produced in a country located in $country5$ (Asian countries), thus they have the relational attribute $\exists ipi(country5)$ and they can be grouped into concept $cereal6$ (cereals produced in an Asian country).
- Step 2: dishes *khaoManKai* and *biryani* are grouped into concept $dish7$ as they share the relational attribute $\exists hmc(cereal6)$, since both have a main cereal in $cereal6$ extent.
- Step 3: countries from $country5$ have the relational attribute $\exists elo(dish7)$ because people from both countries eat a lot of one of the dishes in $dish7$ extent.

The concept lattices show a classification of dishes depending on their main cereals, which in turn are classified through their attributes (e.g. being rice varieties) and

[4] http://dataqual.engees.unistra.fr/logiciels/rcaExplore.

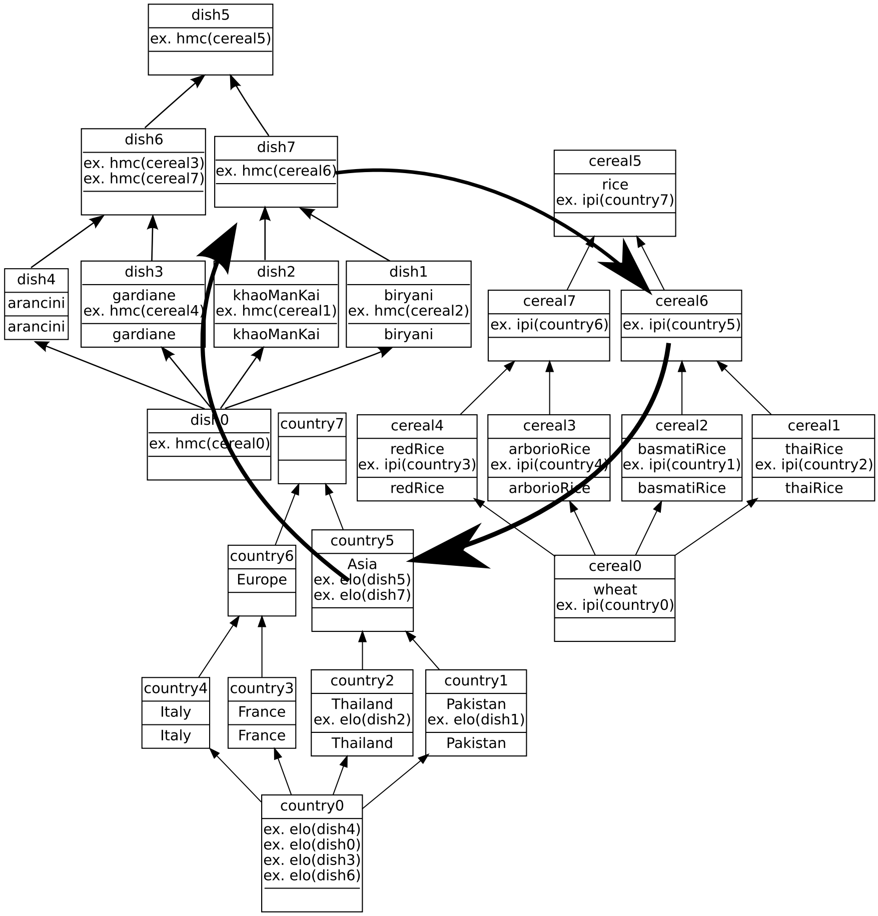


Fig. 5 The three lattices classify objects, i.e. dishes, cereals, and countries, w.r.t. their plain and relational attributes. The label "ex. hmc(cereal3)" stands for "$\exists hasMainCereal(cereal3)$" ("ipi" stands for "isProducedIn" and "elo" for "eatLotOf")

their relations to the countries where they are produced. Finally the countries are classified depending on their attributes, here the continents, and their relations to the dishes that are regularly eaten. While the concepts group sets of similar objects, the relational attributes group sets of similar links. The lattices may also show implications between relational attributes or between relational and non-relational attributes. All analyses that can be made with FCA here apply, with the knowledge of object-object relations.

Actually, RCA is a powerful mechanism for managing relations in the framework of FCA. Compared to FCA approaches based on graph patterns, RCA provides a complementary view, focusing on object classification inside object categories. Some approaches, such as Nica et al. (2017) in the context of sequential data, explore the transformations between the two representations, and extract, from the concept lattices produced with RCA, graph patterns included in a subsumption hierarchy. This hierarchy can be navigated by experts and allows a straightforward interpretation.

There are not many tools implementing RCA. Let us mention Galicia[5] and RCA-explore.[6] RCA has been used for UML class model refactoring (Dao et al. 2004; Guédi et al. 2013), model transformation learning (Dolques et al. 2010), legal document retrieval (Mimouni et al. 2015), ontology construction (Bendaoud et al. 2008), relational rule extraction in hydrobiological data (Dolques et al. 2016). In some applications, AOC-posets (for "Attributes-Objects-Concept partially ordered sets") (Berry et al. 2014) are used rather than concept lattices for efficiency purposes.

4.2 Graph-FCA

Graph-FCA (Ferré 2015; Ferré and Cellier 2016) is another extension of FCA for multi-relational data, and in particular for knowledge graphs of Semantic Web (Hitzler et al. 2009). The specific nature of Graph-FCA is to extract n-ary concepts from a knowledge graph using n-ary relationships. The extents of n-ary concepts are sets of n-ary tuples of graph nodes, and their intents are expressed as graph patterns with n distinguished nodes, which are called the "focus". For instance, in a knowledge graph that represents family members with a "parent" binary relationship, the "sibling" binary concept can be discovered, and described as "a pair of persons having a common parent". Classical FCA corresponds to the case where $n = 1$, i.e. when graph nodes are disconnected, and when concept extents are sets of graph nodes. Graph-FCA differs from Graal (Liquiere and Sallantin 1998) and applications of Pattern Structures to graphs (Kuznetsov 1999; Kuznetsov and Samokhin 2005). Here objects are the nodes of one large knowledge graph instead of having each object being described by a small graph. Graph-FCA shares theoretical foundations with the work of Kötters (2013) and brings in addition algorithms for computing and displaying concepts (Ferré and Cellier 2016).

Whereas FCA defines a formal context as an incidence relation between objects and attributes, Graph-FCA defines a "graph context" as an incidence relation between tuples of objects and attributes. A graph context is a triple $K = (G, M, I)$, where G is a set of objects, M is a set of attributes, and $I \subseteq G^* \times M$ is an incidence relation between object tuples $\overline{g} = (g_1, \ldots, g_n) \in G^*$, for any arity n, and attributes $m \in M$. $G^* = \bigcup_{n \in \mathbb{N}^*} G^n = G \cup (G \times G) \cup (G \times G \times G) \cup \ldots$ denotes the set of all tuples of objects ($\mathbb{N}^*$ denotes the set of natural numbers strictly greater than 0).

The graphical representation of a graph context uses objects as nodes, incidence elements as hyper-edges, and attributes as hyper-edge labels. Attributes can be interpreted as n-ary predicates, and graph contexts as First Order Logic (FOL) model

[5] http://www.iro.umontreal.ca/~galicia/.

[6] http://dataqual.engees.unistra.fr/logiciels/rcaExplore.

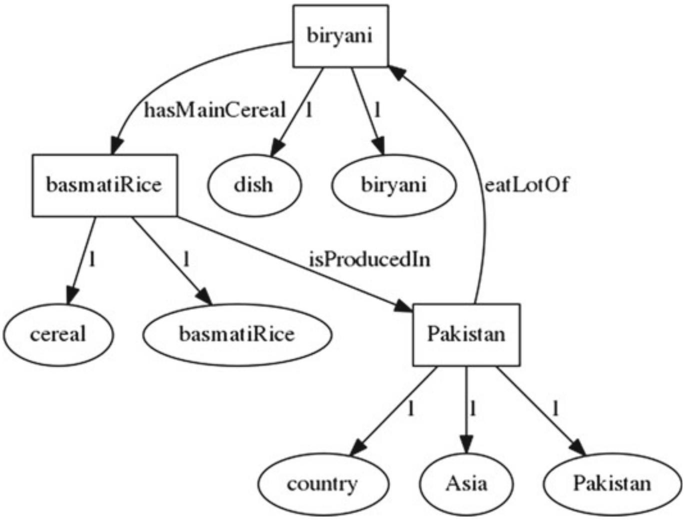


Fig. 6 Graph context (excerpt) about dishes, cereals, and countries. Rectangles are objects, word-labeled links are binary edges, and ellipses are other edges, here unary edges

without functions and constants. For example, a hyper-edge $((g_1, \ldots, g_n), m)$ can be seen as the FOL atom $m(g_1, \ldots, g_n)$.

Different kinds of knowledge graphs, such as Conceptual Graphs (Sowa 1984), RDF graphs, or RCA contexts, can be directly mapped to a graph context. We illustrate Graph-FCA on the same relational data as RCA in the previous section. Fig. 6 shows an excerpt of the graphical representation of the graph context about dishes, cereals, and countries. The objects are dishes (e.g. *biryani*), cereals (e.g. *basmatiRice*), and countries (i.e. *Pakistan*). They are represented as rectangles. The attributes are either unary relations, i.e. classical FCA attributes (e.g., *dish*, *Asia*, *basmatiRice*), or binary relations (e.g., *eatLotOf*, *hasMainCereal*, *isProducedIn*). The former are represented as ellipses attached to objects and the latter are represented as directed edges between objects. More generally, a binary edge $m(x, y)$ is represented by an edge from x to y labeled by m. Other edges $m(x_1, \ldots, x_n)$ are represented as ellipses labeled by m, having an edge labeled i to each node x_i.

Whereas FCA is about finding closed sets of attributes, Graph-FCA is about finding closed graph patterns. A graph pattern is similar to a graph context having variables instead of objects as nodes, in order to generalize over particular objects. A key aspect of Graph-FCA is that closure does not apply directly to graph patterns but to "Projected Graph Patterns" (PGP), i.e. graph patterns with one or several distinguished nodes as focus. Those focus nodes define a projection on the occurrences of the pattern, like a projection in relational algebra. For example, the PGP $(x, y) \leftarrow parent(x, z), parent(y, z)$ defines a graph pattern with two edges, $parent(x, z)$ and $parent(y, z)$, and with focus on variables x, y. It means that for every occurrence of the pattern in the context, the valuation of (x, y) is an occurrence of the PGP. This can be used as a definition of the "sibling" relationship, i.e. the fact that x and y are siblings if they have a common parent z.

PGPs are analogous to anonymous definitions of FOL predicates and to conjunctive SPARQL queries. They play the same role as sets of attributes in FCA, i.e. as concept intents. Set operations are extended from sets of attributes to PGPs. PGP inclusion $\subseteq_q$ is based on graph homomorphisms (Hahn and Tardif 1997). It is similar to the notion of subsumption on queries (Chekol et al. 2012) or rules (Muggleton and De Raedt 1994). PGP intersection $\cap_q$ is defined as a form of graph alignment, where each pair of variables from the two patterns becomes a variable of the intersection pattern. It corresponds to the "categorical product" of graphs (see Hahn and Tardif 1997, p. 116).

The Galois connection underlying the concept construction in Graph-FCA is defined between PGPs $(\mathcal{Q}, \subseteq_q)$ and sets of object tuples $(2^{G^*}, \subseteq)$. The connection from PGP Q to sets of object tuples Q' is analogous to query evaluation, and the connection from sets of object tuples R to PGP R' is analogous to relational learning (Muggleton and De Raedt 1994). In the definitions of Q' and R' below, the PGP $(\overline{g}, I)$ represents the description of an object tuple $\overline{g}$ by the whole incidence relation I seen from the relative position of $\overline{g}$.

$$\begin{array}{ll} Q' := \{\overline{g} \in G^n \mid Q \subseteq_q (\overline{g}, I)\}, & \text{for } Q = (x_1, \ldots, x_n) \leftarrow P \text{ (a PGP)} \\ R' := \cap_q \{(\overline{g}, I)\}_{\overline{g} \in R}, & \text{for } R \subseteq 2^{G^n}, \text{ for } n \in \mathcal{N} \end{array}$$

From there, concepts can be defined in the usual way and proved to be organized into lattices. A concept is a pair (Q, R) such that $Q' = R$ and $R' =_q Q$. The arity of the projected tuple of Q must be the same as the arity of object tuples in R. It determines the arity of the concept. Unary concepts are about sets of objects, while binary concepts are about relationships between objects, and so on. It can be noticed that the intent of a unary concept can combine attributes of different arities. Unlike RCA, there is a concept lattice for each concept arity rather than for each object type. Figure 7 displays a compact representation of the graph concepts about dishes, cereals, and countries. Each rectangle node x identifies a unary concept (e.g., Q2a) along with its extent (here, *Pakistan*, *Thailand*). The concept intent is the PGP $x \leftarrow P$, where P is the subgraph containing node x and all white nodes, which is called the "pattern core". By reading the graph, we learn that Concept Q2a is the concept of "Asian countries, which eat a lot of some dish whose main cereal is produced in the country itself". Formally, its intent is denoted by $c \leftarrow country(c), Asia(c), eatLotOf(c, d), dish(d), hasMainCereal(d, l), cereal(l), isProducedIn(l, c)$. Concepts Q2b and Q2c have the same graph pattern as Q2a but a different focus, on cereals for Q2b and on dishes for Q2c. N-ary concepts are obtained by picking several nodes as focus. For example, (Q2a,Q2c,Q2b) is a ternary concept whose instances are the object triples $(Pakistan, biryani, basmatiRice)$ and $(Thailand, khaoManKai, thaiRice)$. It represents the cyclic relation existing between country, dish, and cereal in Asian countries.

Although the generalization ordering between concepts, hence the concept lattice, is not explicitly represented in Fig. 7, it can be recovered by looking for inclusion

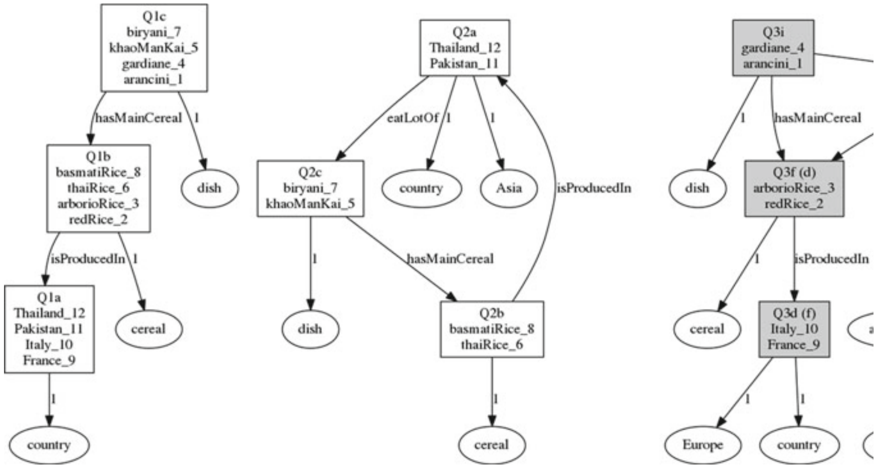


Fig. 7 Compact representation of graph concepts about dishes, cereals, and countries, with minimum support 2

relationships between concept extents. For example, concepts Q2a, Q2b, Q2c are respective specializations, hence sub-concepts, of concepts Q1a, Q1b, Q1c. The latter are general concepts respectively for countries, cereals, and dishes. They show that every dish has a main cereal, and every cereal is produced in some country but that not all countries eat a lot of some dish, and not all dishes are eaten a lot by some country. The (grayed) concepts Q3i, Q3f(d), and Q3d(f), are other specializations of Q1 concepts for European countries (Q3d(f)): Q3f(d) denotes the concept of cereals produced in Europe and Q3i denotes the concept of dishes whose main cereal is produced in Europe. The bracketed letter "(f)" in concept Q3d (with label "Q3d(f)") indicates that concept Q3f belongs to the intent of concept Q3d, in addition to the concepts of the pattern core. Similarly, concept Q3d belongs to the intent of concept Q3f (with label "'Q3f(d)"). Contrasting Asian countries no cycle is observed because rice-based dishes are less popular in Europe. Those concepts are grayed because they do not belong to the pattern core, which is here a copy of a component of the graph context. Concepts in gray represent generalizations over other concepts in the pattern core. For instance, concept Q3i generalizes over concepts Q3g and Q3h (not visible in Fig. 7), which have only *gardiane* and *arancini* in their extents, respectively.

Compared to RCA, Graph-FCA emphasizes relational patterns over the lattice structure. However, relational patterns can be extracted from RCA lattices, and concept lattices can be recovered from Graph-FCA relational patterns. Other differences are about the scaling operators and the meaning of cycles in the intents of concepts. Graph-FCA does not support scaling operators. Moreover a cycle in a Graph-FCA pattern actually corresponds to a cycle in the data, whereas this is not necessarily the case for cycles through RCA concept definitions.

5 Triadic Concept Analysis

Triadic Concept Analysis (TCA Lehmann and Wille 1995) handles ternary relations between objects, attributes and conditions. Such triadic data can be represented as a "triadic context", i.e. a kind of 3-dimensional table or a cube. Accordingly, "triadic concepts" are 3-dimensional and can be seen as maximal sets of objects related to maximal set of attributes under a maximal set of conditions, i.e. a maximal "sub-cube" full of $\times$ in the triadic context (up to rows, columns and layers permutations).

Definition 1 (*Triadic context*) In a triadic context $\mathbb{K} = (G, M, B, Y)$, G, M, and B respectively denote the sets of objects, attributes and conditions, and $Y \subseteq G \times M \times B$. The fact $(g, m, b) \in Y$ is interpreted as the statement "object g has attribute m under condition b".

Example 1 An example of such a triadic context is given in Table 2 where the very first cross (to the left) denotes the fact "object g_2 has attribute m_1 under the condition b_1", i.e. $(g_2, m_1, b_1) \in Y$. In this tabular representation, each table corresponds to the projection of the triadic context for one condition. Projections can be performed for any dimension, i.e. object, attribute and condition.

Definition 2 (*Triadic concept*) A triadic concept (A_1, A_2, A_3) of (G, M, B, Y) is a triple with $A_1 \subseteq G$, $A_2 \subseteq M$ and $A_3 \subseteq B$ and satisfying the two following statements:
(i) $A_1 \times A_2 \times A_3 \subseteq Y$
(ii) for $X_1 \times X_2 \times X_3 \subseteq Y$, $A_1 \subseteq X_1$, $A_2 \subseteq X_2$ and $A_3 \subseteq X_3$ implies that $(A_1, A_2, A_3) = (X_1, X_2, X_3)$ (maximality).

If (G, M, B, Y) is represented as a three dimensional table, (i) means that a concept stands for a rectangular parallelepiped full of $\times$ while (ii) characterizes component-wise maximality of concepts. A_1 is the "extent", A_2 the "intent" and A_3 the "modus" of the triadic concept (A_1, A_2, A_3).

Example 2 $(\{g_3, g_4\}, \{m_2, m_3\}, \{b_1, b_2, b_3\})$ is a triadic concept in the triadic context shown in Table 2. Representing the triadic context as a cube, where each condition is a layer, one can observe that this triadic concept corresponds to a maximal rectangular parallelepiped full of $\times$ (modulo lines, columns and layers permutations).

To describe the derivation operators, it is convenient to represent a triadic context as (K_1, K_2, K_3, Y).

Definition 3 (*Outer derivation operators*) For $\{i, j, k\} = \{1, 2, 3\}$, $j < k$, $X \subseteq K_i$ and $Z \subseteq K_j \times K_k$, the (i)-derivation operators are defined by:

$$\Phi : X \rightarrow X^{(i)} : \{(a_j, a_k) \in K_j \times K_k \mid (a_i, a_j, a_k) \in Y \text{ for all } a_i \in X\}$$
$$\Phi^{'} : Z \rightarrow Z^{(i)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in Z\}$$

Table 2 A triadic context (G, M, B, Y) with the triadic concept $(\{g_3, g_4\}, \{m_2, m_3\}, \{b_1, b_2, b_3\})$

	b_1			b_2			b_3		
	m_1	m_2	m_3	m_1	m_2	m_3	m_1	m_2	m_3
g_1			×	×	×	×	×		×
g_2	×	×		×	×		×		
g_3		×	×	×	×	×	×	×	×
g_4		×	×		×	×		×	×

These two derivation operators lead to 3 dyadic contexts:

$$\mathbb{K}^{(1)} = \langle K_1, K_2 \times K_3, Y^{(1)} \rangle$$
$$\mathbb{K}^{(2)} = \langle K_2, K_1 \times K_3, Y^{(2)} \rangle$$
$$\mathbb{K}^{(3)} = \langle K_3, K_1 \times K_2, Y^{(3)} \rangle$$

where $gY^{(1)}(m, b) \iff mY^{(2)}(g, b) \iff bY^{(3)}(g, m)$.

Example 3 Consider $i = 1$, $j = 2$ and $k = 3$, i.e. $K_1 = G$, $K_2 = M$ and $K_3 = B$. Given the arbitrary set of objects $X = \{g_4\}$, it comes:

$$\Phi(X) = \{(m_2, b_1), (m_3, b_1), (m_2, b_2), (m_3, b_2), (m_2, b_3), (m_3, b_3)\}$$
$$\Phi'\Phi(X) = \{g_3, g_4\}$$

Further derivation operators are defined as follows:

Definition 4 (*Inner derivation operators*) For $\{i, j, k\} = \{1, 2, 3\}$, $X_i \subseteq K_i$, $X_j \subseteq K_j$ and $A_k \subseteq K_k$, the (i, j, A_k)-derivation operators are defined by:

$$\Psi : X_i \to X_i^{(i,j,A_k)} : \{a_j \in K_j \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_i, a_k) \in X_i \times A_k\}$$
$$\Psi' : X_j \to X_j^{(i,j,A_k)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in X_j \times A_k\}$$

This definition yields the derivation operators of dyadic contexts defined by:

$$\mathbb{K}_{A_k}^{ij} = \langle K_i, K_j, Y_{A_k}^{ij} \rangle \text{ where } (a_i, a_j) \in Y_{A_k}^{ij} \iff a_i, a_j, a_k \text{ are related by } Y \text{ for all } a_k \in A_k.$$

Example 4 Consider $i = 1$, $j = 2$ and $k = 3$, i.e. $K_1 = G$, $K_2 = M$ and $K_3 = B$, $A_3 = \{b_1, b_2\}$ and $X = \{g_3\}$:

$$\Psi(X) = \{m_2, m_3\} \qquad \Psi'\Psi(X) = \{g_3, g_4\}$$

Operators Φ and Φ' are called "outer operators", a composition of both operators is called an "outer closure". Operators Ψ and Ψ' are called "inner operators", a composition of them is called "inner closure".

Definition 5 (*Triadic concept formation*) A concept having X_1 in its extent can be constructed as follows:

$$(X_1^{(1,2,A_3)(1,2,A_3)}, X_1^{(1,2,A_3)}, (X_1^{(1,2,A_3)(1,2,A_3)} \times X_1^{(1,2,A_3)})^{(3)})$$

Example 5 In the current example, $(\{g_3, g_4\}, \{m_2, m_3\}, \{b_1, b_2, b_3\})$ is a triadic concept.

From a computational point of view, the algorithm TRIAS is developed in Jäschke et al. (2006) for extracting frequent triadic concepts, i.e. concepts whose extent, intent and modus cardinalities are higher than user-defined thresholds (see also Ji et al. 2006). Cerf et al. from the "pattern mining community" presented a more efficient algorithm called DATA- PEELER which is able to handle n-ary relations (Cerf et al. 2009), following the formal definitions given in terms of "Polyadic Concept Analysis" in Voutsadakis (2002). Some examples of triadic analysis capabilities are given in Kaytoue et al. (2014) with a formalization of biclustering and in Ganter and Obiedkov (2004) for illustrating the use of implications in triadic analysis. Moreover, a comparison of algorithms dealing with triadic analysis is provided in Ignatov et al. (2015).

Contrasting the intuitive graphical representation of a concept lattice in FCA, the exploration of a triadic conceptual structure is not easy task, especially because of the complexity of such a triadic structure. In Rudolph et al. (2015), a new navigation paradigm is proposed for triadic conceptual structures based on a neighborhood notion arising from associated dyadic concept lattices. This visualization capability helps an analyst to understand the construction and the content of a triadic structure.

6 Applications

FCA and extensions are used in many application domains and in many different tasks. Some of these tasks were already mentioned before and we propose below a quick tour of some representative applications for completing the picture. It should also be noticed that we cannot be exhaustive and that some surveys are existing such as Poelmans et al. (2013), Kuznetsov and Poelmans (2013).

Information Retrieval

FCA has been used in a myriad of ways to support a wide variety of information retrieval (IR) techniques and applications (see (Codocedo and Napoli 2015) for a survey). The concept lattice concisely represents the document and query space which can be used as an index for automatic retrieval, and as a navigation structure (Godin et al. 1993; Lindig 1995; Carpineto and Romano 1996; Ducrou et al. 2006). Eklund's team has developed algorithms to efficiently compute neighbor concepts, not only children concepts but also parent concepts and sibling concepts, and have designed user-friendly and multimedia user interfaces for various applications such as email, pictures, or museum collections (Ducrou et al. 2006). However, the Boolean IR model, and consequently FCA, is considered as too limited for modern IR requirements, such as large datasets and complex document representations. This is the motivation for introducing more complex model and in particular Logical Concept Analysis (LCA) and pattern structures, to reconcile search by navigation in a lattice and search by expressive querying.

Logical Concept Analysis (Ferré and Ridoux 2000) was introduced with the purpose to reconcile navigation and querying, combined with rich object descriptions. It led to the paradigm of Logical Information Systems (LIS) (Ferré and Ridoux 2004), where the document and query space can be composed in a flexible way from various components called "logic functors" (Ferré and Ridoux 2001). There are logic functors for different datatypes (e.g., numbers, dates, strings), different structures (e.g., valued attributes, taxonomies, sequences), and different logical operators (e.g., Boolean, epistemology). LIS have been applied to collections of photos (Ferré 2009), to biological sequences (Ferré and King 2004), to geographical data (Bedel et al. 2008), to complex linguistic data (Foret and Ferré 2010), as well as to OLAP cubes (Ferré et al. 2012). LIS have even been implemented as a genuine UNIX file system (Padioleau and Ridoux 2003). In line with relational extensions of FCA (RCA and Graph-FCA), LIS have been extended to semantic web knowledge graphs (Ferré 2010). The richness of the query space has since increased up for covering most of the SPARQL 1.1 query language, including computations such as aggregations, while retaining navigation-based search (Ferré 2017).

Web of Data and Ontology Engineering

There are many connections between FCA, web of data and semantic web. The idea of a "conceptual hierarchy" is fundamental to both Formal Concept Analysis and Description Logics (and thus semantic web). If the concept construction is quite different, there are several attempts to relate FCA-based and DL-based formalisms, and to combine both approaches (Sertkaya 2010).

In Baader et al. (2007), an approach for extending both the terminological and the assertional part of a Description Logic knowledge base is proposed, using information provided by the knowledge base and by a domain expert. Materializing this approach, the "OntoComP" system is a Protégé plugin that supports ontology engineers in completing OWL ontologies using conceptual exploration (Sertkaya 2009). Related approaches were also used in the discovery of axioms or definitions in ontologies (Baader and Distel 2008; Borchmann et al. 2016).

In Kirchberg et al. (2012), authors discuss how to build a "concept layer" above the web of data in an efficient way, for allowing a machine-processable web content. The emphasis should be on creating links in a way that both humans and machines can explore the web of data. Indeed, FCA can bring the level of concept abstraction that makes possible to link together semantically-related facts as meaningful units. Following the same idea, a noticeable application of pattern structures aimed at RDF data completion is discussed in Alam et al. (2015). This shows the great potential of pattern structures to support complex document representations with numerical and heterogeneous indexes (Codocedo et al. 2014; Codocedo and Napoli 2014a).

We should also mention some applications of FCA and pattern structures in text mining. One of the very first approaches to the automatic acquisition of concept hierarchies from a text corpus based on FCA is detailed in Cimiano et al. (2005). Following this way, authors in Bendaoud et al. (2008) combined FCA and RCA to take into account relations within texts and build more realistic DL-based concepts where roles correspond to the extracted relations. Moreover, the learning of expressive ontology axioms from textual definitions with the so-called "relational exploration" (Rudolph 2004) is proposed in Völker and Rudolph (2008). Relational exploration is also based on attribute exploration which is used to interactively clarify underspecified dependencies and increase the quality of the resulting ontological elements. More recently, a specific pattern structure for managing syntactic trees was introduced in Leeuwenberg et al. (2015). This work was aimed at analyzing and classifying relations such as drug-drug interactions in medical texts where sentences are represented as syntactic trees.

Biclustering and Recommendation

Biclustering aims at finding local patterns in numerical data tables. The motivation is to overcome the limitation of standard clustering techniques where distance functions using all the attributes may be ineffective and hard to interpret. Applications are numerous in biology, recommendation, etc. (see references in Kaytoue et al. 2014; Codocedo and Napoli 2014b). In FCA, formal concepts are maximal rectangles of $true$ values in a binary data-table (modulo columns/rows permutations). Accordingly, concepts are binary biclusters with interesting properties: maximality (through a closure operator), overlapping and partial ordering of the local patterns. Such properties are key elements of a mathematical definition of numerical biclusters and the design of their enumeration algorithms. We highlight these links for several types of biclusters with interval (Kaytoue et al. 2011a) and partition pattern structures (Codocedo and Napoli 2014b) and their representation contexts. Next investigations concern dimensionality: a bijection between n-clusters and $n + 1$-concepts is proven (Kaytoue et al. 2014).

Some work about recommendation using plain FCA (du Boucher-Ryan and Bridge 2006) and Boolean Matrix Factorization (Akhmatnurov and Ignatov 2015) should be mentioned.

Database and Functional Dependencies

Characterizing and computing functional dependencies (FDs) are an important topic in database theory (see e.g. references in Baixeries et al. 2014). In FCA, Ganter & Wille proposed a first characterization of FDs as implications in a formal context (binary relation) obtained after a transformation of the initial data (Ganter and Wille 1999). However, one has to create here n^2 objects from the n initial tuples. To overcome this problem, a characterization of functional dependencies is proposed in terms of partition pattern structures offering additional benefits for the computation of dependencies (Baixeries et al. 2014). This method can be naturally generalized to other types of FDs (multi-valued and similarity dependencies Baixeries et al. 2013; Codocedo et al. 2016).

Moreover, an interactive and visual way to discover simultaneously FDs, conditional FDs (i.e., FDs valid under certain conditions), and association rules is proposed in Allard et al. (2010). It is based on navigating a lattice of OLAP cubes whose dimensions correspond to the premise of functional dependencies and association rules.

Software Engineering

A survey by Tilley et al. (2005) gathers and analyzes main research work that applied FCA to the field of Software Engineering (SE) before 2005. One of the oldest is due to R. Godin and H. Mili in the context of Smalltalk class hierarchy reengineering (Godin and Mili 1993), where they introduced ideas that inspired many later approaches.

Another main track has been initiated by Lindig (1995) that aims to classify software components. In the nineties, object identification was also a major topic, due to the spreading of object-orientation and the importance to migrate from procedural code to object-oriented code, for which FCA approaches were proposed (Sahraoui et al. 1997; van Deursen and Kuipers 1999).

Then, FCA continued to expand in software engineering, e.g. to facilitate fault localization in software (Cellier et al. 2008). Abilities of FCA for classifying components or web services have been investigated in Bruno et al. (2005), Aboud et al. (2009), Azmeh et al. (2011). Connections to the software product line representations (feature models) are studied in Carbonnel et al. (2017a, b). Moreover, RCA has been applied in Model-Driven Engineering, for UML class model or use case diagram analysis and reengineering (Dao et al. 2004; Falleri et al. 2008; Guédi et al. 2013), model transformation learning (Dolques et al. 2010), a topic for which an FCA-based graph mining model was also designed (Saada et al. 2014).

Let us conclude with two papers showing the diversity of the applications in software engineering and based on FCA. In the first paper (Obiedkov et al. 2009), the authors are studying the lattice-based access control models using conceptual exploration, for understanding dependencies between different security categories. In the second paper (Priss 2011), the author examines the detection of security threat problems and the way FCA can help in exploring available related data.

Social Network Analysis

The authors of Freeman and White (1993) remarked the visualization power of concept (Galois) lattices and their usefulness for interpretation. Actually, a concept lattice yields a complete and ordered view of the data based on concept extents and intents while in previous models several and separate views are produced. Moreover, the view provided by the concept lattice suggests useful insights about the structural properties of the data.

The book (Missaoui et al. 2017) contains several research papers on recent trends in applying FCA for SNA: acquisition of terminological knowledge from social networks, knowledge communities, individuality computation, FCA-based analysis of two-mode networks, community detection and description in one-mode and multimode networks, multimodal clustering, adaptation of the dual-projection approach to weighted bipartite graphs, and attributed graph analysis.

In particular, the paper (Roth 2017) shows how FCA allows the assessment and analysis of actors and their attributes on an equal basis. FCA can help solve key typical challenges of community detection in SNA such as group hierarchy and overlapping, temporal evolution and stability of networks. The paper (Borchmann and Hanika 2017) defines individuality and introduces a new measure in two-mode (affiliation) networks using FCA by evaluating how many unique groups of users of size *k* can be uniquely defined by a combination of attributes. In Ignatov et al. (2017) the authors present FCA-based biclustering and its extensions to mining multimode communities in social network analysis. The author of Kriegel (2017) describes a technique for the acquisition of terminological knowledge from social networks. The chapter (Soldano et al. 2017) studies social and other complex networks as attributed graphs and addresses attribute pattern mining in such graphs through recent developments in FCA. Finally, in Valverde-Albacete and Peláez-Moreno (2017) the authors adapt a dual-projection approach to weighted two-mode networks using an extension of FCA for incidences with values in a special case of semiring.

Bioinformatics and Chemoinformatics

FCA was used by different research groups as a tool for analyzing biological and chemical data, e.g. for structure-activity relationship studies, where the correlation between chemical structure and biological properties are explored (Bartel and Bruggemann 1998; Blinova et al. 2003; Kuznetsov and Samokhin 2005; Métivier et al. 2015; Quintero and Restrepo 2017). In Gardiner and Gillet (2015), authors describe four data mining techniques, namely Rough Set Theory (RST), Association Rule Mining (ARM), Emerging Pattern Mining (EP), and Formal Concept Analysis (FCA), and give a list of their chemoinformatics applications.

In bioinformatics FCA was also applied to gene expression analysis in Kaytoue et al. (2011c) and in the analysis of metabolomic data in Grissa et al. (2016). In the latter, FCA is combined to numerical classifiers for selecting and visualizing discriminant and predictive features in nutrition data.

In molecular biology, the exploration of potentially interesting substructures within molecular graphs requires proper abstraction and visualization methods. In Bourneuf and Nicolas (2017), the so-called "power graph analysis" based on classes of nodes having similar properties and classes of edges linking node classes is stated in terms of FCA. Then the problem is solved in an alternative way thanks to "answer set programming". Enzymes are macro-molecules, i.e. linear sequences of molecules, whose activity is basic in any biochemical reaction. Predicting the functional activity of an enzyme from its sequence is a crucial task that can be approached by comparing new target sequences with already known source enzymes. In Coste et al. (2014), the authors study the problem in the framework of FCA and define this task as an optimization problem on the set of concepts.

Finally, median networks which generalize the notion of trees in phylogenetic analysis are investigated within FCA in Priss (2013). In standard situations, concept lattices may represent the same information as median networks, but the FCA algorithmic machinery is powerful and operational, and offers efficient algorithms that can be used for evolutionary analysis. Moreover, it should be noticed that median graphs are also related to distributive lattices.

7 Conclusion

FCA is at the heart of knowledge discovery, knowledge representation and reasoning, data analysis and classification. As pointed out in Wille (2002), formal concepts and concept lattices provide a mathematization of real-world concept hierarchies, support reasoning and various other complex tasks, especially using the graphical representation of concept lattices. In such a framework, knowledge discovery can be considered as a whole process, from data to knowledge units, more practically, as pattern discovery associated with knowledge creation. Such a process is guided by the design of concepts and concept lattices, and as well by the representation of concepts within a knowledge representation formalism such as description logics. In particular, relational models in FCA and triadic concept analysis are of main importance in the representation of relations between concepts.

Regarding the wide range of applications in which FCA is involved, it is clear that FCA is now recognized as a formalism for data and knowledge processing that can be really useful for AI practitioners. More than that, we strongly believe that FCA will play a first role in data and knowledge sciences in the forthcoming years.

Acknowledgements The work of Sergei O. Kuznetsov was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Russia.

References

Aboud NA, Arévalo G, Falleri J, Huchard M, Tibermacine C, Urtado C, Vauttier S (2009) Automated architectural component classification using concept lattices. In: joint working IEEE/IFIP conference on software architecture (WICSA/ECSA), pp 21–30

Akhmatnurov M, Ignatov, DI (2015) Context-aware recommender system based on boolean matrix factorisation. In: Proceedings of the twelfth international conference on concept lattices and their applications (CLA), CEUR workshop proceedings, vol 1466, pp 99–110

Alam M, Buzmakov A, Codocedo V, Napoli A (2015) Mining definitions from RDF annotations using formal concept analysis. In: Proceedings of the twenty-fourth international joint conference on artificial intelligence, (IJCAI). AAAI Press, pp 823–829

Alam M, Le TNN, Napoli A (2016) LatViz: a new practical tool for performing interactive exploration over concept lattices. In: Proceedings of the thirteenth international conference on concept lattices and their applications (CLA), CEUR workshop proceedings, vol 1624, pp 9–20

Allard P, Ferré S, Ridoux O (2010) Discovering functional dependencies and association rules by navigating in a lattice of OLAP views. In: Proceedings of the 7th international conference on concept lattices and their applications (CLA), CEUR workshop proceedings, vol 672, pp 199–210

Azmeh Z, Driss M, Hamoui F, Huchard M, Moha N, Tibermacine C (2011) Selection of composable web services driven by user requirements. In: IEEE international conference on web services (ICWS), pp. 395–402
Baader F, Distel F (2008) A finite basis for the set of EL-implications holding in a finite model, vol 4933. Springer, Berlin, pp 46–61
Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider P (eds) (2003) The description logic handbook. Cambridge University Press, Cambridge
Baader F, Ganter B, Sertkaya B, Sattler U (2007) Completing description logic knowledge bases using formal concept analysis. In: Proceedings of the 20th international joint conference on artificial intelligence (IJCAI), pp 230–235
Babin MA, Kuznetsov SO (2012) Approximating concept stability. In: Proceedings of the 10th international conference on formal concept analysis (ICFCA). LNCS, vol 7228. Springer, pp 7–15
Babin MA, Kuznetsov SO (2013) Computing premises of a minimal cover of functional dependencies is intractable. Discret Appl Math 161(6):742–749
Baixeries J, Kaytoue M, Napoli A (2013) Computing similarity dependencies with pattern structures. In: Proceedings of the 10th international conference on concept lattices and their applications (CLA), CEUR workshop proceedings, vol 1062, pp 33–44
Baixeries J, Kaytoue M, Napoli A (2014) Characterizing functional dependencies in formal concept analysis with pattern structures. Ann Math Artif Intell 72(1–2):129–149
Barbut M, Monjardet B (1970) Ordre et classification: algèbre et combinatoire. Hachette, Paris
Bartel H-G, Bruggemann R (1998) Application of formal concept analysis to structure-activity relationships. Fresenius J Anal Chem 361:23–38
Bastide Y, Taouil R, Pasquier N, Stumme G, Lakhal L (2000) Mining frequent patterns with counting inference. SIGKDD Explor Newsl 2(2):66–75
Becker P (2004) Numerical analysis in conceptual systems with ToscanaJ. In: Proceedings of the second international conference on formal concept analysis (ICFCA). LNCS, vol 2961. Springer, pp 96–103
Bedel O, Ferré S, Ridoux O (2008) Handling spatial relations in logical concept analysis to explore geographical data. In: Proceedings of the 6th international conference on formal concept analysis (ICFCA). LNCS, vol 4933. Springer, pp 241–257
Belfodil A, Kuznetsov SO, Robardet C, Kaytoue M (2017) Mining convex polygon patterns with formal concept analysis. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence (IJCAI), pp 1425–1432
Belohlávek R (2004) Concept lattices and order in fuzzy logic. Ann Pure Appl Log 128(1–3):277–298
Belohlávek R (2008) Relational data, formal concept analysis, and graded attributes. Handbook of research on fuzzy information processing in databases. IGI Global, pp 462–489
Belohlávek R (2011) What is a fuzzy concept lattice? (II). In: Proceedings of the 13th international conference on rough sets, fuzzy sets, data mining and granular computing (RSFDGrC). LNCS, vol 6743. Springer, pp 19–26
Belohlávek R, Vychodil V (2005) What is a fuzzy concept lattice? Proceedings of the twelfth international conference on concept lattices and their applications (CLA), pp 34–45
Bendaoud R, Napoli A, Toussaint Y (2008) Formal concept analysis: a unified framework for building and refining ontologies. In: Proceedings of the 16th international conference on knowledge engineering and knowledge management (EKAW), pp 156–171
Berry A, Gutierrez A, Huchard M, Napoli A, Sigayret A (2014) Hermes: a simple and efficient algorithm for building the AOC-poset of a binary relation. Ann Math Artif Intell 72(1–2):45–71
Bertet K, Monjardet B (2010) The multiple facets of the canonical direct unit implicational basis. Theor Comput Sci 411(22–24):2155–2166
Blinova VG, Dobrynin DA, Finn VK, Kuznetsov SO, Pankratova ES (2003) Toxicology analysis by means of the JSM-method. Bioinformatics 19(10):1201–1207

Borchmann D, Hanika T (2017) Individuality in social networks. Formal concept analysis of social networks. Springer, Berlin, pp 19–40
Borchmann D, Distel F, Kriegel F (2016) Axiomatisation of general concept inclusions from finite interpretations. J Appl Non-Class Log 26(1):1–46
Bosc G, Boulicaut J-F, Raïssi C, Kaytoue M (2017) Anytime discovery of a diverse set of patterns with Monte Carlo tree search. Data mining and knowledge discovery (In press)
Bourneuf L, Nicolas, J (2017) FCA in a logical programming setting for visualization-oriented graph compression. In: Proceedings of the 14th international conference on formal concept analysis (ICFCA). LNCS, vol 10308, pp 89–105
Bruno M, Canfora G, Penta MD, Scognamiglio R (2005) An approach to support web service classification and annotation. In: IEEE international conference on e-technology, e-commerce, and e-services (EEE), pp 138–143
Buzmakov A, Kuznetsov SO, Napoli A (2014) Scalable estimates of concept stability, vol 8478. Springer, Berlin, pp 157–172
Buzmakov A, Kuznetsov SO, Napoli A (2015) Revisiting pattern structure projections, vol 9113. Springer, Berlin, pp 200–215
Buzmakov A, Egho E, Jay N, Kuznetsov SO, Napoli A, Raïssi C (2016) On mining complex sequential data by means of FCA and pattern structures. Int J Gen Syst 45(2):135–159
Buzmakov A, Kuznetsov SO, Napoli A (2017) Efficient mining of subsample-stable graph patterns. In: IEEE international conference on data mining (ICDM), pp 757–762
Cabrera IP, Cordero P, Ojeda-Aciego M (2017) Galois connections in computational intelligence: a short survey. In: IEEE symposium series on computational intelligence (SSCI), IEEE, pp 1–7
Carbonnel J, Huchard M, Miralles A, Nebut C (2017a) Feature model composition assisted by formal concept analysis. In: Proceedings of the 12th international conference on evaluation of novel approaches to software engineering (ENASE), pp 27–37
Carbonnel J, Huchard M, Nebut C (2017b) analyzing variability in product families through canonical feature diagrams. In: The 29th international conference on software engineering and knowledge engineering (SEKE), pp 185–190
Carpineto C, Romano G (1996) A lattice conceptual clustering system and its application to browsing retrieval. Mach Learn 24(2):95–122
Carpineto C, Romano G (2004) Concept data analysis: theory and applications. Wiley, Chichester (UK)
Cellier P, Ducassé M, Ferré S, Ridoux O (2008) Formal concept analysis enhances fault localization in software. In: Proceedings of the 6th international conference on formal concept analysis (ICFCA). LNCS, vol 4933. Springer, pp 273–288
Cerf L, Besson J, Robardet C, Boulicaut J-F (2009) Closed patterns meet n-ary relations. ACM Trans Knowl Discov Data 3(1):1–36
Chaudron L, Maille N (2000) Generalized formal concept analysis. In: Proceedings of the 8th international conference on conceptual structures (ICCS). LNCS, vol 1867. Springer, pp 357–370
Chekol MW, Euzenat J, Genevès P, Layaïda N (2012) SPARQL query containment under RDFS entailment regime. In: Proceedings of the 6th international joint conference on automated reasoning (IJCAR), vol 7364. Springer, pp 134–148
Cimiano P, Hotho A, Staab S (2005) Learning concept hierarchies from text corpora using formal concept analysis. J Artif Intell Res 24:305–339
Codocedo V, Napoli A (2014a) A proposition for combining pattern structures an relational concept analysis. In: Proceedings of the 12th international conference on formal concept analysis (ICFCA), vol 8478. Springer, pp 96–111
Codocedo V, Napoli A (2014b) Lattice-based biclustering using partition pattern structures. In: 21st European conference on artificial intelligence (ECAI). IOS Press, pp 213–218
Codocedo V, Napoli A (2015) Formal concept analysis and information retrieval–a survey. In: Proceedings of the 13th international conference on formal concept analysis (ICFCA). LNCS, vol 9113. Springer, pp 61–77

Codocedo V, Lykourentzou I, Napoli A (2014) A semantic approach to concept lattice-based information retrieval. Ann Math Artif Intell 72(1–2):169–195

Codocedo V, Baixeries J, Kaytoue M, Napoli A (2016) Characterization of order-like dependencies with formal concept analysis. In: Proceedings of the thirteenth international conference on concept lattices and their applications (CLA), CEUR workshop proceedings, vol 1624, pp 123–134

Coste F, Garet G, Groisillier A, Nicolas J, Tonon T (2014) Automated enzyme classification by formal concept analysis. In: Proceedings of the 12th international conference on formal concept analysis (ICFCA). LNCS, vol 8478. Springer, pp 235–250

Dao M, Huchard M, Hacene MR, Roume C, Valtchev P (2004) Improving generalization level in UML models iterative cross generalization in practice. In: Proceedings of the 12th international conference on conceptual structures (ICCS), pp 346–360

Davey BA, Priestley HA (1990) Introduction to lattices and order. Cambridge University Press, Cambridge, UK

Distel F, Sertkaya B (2011) On the complexity of enumerating pseudo-intents. Discret Appl Math 159(6):450–466

Dolques X, Huchard M, Nebut C, Reitz P (2010) Learning transformation rules from transformation examples: an approach based on relational concept analysis. In: Workshop proceedings of the 14th IEEE international enterprise distributed object computing conference (EDOCW), pp 27–32

Dolques X, Le Ber F, Huchard M, Grac C (2016) Performance-friendly rule extraction in large water data-sets with AOC posets and relational concept analysis. Int J Gen Syst 45(2):187–210

du Boucher-Ryan P, Bridge DG (2006) Collaborative recommending using formal concept analysis. Knowl-Based Syst 19(5):309–315

Ducrou J, Vormbrock B, Eklund PW (2006) FCA-based browsing and searching of a collection of images. In: Proceedings of the 14th international conference on conceptual structures (ICCS). LNCS, vol 4068. Springer, pp 203–214

Dzeroski S, Lavrac N (eds) (2001) Relational data mining. Springer, Berlin

Eklund PW, Groh B, Stumme G, Wille R (2000) A contextual-logic extension of TOSCANA. In: Proceedings of the 8th international conference on conceptual structures (ICCS), LNCS 1867. Springer, pp 453–467

Eklund PW, Ducrou J, Brawn P (2004) Concept lattices for information visualization: can novices read line-diagrams? In: Proceedings of the second international conference on formal concept analysis (ICFCA). LNCS, vol 2961. Springer, pp 57–73

Falk I, Gardent C (2014) Combining formal concept analysis and translation to assign frames and semantic role sets to French verbs. Ann Math Artif Intell 70(1–2):123–150

Falleri J, Huchard M, Nebut C (2008) A generic approach for class model normalization. In: 23rd IEEE/ACM international conference on automated software engineering (ASE), pp 431–434

Ferré S (2009) Camelis: a logical information system to organize and browse a collection of documents. Int J Gen Syst 38(4):379–403

Ferré S (2010) Conceptual navigation in RDF graphs with SPARQL-like queries. In: Proceedings of the 8th international conference on formal concept analysis (ICFCA), vol 5986. Springer, pp 193–208

Ferré S (2015) A proposal for extending formal concept analysis to knowledge graphs. In: Proceedings of the 13th international conference on formal concept analysis (ICFCA), vol 9113. Springer, pp 271–286

Ferré S (2017) Sparklis: an expressive query builder for SPARQL endpoints with guidance in natural language. Semant Web: Interoperability Usability Appl 8(3):405–418

Ferré S, Cellier P (2016) Graph-FCA in practice. In: Proceedings of the 22nd international conference on conceptual structures (ICCS). LNCS, vol 9717. Springer, pp 107–121

Ferré S, King RD (2004) BLID: an application of logical information systems to bioinformatics. In: Proceedings of the 2nd international conference on formal concept analysis (ICFCA), vol 2961. Springer, pp 47–54

Ferré S, Ridoux O (2000) A logical generalization of formal concept analysis. In: Proceedings of the 10th international conference on conceptual structures (ICCS). LNCS, vol 1867. Springer, pp 371–384

Ferré S, Ridoux O (2001) A framework for developing embeddable customized logics. Selected papers of the 1th international workshop on logic based program synthesis and transformation (LOPSTR). LNCS, vol 2372. Springer, pp 191–215

Ferré S, Ridoux O (2004) An introduction to logical information systems. Inf Process Manag 40(3):383–419

Ferré S, Allard P, Ridoux O (2012) Cubes of concepts: multi-dimensional exploration of multi-valued contexts. In: Proceedings of the 10th international conference on formal concept analysis (ICFCA), vol 7228. Springer, pp 112–127

Foret A, Ferré S (2010) On categorial grammars as logical information systems. In: Proceedings of the 8th international conference on formal concept analysis (ICFCA). LNCS, vol 5986. Springer, pp 225–240

Freeman LC, White DR (1993) Using Galois lattices to represent network data. Sociol Methodol 23:127–146

Ganter B (2010) Two basic algorithms in concept analysis. In: Proceedings of the 8th international conference on formal concept analysis (ICFCA). Lecture notes in computer science, vol 5986. Springer, pp 312–340

Ganter, B, Kuznetsov SO (2001) Pattern structures and their projections. In: International conference on conceptual structures (ICCS). LNCS 2120, pp 129–142

Ganter B, Obiedkov SA (2004) Implications in triadic formal contexts. In: Proceedings of the 12th international conference on conceptual structures (ICCS). Lecture notes in computer science, vol 3127. Springer, pp 186–195

Ganter B, Obiedkov SA (2016) Conceptual exploration. Springer, Berlin

Ganter B, Wille R (1999) Formal concept analysis. Springer, Berlin

Ganter B, Stumme G, Wille R (eds) (2005) Formal concept analysis, foundations and applications. LNCS, vol 3626. Springer

García-Pardo F, Cabrera IP, Cordero P, Ojeda-Aciego M (2013) On Galois connections and soft computing. In: Proceedings of the 12th international work-conference on artificial neural networks (IWANN). LNCS, vol 7903. Springer, pp 224–235

Gardiner EJ, Gillet VJ (2015) Perspectives on knowledge discovery algorithms recently introduced in chemoinformatics: rough set theory, association rule mining, emerging patterns, and formal concept analysis. J Chem Inf Model 55(9):1781–1803

Godin R, Mili H (1993) Building and maintaining analysis-level class hierarchies using Galois lattices. In: Proceedings of the eighth conference on object-oriented programming systems, languages, and applications (OOPSLA), pp 394–410

Godin R, Missaoui R, April A (1993) Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. Int J Man-Mach Stud 38(5):747–767

Grissa D, Comte B, Pujos-Guillot E, Napoli A (2016) A hybrid knowledge discovery approach for mining predictive biomarkers in metabolomic data. In: Proceedings of the European conference on machine learning and knowledge discovery in databases (ECML-PKDD). LNCS, vol 9851. Springer, pp 572–587

Guédi AO, Miralles A, Huchard M, Nebut C (2013) A practical application of relational concept analysis to class model factorization: lessons learned from a thematic information system. In: Proceedings of the tenth international conference on concept lattices and their applications (CLA), pp 9–20

Guigues J-L, Duquenne V (1986) Famile minimale d'implications informatives resultant d'un tableau de données binaire. Mathematique, Informatique et Sciences Humaines 95:5–18

Hahn G, Tardif C (1997) Graph homomorphisms: structure and symmetry. Graph symmetry. Springer, Berlin, pp 107–166

Hitzler P, Krötzsch M, Rudolph S (2009) Foundations of semantic web technologies. Chapman & Hall/CRC

Huchard M, Rouane-Hacene M, Roume C, Valtchev P (2007) Relational concept discovery in structured datasets. Ann Math Artif Intell 49(1–4):39–76
Ignatov DI, Gnatyshak DV, Kuznetsov SO, Mirkin BG (2015) Triadic formal concept analysis and triclustering: searching for optimal patterns. Mach Learn 101(1–3):271–302
Ignatov DI, Semenov A, Komissarova D, Gnatyshak DV (2017) Multimodal clustering for community detection. Formal Concept analysis of social networks. Springer, Berlin, pp 59–96
Jäschke R, Hotho A, Schmitz C, Ganter B, Stumme G (2006) TRIAS - an algorithm for mining iceberg tri-lattices. In: Proceedings of the 6th IEEE International conference on data mining (ICDM), pp 907–911
Ji L, Tan K-L, Tung AKH (2006) Mining frequent closed cubes in 3D datasets. In: Proceedings of the 32nd international conference on very large data bases (VLDB), pp 811–822. ACM
Kaytoue M, Assaghir Z, Napoli A, Kuznetsov SO (2010) Embedding tolerance relations in FCA: an application in information fusion. In: Proceedings of the 19th ACM conference on information and knowledge management (CIKM), pp 1689–1692. ACM
Kaytoue M, Kuznetsov SO, Napoli A (2011a) Biclustering numerical data in formal concept analysis. In: Proceedings of the 9th international conference on formal concept analysis (ICFCA). LNCS, vol 6628. Springer, pp 135–150
Kaytoue M, Kuznetsov SO, Napoli A (2011b) Revisiting numerical pattern mining with formal concept analysis. In: Proceedings of the 22nd international joint conference on artificial intelligence (IJCAI), pp 1342–1347. IJCAI/AAAI
Kaytoue M, Kuznetsov SO, Napoli A, Duplessis S (2011c) Mining gene expression data with pattern structures in formal concept analysis. Inf Sci 181(10):1989–2001
Kaytoue M, Kuznetsov SO, Macko J, Napoli A (2014) Biclustering meets triadic concept analysis. Ann Math Artif Intell 70(1–2):55–79
Kaytoue M, Plantevit M, Zimmermann A, Bendimerad AA, Robardet C (2017) Exceptional contextual subgraph mining. Mach Learn 106(8):1171–1211
Kirchberg M, Leonardi E, Tan YS, Link S, Ko RKL, Lee B (2012) Formal concept discovery in semantic web data. In: Proceedings of the 10th international conference on formal concept analysis (ICFCA). LNCS, vol 7278. Springer, pp 164–179
Kötters J (2013) Concept lattices of a relational structure. In: Proceedings of the 20th international conference on conceptual structures (ICCS). LNCS, vol 7735. Springer, pp 301–310
Kourie DG, Obiedkov SA, Watson BW, van der Merwe D (2009) An incremental algorithm to construct a lattice of set intersections. Sci Comput Program 74(3):128–142
Kriegel F (2017) Acquisition of terminological knowledge from social networks in description logic. Formal concept analysis of social networks. Springer, Berlin, pp 97–142
Kuznetsov SO (1999) Learning of simple conceptual graphs from positive and negative examples. In: Proceedings of the third european conference on principles of data mining and knowledge discovery (PKDD). LNCS, vol 1704. Springer, pp 384–391
Kuznetsov SO (2001) Machine learning on the basis of formal concept analysis. Autom Remote Control 62(10):1543–1564
Kuznetsov SO (2004) On the intractability of computing the Duquenne-Guigues base. J Univers Comput Sci 10(8):927–933
Kuznetsov SO (2007) On stability of a formal concept. Ann Math Artif Intell 49(1–4):101–115
Kuznetsov SO (2009) Pattern structures for analyzing complex data. In: Proceedings of the 12th international conference on rough sets, fuzzy sets, data mining and granular computing (RSFD-GrC), vol 5908. Springer, pp 33–44
Kuznetsov SO (2013) Fitting pattern structures to knowledge discovery in big data. In: Proceedings of the 11th international conference on formal concept analysis (ICFCA), vol 7880. Springer, pp 254–266
Kuznetsov S, Obiedkov S (2002) Comparing performance of algorithms for generating concept lattices. J Exp Theor Artif Intell 14(2/3):189–216
Kuznetsov SO, Makhalova TP (2018) On interestingness measures of formal concepts. Inf Sci 442–443:202–219

Kuznetsov SO, Poelmans J (2013) Knowledge representation and processing with formal concept analysis. Data Min Knowl Discov (Wiley) 3(3):200–215

Kuznetsov SO, Samokhin MV (2005) Learning closed sets of labeled graphs for chemical applications. In: Proceedings of 15th international conference on inductive logic programming (ILP). LNCS, vol 3625. Springer, pp 190–208

Kuznetsov SO, Obiedkov SA, Roth C (2007) Reducing the representation complexity of lattice-based taxonomies. In: Proceedings of the 15th international conference on conceptual structures (ICCS). LNCS, vol 4604. Springer, pp 241–254

Leeuwenberg A, Buzmakov A, Toussaint Y, Napoli A (2015) Exploring pattern structures of syntactic trees for relation extraction, vol 9113. Springer, pp 153–168

Lehmann F, Wille R (1995) A Triadic approach to formal concept analysis, vol 954. Springer, Berlin, pp 32–43

Lindig C (1995) Concept-based component retrieval. In: IJCAI-95 workshop: formal approaches to the reuse of plans, proofs, and programs, pp 21–25

Liquiere M, Sallantin J (1998) Structural machine learning with Galois lattice and graphs. In: Proceedings of the fifteenth international conference on machine learning (ICML), pp 305–313

Métivier J, Lepailleur A, Buzmakov A, Poezevara G, Crémilleux B, Kuznetsov SO, Goff JL, Napoli A, Bureau R, Cuissart B (2015) Discovering Structural alerts for mutagenicity using stable emerging molecular patterns. J Chem Inf Model 55(5):925–940

Mimouni N, Nazarenko A, Salotti S (2015) A conceptual approach for relational IR: application to legal collections. In: Proceedings of the 13th international conference on formal concept analysis (ICFCA), pp 303–318

Missaoui R, Kuznetsov SO, Obiedkov SA (eds) (2017) Formal concept analysis of social networks. Springer, Berlin

Muggleton S, De Raedt L (1994) Inductive logic programming: theory and methods. J Log Program 19–20:629–679

Nica C, Braud A, Le Ber F (2017) Hierarchies of weighted closed partially-ordered patterns for enhancing sequential data analysis. In: Proceedings of the 14th international conference on formal concept analysis (ICFCA), pp 138–154

Obiedkov SA, Duquenne V (2007) Attribute-incremental construction of the canonical implication basis. Ann Math Artif Intell 49(1–4):77–99

Obiedkov SA, Kourie DG, Eloff JHP (2009) Building access control models with attribute exploration. Comput Secur 28(1–2):2–7

Outrata J, Vychodil V (2012) Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data. Inf Sci 185(1):114–127

Padioleau Y, Ridoux O (2003) A logic file system. In: USENIX annual technical conference, pp 99–112

Poelmans J, Ignatov DI, Kuznetsov SO, Dedene G (2013) Formal concept analysis in knowledge processing: a survey on applications. Expert Syst Appl 40(16):6538–6560

Priss U (2011) Unix systems monitoring with FCA. In: Proceedings of the 19th international conference on conceptual structures (ICCS). LNCS, vol 6828. Springer, pp 243–256

Priss U (2013) Representing median networks with concept lattices. In: Proceedings of the 20th international conference on conceptual structures (ICCS). LNCS, vol 7735. Springer, pp 311–321

Quintero NY, Restrepo G (2017) Formal concept analysis applications in chemistry: from radionuclides and molecular structure to toxicity and diagnosis. Partial order concepts in applied sciences. Springer, pp 207–217

Roth C (2017) Knowledge communities and socio-cognitive taxonomies. Formal concept analysis of social networks. Springer, Berlin, pp 1–18

Rouane-Hacene M, Huchard M, Napoli A, Valtchev P (2007). A proposal for combining formal concept analysis and description logics for mining relational data. In: Proceedings of the 5th international conference on formal concept analysis (ICFCA 2007). LNAI, vol 4390. Springer, pp 51–65

Rouane-Hacene M, Huchard M, Napoli A, Valtchev P (2010) Using formal concept analysis for discovering knowledge patterns. In: Proceedings of the 7th international conference on concept lattices and their applications (ICFCA), pp 223–234

Rouane-Hacene M, Huchard M, Napoli A, Valtchev P (2013) Relational concept analysis: mining concept lattices from multi-relational data. Ann Math Artif Intell 67(1):81–108

Rudolph S (2004) Exploring relational structures via FLE. In: Proceedings of the 12th international conference on conceptual structures (ICCS), LNCS, vol 3127. Springer, pp 196–212

Rudolph S, Sacarea C, Troanca D (2015) Towards a navigation paradigm for triadic concepts. In: Proceedings of the 13th international conference on formal concept analysis (ICFCA), LNCS, vol 9113. Springer, pp 252–267

Saada H, Huchard M, Liquiere M, Nebut C (2014) Learning model transformation patterns using graph generalization. In: Proceedings of the eleventh international conference on concept lattices and their applications (CLA), pp 11–22

Sahraoui HA, Melo WL, Lounis H, Dumont F (1997) Applying concept formation methods to object identification in procedural code. In: International conference on automated software engineering (ASE), pp 210–218

Sertkaya B (2009) OntoComP: a protégé plugin for completing OWL ontologies. In: Proceedings of the 6th European semantic web conference (ESWC). LNCS, vol 5554. Springer, pp 898–902

Sertkaya B (2010) A survey on how description logic ontologies benefit from FCA. In: Proceedings of the 7th international conference on concept lattices and their applications (CLA), CEUR workshop proceedings, vol 672, pp 2–21

Soldano H, Santini G, Bouthinon D (2017) Formal concept analysis of attributed networks. Formal concept analysis of social networks. Springer, Berlin, pp 143–170

Sowa J (1984) Conceptual structures. Information processing in man and machine, Addison-Wesley, Reading, US

Staab S, Studer R (eds) (2009) Handbook on ontologies, 2nd edn. Springer, Berlin

Stahl T, Voelter M, Czarnecki K (2006) Model-driven software development: technology, engineering, management. Wiley, New York

Stumme G, Taouil R, Bastide Y, Pasquier N, Lakhal L (2002) Computing iceberg concept lattices with Titanic. Data Knowl Eng 42(2):189–222

Szathmary L, Valtchev P, Napoli A, Godin R, Boc A, Makarenkov V (2014) A fast compound algorithm for mining generators, closed itemsets, and computing links between equivalence classes. Ann Math Artif Intell 70(1–2):81–105

Tatti N, Moerchen F, Calders T (2014) Finding robust itemsets under subsampling. ACM Trans Database Syst 39(3):20:1–20:27

Tilley T, Cole R, Becker P, Eklund PW (2005) A survey of formal concept analysis support for software engineering activities. Formal concept analysis, foundations and applications. LNCS 3626:250–271

Uno T, Asai T, Uchida Y, Arimura H (2004) An efficient algorithm for enumerating closed patterns in transaction databases. In: Proceedings of the 7th international conference on discovery science (DS). LNCS, vol 3245. Springer, pp 16–31

Valverde-Albacete FJ, Peláez-Moreno C (2017) A formal concept analysis look at the analysis of affiliation networks. Formal concept analysis of social networks. Springer, Berlin, pp 171–195

van Deursen A, Kuipers T (1999) Identifying objects using cluster and concept analysis. In: Proceedings of the international conference on software engineering (ICSE), pp 246–255

Völker J, Rudolph S (2008) Lexico-logical acquisition of OWL DL axioms. In: Proceedings of the 6th international conference on formal concept analysis (ICFCA), LNCS, vol 4933. Springer, pp 62–77

Voutsadakis G (2002) Polyadic concept analysis. Order 19(3):295–304

Wille R (2002) Why can concept lattices support knowledge discovery in databases? J Exp Theor Artif Intell 14(2/3):81–92

Wray T, Eklund PW (2011) Exploring the information space of cultural collections using formal concept analysis. In: Proceedings of the 9th international conference on formal concept analysis (ICFCA), LNCS, vol 6628. Springer, pp 251–266

Zaki MJ (2005) Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans Knowl Data Eng 17(4):462–478