



Generalization effect of quantifiers in a classification based on relational concept analysis

Agnès Braud, Xavier Dolques, Marianne Huchard, Florence Le Ber

► To cite this version:

Agnès Braud, Xavier Dolques, Marianne Huchard, Florence Le Ber. Generalization effect of quantifiers in a classification based on relational concept analysis. Knowledge-Based Systems, 2018, 160, pp.119-135. 10.1016/j.knosys.2018.06.011 . hal-01857724

HAL Id: hal-01857724

<https://hal.science/hal-01857724>

Submitted on 21 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Generalization effect of quantifiers in a classification based on Relational Concept Analysis

Agnès Braud^{a,b,*}, Xavier Dolques^{a,b,c},
Marianne Huchard^d, Florence Le Ber^{a,b,c}

^a*Université de Strasbourg, ICube UMR 7357, F-67412 Illkirch-Graffenstaden, France*

^b*CNRS, ICube UMR 7357, F-67412 Illkirch-Graffenstaden, France*

^c*ENGES, ICube UMR 7357, F-67000 Strasbourg, France*

^d*LIRMM, Université de Montpellier, CNRS, F-34392 Montpellier cedex 5, France*

Abstract

Relational Concept Analysis (RCA) has been designed to classify sets of objects described by attributes and relations between these objects. This is achieved by iterating on Formal Concept Analysis (FCA). It can be used to discover knowledge patterns and implication rules in multi-relational datasets. The classification output by RCA is a family of lattices whose graphical representation facilitates the analysis by an expert. However, RCA comes with specific complexity issues. It iterates on the building of interconnected concept lattices, so that each concept in a lattice might be the cause of generating other concepts in other lattices. In complex analyses, it relies on the successive choice of scaling operators which affects the size and the understandability of the results. These operators are based on a set of quantifiers which are studied in this paper: we indeed focus on the comparison of scaling quantifiers and highlight a generality relation between them. Our theoretical proposition is complemented by an experimental evaluation of the exploration space size, based on a real dataset upon watercourses. This work is intended for data analysts, to provide them with an overview on the different strategies offered by RCA.

Keywords: Relational Data Exploration, Relational Concept Analysis,

*Corresponding author

Email addresses: agnes.braud@unistra.fr (Agnès Braud),
xavier.dolques@laposte.net (Xavier Dolques), marianne.huchard@lirmm.fr (
Marianne Huchard), florence.leber@engees.unistra.fr (Florence Le Ber)

1. Introduction

Many data are inherently relational, and several approaches have been implemented to explore such data [1]. Among them, Relational Concept Analysis (RCA) is a descriptive approach based on Formal Concept Analysis (FCA) that both builds a classification (a lattice of formal concepts) for each category of objects contained in a dataset, and allows to obtain implication rules including relations between objects [2, 3]. RCA has been applied to multi-relational datasets from various domains, e.g. for analyzing the quality of watercourses [4], for the fuzzy semantic annotation of web resources [5], or for the analysis and reengineering of software models [6] or semantic wikis [7].

RCA, as FCA, comes with a major challenge, linked to the fact that dealing with complex (relational, but also fuzzy, or multi-polar) data produces huge and complex results. Many methods have been proposed to reduce the lattice size, either by reducing the original data, e.g. by granular reduction [8] or by projection [9], or by reducing the number of concepts to be built, e.g., by thresholding [10], or by using AOC-posets [11] -always trying to keep the principal results. Another way is to help the user navigate within the results and select interesting concepts, based on various measures [12, 13]. In RCA, the user is given the chance to orientate the search while it is running, so that a specific problem is to help him/her making the choice of the direction search. This is the question we focus on, which has received few attention until now.

We are indeed interested in providing experts with tools that can help deepening the understanding of the data and the analysis. Our experience has shown that RCA can be useful for knowledge extraction, and more precisely relational data exploration [14], even by a non computer scientist domain expert with little training. This comes from several features of RCA:

- lattices, which is the output of RCA, have a graphical representation as a Hasse diagram (see for example Fig. 2). Firstly, this graphical represen-

tation facilitates the understanding of the relations between categories of objects. Secondly, this representation organizes relational concepts contained in the dataset in a way that helps the navigation between them when performing the analysis;

- the implication rules highlight existing correlations;
- the process is iterative, providing temporary interpretable results, thus allowing to tune the analysis during the process in case the expert realizes that it is interesting to orientate the analysis in one direction or another.

RCA applies on a set of object-attribute relations that describe objects of a given type by a set of attributes, and object-object relations that represent the links between two categories of objects (possibly the same category). The main idea is to transform object-object relations into relational attributes thanks to so called scaling quantifiers. Scaling quantifiers are actually used to quantify a relation between a given object and a subset of objects (a concept extent) it is linked to through the given relation. The generated relational attributes can then be added to the initial object-attribute relation. The combination of a scaling quantifier, a relation, and a set of concepts is called a scaling operator and it is applied to a given object-attribute relation in order to extend it with new information.

The RCA process thus allows the user to make different choices at each step of the iterative process in order to focus the exploration of the data:

- the object-attribute relations or object-object relations to add or remove from the analysis;
- the object-object relations to transform so as to create new relational attributes;
- the corresponding scaling quantifiers (the scaling quantifier assigned to an object-object relation can be changed and several scaling quantifiers can even be assigned at one step to an object-object relation).

On the one hand, these choices enrich the process, allowing to perform a wider range of analyses. On the other hand, it may be difficult to decide which choice to make, and moreover results including relational information are more abundant and may be harder to understand for data analysts so that they may be lost in results, especially when the scaling quantifiers vary from one step to another. In a previous work, we have proposed an adaptation of RCA to explore relations in a guided way, by defining exploratory paths, allowing to increase the performance and the pertinence of the results [14].

In this paper, our purpose is to further improve the data exploration with RCA. We focus on theoretical aspects deepening the understanding of the RCA process. These results can be used to help the data analyst to understand the current outputs and to make choices for the next step, thus guiding her/him in its analysis. This research is also a theoretical basis for improving the navigation and visualization of RCA results (some tuning facilities are already integrated in the RCAexplore tool¹). More precisely, the aim of this paper is to compare the scaling quantifiers used in RCA, focusing on a generality relation between them. Variants of the scaling quantifiers (existential, universal and contains) are considered and their effects on RCA results are studied. The generality relation on scaling quantifiers is formalized and proved for the different quantifiers. The resulting hierarchy of quantifiers can then be used as a guide to help an analyst forecast the kind of results she/he will obtain. An illustrative example, and some numerical results are given on a dataset from the Fresqueau² project.

The paper is organized as follows. Section 2 motivates the usage of formal concepts provided with their specialization partial order, gives the background of this work and the principles of RCA-based data exploration. Section 3 presents the core of our proposition. Section 4 describes experimental results on a real dataset upon watercourses. Related work is discussed in Section 5 and Section 6 concludes the paper with some perspectives.

¹<http://dolques.free.fr/rcaexplore/>

²<http://engees-fresqueau.unistra.fr/presentation.php?lang=en>

2. Background

In this section, we motivate the use of formal concepts in knowledge discovery (Section 2.1), then we introduce the basics for Formal Concept Analysis (Section 2.2) and for Relational Concept Analysis (Section 2.3).

2.1. Formal concepts in knowledge discovery

In her paper in the 40th anniversary volume of Annual Review of Information Science and Technology [15], Uta Priss nicely introduces formal concepts as describing *"a natural feature of information representation which is as fundamental to hierarchies and object/attribute structures as set theory or relational algebra are for relational databases"*. She explains that the notions of concept “extension” and concept “intension” are adapted from philosophy, and that *"the basic FCA structures have been rediscovered over and over by different researchers and in different settings."*, fostering the idea of its inherent universality. Roots of formal concepts and concept lattices were introduced in 1940 by G. Birkhoff [16], and developed by M. Barbut and B. Monjardet [17] to cite just a few, and the FCA terminology and conceptual approach have been fully developed since 1982 following R. Wille [18] vision.

Formal concepts help in forming object/attribute groups and building hierarchical structures on these groups, with strong mathematical properties, such as the duality of objects versus attributes or the canonicity of the building: for a given object set description, a unique concept lattice is built, contrarily to most approaches in conceptual clustering that are approximate. The hierarchical structure and the groups highlight many properties such as frequent patterns, attribute implications, mutual exclusion within attribute groups, or object similarities. Querying through concept lattices helps organize groups (concepts) of similar answers that can be navigated through the specialization structure [19]. They also help in reformulating queries when the answer set is empty. All these qualities explain the spread of FCA in domains such as information retrieval [20], recommendation systems [21], or machine learning [22, 23]. Furthermore, FCA emphasizes the visualization of concepts and their partial ordering which

is meaningful for knowledge discovery and data exploration. When lattices are too complex, they are not fully displayed to the user. To give a few examples, in Credo [24], the lattices are displayed through a kind of explorer akin to file explorers, and the hierarchy can be expanded; in Carpineto and Romano [25] fish-eye displays help to focus on specific parts; in [26], smart tag clouds allow the expert to navigate into the lattice.

FCA can be applied to various forms of data, going from the simplest one (objects described by binary attributes) to more complex ones, including numerical data [27], labeled graphs [28], interval data [29], hierarchical attributes [24], composite attributes [30], fuzzy data [31], multi-dimensional data [32, 33] or pattern structures [34]. Multi-relational data are taken into account within the RCA framework, which extends the purpose of FCA [2, 3].

The Uta Priss’s web page³ gathers a number of FCA softwares, some of them being available online. There is a rich literature about algorithms as [35], as well as analysis tools such as specific metrics dedicated to formal concepts relevance evaluation [36]. Thanks to its properties and available softwares, FCA has been used for applications in a wide range of domains, such as Software engineering [37], Linguistics [38], chemical graphs [28], water datasets [30], candidate CV exploration in recruitment [39], Organised Crime monitoring [40] or symbol recognition [41].

2.2. Concept Lattices

Formal Concept Analysis [27] is based on lattice theory and can be used in particular to discover knowledge in data.

This section begins by explaining the simplest form, which extracts an ordered set of concepts from a binary dataset, called an *object-attribute context* (or *formal context*), composed of objects described by attributes. An object-attribute context \mathcal{K} is a 3-tuple (G, M, I) , where G is an object set, M an attribute set, and $I \subseteq G \times M$ an object-attribute relation. Figure 1 (left-hand

³<http://www.upriss.org.uk/fca/fcasoftware.html>

side) shows an object-attribute context $\mathcal{K}_{Ingredients} = (G_{Ingredients}, M_{Ingredients}, I_{Ingredients})$ which describes ingredients by their origin (e.g. vegetable, meat, etc.) or preparation form (e.g. sauce).

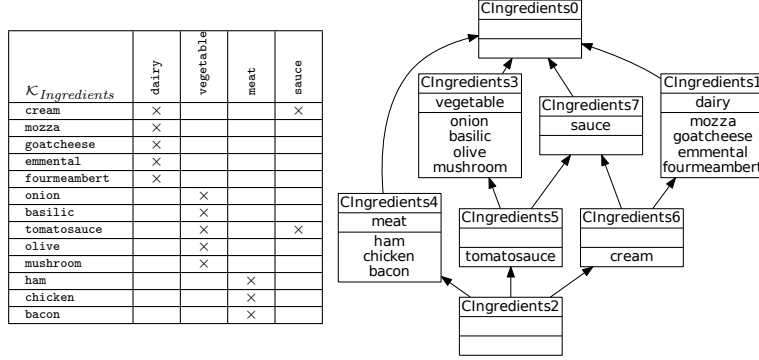


Figure 1: Object-attribute context $\mathcal{K}_{Ingredients}$, lattice $\mathcal{L}^0_{Ingredients}$

Given a $\mathcal{K} = (G, M, I)$ object-attribute context, let $X \subseteq G$, X' is the set of all attributes of M that all objects of X share. Let $Y \subseteq M$, Y' is the set of all objects of G that have all attributes of Y . A *formal concept* associates a maximal set of objects with the maximal set of attributes they share. A formal concept is thus a pair $C = (X, Y)$, where $Y = X'$ and $X = Y'$. $X = Extent(C) = \{g \in G | \forall m \in Intent(C), (g, m) \in I\}$ is called the extent of the concept (objects covered by the concept). $Y = Intent(C) = \{m \in M | \forall g \in Extent(C), (g, m) \in I\}$ is the intent of the concept (shared attributes). With these notations, $Extent(C) = Intent(C)'$ and $Extent(C)' = Intent(C)$.

Let $\mathcal{C}_{\mathcal{K}}$ be the set of all concepts built from \mathcal{K} . Given two formal concepts $C_1 = (E_1, I_1)$ and $C_2 = (E_2, I_2)$ of $\mathcal{C}_{\mathcal{K}}$, the concept generalization order \preceq_C is here defined by $C_1 \preceq_C C_2$ if and only if $E_2 \subseteq E_1$ (and equivalently $I_1 \subseteq I_2$)⁴. $C_1 \preceq_C C_2$ means that C_1 is more general than C_2 , or equivalently C_1 is a superconcept of C_2 , or C_2 is a subconcept of C_1 . The set $\mathcal{C}_{\mathcal{K}}$, provided with the generalization order is the concept lattice $\mathcal{L}_{\mathcal{K}} = (\mathcal{C}_{\mathcal{K}}, \preceq_C)$ associated with \mathcal{K} .

⁴We borrow the \preceq_C notation for subsumption from Inductive Logic Programming.

Besides, the fact that $C_1 \preceq_C C_2$ is represented with C_1 above C_2 in the lattice diagrams.

Figure 1 (right-hand side) shows the concept lattice $\mathcal{L}_{Ingredients}^0$ associated with context $\mathcal{K}_{Ingredients}$ (the 0 exponent will be explained later in the paper). Let us notice that to simplify the notation, we will write $\mathcal{L}_{Ingredients}^0$ rather than $\mathcal{L}_{\mathcal{K}_{Ingredients}}^0$. As every attribute is top-down inherited, and every object is bottom-up inherited, the representation is usually simplified. It shows an attribute (*resp.* an object) only in the highest (*resp.* lowest) concept where it appears (this concept is the introducing concept). In Fig. 1, the **vegetable** attribute is introduced in **CIngredients3** and is inherited by its subconcepts: **CIngredients5** and **CIngredients2**. Furthermore the **cream** object is introduced in **CIngredients6** concept, and inherited by **CIngredients6** superconcepts: **CIngredients7**, **CIngredients1** and also **CIngredients0**.

2.3. Principles of Relational Concept Analysis (RCA)

Relational Concept Analysis [2] has been introduced to extend the scope of FCA to multi-relational data encoded in a Relational Context Family (RCF). An RCF is composed of several categories of objects and relations between these objects, respectively described within object-attribute contexts and object-object contexts (also called relational contexts).

Definition 1 (Relational Context Family (RCF)). A Relational Context Family is a (\mathbf{K}, \mathbf{R}) pair where: $\mathbf{K} = \{\mathcal{K}_i\}_{i=1,\dots,n}$ is a set of $\mathcal{K}_i = (G_i, M_i, I_i)$ object-attribute contexts. $\mathbf{R} = \{r_j\}_{j=1,\dots,p}$ is a set of r_j relations where $r_j \subseteq G_k \times G_l$ for some $k, l \in \{1, \dots, n\}$. In the following, for a given relation r_j , its domain is denoted by $dom(r_j)$ and its range is denoted by $ran(r_j)$.

To illustrate RCA, we use the RCF presented in Tab. 1, composed of: three object-attribute contexts, \mathcal{K}_{People} , \mathcal{K}_{Pizzas} , and $\mathcal{K}_{Ingredients}$, corresponding to three categories of objects; two object-object contexts, r_{likes} and $r_{hastopping}$, that represent in extension the corresponding object-object relations and that show respectively which pizzas people like, and which ingredients are on top of pizzas. Let us note that \mathcal{K}_{People} and \mathcal{K}_{Pizzas} do not have any specific attribute, however for \mathcal{K}_{Pizzas} identity attributes have been created leading to a diagonal

Table 1: RCF composed of people, pizza and ingredient categories, with object-object relations **likes** (or r_{likes}) and **hastopping** (or $r_{hastopping}$)

\mathcal{K}_{People}											
Arthur	John	Alice	Juliet	Nancy	Ellen						
\mathcal{K}_{Pizzas}		auvergne	margherita	threecheese	fourcheese	futrader	vege	simple			
		×									
			×								
				×							
					×						
						×					
							×				
								×			

$\mathcal{K}_{Ingredients}$		dairy	vegetable	meat	sauce
cream		×			×
mozza		×			
goatcheese		×			
emmental		×			
fourmeambert		×			
onion			×		
basilic			×		
tomatoesauce			×		×
olive			×		
mushroom			×		
ham				×	
chicken				×	
bacon				×	

\mathcal{K}_{Likes}		auvergne	margherita	threecheese	fourcheese	futrader	vege	simple
Arthur		×	×	×	×	×		
John					×	×		
Alice			×				×	×
Juliet			×		×	×		×
Nancy							×	×
Ellen				×	×			

$\mathcal{K}_{hasopping}$		cream	mozza	goatcheese	emmental	fourmeambert	onion	basilic	tomatoesauce	olive	mushroom	ham	chicken	bacon
auvergne		×	×			×						×		
margherita			×					×	×	×				
threecheese		×	×	×	×									
fourcheese		×	×	×	×	×								
futrader		×	×	×	×				×			×	×	×
vege							×	×			×			
simple									×					

table. This is because \mathcal{K}_{Pizzas} is the range of a relation, and we will explain later in this section the benefits of such a modeling for the RCA process.

To deal with relational data, RCA works by enriching object-attribute contexts with new attributes, called *relational attributes*. For each \mathcal{K} object-attribute context, the idea is to capture in these attributes an additional information coming from the relations that have \mathcal{K} as domain. For example, one could add in \mathcal{K}_{People} attributes about the kind of pizzas liked thanks to r_{likes} object-object relation that has $dom(r_{likes}) = \mathcal{K}_{People}$.

More precisely, RCA transforms an object-object context r , that has \mathcal{K} as domain, into new relational attributes in \mathcal{K} quantifying the relation between objects of r and concepts formed on objects of $ran(r)$. The type of quantification captured depends on the choice of *scaling quantifier*.

Definition 2 (Relational attribute). A relational attribute is an expression $qr(\mathbf{C})$, where $q \in \mathbf{Q}$ is a scaling quantifier, r a specific relation, and \mathbf{C} a specific concept whose extent contains objects from $\text{ran}(r)$.

$\mathbf{Q} \supseteq \{\exists, \exists\forall, \exists\supseteq\}$ and other quantifiers will be introduced later (Section 2.5). The objective is to go beyond a simple description by attributes like “has-topping: mozza”. For example, RCA allows to identify the group of dairy ingredients, then the group of pizzas that have at least one dairy ingredient, and finally, the group of people that like pizzas that have at least one dairy ingredient by composition of several object-object relations. In this example, “at least” is a kind of scaling quantifier (\exists quantifier). Other quantifiers would give attributes like “all ingredients are dairy ingredients” (“for all” quantifier: $\exists\forall$) or “contains all the dairy ingredients” (“contains all” quantifier: $\exists\supseteq$). Let us note that in a more general version of RCA, several quantifiers could be applied to r and $\rho(r)$ would denote a set of quantifiers associated with r .

Table 2 presents the result of the application of the existential scaling quantifier to relation $r_{\text{hastopping}}$, using the concepts of $\mathcal{L}_{\text{Ingredients}}^0$ shown in Fig. 1. First columns correspond to the initial object-attribute context on pizzas, namely $\mathcal{K}_{\text{Pizzas}}$. The attributes of the last 8 columns are the obtained relational attributes, which encode relations between pizzas and groups of ingredients. Relational attribute $\exists r_{\text{hastopping}}(\mathbf{C}\text{Ingredients1})$ is thus assigned to **threecheese** pizza because this pizza has at least one of the (dairy) ingredients grouped in concept **CIngredients1**. Figure 2 shows the corresponding lattice $\mathcal{L}_{\text{Pizzas}}^1$. The pizzas classification initially showed only trivial (singleton) groups, each one corresponding to a pizza, because $\mathcal{K}_{\text{Pizzas}}$ is a diagonal table. Thanks to the added relational attributes, the lattice now contains new relevant groups such as **CPizzas9** which groups pizzas with at least one dairy ingredient (as we said, these ingredients are grouped in **CIngredients1**). Let us note that thanks to the diagonal of $\mathcal{K}_{\text{Pizzas}}$, single pizzas can be used in relational attributes based on relation $r_{\text{hastopping}}$ through singleton concepts, so that relational attributes can bring information related to a single pizza.

2.4. Extending an object-attribute context

Adding relational attributes to an object-attribute context is done through a scaling operation, based on a scaling quantifier. In order to provide a generic definition of this operation, a generic function κ is introduced which maps a scaling quantifier q , a relation r and an object subset in the range of r to an object subset from the domain of r .

$$\kappa : \mathbf{Q} \times \mathbf{R} \times \bigcup_{i=1,\dots,n} 2^{G_i} \rightarrow \bigcup_{i=1,\dots,n} 2^{G_i}$$

κ returns the group of domain objects that are connected for r and q with a certain group of range objects. For example, $\kappa(\exists, r_{\text{hastopping}}, \{\text{mozza}, \text{goatcheese}, \text{emmental}, \text{fourmeambert}\}) = \{\text{auvergne}, \text{margherita}, \text{threecheese}, \text{fourcheese}, \text{furtrader}\}$. The reader can check that every pizza of the image $\{\text{auvergne}, \text{margherita}, \text{threecheese}, \text{fourcheese}, \text{furtrader}\}$ has at least one of the toppings of $\{\text{mozza}, \text{goatcheese}, \text{emmental}, \text{fourmeambert}\}$. More specifically, for r , q and a concept C , $\kappa(r, q, C)$ is the set of objects that own the relational attribute $qr(C)$. In our example, C is **CIngredients1**.

Table 2: Relational extension of \mathcal{K}_{Pizzas} : \mathcal{K}_{Pizzas} is concatenated with its derived context $\mathbb{S}_{(r_{\text{hastopping}}, \exists), \text{CIngredients}}(\mathcal{K}_{Pizzas})$, based on the existential scaling of $r_{\text{hastopping}}$

\mathcal{K}_{Pizzas}	auvergneId	margheritaId	threecheeseId	fourcheeseId	furtraderId	vegeId	simpleId	\exists hastopping(CIngredients0)	\exists hastopping(CIngredients1)	\exists hastopping(CIngredients2)	\exists hastopping(CIngredients3)	\exists hastopping(CIngredients4)	\exists hastopping(CIngredients5)	\exists hastopping(CIngredients6)	\exists hastopping(CIngredients7)
auvergne	x							x	x		x			x	x
margherita		x						x	x		x			x	x
threecheese			x					x	x					x	x
fourcheese				x				x	x					x	x
furtrader					x			x	x		x	x		x	x
vege						x		x			x			x	x
simple							x	x			x			x	x

The generic notion of scaling operator, an operator that yields a relational extension of a context $\mathcal{K}_k = (G_k, M_k, I_k)$ along a relation r , a scaling quantifier q and a concept set \mathcal{C}_l , can now be defined as follows.

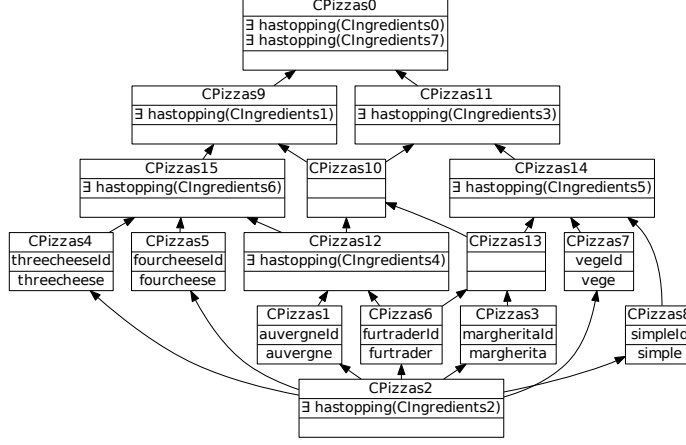


Figure 2: Lattice of pizzas at step 1 \mathcal{L}_{Pizzas}^1 , built from Tab. 2

Definition 3 (Scaling operator and partial relational extension). Let us consider $\mathcal{K}_k = (G_k, M_k, I_k)$ and $\mathcal{K}_l = (G_l, M_l, I_l)$ two object-attribute contexts, $r \subseteq G_k \times G_l$ a relation, \mathcal{C}_l the set of concepts on \mathcal{K}_l , and q a scaling quantifier. The scaling operator $\mathbb{S}_{(r,q),\mathcal{C}_l}$ over \mathcal{K}_k yields the partial relational extension $\mathbb{S}_{(r,q),\mathcal{C}_l}(\mathcal{K}_k) = (G^+, M^+, I^+)$, where the elements are as follows: $G^+ = G_k$, $M^+ = \{q \ r(c) \mid c \in \mathcal{C}_l\}$, and $I^+ = \bigcup_{c \in \mathcal{C}_l} \kappa(q, r, \text{Extent}(c)) \times \{q \ r(c)\}$.

The *relational extension* of an object-attribute context \mathcal{K}_k is obtained by extending it with all the partial relational extensions generated from the relations that have \mathcal{K}_k as a domain and their assigned scaling operator. Thus, let us consider an RCF that contains only the above-mentioned context \mathcal{K}_k , the context \mathcal{K}_l and relation r with $\text{dom}(r) = \mathcal{K}_k$ and $\text{ran}(r) = \mathcal{K}_l$. The relational extension of \mathcal{K}_k when assigning quantifier q to r is the union of the initial and derived contexts $\mathcal{K}_k \cup \mathbb{S}_{(r,q),\mathcal{C}_l}(\mathcal{K}_k) = (G_k, M_k \cup M^+, I_k \cup I^+)$.

Table 2 shows the relational extension of context \mathcal{K}_{Pizzas} , when considering $\rho(r_{\text{hastopping}}) = \exists$ and the concept set of $\mathcal{L}_{Ingredients}^0$ lattice of Fig. 1. If an additional relation connecting pizzas to another kind of objects, for example, r_{servedin} , connecting pizzas to restaurants, was in the dataset, then the relational extension of \mathcal{K}_{Pizzas} would also include the scaling upon r_{servedin} .

The general case is defined as follows:

Definition 4 (Relational extension of a context $\mathcal{K}_k = (G_k, M_k, I_k)$). Under Def. 3, let $R_k = \{r_j, 1 \leq j \leq p_k | \text{dom}(r_j) = G_k\}$ be the set of relations with domain G_k . Let ρ be a mapping from \mathbf{R} to \mathbf{Q} which associates a scaling quantifier to each object-object relation r_j . Let \mathbf{L} be a set of concept lattices containing, for every $r_j \in R_k$, a lattice $\mathcal{L}_{\text{ran}(r_j)}$ where the concepts group objects of the range of r_j . The relational extension of context \mathcal{K}_k consists in apposing to \mathcal{K}_k the respective scaling upon each $r_j \in R_k$:

$$\mathbb{E}_{\rho, \mathbf{L}}(\mathcal{K}_k) = \mathcal{K}_k \mid \mathbb{S}_{(r_1, \rho(r_1)), \mathcal{C}_{\text{ran}(r_1)}}(\mathcal{K}_k) \mid \dots \mid \mathbb{S}_{(r_{p_k}, \rho(r_{p_k})), \mathcal{C}_{\text{ran}(r_{p_k})}}(\mathcal{K}_k)$$

The relational extension of the whole \mathbf{K} is composed of all the relational extensions of all \mathcal{K}_i in \mathbf{K} .

Definition 5 (Relational extension of an RCF). Under the previous definitions, the relational extension of \mathbf{K} is:

$$\mathbb{E}_{\rho, \mathbf{L}}^*(\mathbf{K}) = \{\mathbb{E}_{\rho, \mathbf{L}}(\mathcal{K}_1), \dots, \mathbb{E}_{\rho, \mathbf{L}}(\mathcal{K}_n)\}$$

With the lattices built at the initial step (step 0), and with the mapping $\rho = \{(r_{\text{likes}}, \exists\forall), (r_{\text{hasstopping}}, \exists)\}$, the relational extension of \mathbf{K} would be composed of the relational extensions of $\mathcal{K}_{\text{People}}$, $\mathcal{K}_{\text{Pizzas}}$ and $\mathcal{K}_{\text{Ingredients}}$. The relational extension of $\mathcal{K}_{\text{People}}$ is presented in Fig. 3 (bottom-left-hand side), with the corresponding lattice $\mathcal{L}_{\text{People}}^1$ on the right-hand side (exponent 1 stands for step 1). The relational extension of $\mathcal{K}_{\text{Pizzas}}$ has been shown in Tab. 2. The relational extension of $\mathcal{K}_{\text{Ingredients}}$ is simply $\mathcal{K}_{\text{Ingredients}}$, because there is no outgoing relation. At this step, there is no pizza groups; since everybody likes more than one pizza, the relational extension of $\mathcal{K}_{\text{People}}$ obtained after the application of scaling operator $\mathbb{S}_{(r_{\text{likes}}, \exists\forall), \mathcal{C}_{\text{Pizzas}}^0}$ ($\mathcal{C}_{\text{Pizzas}}^0$ being the set of concepts corresponding to $\mathcal{K}_{\text{Pizzas}}$ at step 0) has only empty columns except one, corresponding to $\exists\forall r_{\text{likes}}(\text{CPizzas0})$ (Fig. 3, bottom-left-hand side).

The whole construction process consists in building a finite sequence of contexts and concept lattices. The lattices at one step are used to calculate the relational extensions for the next step. The last sequence is obtained when the lattice family at one step is isomorphic to the one from the previous step and the extended contexts are unchanged (keeping the same scaling quantifier for a given relation).

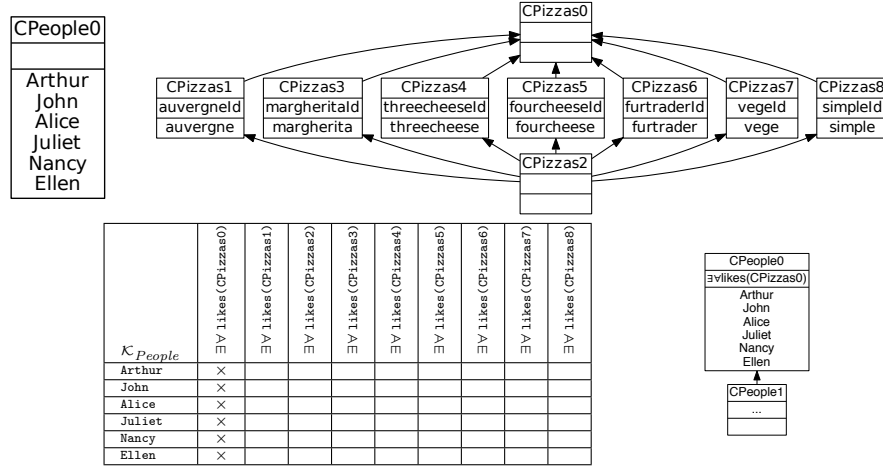


Figure 3: Lattice \mathcal{L}_{People}^0 (top/left), lattice \mathcal{L}_{Pizzas}^0 (top/right), context resulting from the scaling operation $\mathbb{S}_{(r_{likes}, \exists \forall), C_{Pizzas}^0}(\mathcal{K}_{People})$ (bottom/left), lattice \mathcal{L}_{People}^1 (bottom/right)

Algorithm 1, adapted from [43], describes the construction process of RCA. Its inputs are a relational context family (\mathbf{K}, \mathbf{R}) and a scaling quantifier mapping which associates a scaling quantifier with each relation of \mathbf{R} . Its output is a set of concept lattices, one for each context of \mathbf{K} . These concept lattices are stored in an array whose indices correspond to the context indices in \mathbf{K} . The algorithm uses a variable \mathbf{K} to store the current relational extensions of all contexts of \mathbf{K} . The variable $\mathbf{L}_{previous}$ stores the concept lattices built at the previous step. Two external algorithms, not described here, compute respectively a concept lattice associated with a formal context (FCA algorithm) and whether two concept lattices are isomorphic (ISOMORPHIC algorithm). The reader can easily find these algorithms in the literature [44, 45].

Lines 10-11, each \mathbf{L} is initialized with an empty concept lattice (represented by an empty set).

The instructions from lines 12-17 describe the RCA iterative process. First (Line 13), $\mathbf{L}_{previous}$ keeps the current lattices before their modification. Then for every $i, 1 \leq i \leq n$, $\mathbf{K}[i]$ receives the relational extension computed with \mathcal{K}_i and the current concept lattices (Line 15). New lattices are built (Line 16),

based on the current relational extensions. The algorithm stops when all the built lattices are the same at two successive steps. For efficiency purpose, due to its incremental monotonous nature, the algorithm can be implemented by updating at each step the lattices and the relational extensions, rather than computing them from scratch at lines 15-16.

```

1: Algorithm RCA
2: In:  ( $\mathbf{K}, \mathbf{R}$ ) an RCF,  $\rho$  a scaling quantifier mapping
3: Out:  $\mathbf{L}$  array  $[1, \dots, n]$  of lattices
4: Internal:  $\mathbf{K}$  array  $[1, \dots, n]$  of contexts
5: Internal:  $\mathbf{L}_{\text{previous}}$  array  $[1, \dots, n]$  of lattices of previous step
6: Calls: FCA algorithm, which returns the concept lattice associated
    with a context
7: Calls: ISOMORPHIC algorithm, which returns true if the two input
    lattices are isomorphic
8: -----
9: BEGIN
10: for  $i$  from 1 to  $n$  do
11:    $\mathbf{L}[i] \leftarrow \emptyset$ 
12: repeat
13:    $\mathbf{L}_{\text{previous}} \leftarrow \mathbf{L}$ 
14:   for  $i$  from 1 to  $n$  do
15:      $\mathbf{K}[i] \leftarrow \mathbb{E}_{\rho, \mathbf{L}_{\text{previous}}}(\mathcal{K}_i) \ // \ \mathcal{K}_i \in \mathbf{K}$ 
16:      $\mathbf{L}[i] \leftarrow \text{FCA}(\mathbf{K}[i])$ 
17: until  $\forall i \in \{1, \dots, n\}, \text{ISOMORPHIC}(\mathbf{L}[i], \mathbf{L}_{\text{previous}}[i])$ 
18: END

```

Algorithm 1: RCA process, adapted from [43]

The RCA process time complexity is detailed in [2]. By using the worst-case complexity known for concept lattice building and adding the cost of the scaling operations, the time complexity is in $O(n_{\text{step}} * n_{\text{cm}} * n_{\text{om}} * (n_{\text{am}} + n_{\text{om}}))$, where: n_{step} is the step number (which is the least common multiple of the circuit lengths in the data model), n_{cm} is the number of concepts in the largest lattice (in the whole process), n_{om} is the number of objects in the largest extended context, n_{am} is the number of objects in the largest context.

For the Pizza example, the fixed point is obtained after four steps starting from step 0. The final lattice for people at step 3, $\mathcal{L}_{\text{People}}^3$ is shown in Fig. 4. CPeople2 represents the group of people (Arthur, John and Ellen) who own relational attribute $\exists \forall n_{\text{likes}}(\text{CPizzas9})$ (CPizzas9 comes from Fig. 2), meaning

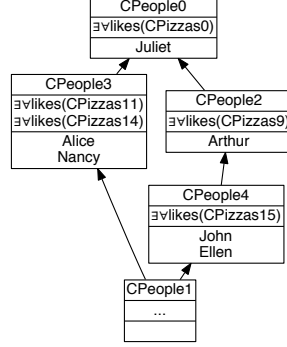


Figure 4: Lattice of people (\mathcal{L}^3_{People}) (step 3 of RCA)

that all the pizzas they like contain at least one individual from **CIngredients1** (dairy ingredients).

2.5. Various Scaling Quantifiers

We define here some quantifiers that we have informally introduced in the previous section, and their variants: the existential quantifier \exists , the universal strict scaling quantifier $\exists\forall$, the contains strict quantifier $\exists\supseteq$, a general contains-percent quantifier $\exists\supseteq_{\geq n\%}$ and a general universal-percent quantifier $\exists\forall_{\geq n\%}$. These last scaling quantifiers come in various levels, according to application needs, to quantify the strength of the link between objects from a set and objects from another set, or to handle noisy results. In the following definitions, we consider two object-attribute contexts $\mathcal{K} = (G, M, I)$ and $\mathcal{K}_r = (G_r, M_r, I_r)$, and a relation r , where $dom(r) = G$, and $ran(r) = G_r$; \mathcal{C}_r is the concept set built on \mathcal{K}_r , n a percent-value, $n \in [0, 100]$. For $o \in G$, the image set of o is denoted by $r(o) = \{o_2 \in G_r \mid (o, o_2) \in r\}$.

Definition 6 (Existential scaling). For every object $o \in G$ and every concept $C_i \in \mathcal{C}_r$, if $r(o) \cap Extent(C_i) \neq \emptyset$, then the relational attribute $\exists r(C_i)$ is added to the attribute set of o . This operation is called existential scaling on \mathcal{K} , \mathcal{C}_r and r .

For example, $\exists r_{hasopping}(\mathbf{CIngredients1})$ from Tab. 2 is assigned to pizzas

(objects from $dom(r_{hastopping})$) that have at least one of the dairy ingredients grouped in **CIngredients1**.

Definition 7 (Universal strict Scaling). For every object $o \in G$ and every concept $C_i \in \mathcal{C}_r$, if $r(o) \neq \emptyset$ and $r(o) \subseteq Extent(C_i)$, then relational attribute $\exists \forall r(C_i)$ is added to the attributes of o .

For example, in Fig. 4 relational attribute $\exists \forall r_{likes}(\text{CPizzas9})$ in the intent of concept **CPeople2**, indicates that objects (people) in the extent of this concept only like pizzas grouped in concept **CPizzas9**, namely pizzas that have at least one of the dairy ingredients (grouped in **CIngredients1**).

Definition 8 (Contains strict Scaling). For every object $o \in G$ and every concept $C_i \in \mathcal{C}_r$, if $Extent(C_i) \neq \emptyset$ and $Extent(C_i) \subseteq r(o)$, then the relational attribute $\exists \supseteq r(C_i)$ is added to the attributes of o .

Definition 9 (Universal-Percent Scaling). For every object $o \in G$ and every concept $C_i \in \mathcal{C}_r$, if more than n percent of $r(o)$ is included in $Extent(C_i)$ (i.e. $|r(o) \cap Extent(C_i)| \geq n|r(o)|/100$), and $|r(o) \cap Extent(C_i)| > 0$ then the relational attribute $\exists \forall_{\geq n\%} r(C_i)$ is added to the attributes of o .

Definition 10 (Contains-Percent Scaling). For every object $o \in G$ and every concept $C_i \in \mathcal{C}_r$, if more than n percent of $Extent(C_i)$ is included in $r(o)$ (i.e. $|r(o) \cap Extent(C_i)| \geq n|Extent(C_i)|/100$), and $|r(o) \cap Extent(C_i)| > 0$, then the relational attribute $\exists \supseteq_{\geq n\%} r(C_i)$ is added to the attributes of o .

2.6. Exploring data with RCA

As explained above, RCA is designed for extracting knowledge patterns and classifications in data that conform to a model similar to a UML class diagram, given through a relational context family. For example, a data analyst may begin with a query pattern which assigns \exists to r_{likes} and \exists to $r_{hastopping}$. Let us recall that (non-trivial) ingredient concepts correspond to categories (dairy, vegetable, etc.). When the corresponding relational extensions are built, the obtained (non-trivial) pizza concepts are groups of pizzas that have at least one ingredient from a specific category; the obtained (non-trivial) people concepts are groups of people who like at least one pizza in a group that has at least one ingredient from a specific category. The concept lattices exhaustively give these people and pizza groups, in a classified way, that shows for example (see Fig. 2)

that having at least one meat ingredient ($\exists r_{\text{hasopping}}(\text{CIngredients4})$) implies having at least one dairy ingredient ($\exists r_{\text{hasopping}}(\text{CIngredients1})$); such knowledge is supported by the lattice generalization order on the concepts introducing the relational attributes ($\text{CPizzas9} \preceq_C \text{CPizzas12}$ in lattice $\mathcal{L}_{\text{Pizzas}}^1$).

However, the data analyst may consider that the information obtained is not satisfactory and decide to use different quantifiers, but for that purpose it is useful to have an idea of the effect of such a change. Thus, when exploring data with RCA, a main problem is to choose the scaling quantifiers to use, and eventually to compare lattices built with different combinations of quantifiers.

We propose below a variation on the primitive RCA algorithm which aims to sketch the way the impact of the changes in the scaling quantifiers can be measured. It is presented in two parts. Algorithm 2 presents a skeleton of exploratory RCA. It includes at Line 21 an optional sub-algorithm (Algorithm 3) which allows the expert to measure the impact that different scaling quantifier selections would have in the next step, before choosing one. In Algorithm 2, Lines 16-19 consist in asking the expert to select the object-attribute contexts, the relations and the algorithms to build the conceptual structure (the concept lattice, the AOC-poset or an Iceberg concept lattice), that she/he finds appropriate for her/his analysis. Line 20 is used to memorize in $\mathbf{IC}_{\text{previous}}$ the indices of the object-attribute contexts that have already been computed in one previous step and check the consistency of the chosen relations. Indeed, a constraint for selecting relations at Line 19 is that for any $r \in \mathbf{SR}$, with $r \subseteq O_i \times O_j$, the respective K_i , K_j are in \mathbf{SC} and $j \in \mathbf{IC}_{\text{previous}}$, meaning that a conceptual structure (concept lattice, AOC-poset, Iceberg lattice) has been built in a previous step (which is not necessarily the immediately preceding step). Then Line 22, the expert selects a scaling quantifier for each selected relation. Lines 23-25, the relational extensions and the new conceptual structures are computed. Line 26 shows that the expert can stop at every step.

Algorithm 3 (SEEIMPACT) measures the impact of changes of scaling quantifiers. It computes a table `UNCERTAINTYCHANGES`. `UNCERTAINTYCHANGES[e][i]` is the difference between the concept number of lattice i at the preceding step

```

1: Algorithm RCA-EXPLORE
2: In:  $(\mathbf{K}, \mathbf{R})$  an RCF
3: Out:  $\mathbf{L}$  array  $[1, \dots, n]$  of lattices
4: Internal:  $\mathbf{K}$  array  $[1, \dots, n]$  of contexts
5: Internal:  $\mathbf{L}_{\text{previous}}$  array  $[1, \dots, n]$  of lattices of the preceding step
6: Internal:  $\mathbf{IC}_{\text{previous}}$  indices of the object-attribute contexts that
   have already been computed in any previous step
7: Internal:  $\mathbf{SR}$  set of object-object contexts selected by the user
8: Internal:  $\mathbf{SC}$  set of object-attribute contexts selected by the user
9: Internal:  $\mathbf{ISC}$  indices of object-attribute contexts selected by the
   user
10: Calls: CONCEPTSTRUCT algorithm, which returns the chosen conceptual
   structure: the concept lattice, the AOC-poset or an Iceberg concept
   lattice associated with a context
11: Note: SELECT is an expert action
12: -----
13: BEGIN
14:  $\mathbf{IC}_{\text{previous}} \leftarrow \emptyset$ 
15: for  $i$  from 1 to  $n$  do
16:    $\mathbf{L}[i] \leftarrow \emptyset$ 
17: repeat
18:    $\mathbf{L}_{\text{previous}} \leftarrow \mathbf{L}$ 
19:   SELECT a subset  $\mathbf{SC} \subseteq \mathbf{K}$  of object-attribute contexts
20:   SELECT a subset  $\mathbf{SR} \subseteq \mathbf{R}$  of relations
21:   SELECT an algorithm CONCEPTSTRUCT for each  $\mathcal{K}_i \in \mathbf{SC}$ 
22:    $\mathbf{IC}_{\text{previous}} \leftarrow \mathbf{IC}_{\text{previous}} \cup \mathbf{ISC}$ 
23:   CALL SEEIMPACT - ALGORITHM 3
24:   SELECT the scaling quantifier  $\rho(r)$  for each  $r \in \mathbf{SR}$ 
25:   for  $i \in \mathbf{ISC}$  do
26:      $\mathbf{K}[i] \leftarrow \mathbb{E}_{\rho, \mathbf{L}_{\text{previous}}}(\mathcal{K}_i)$ 
27:      $\mathbf{L}[i] \leftarrow \text{CONCEPTSTRUCT}(\mathbf{K}[i])$ 
28: until SELECT STOP
29: END

```

Algorithm 2: RCA process adapted to exploration and measure impact of changes in scaling quantifiers, adapted from [43]

and the concept number of lattice i at the current step using the set of scaling quantifiers of step e . Line 11, the expert selects a number of scaling quantifier alternatives he wants to analyze. Line 12-17, an alternative e is examined. Then the relational extension and the conceptual structures are built (resp. Line 15 and Line 16), then $\text{UNCERTAINTYCHANGES}[e][i]$ can be computed. Line 18 shows results to the expert.

```

1: Algorithm SEEIMPACT
2: In: Variables of Algorithm RCA-EXPLORE Lines 2-25
3: Out: UNCERTAINTYCHANGES array  $[1, \dots, n_p][1, \dots, n]$  of values indicating
      the difference between the concept number at the preceding step and
      at the current step using an alternative set of scaling quantifiers
4: Internal:  $n_p$  the selected number of alternatives that the expert
      wants to investigate
5: Internal:  $\mathbf{K}_{\text{impact}}$  array  $[1, \dots, n_p][1, \dots, n]$  of contexts
6: Internal:  $\mathbf{L}_{\text{impact}}$  array  $[1, \dots, n_p][1, \dots, n]$  of lattices
7: Calls: CONCEPTSTRUCT algorithm, which returns the chosen conceptual
      structure: the concept lattice, the AOC-poset or an Iceberg concept
      lattice associated with a context
8: Note: SELECT is an expert action
9: -----
10: BEGIN
11: SELECT  $n_p$  a number of alternatives the expert wants to investigate
12: for  $e$  from 1 to  $n_p$  do
13:   SELECT the quantifier  $\rho_e(r)$  for each  $r \in SR$ 
14:   for  $i \in \text{ISC}$  do
15:      $\mathbf{K}_{\text{impact}}[e][i] \leftarrow \mathbb{E}_{\rho_e, L_{\text{previous}}}(\mathcal{K}_i)$ 
16:      $\mathbf{L}_{\text{impact}}[e][i] \leftarrow \text{CONCEPTSTRUCT}(\mathbf{K}_{\text{impact}}[e][i])$ 
17:      $\text{UNCERTAINTYCHANGES}[e][i] = \#Cpts(\mathbf{L}_{\text{impact}}[e][i]) - \#Cpts(L_{\text{previous}}[i])$ 
18:   PRINT UNCERTAINTYCHANGES and metrics min, max, average, standard
      deviation for  $e$  varying on  $i$ 
19: END

```

Algorithm 3: Process for measuring the impact of different scaling quantifier selections ("Cpts" is a shortcut for Concepts)

These algorithms aim to give a support for the expert to measure the impact of the scaling quantifiers when RCA is conducted in an exploratory manner. This paper will now focus on understanding the foundations of the quantifiers behaviors and guiding the navigation.

3. Understanding quantifier generalization

In this section, we examine a generality relation between the previously introduced scaling quantifiers. We first define a generality relation \preceq_A between relational attributes. The comparison of relational attributes can be defined on different relations and concepts as long as all relations have the same domain. We first introduce the generality relation, and we illustrate it on an example restricted to one relation. Then we examine how this generality relation enlightens the relations between the lattices built with the different scaling operators.

3.1. The generality relation

Let us first define the generality relation on relational attributes.

Definition 11 (Generality relation \preceq_A on relational attributes). Let $a_1 = q_1 r_1(\mathbf{C}_1)$ and $a_2 = q_2 r_2(\mathbf{C}_2)$ be two relational attributes such that $r_1 \subseteq G_3 \times G_1$, $r_2 \subseteq G_3 \times G_2$, $\text{Extent}(\mathbf{C}_1) \subseteq G_1$ and $\text{Extent}(\mathbf{C}_2) \subseteq G_2$: a_1 is more general than a_2 (written $a_1 \preceq_A a_2$) if $\{a_2\}' \subseteq \{a_1\}'$.

The generality relation on scaling quantifiers can then be introduced as follows. A scaling quantifier q_1 is more general than a scaling quantifier q_2 , if the combination of q_1 with any relation r /concept \mathbf{C} couple yields a relational attribute more general than with the other quantifier q_2 .

Definition 12 (Generality relation \preceq_S on scaling quantifiers). Let q_1 and q_2 be two scaling quantifiers; q_1 is more general than q_2 (written $q_1 \preceq_S q_2$) if $\forall r, \forall \mathbf{C}, q_1 r(\mathbf{C}) \preceq_A q_2 r(\mathbf{C})$.

Some generality relations can be observed between the previously defined scaling quantifiers as depicted by Fig. 5. We detail in the following these generality relations for the various scaling quantifiers.

Proposition 1. $\forall n_1, n_2 \in [0, 100[, \exists \forall_{\geq n_1\%} \preceq_S \exists \forall_{\geq n_2\%}$ if $n_1 \leq n_2$.

PROOF. Let $r \subseteq G_1 \times G_2$ be an object-object relation, C a concept from the lattice built on the object-attribute context (G_2, M_2, I_2) , $a_1 = \exists \forall_{\geq n_1\%} r(C)$, $a_2 = \exists \forall_{\geq n_2\%} r(C)$ with $n_1 \leq n_2$.

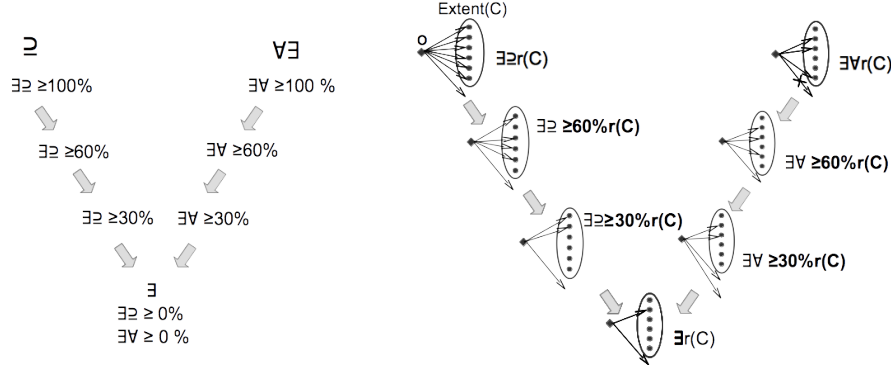


Figure 5: Generality relation on quantifiers (left), and illustration: relation r between object o and concept C extent, where C is a concept of a concept lattice on $\text{ran}(r)$ (right)

According to the definition of $\exists \forall$: $\{a_1\}' = \left\{ o \mid |r(o) \cap \text{Extent}(C)| \geq \frac{n_1 |r(o)|}{100} \right\}$ and $\{a_2\}' = \left\{ o \mid |r(o) \cap \text{Extent}(C)| \geq \frac{n_2 |r(o)|}{100} \right\}$

Let $o \in \{a_2\}'$, by definition of $\{a_2\}'$, $|r(o) \cap \text{Extent}(C)| \geq \frac{n_2 |r(o)|}{100}$, and since $n_1 \leq n_2$ this implies that $|r(o) \cap \text{Extent}(C)| \geq \frac{n_2 |r(o)|}{100} \geq \frac{n_1 |r(o)|}{100}$, thus $o \in \{a_1\}'$.

Now let $o \in \{a_1\}'$, by definition of $\{a_1\}'$, $|r(o) \cap \text{Extent}(C)| \geq \frac{n_1 |r(o)|}{100}$. Let $v = |r(o) \cap \text{Extent}(C)|$, we have $v \geq \frac{n_1 |r(o)|}{100}$ and since $n_1 \leq n_2$ we may have $v \leq \frac{n_2 |r(o)|}{100}$ in some cases, so that $o \notin \{a_2\}'$. Thus $\{a_2\}' \subseteq \{a_1\}'$, but we do not have $\{a_2\}' = \{a_1\}'$.

Proposition 2. $\forall n_1, n_2 \in [0, 100[, \exists \supseteq_{\geq n_1\%} \preceq_S \exists \supseteq_{\geq n_2\%}$ if $n_1 \leq n_2$.

The proof follows the same schema as for the previous proposition 1 except that $r(o)$ is replaced by $\text{Extent}(C)$ in the right part of the conditions.

Proposition 3. The scaling quantifiers \exists , $\exists \forall_{\geq 0\%}$ and $\exists \supseteq_{\geq 0\%}$ are equivalent according to \preceq_S .

PROOF. Let $r \subseteq G_1 \times G_2$ be an object-object relation, C a concept from the lattice built on the object-attribute context (G_2, M_2, I_2) .

$$\{\exists \forall_{\geq 0\%} r(C)\}' = \{o \mid |r(o) \cap \text{Extent}(C)| > 0\} = \{o \mid r(o) \cap \text{Extent}(C) \neq \emptyset\} =_{\text{def}} \{\exists r(C)\}' \text{ and } \{\exists \supseteq_{\geq 0\%} r(C)\}' = \{o \mid |r(o) \cap \text{Extent}(C)| > 0\} =_{\text{def}} \{\exists r(C)\}'$$

Proposition 4. $\forall n_1, n_2 \in [0, 100[$ the relational scaling quantifiers $\exists \forall_{\geq n_1\%}$ and $\exists \exists_{\geq n_2\%}$ are not comparable according to \preceq_S .

PROOF. According to definition 12 the quantifiers are not comparable if both $\exists \forall_{\geq n_1\%} \not\preceq_S \exists \exists_{\geq n_2\%}$ and $\exists \exists_{\geq n_2\%} \not\preceq_S \exists \forall_{\geq n_1\%}$ hold for at least one counter-example. Let us define an RCF where both properties are verified: G_1 is an object set from an object-attribute context \mathcal{K}_1 , $o_1, o_2 \in G_1$ two objects, $r \subseteq G_1 \times G_2$ a relation where G_2 is the object set of a context \mathcal{K}_2 , and C a concept from the concept set \mathcal{C}_2 of \mathcal{K}_2 .

If o_1 is such that $|r(o_1)| = |r(o_1) \cap \text{Extent}(C)| = 1$ then $o_1 \in \{\exists \forall r(C)\}'$. If $|\text{Extent}(C)| > \frac{100}{n_2}$ then we have a counter-example for $\exists \exists_{\geq n_2\%} \preceq_S \exists \forall_{\geq n_1\%}$.

If o_2 is such that $|r(o_2) \cap \text{Extent}(C)| = |\text{Extent}(C)|$ then $o_2 \in \{\exists \exists r(C)\}'$. If $|r(o_2)| > \frac{100|\text{Extent}(C)|}{n_1}$ then we have a counter-example for $\exists \forall_{\geq n_1\%} \preceq_S \exists \exists_{\geq n_2\%}$.

By assigning $|\text{Extent}(C)| = \left\lfloor \frac{100}{n_2} \right\rfloor + 1$ and $|r(o_2)| = \left\lfloor \frac{100}{n_1} \left(\left\lfloor \frac{100}{n_2} \right\rfloor + 1 \right) \right\rfloor + 1$ we are able to build a counter-example for any combination of n_1 and n_2 , hence proving that the quantifiers are not comparable.

3.2. Illustration of the generality relation

In this section, we introduce an example to illustrate the generality relation presented in the previous section and the way its understanding may help exploring data. It is reduced to the observation of a single relation for which we can show several quantifier combinations. The example will follow the evolution of the extent of a concept introducing a relational attribute involving both this relation and the same target concept when using different scaling quantifiers. For a given relation r , this corresponds to the task of an expert interested in knowing the group of objects of the r domain that are connected (in various ways determined by the scaling quantifiers) to a specific group of objects (concept) from the r range.

Table 3 shows an object-attribute context of dishes described by their main ingredient type (fish, meat, cheese) and an object-attribute context of wines described by their color, production area and flavor. Table 4 presents the

`matchesWith` relation between dishes and wines, which is alternatively represented graphically in Fig. 6.

Table 3: Dish and Wine object-attribute contexts

\mathcal{K}_{Dish}	fish	meat	cheese
FriedChicken		×	
Paella	×		
BeefStew		×	
Raclette			×
Ossobucco		×	
SalmonPekin	×		
CodBrandade	×		
CheeseSouffle			×
VealBlanquette		×	

\mathcal{K}_{Wine}	white	red	rosy	alsacian	burgundy	languedoc	flowery	fruity
Riesling	×			×				
Gewurztraminer	×			×				
PinotGris	×			×				
Sylvaner	×			×				
Tokay	×			×				
Klevner	×			×			×	
RomaneeConti		×			×			×
Mandourelle			×			×	×	
Lestabel			×			×	×	
Montrose			×			×	×	
Bressades			×			×	×	

Figure 7 presents the concept lattice of the object-attribute context of wines (Tab. 3 bottom). In this concept lattice, `Concept_Wine_5` represents famous white wines produced in Alsace region (France). We will focus on relational attributes with that concept as target but based on different quantifiers, of the form $qr_{matchesWith}(\text{Concept_Wine_5})$.

The observed scaling quantifiers are in $\mathbf{Q} = \{ \exists \supset = \exists \supset \geq 100\%, \exists \supset \geq 60\%, \exists \supset \geq 30\%, \exists \forall = \exists \forall \geq 100\%, \exists \forall \geq 60\%, \exists \forall \geq 30\%, \exists = \exists \supset \geq 0\% = \exists \forall \geq 0\% \}$ (see Fig. 5). Figure 8 shows 7 concepts, extracted from the 7 concept lattices built with the scaling opera-

Table 4: `matchesWith` relation between dishes and wines

$r_{matchesWith}$	Riesling (Ri)	Gewurztraminer (G)	PinotGris (PG)	Sylvaner (S)	Tokay (T)	Klevner (K)	RomaneeConti (Ro)	Mandourelle (Ma)	Lestabel (L)	Montrose (Mo)	Bressades (B)
FriedChicken	×						×	×	×		
Paella	×	×	×	×			×				
BeefStew			×		×			×			
Raclette	×		×	×	×			×	×	×	×
Ossobucco							×		×		
SalmonPekan	×	×									
CodBrandade	×	×	×	×	×	×	×	×	×	×	×
CheeseSouffle				×		×	×	×	×	×	×
VealBlanquette						×			×		

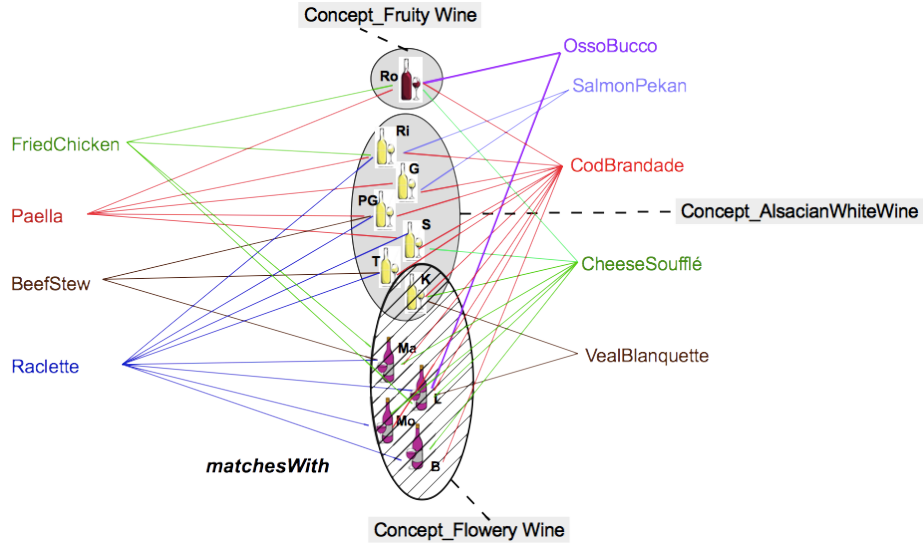


Figure 6: Graphical representation of `matchesWith` relation

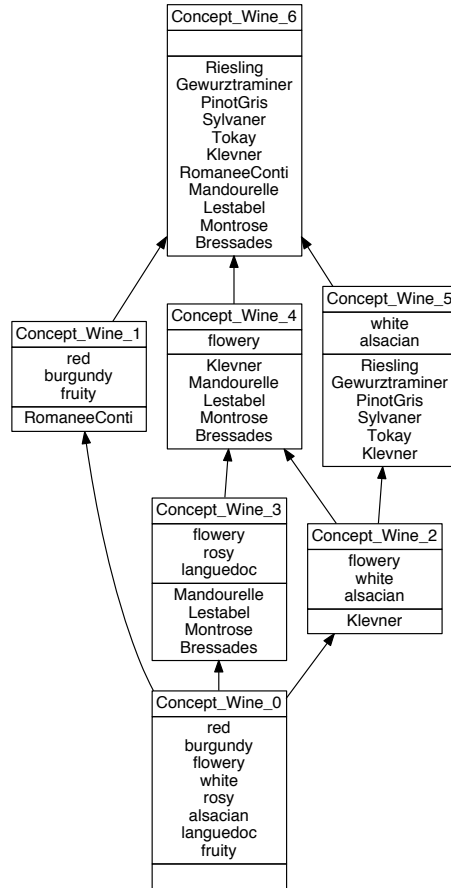


Figure 7: Concept lattices of wines built from object-attribute context in Tab. 3 (right-hand side)

tors based on these 7 scaling quantifiers; each of these concepts introduces the relational attribute $qr_{matchesWith}(\text{Concept_Wine_5})$. In the left branch of the

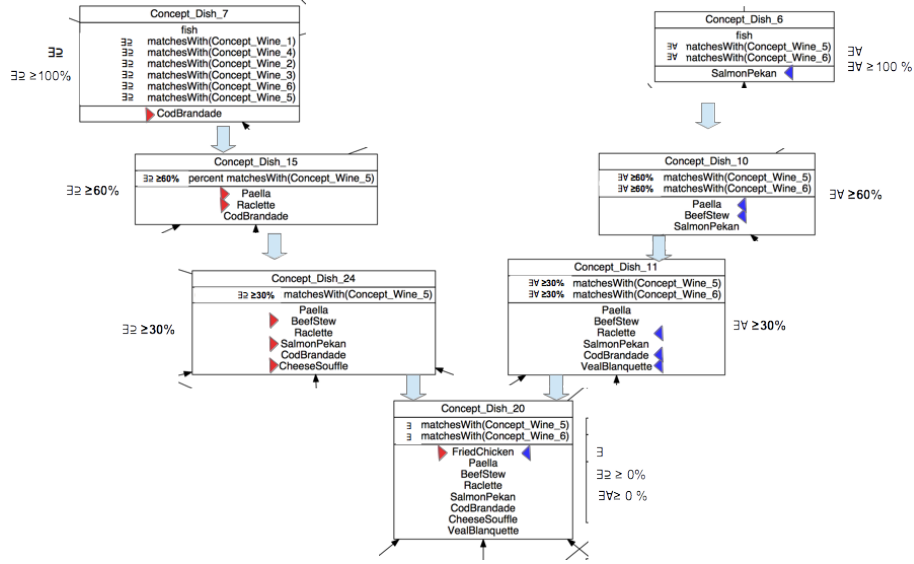


Figure 8: Concepts introducing the relational attribute $q \text{ matchesWith}(\text{Concept_Wine_5})$ in the different lattices of $\mathbf{Q} = \{ \exists\geq, \exists\geq 60\%, \exists\geq 30\%, \exists V, \exists V \geq 60\%, \exists V \geq 30\%, \exists \}$

diagram, going from the top to the bottom (from specific to general quantifiers):

- **Concept_Dish_7** introduces $\exists\geq r_{matchesWith}(\text{Concept_Wine_5})$; it represents the dishes that match with all Alsatian white wines.
- **Concept_Dish_15** introduces $\exists\geq 60\% r_{matchesWith}(\text{Concept_Wine_5})$; it represents the dishes that match with at least 60% of the Alsatian white wines.
- **Concept_Dish_24** introduces $\exists\geq 30\% r_{matchesWith}(\text{Concept_Wine_5})$; it represents the dishes that match with at least 30% of the Alsatian white wines.
- **Concept_Dish_20** introduces $\exists r_{matchesWith}(\text{Concept_Wine_5})$; it represents the dishes that match with at least one Alsatian white wine.

In the right branch of the diagram, going again from the top to the bottom (from specific to general quantifiers):

- **Concept_Dish_6** introduces $\exists\forall r_{\text{matchesWith}}(\text{Concept_Wine_5})$; it represents the dishes that match only with Alsatian white wines.
- **Concept_Dish_10** introduces $\exists\forall_{\geq 60\%} r_{\text{matchesWith}}(\text{Concept_Wine_5})$; it represents the dishes such that more than 60% of the matches are with Alsatian white wines.
- **Concept_Dish_11** introduces $\exists\forall_{\geq 30\%} r_{\text{matchesWith}}(\text{Concept_Wine_5})$; it represents the dishes such that more than 30% of the matches are with Alsatian white wines.

In this diagram, the triangles indicate the dishes which are newly introduced by the concept compared to the upper (more specific) ones. It is easy to see the phenomenon of extent increase when going top-down, that is, from the use of a specific to a more general quantifier. Additionally, it can be seen that all extents are different which is the general case. In some special cases, two different $qr(\mathcal{C})$, for the same r and \mathcal{C} , may be introduced in concepts with the same extent. The extent increase corresponds to constraint relaxing, for example when the user finds that the answer he obtained with a quantifier is too restricted and he accepts to use a different, more general, quantifier.

For example, a waiter may want to propose a set of dishes to a group of tourists in a restaurant in Strasbourg, which matches with all Alsatian white wines, in order to offer a large local wine choice. He would first apply quantifier $\exists\supseteq$. But the only dish which matches with all Alsatian white wines is *CodBrandade*. Notice that *CodBrandade* matches also with some red and rosy wines (so that it is not in **Concept_Dish_6**), meaning that if one tourist does not like white wine, he can have a wine adapted to his dish. Thus the waiter could relax the constraint and use the quantifier $\exists\supseteq_{\geq 60\%}$, and obtain *CodBrandade*, *Paella*, and *Raclette*. This means that these dishes, even not matching with all the (same) Alsatian white wines, have great affinity with that type of wine.

The constraints may yet be relaxed by using $\exists \supseteq_{\geq 30\%}$ or \exists . The same process can be done in the other generalization branch, but the interpretation is slightly different. If the waiter applies $\exists \forall$, he will obtain dishes that match only with Alsatian white wines. This means an even greater affinity of the dish (here *SalmonPekan*) to Alsatian white wines, thus no possibility to propose another wine to the tourist who does not like white wine. If this is too restrictive, the $\exists \forall_{\geq 60\%}$ quantifier will allow to propose dishes that match with many white wines and not many other types of wines, thus allowing to focus the choice for the tourist who does not like white wine.

The observed property about extent increase when generalizing a scaling quantifier is given below.

Property 1 (Extent inclusion based on generalization). *For two relational attributes $q_g r(\mathbb{C})$ and $q_s r(\mathbb{C})$ with $q_g \preceq_S q_s$, the extent of the concept introducing $q_g r(\mathbb{C})$ includes the extent of the concept introducing $q_s r(\mathbb{C})$.*

This property enables us to compare the lattices obtained with two scaling quantifiers q_g and q_s such that $q_g \preceq_S q_s$.

3.3. Impact of the generality relation on the concept lattices

Let us denote by $\mathcal{L}_K^{final}(q)$ the final concept lattice computed by RCA on object-attribute context $\mathcal{K} = (G, M, I)$ with quantifier q applied to relation r (with domain G). We have the following property that shows how concept lattices built using q_g and q_s can be compared. It is then exemplified based on Fig. 9 that shows the projection of $\mathcal{L}_{Dish}^{final}(\exists \forall_{\geq 60\%})$ to $\mathcal{L}_{Dish}^{final}(\exists \forall_{\geq 30\%})$.

Property 2 (Projection). *Let us consider an RCF, an object-attribute context $K = (G, M, I)$ of the considered RCF, and two scaling quantifiers q_g and q_s , with $q_g \preceq_S q_s$. There is a mapping from concepts of $\mathcal{L}_K^{final}(q_s)$ to concepts of $\mathcal{L}_K^{final}(q_g)$ such that:*

- A concept of $\mathcal{L}_K^{final}(q_s)$ which introduces a simple (non relational) attribute is mapped to the concept of $\mathcal{L}_K^{final}(q_g)$ which introduces the same simple

attribute. The concept extents are the same. The concept intent may be different.

- A concept of $\mathcal{L}_K^{final}(q_s)$ which introduces a relational attribute $q_sr(\mathbb{C})$ is mapped in $\mathcal{L}_K^{final}(q_g)$ to the concept which introduces the relational attribute $q_g r(\mathbb{C})$. The concept extent may be larger for the concept introducing $q_g r(\mathbb{C})$.
- A concept \mathbb{C} of $\mathcal{L}_K^{final}(q_s)$ with an empty simplified intent is mapped in $\mathcal{L}_K^{final}(q_g)$ to the lowest concept which includes \mathbb{C} extent.

Let us illustrate the different mappings and justify the property. For the first category of concepts of $\mathcal{L}_K^{final}(q_s)$, introducing no relational attribute, we have for example: **Concept_Dish_2** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ mapped to **Concept_Dish_2** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$. Both introduce the attribute *fish*. For this category, the set of objects that have the simple attribute cannot change. This is sufficient to establish the property.

For the second category of concepts of $\mathcal{L}_K^{final}(q_s)$, introducing a relational attribute $q_sr(\mathbb{C})$, we have for example: **Concept_Dish_10** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ mapped to **Concept_Dish_11** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$. **Concept_Dish_10** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ introduces $\exists\forall_{\geq 60\%} matchesWithConcept_Wine_5$ and **Concept_Dish_5** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$ introduces $\exists\forall_{\geq 30\%} matchesWithConcept_Wine_5$. In this category, due to the generality relation between relational attributes, the set of objects that have $q_sr(\mathbb{C})$ will also have $q_g r(\mathbb{C})$. Thus there will be a concept in $\mathcal{L}_K^{final}(q_g)$ introducing $q_g r(\mathbb{C})$ with an extent larger than the extent of the concept in $\mathcal{L}_K^{final}(q_s)$ which introduces $q_sr(\mathbb{C})$.

For the third concept category of $\mathcal{L}_K^{final}(q_s)$, an example is given by the mapping of **Concept_Dish_9** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ to **Concept_Dish_2** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$. **Concept_Dish_9** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ extent is {Paella, SalmonPekan}, **Concept_Dish_2** of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$ extent is {Paella, CodBrandade, SalmonPekan}. In this case, the objects that are in the extent of the concept will have the inherited simple attributes and the inherited relational attributes of the form $q_sr(\mathbb{C})$. Then they

will also inherit the relational attributes of the form $q_g r(\mathbb{C})$ in $\mathcal{L}_K^{final}(q_g)$. Thus they will continue sharing attributes in $\mathcal{L}_K^{final}(q_g)$ and there will be a concept in $\mathcal{L}_K^{final}(q_g)$ whose extent will contain them.

Table 5 sums up the projections between concepts of $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ (left column) and $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$ (right column).

Table 5: Mapping from $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$ to $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$

Concept number in $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 60\%})$	Concept number in $\mathcal{L}_{Dish}^{final}(\exists\forall_{\geq 30\%})$
Introducing a simple attribute	
2	2
3	3
1	1
Introducing relational attributes	
4	4
10	11
8	12
0	0
Empty simplified intent	
9	2
7	9
5	5
6	1

This projection organizes the space in which a final user can move, by relaxing, or enforcing, or changing the constraints on the groups of objects. It also guides (and traces) the user to move between the concepts of the different concept lattices. The lattices are useful for exploring around a concept with the same scaling quantifier.

4. Implementation

The quantifiers presented in this paper have been implemented and tested in RCAExplore⁵ tool. It implements a modular version of RCA where core elements, such as concept generator algorithms and scaling operators, have several

⁵<http://dolques.free.fr/rcaexplore>

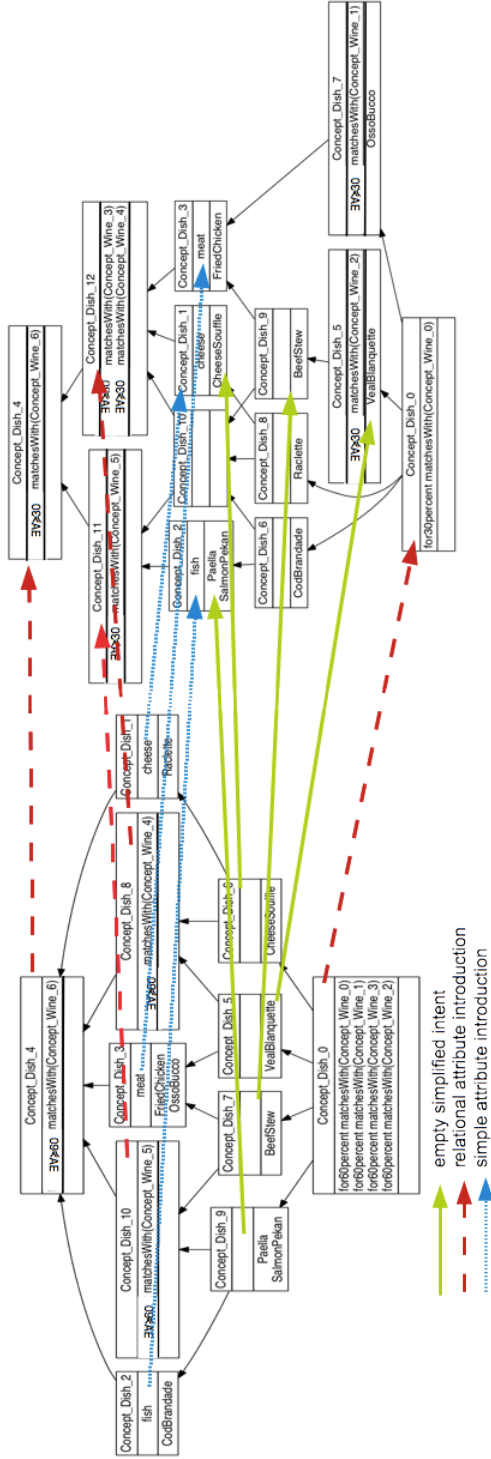


Figure 9: Projection of concept lattice of dishes with $\exists V_{\ge 60\%}$ (left) to the concept lattice of dishes with $\exists V_{\ge 30\%}$ (right)

alternatives that can be controlled interactively by the user. RCAExplore has been developed in Java as part of the Fresqueau project to explore new ways for extracting knowledge from watercourses quality data. Some features of the tool have already been published. In [14] we present the interactive process and in [11] we present the use of AOC-posets as an alternative to concept lattices. AOC-posets have been introduced for several FCA applications. Given a concept lattice $(\mathcal{C}_K, \preceq_C)$, its associated AOC-poset (for Attribute-Object-Concept poset) is the suborder restricted to the set of *object-concepts* (which introduce at least one object) and *attribute-concepts* (which introduce at least one attribute). Lattice computation may have high complexity in time and space, as the size of the lattice may rise up to $2^{\min(|G|, |M|)}$ concepts. The size of an AOC-poset is bounded by $|G| + |M|$. This significant size difference is observed in practice [11, 42]. Particular implication rules (implication rules having a premise containing one attribute) are extracted in the context of Fresqueau project in [11], on a variant of the dataset used in the current paper. AOC-posets and Iceberg-lattices are compared for this task, and the results highlight that, although RCA is impracticable when Iceberg-lattices are used, the use of AOC-posets makes RCA perfectly applicable in this case. In [42], with additional figures given in the companion presentation⁶, RCA enables to normalize UML class models in order to remove their redundancies and highlight relevant new abstractions. The method is applied to 15 UML class models corresponding to the successive versions of an information system on Pesticides and to 4 Java programs in the domain of human resource management. The models are true models coming from a public organism which works in the environment domain (IRSTEA⁷) and from industry (Berger Levrault⁸). The results show that there are no time computation problems and that the use of the AOC-posets makes the method practical for design experts (that have to analyze the extracted new abstractions) and far more efficient than with lattices.

⁶<http://www.lirmm.fr/~huchard/Documents/Papiers/miralles2015.pdf>

⁷<http://www.irstea.fr/en/accueil>

⁸<https://www.berger-levrault.fr/>

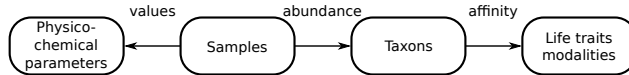


Figure 10: Relational schema used for experiments on the watercourse quality dataset

The AOC-poset is sufficient to recover the entire object-attribute context, and for several tasks, such as building non-redundant classifications or finding implication rules with premise composed of a single attribute. Note that Iceberg lattices can also be used in order to limit the exploration space, using threshold on concepts and thus losing more or less information [10, 46].

For illustration purpose, we rely on data collected during the Fresqueau project which aimed to study biological and physico-chemical parameters sampled in waterbodies, in order to assess and monitor the hydrobiologic quality of running waters [47]. Data were collected from various French databases such as the ones proposed by the Eau France portal⁹. In the following we consider a dataset dealing with taxons¹⁰, life traits which are taxon characteristics, physico-chemical parameters, and water samples. This dataset was used in the Fresqueau project to extract knowledge about physico-chemical values and their link to life traits modalities, e.g. finding the characteristics of taxons that live in water streams with a given physico-chemical quality. The water samples have been taken from monitoring stations at different dates between 2005 and 2009, that were considered representative by the biologists. The station names are anonymized. The dataset is available online¹¹, it has been analyzed with the 2015 RCAexplore version¹².

Based on this dataset, we study the link between the scaling quantifiers and the number of generated concepts, that represent the exploration space in which an expert moves during his/her analysis tasks. We compare the number

⁹<http://www.eaufrance.fr>

¹⁰taxons represent animals or plants, at different levels as species, genres or families

¹¹<https://www.lirmm.fr/~huchard/Documents/Papiers/latticedataset-taxons-hierarchy.rcft>

¹²<https://www.lirmm.fr/~huchard/Documents/Papiers/rcaexplore-20151012.jar>

of generated concepts when using the various quantifiers.

The relational schema used to parameterize the experiments is presented in Fig. 10. Samples have been taken at various river sites and various dates. Water samples are described by five object-object relations representing different levels of physico-chemical parameter values and five object-object relations representing different levels of abundance of taxons. Physico-chemical parameters can be magnesium, chloride, organic carbon, ... and are described (in an object-attribute relation) by several categories (metals, pesticides, ...). Taxons are described by their parents at different hierarchical levels, e.g. *Acanthoceras* has for parent *Chaetocerotales* and *Coscinodiscophyceae* (the direct parent of *Chaetocerotales*). Life traits are for example microhabitats, maximal sizes, reproduction modes, respiration modes, etc. Taxons are described by five object-object relations representing different levels of affinity with the life traits modalities, e.g. *Acentrella* has an affinity level 1 for the modality "tegument" and an affinity level 2 for the modality "gill" of trait "respiration modes".

In these experiments we use AOC-posets. Table 6 presents the size of the contexts and AOC-posets for physico-chemical parameters and life traits modalities at step 0 (they will not change in the following steps because they have no outgoing relations). The 40 physico-chemical parameters are described by their identity (40 attributes) and 12 classes (metals, pesticides, ...). The 116 life traits modalities are described only by their identity (the relation is a diagonal).

Table 6: Size of the contexts and AOC-posets for physico-chemical parameters and life traits modalities at step 0

	#objects	#attributes	#concepts
Physico-chemical parameters	40	52	49
Life traits modalities	116	116	116

Table 7 presents the results obtained for the 392 taxons and 1702 water samples while following the schema of Fig. 10 with different quantifiers. The first table indicates the object and attribute numbers of the taxons and water samples object-attribute contexts. Water samples are not described by any

attribute, while taxons are described by their taxon parents.

Table 7: Size of the contexts and AOC-posets for Taxons and Water samples at step 3

	Taxons	Water samples
#objects	392	1702
#attributes	392	0

	Taxons #concepts	Water samples #concepts	Generalization
$\exists \sqsubseteq$	832	2005	\downarrow
$\exists \sqsubseteq_{>60\%}$	832	2015	\downarrow
$\exists \sqsubseteq_{\geq 30\%}$	832	2071	\downarrow
\exists	832	3742	\square
$\exists \forall_{\geq 30\%}$	368	2183	\uparrow
$\exists \forall_{>60\%}$	293	2013	\uparrow
$\exists \forall$	293	2013	\uparrow

For high levels (level5)	Taxons #concepts	Water samples #concepts	Generalization
$\exists \sqsubseteq$	288	731	\downarrow
$\exists \sqsubseteq_{>60\%}$	288	733	\downarrow
$\exists \sqsubseteq_{\geq 30\%}$	288	812	\downarrow
\exists	288	923	\square
$\exists \forall_{\geq 30\%}$	287	626	\uparrow
$\exists \forall_{>60\%}$	252	270	\uparrow
$\exists \forall$	252	247	\uparrow

Then the second table shows the numbers of concepts obtained with different quantifier choices on the whole RCF. The highest numbers are obtained, as expected, for \exists quantifier (resp. 832 taxon concepts and 3742 water sample concepts), and they decrease when more general quantifiers are used. Using $\exists \sqsubseteq$ and its percent versions, the numbers of concepts decrease until reaching respectively 832 for taxons and 2005 for water samples. Using $\exists \forall$ and its percent versions, the numbers of concepts decrease until reaching respectively 293 for taxons and 2013 for stations.

The third table shows the numbers of concepts obtained with different quantifier choices on the RCF restricted to the object-object relations corresponding to the highest value (5) for physico-chemical, abundance and affinity levels. The highest numbers are obtained, as expected, for \exists quantifier (resp. 288

taxon concepts and 923 station concepts), and they decrease when more general quantifiers are used. Using $\exists\supseteq$ and its percent versions, the numbers decrease until reaching respectively 288 for taxons and 731 for water samples. Using $\exists\forall$ and its percent versions, the number of water sample concepts decreases until reaching 247, while the number of taxon concepts reaches 252.

The last column in these two tables indicates the generalization order on quantifiers as shown in Fig. 5.

The variations show the effect of scaling quantifiers on relaxing or reinforcing the constraints, and thus on expanding or reducing the data exploration space. To simplify our illustration, we applied the same quantifier to all object-object relations, but of course, during an analysis, the object-object relations may be associated to different quantifiers that besides may change according to the analysis steps. Nevertheless, using the "extreme" combination of quantifiers (\exists versus $\exists\forall$), our study shows the borders of the exploration space.

5. Related Work

Multi-relational data have been studied in the domain of Formal Concept Analysis from several perspectives. In the earliest approaches [48, 49, 34], graphs connecting objects are classified, while in RCA, objects are classified depending on their relations to other objects. More recently, relational data have been dealt with logical concept analysis [50]. The concepts' intents may then contain relational attributes such as $(\exists r.f)$. Meta-relations are also built for navigating from a concept to another. But, contrarily to RCA, no iteration is performed. In [51], authors compute a basis of general concept inclusions in Description Logics \mathcal{EL}_{gfp} where cyclic concept definition has close connections with RCA. In [52], they extract tuples of boolean factors from the various data tables. Several connection schemas can be applied that are similar to the scaling operators of RCA. Compared to RCA, the boolean factors are only a part of the formal concepts that can be built from the object-attribute tables, and the approach does not iterate and cannot propagate knowledge. In [53], a Galois connection (and the derived concept lattice) is introduced to query sets of objects connected

by relations. Only existential queries are expressed and there is no iteration, thus no possibility to progressively find the concepts. Ferré [54] proposes an extension of FCA to knowledge graphs: in this framework, a context is a conceptual graph G with various relations and entities; a concept extent is a set of k -tuples of entities ($k = 1, \dots, n$) and the corresponding intension is a graph pattern, i.e. a generalized graph from G that is matched by the extent tuples. A family of lattices is built from G , with a lattice for each arity k . Kötters [55] investigates the relations between power context families in PCF_M and intension graphs in IG_M , both based on the set of attribute sets M . He shows that the pattern structure $((G)^n, \text{IG}_M^n, \delta^n)$ where G is a set of objects (from a power context family), and δ^n maps a n -uplet to a windowed intension graph, allows to build a lattice that corresponds to the lattice built from an RCA process.

Generalization is widely used in Machine Learning in order to efficiently search the space of possible solutions (hypotheses) [56]. Besides, quantifiers are commonly used to query or extract knowledge from relational data. The Inductive Logic Programming (ILP) field is concerned with the development of methods for relational data mining [1]. Methods that are based on logical formalisms use the existential quantifier. Nevertheless, the structuration of the search space in ILP while using both the existential and the universal quantifier has been studied theoretically in [57]. Our work has been inspired from this one, but while ILP quantifies variables, RCA quantifies relations. Aggregates and complex aggregates [58, 59] also allow to perform quantification. They rely on aggregate functions (e.g., *max*, *count*) that can have some restrictions on the objects whose value is aggregated. Our definitions on generalization of scaling operators follow the idea of the one used for hypotheses in ILP [1] and the one used for concepts in FCA. A set of quantifiers similar to the one of RCA has been introduced a long time ago in logic and linguistics as generalized quantifiers, and are studied in the field of databases to extend SQL [60].

Many variants of FCA have been developped to deal with multivariate, interval or fuzzy or rough data [61]. Similarly, variants of RCA could be developped to deal with e.g. fuzzy relational data, but as far as we know, no work has been

done in this field. Links between FCA and rough sets (RS), when formulated between two universes, have been widely studied [62, 63]. Indeed, both theories attempt to formalize the notion of concept. In [62], the operators of FCA and of RS are compared: the author states that the set X' consists of necessary properties of an object in X while $X^\square = \{y \in M | \forall x \in G(xIy \Rightarrow x \in X)\}$ consists of sufficient properties, and an object of X has at least one property in the set $X^\diamond = \{y \in M | \exists x \in G(xIy \wedge x \in X)\}$. Object, resp. attribute oriented formal concepts can be defined by using both operators $^\square$ and $^\diamond$. The set of pairs (X, Y) where $X = Y^\diamond$ and $Y = X^\square$, resp. $X = Y^\square$ and $Y = X^\diamond$ forms a lattice, as does the set of pairs (X, Y) where $Y = X'$ and $X = Y'$ in the FCA framework.

Furthermore using scaling quantifiers to build relational attributes between objects and a concept in RCA can be related to the theory of rough sets over two universes. More precisely, let us consider G and G_r two object sets in a relational context family, and r the binary relation on $G \times G_r$. Then, the lower approximation of a concept extent $Extent(C_i) \subseteq G_r$ is $Extent(C_i)^\square = \{o \in G : r(o) \subseteq Extent(C_i)\}$, that is all the objects of G having $\exists r(C_i)$ as relational attribute; the upper approximation of $Extent(C_i)$ is: $Extent(C_i)^\diamond = \{o \in G : r(o) \cap Extent(C_i) \neq \emptyset\}$, that is all the objects of G having $\exists r(C_i)$ as relational attribute. Accordingly, the set of objects linked to a concept with a given relational attribute can be seen as an approximation of the extent of this concept. Besides, since RCA allows relations between G_r and several objects sets, which are themselves linked, several nested approximations are built for $Extent(C_i)$. Furthermore, using various scaling quantifiers can be related to probabilistic rough sets [64] or the variable precision rough set model described in [65]. Finally, our aim is to analyse the effect of these scaling quantifiers in the RCA framework, where iterative algorithms allow to build explicit hierarchies of concepts from complex relation schemas, even with cycles.

RCA has been used for analysis and reengineering of UML elements, namely in class diagrams and in use case diagrams [6]. In [66], RCA is used to exploit relations between methods and attributes to detect and fix design defects. It is

used to learn model transformations in [67]. In [68], relations between abstract tasks in an abstract orchestration are used to classify relevant Web services to instantiate the tasks. Other applications can be found in ontology engineering [69, 70]. In [14], we proposed an adaptation of RCA to explore relations in a guided way in order to increase the performance and the pertinence of the results. A variant of RCA is used in [11] for extracting rules from watercourse datasets. These applications show that RCA is useful for data analysis, and it is thus interesting to provide tools to better explain its functioning, especially when interacting with an analyst.

6. Conclusion

Relational data are rich of many information for data analysts. Among the possible ways to analyze such data, Relational Concept Analysis focuses on classifying objects of several categories taking into account their connections in an interdependent manner. It builds hierarchies of concepts, that are groups of objects (of the same category) similar with respect to their attributes and their relationships to other groups of objects (of another or the same category). The relationships between objects and groups, or among groups, are captured by relational attributes that conform to specific scaling quantifiers. The choice of scaling quantifiers by the analyst determines the kind of knowledge patterns and classifications that are learnt from the dataset.

In this paper, we have compared the scaling operators according to a generality relation. Understanding this generality relation is a necessary step for helping the data analyst to forecast the kind of results she/he may obtain and to navigate between the multiple analysis possibilities. Based on a real dataset, we have shown that the possibility to use different scaling quantifiers is relevant for extracting various information levels. To summarize, our main contributions concern:

- A formal definition of all percent scaling quantifiers used in RCA.

- A generality relation between relational attributes in RCA, based on the derived relational attributes.
- An illustrative scenario which shows how the method can be used.
- An illustration on real data, showing the effect of the chosen quantifiers on the result size.

Furthermore, this theoretical exploration is a first step to a guide for the RCA user with quantifiers. Indeed, it provides a basis for visualization tools that show the lattices and their connections, or that highlight the analysis path (summaries of the choices that are made at each step).

As future work, we would like to study the generality relation between more complex sequences of scaling operators, and to analyze the case where data contain cycles. We also plan to enhance the existing software RCAExplore for exploiting these results on generalization in RCA. In particular we are developing an interface, in order to guide the experts in the choice of a scaling operator at each step of the RCA process in an interactive way. We plan to study how anticipating the effect of scaling quantifiers on the size of lattices can guide the analysis process of various datasets. We are also studying a possible combination between RCA and propositionalization approaches and will deepen the comparison between RCA, granularity computing, and rough sets.

Acknowledgements.. This work was partially funded by ANR11_MONU14 and ONEMA. We warmly acknowledge Corinne Grac (UMR7362 LIVE - ENGEES) for advices about the dataset, and Amirouche Ouzerdine (Université de Montpellier) for preparing the dataset.

- [1] S. Džeroski, Multi-relational data mining: an introduction, ACM SIGKDD Explorations Newsletter 5 (1) (2003) 1–16.
- [2] M. R. Hacene, M. Huchard, A. Napoli, P. Valtchev, Relational concept analysis: mining concept lattices from multi-relational data, Ann. Math. Artif. Intell. 67 (1) (2013) 81–108.

- [3] U. Priss, Relational concept analysis: Semantic structures in dictionaries and lexical databases, Ph.D. thesis, Technische Universität Darmstadt (1996).
- [4] C. Nica, A. Braud, X. Dolques, F. Le Ber, M. Huchard, Exploring temporal data using relational concept analysis – an application to hydroecology, in: Concept lattices and their applications (CLA 2016), CEUR Workshop Proceedings, 2016, pp. 1–13.
- [5] C. De Maio, G. Fenza, M. Gallo, V. Loia, S. Senatore, Formal and relational concept analysis for fuzzy-based automatic semantic annotation, *Applied Intelligence* 40 (1) (2014) 154–177.
- [6] X. Dolques, M. Huchard, C. Nebut, P. Reitz, Fixing Generalization Defects in UML Use Case Diagrams, *Fundam. Inform.* 115 (4) (2012) 327–356.
- [7] L. Shi, Y. Toussaint, A. Napoli, A. Blansch  , Mining for Reengineering: An Application to Semantic Wikis Using Formal and Relational Concept Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 421–435.
- [8] W. Z. Wu, Y. Leung, J. S. Mi, Granular computing and knowledge reduction in formal contexts, *IEEE Transactions on Knowledge and Data Engineering* 21 (10) (2009) 1461–1474.
- [9] P. K. Singh, m-polar fuzzy graph representation of concept lattice, *Engineering Applications of Artificial Intelligence* 67 (Supplement C) (2018) 52–62.
- [10] G. Stumme, Efficient data mining based on formal concept analysis, in: *Database and Expert Systems Applications*, Springer, 2002, pp. 534–546.
- [11] X. Dolques, F. Le Ber, M. Huchard, C. Grac, Performance-friendly rule extraction in large water data-sets with AOC posets and relational concept analysis, *Int. J. General Systems* 45 (2) (2016) 187–210.

- [12] A. Buzmakov, S. O. Kuznetsov, A. Napoli, Is Concept Stability a Measure for Pattern Selection?, *Procedia Computer Science* 31 (2014) 918–927.
- [13] R. Belohlavek, J. Macko, Selecting important concepts using weights, in: P. Valtchev, R. Jäschke (Eds.), *Formal Concept Analysis: 9th International Conference, ICFCA 2011, Nicosia, Cyprus, May 2-6, 2011. Proceedings*, Springer Berlin Heidelberg, 2011, pp. 65–80.
- [14] X. Dolques, F. Le Ber, M. Huchard, C. Nebut, Relational Concept Analysis for Relational Data Exploration, *Advances in Knowledge Discovery and Management* 5 (2015) 55–77.
- [15] U. Priss, Formal concept analysis in information science, *ARIST* 40 (1) (2006) 521–543. doi:10.1002/aris.1440400120.
URL <https://doi.org/10.1002/aris.1440400120>
- [16] G. Birkhoff, *Lattice Theory*, Colloquium publications, American Mathematical Society, 1940.
- [17] M. Barbut, B. Monjardet, *Ordre et Classification*, Hachette, 1970.
- [18] R. Wille, Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts, *Ordered Sets*, Ivan Rival Ed., NATO Advanced Study Institute 83 (1982) 445–470.
- [19] S. Ferré, Reconciling expressivity and usability in information access - from filesystems to the semantic web, *Habilitation thesis*, Matisse, Univ. Rennes 1, habilitation à Diriger des Recherches (HDR), defended on November 6th (2014).
- [20] V. Codocedo, A. Napoli, Formal concept analysis and information retrieval - A survey, in: *Formal Concept Analysis - 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings*, 2015, pp. 61–77.
- [21] D. I. Ignatov, A. Y. Kaminskaya, N. Konstantinova, A. Malyukov, J. Poelmans, Fca-based recommender models and data analysis for crowdsourcing

- platform witology, in: Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iași, Romania, July 27-30, 2014, Proceedings, 2014, pp. 287–292.
- [22] S. O. Kuznetsov, Learning of simple conceptual graphs from positive and negative examples, in: J. M. Zytkow, J. Rauch (Eds.), Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings, Vol. 1704 of Lecture Notes in Computer Science, Springer, 1999, pp. 384–391. doi:10.1007/978-3-540-48247-5_47.
URL http://dx.doi.org/10.1007/978-3-540-48247-5_47
- [23] S. O. Kuznetsov, Machine learning and formal concept analysis, in: P. W. Eklund (Ed.), Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004, Proceedings, Vol. 2961 of Lecture Notes in Computer Science, Springer, 2004, pp. 287–312. doi:10.1007/978-3-540-24651-0_25.
URL http://dx.doi.org/10.1007/978-3-540-24651-0_25
- [24] C. Carpineto, G. Romano, Exploiting the potential of concept lattices for information retrieval with CREDO, J. UCS 10 (8) (2004) 985–1013.
- [25] C. Carpineto, G. Romano, ULYSSES: A lattice-based multiple interaction strategy retrieval interface, in: Human-Computer Interaction, 5th International Conference, EWHCI '95, Moscow, Russia, July 3-7, 1995, Selected Papers, 1995, pp. 91–104.
- [26] G. J. Greene, M. Esterhuizen, B. Fischer, Visualizing and exploring software version control repositories using interactive tag clouds over formal concept lattices, Information & Software Technology 87 (2017) 223–241. doi:10.1016/j.infsof.2016.12.001.
URL <https://doi.org/10.1016/j.infsof.2016.12.001>
- [27] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer Verlag, 1999.

- [28] S. O. Kuznetsov, M. V. Samokhin, Learning closed sets of labeled graphs for chemical applications, in: S. Kramer, B. Pfahringer (Eds.), *Inductive Logic Programming, 15th International Conference, ILP 2005, Bonn, Germany, August 10-13, 2005, Proceedings*, Vol. 3625 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 190–208. doi:10.1007/11536314_12.
URL http://dx.doi.org/10.1007/11536314_12
- [29] M. Kaytoue-Uberall, Z. Assaghir, N. Messai, A. Napoli, Two complementary classification methods for designing a concept lattice from interval data, in: S. Link, H. Prade (Eds.), *Foundations of Information and Knowledge Systems, 6th International Symposium, FoIKS 2010, Sofia, Bulgaria, February 15-19, 2010. Proceedings*, Vol. 5956 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 345–362. doi:10.1007/978-3-642-11829-6_22.
URL http://dx.doi.org/10.1007/978-3-642-11829-6_22
- [30] A. Bertaux, F. L. Ber, A. Braud, M. Trémolières, Identifying Ecological Traits: A Concrete) FCA-Based Approach, in: S. Ferré, S. Rudolph (Eds.), *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*, Vol. 5548 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 224–236. doi:10.1007/978-3-642-01815-2_17.
URL http://dx.doi.org/10.1007/978-3-642-01815-2_17
- [31] R. Belohlávek, Fuzzy galois connections, *Math. Log. Q.* 45 (1999) 497–504. doi:10.1002/malq.19990450408.
URL <http://dx.doi.org/10.1002/malq.19990450408>
- [32] J. Konecny, P. Osicka, Triadic concept lattices in the framework of aggregation structures, *Inf. Sci.* 279 (2014) 512–527.
- [33] D. I. Ignatov, D. V. Gnatyshak, S. O. Kuznetsov, B. G. Mirkin, Triadic formal concept analysis and triclustering: searching for optimal patterns, *Machine Learning* 101 (1-3) (2015) 271–302.

- [34] B. Ganter, S. O. Kuznetsov, Pattern structures and their projections, in: ICCS'01, Stanford, CA, USA, 2001, pp. 129–142.
- [35] S. O. Kuznetsov, S. A. Obiedkov, Comparing performance of algorithms for generating concept lattices, *J. Exp. Theor. Artif. Intell.* 14 (2-3) (2002) 189–216.
- [36] S. O. Kuznetsov, T. P. Makhalova, On interestingness measures of formal concepts, *Inf. Sci.* 442-443 (2018) 202–219.
- [37] T. Tilley, R. Cole, P. Becker, P. W. Eklund, A survey of formal concept analysis support for software engineering activities, in: *Formal Concept Analysis, Foundations and Applications*, 2005, pp. 250–271.
- [38] R. Osswald, W. Pedersen, Induction of Classifications from Linguistic Data, in: *Proc. of ECAI'02 Workshop*, 2002.
- [39] G. J. Greene, B. Fischer, Cvexplorer: identifying candidate developers by mining and exploring their open source contributions, in: *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, ASE 2016, Singapore, September 3-7, 2016*, 2016, pp. 804–809.
- [40] K. Domdouzis, B. Akhgar, S. Andrews, H. Gibson, L. Hirsch, A social media and crowdsourcing data mining system for crime prevention during and post-crisis situations, *J. Systems and IT* 18 (4) (2016) 364–382.
- [41] S. Guillas, K. Bertet, Ogier, A generic description of the concept lattices' classifier: Application to symbol recognition, in: W. Liu, J. Lladós (Eds.), *GREC 2005, LNCS 3926*, 2005, pp. 47–60.
- [42] A. Miralles, G. Molla, M. Huchard, C. Nebut, L. Deruelle, M. Derras, Class model normalization - outperforming formal concept analysis approaches with aoc-posets, in: *Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications, Clermont-Ferrand, France, October 13-16, 2015*, 2015, pp. 111–122.
URL <http://ceur-ws.org/Vol-1466/paper09.pdf>

- [43] M. R. Hacene, M. Huchard, A. Napoli, P. Valtchev, Soundness and completeness of relational concept analysis, in: ICFCA 2013, Dresden, Germany, LNCS 7880, 2013, pp. 228–243.
- [44] S. Kuznetsov, S. Obědkov, Comparing Performance of Algorithms for Generating Concept Lattices, in: Proceedings of Int. Workshop on Concept Lattice-based Theory, Methods and Tools for Knowledge Discovery in Databases, 2001.
- [45] I. Haviv, O. Regev, On the Lattice Isomorphism Problem, CoRR abs/1311.0366. arXiv:1311.0366.
URL <http://arxiv.org/abs/1311.0366>
- [46] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, Computing iceberg concept lattices with Titanic, Data & Knowledge Engineering 42 (2) (2002) 189 – 222.
- [47] L. Berrahou, N. Lalande, E. Serrano, G. Molla, L. Berti-Équille, S. Bimonte, S. Bringay, F. Cernesson, C. Grac, D. Ienco, F. Le Ber, M. Teisseire, A quality-aware spatial data warehouse for querying hydroecological data, Computers & Geosciences 85, Part A (2015) 126–135.
- [48] M. Liquière, J. Sallantin, Structural Machine Learning with Galois Lattice and Graphs, in: ICML, Madison, Wisconsin, 1998, pp. 305–313.
- [49] S. Prediger, R. Wille, The Lattice of Concept Graphs of a Relationally Scaled Context, in: ICCS’99, Blacksburg, Virginia, LNCS 1640, Springer, 1999, pp. 401–414.
- [50] S. Ferré, O. Ridoux, B. Sigonneau, Arbitrary Relations in Formal Concept Analysis and Logical Information Systems, in: ICCS’05, Kassel, Germany, LNAI 3596, Springer, 2005, pp. 166–180.
- [51] F. Baader, F. Distel, A finite basis for the set of \mathcal{EL} -implications holding in a finite model, in: ICFCA, LNCS 4933, 2008, pp. 46–61.

- [52] M. Krmelova, M. Trnecka, Boolean Factor Analysis of Multi-Relational Data, in: CLA 2013, La Rochelle, France, CEUR Workshop Proc. 1062, 2013, pp. 187–198.
- [53] J. Kötters, Concept Lattices of a Relational Structure, in: ICCS 2013, Mumbai, India, LNCS 7735, 2013, pp. 301–310.
- [54] S. Ferré, A Proposal for Extending Formal Concept Analysis to Knowledge Graphs, in: Formal Concept Analysis, ICFCA 2015, Vol. LNCS 9113, Nerja, Spain, 2015, pp. 271–286.
- [55] J. Kötters, Intension Graphs as Patterns over Power Context Families, in: Concept Lattices and Their Applications, CLA 2016, CEUR Workshop Proceedings 1624, Moscow, Russia, July 18-22, 2016, 2016, pp. 203–216.
- [56] T. M. Mitchell, Generalization as search, *Artif. Intell.* 18 (2) (1982) 203–226.
- [57] M.-E. Goncalves, Handling Quantifiers in ILP, in: Selected Papers from ILP '96, Springer-Verlag, 1997, pp. 337–357.
- [58] C. Vens, J. Ramon, H. Blockeel, Refining aggregate conditions in relational learning, in: Knowledge Discovery in Databases: PKDD 2006, LNCS 4213, Springer, 2006, pp. 383–394. doi:10.1007/11871637_37.
- [59] C. Charnay, N. Lachiche, A. Braud, Incremental construction of complex aggregates: Counting over a secondary table, in: LBP of ILP 2013, 2013, pp. 1–6.
- [60] A. Badia, B. Cao, Efficient implementation of generalized quantification in relational query languages, *Proc. VLDB Endow.* 6 (4) (2013) 241–252.
- [61] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, G. Dedene, Fuzzy and rough formal concept analysis: a survey, *Int. J. General Systems* 43 (2) (2014) 105–134. doi:10.1080/03081079.2013.862377.
URL <https://doi.org/10.1080/03081079.2013.862377>

- [62] Y. Yao, A comparative study of formal concept analysis and rough set theory in data analysis, in: *Rough Sets and Current Trends in Computing*, Vol. LNCS 3066, 2004, pp. 59–68.
- [63] J. Atif, I. Bloch, C. Hudelot, Some relationships between fuzzy sets, mathematical morphology, rough sets, f-transforms, and formal concept analysis, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 24 (2) (2016) 1–32.
- [64] W. Ma, B. Sun, Probabilistic rough set over two universes and rough entropy, *International Journal of Approximate Reasoning* 53 (4) (2012) 608 – 619.
- [65] Y. Shen, F. Wang, Variable precision rough set model over two universes and its properties, *Soft Computing* 15 (3) (2011) 557–567.
- [66] N. Moha, A. R. Hacene, P. Valtchev, Y.-G. Guéhéneuc, Refactorings of Design Defects Using Relational Concept Analysis, in: *ICFCA 2008*, 2008, pp. 289–304.
- [67] H. Saada, X. Dolques, M. Huchard, C. Nebut, H. A. Sahraoui, Generation of Operational Transformation Rules from Examples of Model Transformations, in: *MoDELS 2012*, 2012, pp. 546–561.
- [68] Z. Azmeh, M. Driss, F. Hamoui, M. Huchard, N. Moha, C. Tibermacine, Selection of Composable Web Services Driven by User Requirements, in: *ICWS 2011*, 2011, pp. 395–402.
- [69] R. Bendaoud, A. Napoli, Y. Toussaint, Formal Concept Analysis: A unified framework for building and refining ontologies, in: *EKAU 2008*, LNCS 5268, 2008, pp. 156–171.
- [70] M. Rouane-Hacene, P. Valtchev, R. Nkambou, Supporting Ontology Design through Large-Scale FCA-Based Ontology Restructuring, in: *ICCS 2011*, 2011, pp. 257–269.