Ann Math Artif Intell (2014) 72:151–168
DOI 10.1007/s10472-014-9411-0

Generalized pattern extraction from concept lattices

Léonard Kwuida · Rokia Missaoui ·
Abdélilah Balamane · Jean Vaillancourt

Published online: 23 March 2014

Abstract In this paper, we show how the existence of taxonomies on objects and/or attributes can be used in Formal Concept Analysis to help discover *generalized* concepts. To that end, we analyze three generalization cases (∃, ∀, and α) and present different scenarios of a simultaneous generalization on both objects and attributes. We also discuss the cardinality of the generalized pattern set against the number of simple patterns produced from the initial data set.

Keywords Concept lattices · Taxonomy · Generalization · Formal Concept Analysis

Mathematics Subject Classifications (2010) 68T30 · 06A15

1 Introduction

In many real-life applications and research trends in Computer Science, the semantics of data can be advantageously exploited to better retrieve and efficiently manage information and discover unexpected and relevant patterns which are a concise and semantically rich representation of data. Patterns can be clusters, concepts, association rules, outliers, and so on. In this work we present alternate ways to abstract or group objects such as communities

This paper is an extended version of [22]

L. Kwuida
Department of Business, Bern University of Applied Sciences, Bern, Switzerland
e-mail: leonard.kwuida@bfh.ch

R. Missaoui (✉) · A. Balamane · J. Vaillancourt
Université du Québec en Outaouais, Gatineau, Canada
e-mail: rokia.missaoui@uqo.ca

A. Balamane
e-mail: bala04@uqo.ca

J. Vaillancourt
e-mail: jean.vaillancourt@uqo.ca

of individuals and/or attributes like geographical locations at different levels of granularity to get generalized concepts by using taxonomies on attributes and/or objects.

Formal Concept Analysis (FCA) is a formalism for knowledge representation which is based on the formalization of "concepts" and "concept hierarchies" [16]. One recurrent problem in FCA is the number of concepts that can be exponential in the size of the context. To control the size of the context and the corresponding concept lattice many techniques have been proposed [5]. One of them is to show the iceberg lattice with frequent concepts only rather than the whole lattice [29]. Other techniques aim to produce reduced lattices through selection on objects and/or projection on attributes [20] or by exploiting a taxonomy on attributes or objects. Beside the ability to potentially reduce the size of the lattice, the generalization on attributes or objects can lead to new meaningful and possibly unexpected generalized patterns as illustrated in Section 6.

The rest of this contribution is organized as follows. In Section 2 we introduce the basic notions of FCA. Section 3 presents three different generalization schemes, presents different scenarios of generalizing both objects and attributes. It also discusses the visualization issue of generalized patterns and provides the real meaning of the three generalization cases. In Section 4 the size of the generalized concept set is compared to the size of the initial concept set (i.e., before generalization). Existing work about combining FCA with ontology is briefly described in Section 5. Finally, an empirical study about the link between the size of a lattice before and after an $\exists$ or α generalization is given in Section 6.

2 Formal Concept Analysis and data mining

2.1 Elementary information systems, contexts and concepts

In Formal Concept Analysis, a context is a triple $\mathbb{K} := (G, M, \mathrm{I})$ where G, M and I stand for a set of objects, a set of attributes, and a binary relation between G and M respectively. A formal concept is a pair (A, B) such that B is exactly the set of all properties shared by the objects in A and A is the set of all objects that have all the properties in B. We set $A' := \{m \in M \mid a\mathrm{I}m \text{ for all } a \in A\}$ and $B' := \{g \in G \mid g\mathrm{I}b \text{ for all } b \in B\}$. Then (A, B) is a concept of $\mathbb{K}$ iff $A' = B$ and $B' = A$. The extent of the concept (A, B) is A (also denoted by $\mathrm{ext}(A, B)$) while its intent is B. We denote by $\mathfrak{B}(\mathbb{K})$, $\mathrm{Int}(\mathbb{K})$ and $\mathrm{Ext}(\mathbb{K})$ the set of concepts, intents and extents of $\mathbb{K}$, respectively. A subset X is closed if $X'' = X$. Closed subsets of G are exactly extents while closed subsets of M are intents of $\mathbb{K}$. Fig. 1 describes

$\mathbb{K}$	a	b	c	d	e	f	g	h
1	×				×		×	
2	×				×	×		×
3	×	×			×	×	×	
4		×			×	×	×	×
5	×		×	×				
6	×	×	×	×				
7		×	×				×	
8		×	×	×			×	

Fig. 1 A formal context

items $a, \dots, h$ that appear in eight transactions (customers) of a *basket market analysis* application. Such a setting defines a binary relation I between the set G of transactions and the set M of items.

The concept hierarchy is formalized with a relation $\leq$ defined on $\mathfrak{B}(\mathbb{K})$ by $A \subseteq C \iff: (A, B) \leq (C, D) :\iff B \supseteq D$. This is an order relation and is also called a *specialization/generalization* relation on concepts. In fact, a concept (A, B) is a specialization of a concept (C, D), or (C, D) is a generalization of (A, B) iff $(A, B) \leq (C, D)$ holds. For any list $\mathcal{C}$ of concepts of $\mathbb{K}$, there is a concept $\mathfrak{u}$ of $\mathbb{K}$ which is more general than every concept in $\mathcal{C}$ and is a specialization of every generalization of all concepts in $\mathcal{C}$ ($\mathfrak{u}$ is the *supremum* of $\mathcal{C}$ and is denoted by $\bigvee \mathcal{C}$), and there is a concept $\mathfrak{v}$ of $\mathbb{K}$ which is a specialization of every concept in $\mathcal{C}$ and a generalization of every specialization of all concepts in $\mathcal{C}$ ($\mathfrak{v}$ is the *infimum* of $\mathcal{C}$ and is denoted by $\bigwedge \mathcal{C}$).[1] Hence, $\mathfrak{B}(\mathbb{K})$ is a *complete lattice* called the concept lattice of the context $\mathbb{K}$.

For $g \in G$ and $m \in M$ we set $g' := \{g\}'$ and $m' := \{m\}'$. The object concepts $(\gamma g := \big(g'', g'\big))_{g \in G}$ and the attribute concepts $(\mu m := \big(m', m''\big))_{m \in M}$ form the "building blocks" of $\mathfrak{B}(\mathbb{K})$. In fact, every concept of $\mathbb{K}$ is a supremum of some γg's and infimum of some μm's.[2]

The size of a concept lattice can be extremely large, even exponential in the size of the context. To handle such large sets of concepts many techniques have been proposed [16], based on context decomposition or lattice pruning/reduction (atlas decomposition, direct or subdirect decomposition, iceberg concept lattices, nested line diagrams, ...). We believe that using taxonomies on objects and attributes can contribute to the extraction of unexpected and relevant generalized patterns and in most cases to the reduction of the size of discovered patterns.

2.2 Labeled line diagrams of concept lattices

One of the strengths of FCA is the ability to pictorially display knowledge, at least for contexts of reasonable size. Finite concept lattices can be represented by reduced labeled Hasse diagrams (see Fig. 2). Each node represents a concept. The label g is written below γg and m above μm. The extent of a concept represented by a node a is given by all labels in G from the node a downwards, and the intent by all labels in M from a upwards. For example, the label 5 in the left side of Fig. 2 represents the object concept $\gamma 5 = (\{5, 6\}, \{a, c, d\})$. Diagrams are valuable tools for visualizing data. However drawing a good diagram for complex structures is a big challenge [7]. Therefore, we need tools to abstract the output by reducing the size of the input, making the structure nicer, or by exploring the diagram layer by layer. For the last case, FCA offers nested line diagrams as a means to visualize the concepts level-wise [16]. Before we move to generalized patterns, let us see how data are transformed into binary contexts, the suitable format for our data.

2.3 Scaling

Frequently, data are not directly encoded in a "binary" form, but rather as a many-valued context, i.e., a tuple (G, M, W, I) such that G is the set of objects, M the set of attribute

[1] For two concepts x_1 and x_2 we set $x_1 \vee x_2 := \bigvee\{x_1, x_2\}$ and $x_1 \wedge x_2 := \bigwedge\{x_1, x_2\}$.

[2] For $(A, B) \in \mathfrak{B}(G, M, \mathrm{I})$ we have $\bigvee_{g \in A} \gamma g = (A, B) = \bigwedge_{m \in B} \mu m$.

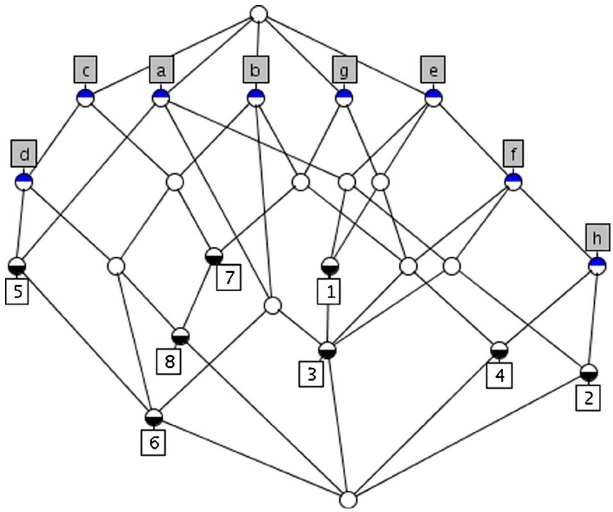

Fig. 2 Concept lattice of the context in Fig. 1

names, W the set of attribute values, $I \subseteq G \times M \times W$ and every $m \in M$ is a partial map from G to W with $(g, m, w) \in \mathrm{I}$ iff $m(g) = w$. Many-valued contexts can be transformed into binary contexts via conceptual scaling. A conceptual scale for an attribute m of (G, M, W, I) is a binary context $\mathbb{S}_m := (G_m, M_m, \mathrm{I}_m)$ such that $m(G) \subseteq G_m$. Intuitively, M_m discretizes or groups the attribute values into $m(G)$, and I_m describes how each attribute value $m(g)$ is related to the elements in M_m. For an attribute m of (G, M, W, I) and a conceptual scale $\mathbb{S}_m$ we derive a binary context $\mathbb{K}_m := (G, M_m, \mathrm{I}^m)$ with $g\, \mathrm{I}^m\, s_m \ :\iff\ m(g)\, \mathrm{I}_m\, s_m$, where $s_m \in M_m$. This means that an object $g \in G$ is in relation with a scaled attribute s_m iff the value of m on g is in relation with s_m in $\mathbb{S}_m$. With a conceptual scale for each attribute we get the derived context $\mathbb{K}^S := (G, N, \mathrm{I}^S)$ where $N := \bigcup\{M_m \mid m \in M\}$ and $g\, \mathrm{I}^S\, s_m \iff m(g)\, \mathrm{I}^m\, s_m$. In practice, the set of objects remains unchanged; each attribute name m is replaced by the scaled attributes $s_m \in M_m$. The choice of a suitable set of scales depends on the interpretation, and is usually done with the help of a domain expert. A *Conceptual Information System* is a many-valued context together with a set of conceptual scales [25, 27]. The methods presented in Section 3 are actually a form of scaling.

3 Generalized patterns

In the field of data mining, generalized patterns are pieces of knowledge extracted from data when an ontology is used. In the following we formalize the way generalized patterns are produced. Let $\mathbb{K} := (G, M, \mathrm{I})$ be a context. The attributes of $\mathbb{K}$ can be grouped together to form another set S of attributes. For the basket market analysis example, items/products can be generalized into product lines and then product categories, and customers may be generalized to groups according to some specific features (e.g., income, education). The context $\mathbb{K}$ is then replaced with a context (G, S, J) as in the scaling process where S can be seen as an index set such that $\{m_s \mid s \in S\}$ covers M. We will usually identify the group m_s with the index s.

3.1 Types of generalization

There are mainly three ways to express the relation J (see Table 1):

(∃) $g \,\mathrm{J}\, s :\iff \exists m \in s,\; g \,\mathrm{I}\, m$. Consider an information table describing companies and their branches in USA. We first set up a context whose objects are companies and whose attributes are the cities where these companies have branches. If there are too many cities, we can decide to group them into states to reduce the number of attributes. Then, the (new) set of attributes is now a set S whose elements are states. It is quite natural to assert that a company g has a branch in a state s if g has a branch in a city m which belongs to the state s.

(∀) $g \,\mathrm{J}\, s :\iff \forall m \in s,\; g \,\mathrm{I}\, m$. Consider an information system about Ph.D. students and the components of the comprehensive exam (CE). Assume that components are: the written part, the oral part, and the thesis proposal, and a student succeeds in his exam if he succeeds in the three components of that exam. The objects of the context are Ph.D. students and the attributes are the different exams taken by students. If we group together the different components, for example

$$CE.written, CE.oral, CE.proposal \mapsto CE.exam,$$

then it becomes natural to state that a student g succeeds in his comprehensive exam $CE.exam$ if he succeeds in *all* the exam parts of CE.

(α%) $g \,\mathrm{J}\, s :\iff \frac{|\{m \in s \mid g \,\mathrm{I}\, m\}|}{|s|} \geq \alpha_s$ where α_s is a threshold set by the user for the generalized attribute s. This case generalizes the (∃)-case with $\alpha = \frac{1}{|M|}$ and the (∀)-case with $\alpha = 1$. To illustrate this case, let us consider a context describing different specializations in a given Master degree program. For each program there is a set of mandatory courses and a set of optional ones. Moreover, there is a predefined number of courses that a student should succeed to get a degree in a given specialization. Assume that to get a Master in Computer Science with a specialization in "computational logic", a student must succeed seven courses from a set s_1 of mandatory courses and three courses from a set s_2 of optional ones. Then, we can introduce two generalized attributes s_1 and s_2 so that a student g succeeds in the group s_1 if he succeeds in at least seven courses from s_1, and succeeds in s_2 if he succeeds in at least three courses from s_2. So, $\alpha_{s_1} := \frac{7}{|s_1|}$, $\alpha_{s_2} := \frac{3}{|s_2|}$, and

$$g \,\mathrm{J}\, s_i \iff \frac{|\{m \in s_i \mid g \,\mathrm{I}\, m\}|}{|s_i|} \geq \alpha_{s_i},\; 1 \leq i \leq 2.$$

Table 1 Three generalizations of the context in Fig. 1

	Initial context								∃-generalization				∀-generalization				α-generalization		
	a	b	c	d	e	f	g	h	A	B	C	D	S	T	U	V	E	F	H
1	×				×		×		×		×		×						
2	×				×	×		×	×		×	×				×		×	
3	×	×			×	×	×		×	×	×	×	×				×	×	
4		×			×	×	×	×	×	×		×	×			×		×	×
5	×		×	×						×	×				×		×		
6	×	×	×	×						×	×			×	×		×		
7		×	×					×		×	×			×			×		
8		×	×	×			×		×	×	×			×			×		

The ∃-generalized attributes are $A := \{e, g\}$, $B := \{b, c\}$, $C := \{a, d\}$ and $D := \{f, h\}$. The ∀-generalized attributes are $S := \{e, g\}$, $T := \{b, c\}$, $U := \{a, d\}$ and $V := \{f, h\}$. The α-generalized attributes are $E := \{a, b, c\}$, $F := \{d, e, f\}$ and $H := \{g, h\}$ with $\alpha = 60\ \%$

Attribute generalization reduces the number of attributes in the partition case. One may therefore expect a reduction in the number of concepts. Unfortunately, this is not always the case. Therefore, it is interesting to investigate under which condition generalizing patterns leads to a "generalized" lattice of smaller size than the initial one. Moreover, finding the connections between the implications and more generally association rules of the generalized context and the initial one is also an important problem to be considered in the future.

If data represent customers (transactions) and items (products), the usage of a taxonomy on attributes leads to new useful patterns that could not be seen before generalizing attributes. For example, the ∃-case (see Fig. 3, left) helps the user acquire the following knowledge:

- Customer 3 (at the bottom of the lattice) buys at least one item from each product line
- Whenever a customer buys at least one item from the product line D, then he/she necessarily buys at least one item from the product line A.

From the ∀-case in Fig. 3 (middle), one may learn for example that Customers 4 and 6 have distinct behaviors in the sense that the former buys all the items of the product lines V and S while the latter purchases all the items of the product lines U and T.

An illustration of the α-case is shown in Fig. 3 (right). One can learn that any customer who buys at least 60 % of items in H necessarily purchases at least 60 % of items in F. Moreover, the product line E (respectively H) seems to be the most (resp. the least) popular among the four product lines since five out of eight customers (resp. only one customer) bought at least 60 % of items in E (resp. H). The fact that Customer 1 appears at the supremum of the lattice means that he buys less than 60 % of items in each one of the product lines. Note that if all groups in an α-case have two elements, then any α-generalization would be either an ∃-generalization ($\alpha \leq 0.5$) or a ∀-generalization ($\alpha > 0.5$).

Generalization can also be conducted on objects to replace some (or all) of them with generalized objects, or even more, can be done simultaneously on objects and attributes.

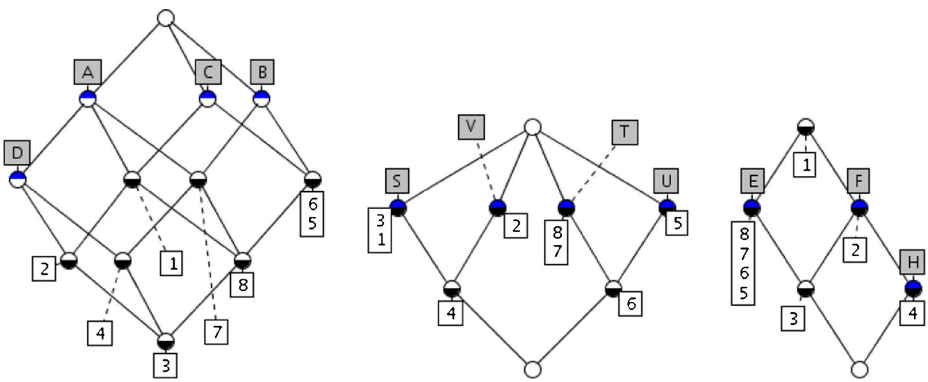

Fig. 3 Concept lattices of generalized contexts in Table 1. ∃-generalization (*left*), ∀-generalization (*middle*) and α-generalization with $\alpha := 60$ % (*right*)

3.2 Generalization on objects and attributes

Done simultaneously on attributes and on objects, the generalization will give a kind of *hypercontext* (similar to hypergraphs [4]), since the objects are subsets of G and attributes are subsets of M. Let A be a group of objects and B a group attributes in $\mathbb{K}$. A relation J between groups of objects and groups of attributes can be defined using one or a combination of the following cases:

(1) $A \, \mathrm{J}_1 \, B$ iff $\exists a \in A, \exists b \in B$ such that $a \, \mathrm{I} \, b$, i.e. some objects from the group A are in relation with some attributes in the group B; i.e. $A \times B \cap \mathrm{I} \neq \emptyset$. This is the weakest condition one can impose on A and B, and is probably not useful in real life situations. Its dual, J_1^d, is defined by: $A \, \mathrm{J}_1^d \, B$ iff $\forall a \in A, \forall b \in B \; a \, \mathrm{I} \, b$, i.e. $A \times B \cap \mathrm{I} = A \times B$. The following statements are equivalent: (a) $A \, \mathrm{J}_1^d \, B$, (b) $A \times B \subseteq I$, (c) $A \subseteq B'$, (d) $B \subseteq A'$, and (e) (A, B) is a preconcept [32].

(2) $A \, \mathrm{J}_2 \, B$ iff $\forall a \in A, \exists b \in B$ such that $a \, \mathrm{I} \, b$, i.e. every object in the group A has at least one attribute from the group B; Thus $A \, \mathrm{J}_2 \, B$ iff $A \subseteq \bigcup \{b' \mid b \in B\}$. The dual, J_2^d is defined by: $A \, \mathrm{J}_2^d \, B$ iff $B \subseteq \bigcup \{a' \mid a \in A\}$. Note that

$$\begin{cases} A \, \mathrm{J}_2 \, B \implies |A \times B \cap \mathrm{I}| \geq |A| \\ A \, \mathrm{J}_2^d \, B \implies |A \times B \cap \mathrm{I}| \geq |B|. \end{cases} \quad \text{The converse is not true.}$$

(3) $A \, \mathrm{J}_3 \, B$ iff $\exists b \in B$ such that $\forall a \in A \; a \, \mathrm{I} \, b$, i.e. there is an attribute in the group B that belongs to all objects of the group A. (b is a kind of *federating* attribute for the objects in A.) Thus $A \, \mathrm{J}_3 \, B$ iff $B \cap A' \neq \emptyset$. Its dual, J_3^d is defined by: $A \, \mathrm{J}_3^d \, B$ iff $A \cap B' \neq \emptyset$. Moreover,

$$\begin{cases} A \, \mathrm{J}_3 \, B \implies |A \times B \cap \mathrm{I}| \geq |A| \\ A \, \mathrm{J}_3^d \, B \implies |A \times B \cap \mathrm{I}| \geq |B|. \end{cases} \quad \text{The converse is not true.}$$

(4) $A \, \mathrm{J}_4 \, B$ iff $\frac{\left|\{a \in A \mid \frac{|a' \cap B|}{|B|} \geq \beta_B\}\right|}{|A|} \geq \alpha_A$, i.e. at least α_A fraction of objects in the group A have each at least β_B fraction of the attributes in the group B. Setting $\mathcal{A}_{\beta_B} := \{a \in A \mid |a' \cap B| \geq |B|\beta_B\}$, we get $A \, \mathrm{J}_4 \, B$ iff $|\mathcal{A}_{\beta_B}| \geq \alpha_A |A|$. Thus $A \, \mathrm{J}_4 \, B$ implies

$$|A \times B \cap \mathrm{I}| = \sum_{a \in A} |a' \cap B| \geq \sum_{a \in \mathcal{A}_{\beta_B}} |a' \cap B| \geq |\mathcal{A}_{\beta_B}||B|\beta_B \geq \alpha_A |A||B|\beta_B.$$

i.e. the density of $A \times B$ is at least $\alpha_A \beta_B$. The dual of J_4 is defined by: $A \, \mathrm{J}_4^d \, B$ iff $\left|\left\{b \in B \mid \frac{|b' \cap A|}{|A|} \geq \alpha_A\right\}\right| \geq |B| \cdot \beta_B$. Similarly, $A \, \mathrm{J}_4^d \, B \implies \frac{|A \times B \cap \mathrm{I}|}{|A \times B|} \geq \alpha_A \beta_B$.

(5) $A \, \mathrm{J}_5 \, B$ iff $\frac{|A \times B \cap \mathrm{I}|}{|A \times B|} \geq \alpha$, i.e. the density of the rectangle $A \times B$ is at least α. All the cases above satisfy J_5 for some given α, but these cases cannot be recovered just by setting a value for α.

Remark 1 One particular annoying situation that we could encounter is when we have to declare (apart from J_1^d) that a group A of objects is in relation with a group B of attributes although some objects in A have none of the attributes in B or some attributes of B are satisfied by none of the objects in A. To avoid this situation we could require that all relations we defined on generalized objects/attributes should be a subrelation of $\tilde{\mathrm{J}}_2 := \mathrm{J}_2 \cap \mathrm{J}_2^d$. This means that $A\tilde{\mathrm{J}}_2 B$ iff $A \times B$ contains no empty row and no empty column. Thus $A\tilde{\mathrm{J}}_2 B$ iff $A \subseteq \bigcup_{b\in B} b'$ and $B \subseteq \bigcup_{a\in A} a'$.

Remark 2 The relations J_4 and J_4^d are enough to generalize $\mathrm{J}_1, \mathrm{J}_2, \mathrm{J}_3$ and their dual. Moreover, if $A\,\mathrm{J}_4\,B$ or $A\,\mathrm{J}_4^d\,B$ then $A\,\mathrm{J}_5\,B$ for every $\alpha \leq \alpha_A\beta_B$.

An example of generalization on both objects and attributes would be one of customers grouped according to common features and items grouped into product lines. We can also assign to each group all items bought by their members (an ∃-generalization) or only their common items (a ∀-generalization), or just some of the frequent items among their members (similar to an α-generalization). We could also decide, as in $\tilde{\mathrm{J}}_2$, to assign a product line B to a group of customers A only if every customer from A buys at least one product in line B and every product from line B is bought by at least one customer in A.

Remark 3 In [30] the authors discuss α-lattices, an approach to reduce the size of the concept lattice by grouping the attributes and/or objects and defining an α-satisfaction $\models_\alpha$ between objects or groups of objects and groups of attributes. The starting point was to state that the extent of the attribute concept μm_s in the new context (G, S, J) is equal to the extent of the concept generated by m_s in the initial context; i.e., $m_s^{\mathrm{J}} = m_s' = \bigcap_{m\in m_s} m'$. This corresponds to the ∀-generalization as presented in Section 3.1. To generalize this case, they define an α-satisfaction (with $\alpha \in [0, 1]$) between a set of objects A and a set of attributes B by: $A \models_\alpha B \iff |B' \cap A| \geq \alpha|A|$. This is a generalization of J_3^d, since instead of just requiring $B' \cap A$ to be non-empty, they set a threshold α for the proportion of $B' \cap A$ in A. To bring this α-satisfaction to the object level, they assumed that the grouping of objects form a partition (then each object g belongs to a unique group g_o) and state that $g \models_\alpha B \iff g \in B'$ and $|g_o \cap B'| \geq \alpha|g_o|$; i.e., g should satisfy all properties in B and the proportion of objects in g_o satisfying all attributes in B is at least equal to α. This is a bit different from the α-case as presented in the present paper.

3.3 Visualizing generalized patterns on line diagrams

Let $\mathbb{K}$ be a formal context and (G, S, J) a context obtained from $\mathbb{K}$ via a generalization on attributes. The usual action is to directly construct a line diagram of $\mathfrak{B}(G, S, \mathrm{J})$. (See Fig. 3). One may be interested in refining the line diagram of $\mathfrak{B}(G, S, \mathrm{J})$ further to the attributes in M, for example in order to recover the concept lattice of $\mathbb{K}$.

If storage space is not a constraint, the attributes in M and the generalized attributes can be kept all together. This is done using the apposition of (G, M, I) and (G, S, J) to get $(G, M \cup S, \mathrm{I} \cup \mathrm{J})$.

A nested line diagram [16] can be used to display the resulting lattice, with (G, S, J) at the first level and $\mathbb{K}$ at the second one, i.e., a line diagram is constructed for $\mathfrak{B}(G, S, \mathrm{J})$ with nodes large enough to contain copies of the line diagram of $\mathfrak{B}(\mathbb{K})$. This is probably here not so interesting since each node of $\mathfrak{B}(G, S, \mathrm{J})$ will contain a copy of $\mathfrak{B}(\mathbb{K})$, which is probably already large enough.

An alternate way to visualize generalized concepts is to conduct a projection of $(G, M \cup S, \mathrm{I} \cup \mathrm{J})$ on the generalized attributes in S and attach to each node of $\mathfrak{B}(G, M \cup S, \mathrm{J})$ its equivalence class. Here, two nodes are equivalent iff their intents have the same restriction on S [20]. For our example of ∀-generalization, we display in Fig. 4 the projection of

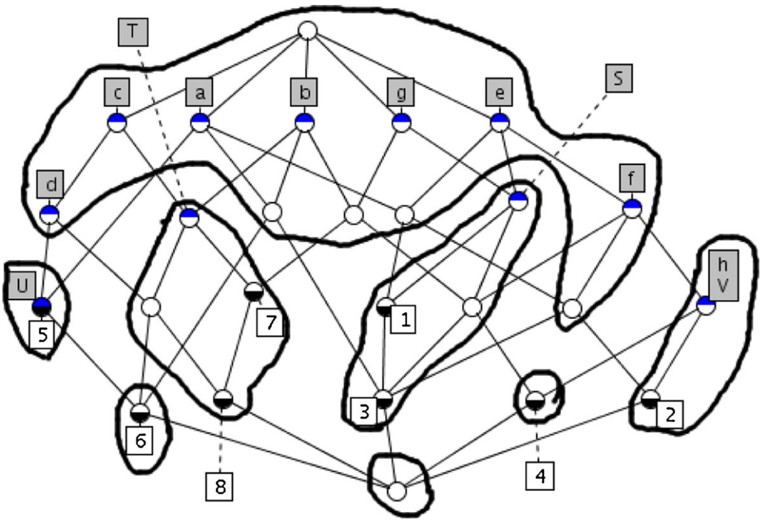

Fig. 4 Projection of the lattice in Fig. 2 onto the ∀-generalized attributes

$\mathfrak{B}(G, M \cup S, \mathrm{I} \cup \mathrm{J})$ on S by marking the equivalence classes on $\mathfrak{B}(G, M \cup S, \mathrm{I} \cup \mathrm{J})$. This is a refinement of the lattice in the middle of Fig. 3.

3.4 Are generalized attributes really generalizations?

For attributes $a, b \in M \cup S$, we should normally assert that a is a generalization of b (or b is a specialization of a) whenever more objects satisfy a than b, i.e. $\mu a \geq \mu b$.

For the ∃-case we have, $m_s' = \bigcup\{m' \mid m \in m_s\}$. Thus, $\mu m_s \geq \mu m$ for all $m \in m_s$; i.e. m_s really *generalizes* every attribute $m \in m_s$.

For the ∀-case we have, $m_s' = \bigcap\{m' \mid m \in m_s\}$. Thus, $\mu m_s \leq \mu m,\ \forall m \in m_s$; i.e. m_s *specializes* every attribute $m \in m_s$.

For the α-case, $g\,\mathrm{J}\,m_s$ iff $\alpha \leq \frac{|\{m \in m_s \mid g\,\mathrm{I}\,m\}|}{|m_s|}$. For the α-generalization in Table 1 the concept lattice of $(G, M \cup S, \mathrm{I} \cup \mathrm{J})$ is depicted in Fig. 5 and shows that:

- There is a generalized attribute $m_s \in S$ with at least one attribute $m \in m_s$ such that $\mu m_s \not\leq \mu m$ in $\mathfrak{B}(G, M \cup S, I \cup J)$; i.e μm_s is not a *specialization* of μm. E.g., $m_s := E = \{a, b, c\}$ and $m = b$.
- There is a generalized attribute $m_s \in S$ with at least one attribute $m \in m_s$ such that $\mu m \not\leq \mu m_s$ in $\mathfrak{B}(G, M \cup S, I \cup J)$; i.e μm_s is not a *generalization* of μm. E.g., $m_s := E = \{a, b, c\}$ and $m = b$.

Therefore, in α-case, there are generalized attributes m_s that are neither a *generalization* of the m's nor a *specialization* of the m's. Thus, we should better call the α-case an attribute *approximation*, the ∀-case a *specialization* and only the ∃-case a *generalization*.

4 Controlling the size of generalized concepts

A generalized concept is a concept whose intent (or extent) contains generalized attributes (or objects). Fig. 6 displays an ∃-generalization that leads to a larger number of concepts. In

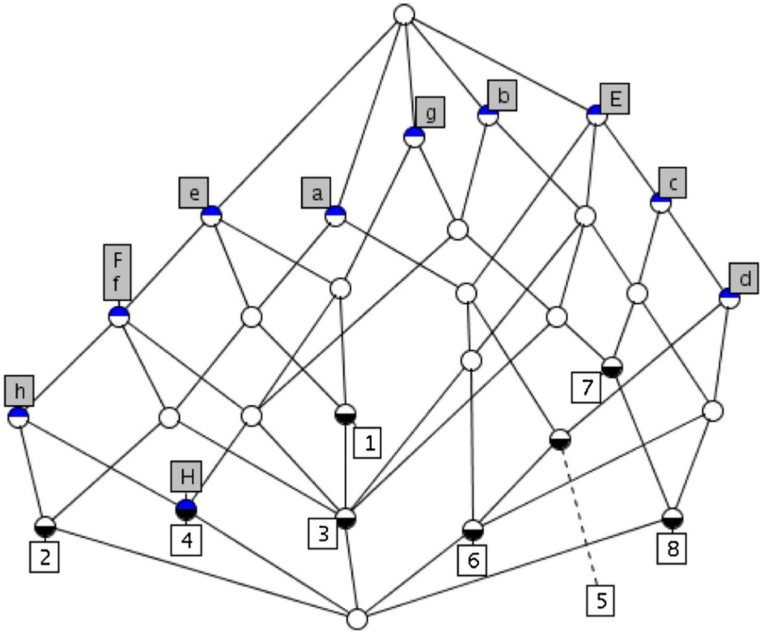

Fig. 5 α-generalization with $\mu E \| \mu b$. $E = \{a, b, c\}$, $F = \{d, e, f\}$, $H = \{g, h\}$, $\alpha = 0.6$, where $x \parallel y$ means that x and y are incomparable

the following, we analyze the impact of ∃ and ∀ attribute generalizations on the size of the resulting set of generalized concepts.

4.1 An ∃-generalization on attributes

Let (G, M, I) be a context and (G, S, J) a context obtained by an ∃-generalization on attributes. We set $S = \{m_s \mid s \in S\}$, with $m_s \subseteq M$. Then $g \,\mathrm{J}\, m_s$ iff $\exists m \in m_s, g \,\mathrm{I}\, m$. To compare the sizes of the corresponding concept lattices, we define some mappings. We assume that $(m_s)_{s \in S}$ forms a partition of M. Then for each $m \in M$ there is a unique generalized attribute m_s such that $m \in m_s$, and $g \,\mathrm{I}\, m$ implies $g \,\mathrm{J}\, m_s$, for every $g \in G$. To distinguish

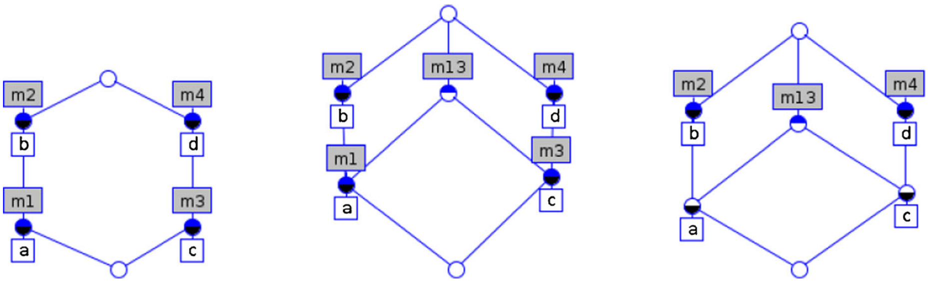

Fig. 6 B_4 (*left*) and an ∃-generalization: m_1 and m_3 are generalized to m_{13} and then removed from the context. The size does increase

between derivations in (G, M, I) and in (G, S, J), we will replace $'$ by the name of the corresponding relation. For example $g^{\mathrm{I}} = \{m \in M \mid g \,\mathrm{I}\, m\}$ and $g^{\mathrm{J}} = \{s \in S \mid g \,\mathrm{J}\, s\}$. Two canonical maps $\bar{\gamma}$ and $\bar{\mu}$ defined by

$$\begin{aligned} \bar{\gamma}\colon G &\to \mathfrak{B}(G, S, J) \\ g &\mapsto (g^{\mathrm{JJ}}, g^{\mathrm{J}}) \end{aligned} \quad \text{and} \quad \begin{aligned} \bar{\mu}\colon M &\to \mathfrak{B}(G, S, J) \\ m &\mapsto (s^{\mathrm{J}}, s^{\mathrm{JJ}}), \text{ where } m \in m_s \end{aligned}$$

induce two order preserving maps φ and ψ defined by

$$\begin{aligned} \varphi : \mathfrak{B}(G, M, I) &\to \mathfrak{B}(G, S, J) \\ (A, B) &\mapsto \bigvee\{\bar{\gamma} g \mid g \in A\} \end{aligned} \quad \text{and} \quad \begin{aligned} \psi : \mathfrak{B}(G, M, I) &\to \mathfrak{B}(G, S, J) \\ (A, B) &\mapsto \bigwedge\{\bar{\mu} m \mid m \in B\}. \end{aligned}$$

If φ or ψ is surjective, then the concept lattice of the generalized context is of smaller or equal cardinality. As we have seen on Fig. 6 these maps can be both not surjective. Obviously $\varphi(A, B) \leq \psi(A, B)$ since $g \,\mathrm{I}\, m$ implies $g \,\mathrm{J}\, m_s$ and $\bar{\gamma} g \leq \bar{\mu} m_s$. When do we have the equality? Does the equality imply surjectivity?

Here are some special cases where the number of concepts does not increase after a generalization.

Case 1 Every m_s has a greatest element $\top_s$. Then the context (G, S, J) is a projection of (G, M, I) on the set $M_S := \{\top_s \mid s \in S\}$ of greatest elements of m_s. Thus $\mathfrak{B}(G, S, J) \cong \mathfrak{B}(G, M_S, I \cap (G \times M_S))$ and is a sub-order of $\mathfrak{B}(G, M, I)$. Hence $|\mathfrak{B}(G, S, J)| = |\mathfrak{B}(G, M_S, I \cap G \times M_S)| \leq |\mathfrak{B}(G, M, I)|$.

Case 2 $\bigcup\{m^{\mathrm{I}} \mid m \in m_s\}$ is an extent, for any $m_s \in S$. Then any grouping does not produce a new concept. Hence the number of concepts cannot increase.

The following result (Theorem 1) gives an important class of lattices for which the $\exists$-generalization does not increase the size of the lattice. A context is object-reduced if no row can be obtained as the intersection of some other rows. In this case, the object concepts γg are $\bigvee$-irreducible.

Theorem 1 *The $\exists$-generalizations on distributive concept lattices whose contexts are object-reduced do not increase the size of the concept lattice.*

Proof Let $\mathbb{K} := (G, M, \mathrm{I})$ be an object-reduced context such that $L := \mathfrak{B}(\mathbb{K})$ is a distributive lattice. Let (G, S, J) be a context obtained by an $\exists$-generalization on the attributes in M. Let m_s be a generalized attribute, i.e. a group of attributes of M. It is enough to prove that m_s^{J} is an extent of (G, M, I). By definition,

$$m_s^{\mathrm{J}} = \bigcup\{m^{\mathrm{I}} \mid m \in m_s\} \subseteq \left(\bigcup\{m^{\mathrm{I}} \mid m \in m_s\}\right)^{\mathrm{II}} = \mathrm{ext}\left(\bigvee\{\mu m \mid m \in m_s\}\right)$$

For any $g \in \mathrm{ext}(\bigvee\{\mu m \mid m \in m_s\})$ we have $\gamma g \leq \bigvee\{\mu m \mid m \in m_s\}$ and

$$\begin{aligned} \gamma g &= \gamma g \wedge \bigvee\{\mu m \mid m \in m_s\}, && \text{since } \gamma g \leq \bigvee\{\mu m \mid m \in m_s\} \\ &= \bigvee\{\gamma g \wedge \mu m \mid m \in m_s\}, && \text{since } L \text{ is distributive} \\ &= \gamma g \wedge \mu m \text{ for some } m \in m_s, && \text{since } \mathbb{K} \text{ is object-reduced.} \end{aligned}$$

Therefore $\gamma g \leq \mu m$, and $g \in m^{\mathrm{I}}$. This proves that $\mathrm{ext}(\bigvee\{\mu m \mid m \in m_s\}) \subseteq m_s^{\mathrm{J}}$, and $m_s^{\mathrm{J}} = \mathrm{ext}\left(\bigvee\{\mu m \mid m \in m_s\}\right)$.

□

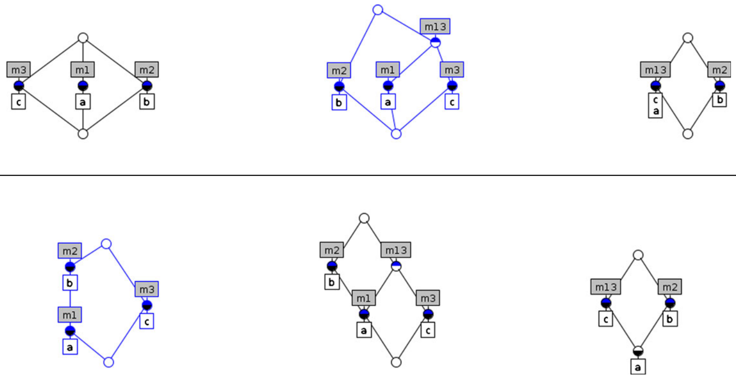

Fig. 7 ∃-generalization on M_3 (*above the line, left*) and on N_5 (*below the line, left*): m_1 and m_3 are generalized to m_{13} and then removed from the context. The size does not increase

In [9] Marcel Erné found necessary and sufficient conditions on contexts for a concept lattice to be distributive (see also [16]). The above discussed cases are not the only ones where the size does not increase. It would be interesting to describe the classes of lattices on which ∃-generalizations do not increase the size. Figure 7 provides some examples satisfying none of the conditions above.

The lattices M_3 and N_5 are the minimal non-distributive lattices. Any generalization on these lattices does not increase the size. On the lattice B_4 (Fig. 6) there is an ∃-generalization that increases the size of the concept lattice. In fact, putting the attribute m_1 and m_3 together generates exactly one new concept: μm_{13} in the context $(\{a, b, c, d\}, \{m_1, m_2, m_3, m_4, m_{13}\}, \mathrm{I})$.[3] However, the attributes m_1 and m_3 become reducible. Removing these does not reduce the size of the concept lattice. This seems to be the main configuration that forces ∃-generalizations to increase the size, as we can see in Proposition 1.

Proposition 1

i) *The lattice B_4 is the smallest lattice on which there is an ∃-generalization that increases the size of the initial concept lattice.*

ii) *If a context contains attributes m_1, m_2, m_3, m_4 such that $\mu m_1 < \mu m_2$, $\mu m_3 < \mu m_4$, $\mu m_2 \wedge \mu m_3 \leq \mu m_1$ and $\mu m_1 \wedge \mu m_4 \leq \mu m_3$, then there is an ∃-generalization that does not decrease the size of the concept lattice.*

Proof Let (G, M, I) be a context and $m_1, m_2, m_3, m_4 \in M$ such that $\mu m_1 < \mu m_2$, $\mu m_3 < \mu m_4$, $\mu m_2 \wedge \mu m_3 \leq \mu m_1$, $\mu m_1 \wedge \mu m_4 < \mu m_3$. Generalizing m_1 and m_3 to m_{13} will produce a new context $(G, M \cup \{m_{13}\})$. The attributes m_1 and m_3 become reducible since $\mu m_1 = \mu m_2 \wedge \mu m_{13}$ and $\mu m_3 = \mu m_4 \wedge \mu m_{13}$. In fact, for any $g \in G$ we have $g \in m_1' \implies g \in m_2' \cap m_{13}'$. Conversely $g \in m_2' \cap m_{13}' = m_2' \cap (m_1' \cup m_3') = (m_2' \cap m_1') \cup (m_2' \cap m_3') \subseteq m_1'$ since $\mu m_2 \wedge \mu m_3 \leq \mu m_1$. The proof that m_3 is reducible is similar. Therefore $|\mathfrak{B}(G, M \cup \{m_{13}\} \setminus \{m_1, m_2\}, \mathrm{I})| = |\mathfrak{B}(G, M \cup \{m_{13}\}, \mathrm{I})| \geq |\mathfrak{B}(G, M, \mathrm{I})|$. □

[3] We will not distinguish here between I and its restriction or extension.

Proposition 1 states that if a concept lattice contains a copy of B_4 labelled as indicated in Fig. 6 then there is a $\exists$-generalization that does not decrease the size of the lattice. This copy must not be a sublattice as we can see on Fig. 8.

To complete the characterization a couple of questions are still to be investigated:

– How many new concepts can be generated by an $\exists$-generalization on just two attributes?
– Does the converse of Proposition 1 ii) hold? i.e. Is there an $\exists$-generalization that does not decrease the size of the initial concept lattice only if it contains a copy of B_4? Although this seems to be plausible, we have not yet succeeded to write down a correct proof. We will investigate this further in the future work.

4.2 A $\forall$-generalization on attributes

Let (G, S, J) be a context obtained from (G, M, I) by a $\forall$-generalization. In the context $(G, M \cup S, \mathrm{I} \cup \mathrm{J})$, each attribute concept μm_s is reducible. This means that $m_s^{\mathrm{J}} = \bigcap\{m^{\mathrm{J}} \mid m \in m_s\} = \bigcap\{m^{\mathrm{I}} \mid m \in m_s\}$, and is an extent of (G, M, I). Therefore, $|\mathfrak{B}(G, S, \mathrm{J})| \leq |\mathfrak{B}(G, M \cup S, \mathrm{I} \cup \mathrm{J})| = |\mathfrak{B}(G, M, \mathrm{I})|$.

Theorem 2 *The $\forall$-generalizations on attributes do not increase the size of the concept lattice.*

5 Related work

The present section gives an overview about related work either in terms of processing generalized descriptions within the Formal Concept Analysis framework [11, 15, 19, 23] or with respect to the exploitation of ontology (including taxonomy) to discover "generalized" patterns from data. Our present work is much more concerned with the second topic. In the first research topic, the generalization of concept lattice construction to contexts with an additional order structure on the set of objects and/or attributes is proposed in [19, 23]. Ganter and Kuznetsov [15] propose an approach where objects together with their partially ordered data descriptions (e.g., labeled graphs) form "pattern structures" that are exploited

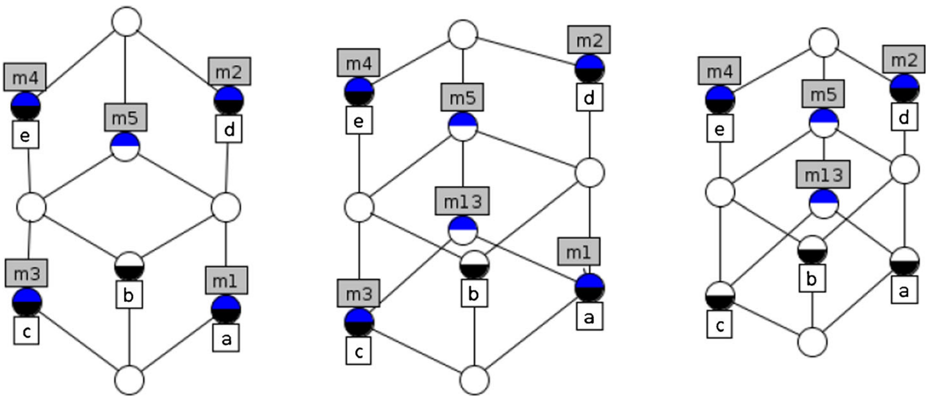

Fig. 8 An $\exists$-generalization with B_4 as subposet but not a sublattice

in FCA. To simplify computation, projection is used on pattern structures to produce concepts, implications and classification rules. The approach has been successfully used in [18] to analyze gene expression data where patterns are expressed as tuples of intervals. Pernelle et al. [24] independently used the notion of projection with the same motivation of getting smaller lattices as in [15]. To that end, they first introduced *extensional* projections by defining a function *ext* as a mapping of an object to a predefined type (or class) and introducing object set partition into clusters called basic classes. The notion of nested Galois lattices has also been introduced where nesting makes use of not only extensional but also intensional projections. Ferré [11] defines "Logical Concept Analysis" as a generalization of FCA, where sets of attributes are expressions of an (almost) arbitrary logic. In such a logical framework [12], subsumption relations can be defined and operations on concept lattices like navigation, querying and updating can be conducted in a uniform and powerful way.

For the second research topic, there are a set of studies [5, 6, 10, 13, 17, 28, 31] about the possible collaborations between Formal Concept Analysis and ontology engineering (e.g., ontology merging and mapping) to let the two formalisms benefit from each other's strengths. For example, starting from the observation that both domain ontologies and FCA aim at modeling concepts, [5] shows how FCA can be exploited to support ontology engineering (e.g., ontology construction and exploration), and conversely how ontologies can be fruitfully used in FCA applications (e.g., extracting new knowledge). In [28], the authors propose a bottom-up approach called $FCA - MERGE$ for merging ontologies using a set of documents as input. The method relies on techniques from natural language processing and FCA to produce a lattice of concepts. [13] studies the role of FCA in reusing independently developed domain ontologies. To that end, an ontology-based method for evaluating similarity between FCA concepts is defined to perform some Semantic Web activities such as ontology merging and ontology mapping. In [31] an approach towards the construction of a domain ontology using FCA is proposed. The resulting ontology is represented as a concept lattice and expressed via the Semantic Web Rule Language to facilitate ontology sharing and reasoning.

There are many efforts to integrate knowledge in a data mining process. For example, the study in [26] uses a taxonomy on attributes to produce generalized rules while [2] uses a domain ontology, including relations between concepts, to discover generalized sequential patterns. In the field of FCA, Ganter [14] exploits attribute implications for attribute exploration while in [3], the authors use attribute-dependency formulae for constraint-based data mining, and more precisely, to only select formal concepts that meet the constraints expressed by these formulae.

6 Tests

The goal of the experiments is twofold: (i) to show that generalization can bring new *semantically rich patterns*, and (ii) to highlight the fact that in the α and $\exists$ cases, the *size of the lattice* after attribute generalization is *generally reduced* except in some specific cases such as the $\exists$-generalization on groups of two attributes that do not share common attributes.

We carried out experiments on the well-known data set called *mushroom* from the UCI Repository [1]. This data collection describes samples representing 23 species of 8126 gilled mushrooms in terms of 22 nominally valued attributes. Each species is identified as either edible or poisonous. The converted set into a binary context contains 119 attributes, including two attributes for the two classes of mushrooms: edible (e) and poisonous (p). The binary context has a density of 19 % and leads to a large lattice of 238711 concepts.

6.1 New generalized patterns

To highlight the potential of generalization in discovering new and relevant patterns, we have performed different generalizations on attributes in the *mushroom* data set. Figure 9 shows one of them where two generalized attributes are created. The first one existentially generalizes habitat = $\{d, g, m\}$ and cap surface = $\{g, s, y\}$ to represent the property of a mushroom to have a habitat of either woods (d), grasses (g), or meadows (m) among the seven possible values, or to have a cap surface of one of the three sorts: grooves (g), scaly (y), or smooth (s). The second generalization concerns the property of having an odor of almond (a) or anise (l) among the nine possible mushroom odors. After generalization and projection on the generalized attributes as well as on the two classification attribute values and the narrow (n) gill-size attribute, we get Fig. 9 which clearly shows that the first generalized attribute is owned by 96 % of mushrooms but does not allow a discrimination between the two classes: edible or poisonous mushrooms. However, we can conclude that mushrooms with almond or anise odor are necessarily edible. Such pattern can be hardly seen in the initial large lattice.

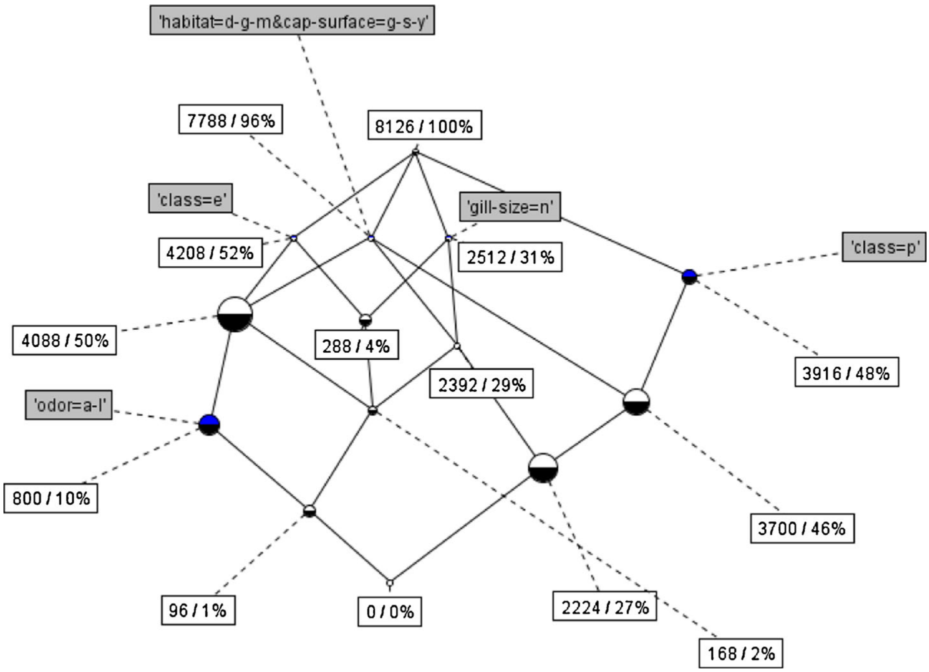

Fig. 9 Two generalized attributes in the mushroom data set

Table 2 α-generalization on 117 attributes of the mushroom data set

Nb. of generalized attributes	Nb. of generalized concepts	Percentage of reduction	Density
6	145	99.9	0.60
9	477	99.8	0.46
20	23138	90.3	0.68
58	185138	22.4	0.34

6.2 Lattice size variation

In order to empirically analyze the variation in the size of a lattice after an $\exists$ or α generalization, we have conducted two types of experiments on the mushroom data set. The first type of tests consists in random generation of 6, then 9, 20 and finally 58 generalized attributes among the 117 predictable elementary attributes and applying an α-generalization with a value $\alpha = 19\ \%$ within each group of attributes in the whole set of 8126 mushrooms. For example, in the case of nine generalized attributes, each group represents 13 elementary attributes. Every random generation was done ten times and an average value is computed for the number of generalized concepts and the density of the generalized context as shown in Table 2. One can expect a lesser reduction in the size of the lattice as the number of elementary attributes per generalized attribute is getting smaller and the number of generalized attributes is increasing. This is indeed observed when we took 58 generalized attributes per two elementary attributes randomly identified ten times.

Our previous tests [21] on large synthetically generated contexts show that the generalization process does not only reduce the context size but can also considerably reduce the size of the corresponding lattice. Moreover, the number of generalized concepts is almost inversely proportional to the fanout, i.e., the number of simple attributes per generalized attribute. However, we have noticed that when the fanout is equal to 2, the number of generalized concepts can be greater than the number of original concepts. This is observed in the second type of tests (see Table 3).

The second type of empirical tests related to lattice size variation aims at observing the impact of an $\exists$-generalization applied to only one pair of attributes of the whole set of 8126 mushrooms. Twenty attributes among the 117 predictable attributes are first randomly selected five times. For each group of twenty attributes, two of them are randomly identified ten times to form a generalized attribute. We therefore get 50 contexts of 19 attributes and 8126 objects. Table 3 shows that among the fifty contexts with only one generalized attribute obtained from two elementary attributes, there are situations where the size of the lattice decreases and other ones where the size increases. For example, among the fifty generated

Table 3 Lattice size variation after an $\exists$-generalization on one pair of attributes in the mushroom data set

Size decrease	Nb of tests
$[-0.007, -0.004[$	8
$[-0.004, 0[$	7
$[0, 6[$	17
$[6, 13[$	7
$[13, 41[$	11

contexts, we observe a decrease of 13 to 41 % in the size of the lattice while there are eight cases where the lattice hardly increases by 0.004 to 0.007.

7 Conclusion

In this paper we have studied the problem of using a taxonomy on objects and/or attributes in the framework of Formal Concept Analysis under three main cases of generalization (∃, ∀, and α) and have shown that (i) the set of *generalized* concepts is in some cases smaller than the set of patterns extracted from the *original* set of attributes (before generalization), and (ii) the generalized concept lattice not only embeds new patterns on generalized attributes but also reveals particular features of objects and may unveil a new taxonomy on objects. A careful analysis of the three cases of attribute generalization led to the following conclusion: the α-case is an attribute *approximation*, the ∀-case is an attribute *specialization* while only the ∃-case is actually an attribute *generalization*. Different scenarios of a simultaneous generalization on objects and attributes are also discussed based on the three cases of generalization.

Since we focused our analysis on the integration of taxonomies in FCA to produce generalized concepts, our further research concerns the theoretical study of the mapping between a rule set on original attributes and a rule set of generalized attributes as well as the exploitation of other components of an ontology such as general links (other than *is-a* hierarchies) between concepts/entities.

Acknowledgments The second and fourth authors acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors express their gratitude to the anonymous reviewers for their suggestions.

References

1. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA. [http://archive.ics.uci.edu/ml] (2013)
2. Adda, M., Valtchev, P., Missaoui, R., Djeraba, C.: Toward recommendation based on ontology-powered web-usage mining. IEEE Int. Comput. **11**(4), 45–52 (2007)
3. Belohlávek, R., Vychodil, V.: Adding background knowledge to Formal Concept Analysis via attribute dependency formulas. In: SAC, pp. 938–943 (2008)
4. Berge, C.: Graphs and Hypergraphs, North Holland Mathematical Library, vol. 6. North Holland, Elsevier, Amsterdam (1979). repr. of the 2., rev. ed. edition
5. Cimiano, P., Hotho, A., Stumme, G., Tane, J.: Conceptual knowledge processing with formal concept analysis and ontologies. In: Eklund P.W. (ed.) Concept Lattices, 2nd International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, 23–26 Feb 2004. Proceedings of Lecture Notes in Computer Science, vol. 2961, pp. 189–207. Springer (2004)
6. Curé, O., Jeansoulin, R.: An FCA-based solution for ontology mediation. In: ONISW '08: Proceeding of the 2nd International Workshop on Ontologies and Information Systems for the Semantic web, pp. 39–46. ACM, New York (2008)
7. Duquenne, V.: Lattice drawings and morphisms. In: 8th International Conference on Formal Concept Analysis, pp. 88–103 (2010)
8. Elkund, P.W. (ed.): Concept Lattices, 2nd International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, 23–26 Feb 2004. Proceedings, of Lecture Notes in Computer Science, vol. 2961. Springer (2004)
9. Erné, M.: Distributive laws for concept lattices. Algebra Univers. **30**, 538–580 (1993)

10. Fan, L., Xiao, T.: An automatic method for ontology mapping. In: Apolloni, B., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes of Computer Science, vol. 4964, pp. 661–669. Springer (2007)
11. Ferre, S., Ridoux, O.: A logical generalization of formal concept analysis. In: 8th International Conference on Conceptual Structures, pp. 371–384 (2000)
12. Ferre, S., Ridoux, O.: Introduction to logical information systems. Inf. Process. Manage. **40**(3), 383–419 (2004)
13. Formica, A.: Ontology-based concept similarity in formal concept analysis. Inf. Sci. **176**(18), 2624–2641 (2006)
14. Ganter, B.: Attribute exploration with background knowledge. Theor. Comput. Sci. **217**(2), 215–233 (1999)
15. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: 9th International Conference on Conceptual Structures, pp. 129–142 (2001)
16. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, New York (1999). Translator-C. Franzke
17. Haav, H.-M.: A semi-automatic method to ontology design by using FCA. In: Snásel, V., Belohlávek, R. (eds.) Concept Lattices and Applications (CLA), CEUR Workshop Proceedings, vol. 110. CEUR-WS.org (2004)
18. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci. **181**(10), 1989–2001 (2011)
19. Kuznetsov, S.O.: Learning of simple conceptual graphs from positive and negative examples. In: PKDD, pp. 384–391 (1999)
20. Kwuida, L., Missaoui, R., Amor, B.B., Boumedjout, L., Vaillancourt, J.: Restrictions on concept lattices for pattern management. In: Kryszkiewicz, M., Obiedkov, S.A., (eds.) Concept Lattices and Applications (CLA), CEUR Workshop Proceedings, vol. 672, pp. 235–246. CEUR-WS.org (2010)
21. Kwuida, L., Missaoui, R., Boumedjout, L., Vaillancourt, J.: Mining generalized patterns from large databases using ontologies. CoRR, arXiv:abs/09054713 (2009)
22. Kwuida, L., Missaoui, R., Vaillancourt, J.: Using taxonomies on objects and attributes to discover generalized patterns. In: Szathmary, L., Priss U. (eds.) CLA, pp. 327–338 (2012)
23. Liquiere, M., Sallantin, J.: Structural machine learning with Galois lattice and graphs. In: 15th International Conference on Machine Learning (ICML 1998), pp. 305–313. Madison (1998)
24. Pernelle, N., Rousset, M.-C., Soldano, H., Ventos, V.: Zoom: a nested Galois lattices-based system for conceptual clustering. J. Exp. Theor. Artif. Intell. **14**(2–3), 157–187 (2002)
25. Scheich, P., Skorsky, M., Vogt, F., Wachter, C., Wille, R.: Conceptual data systems. In: Opitz, O., Lausen, B., Klar, R. (eds) Information and Classification, pp. 72–84. Springer, Berlin-Heidelberg (1993)
26. Srikant, R., Agrawal, R.: Mining generalized association rules. In: Dayal, U., Gray, P.M.D., Nishio, S. (eds.) VLDB, pp. 407–419. Morgan Kaufmann (1995)
27. Stumme, G.: Conceptual on-line analytical processing. In: Tanaka, K., Ghandeharizadeh, S., Kambayashi, Y. (eds.) Information Organization and Databases, chap. 14, pp. 191–203. Kluwer (2000)
28. Stumme, G., Alexander, M.: FCA-MERGE: Bottom-up merging of ontologies. In 17th International Joint Conference on Artificial Intelligence (IJCAI) pp. 225–234. Seattle (2001)
29. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with titanic. Data Knowl. Eng. **42**(2), 189–222 (2002)
30. Ventos, V., Soldano, H.: Alpha Galois lattices: an overview. In: Ganter, B., Godin, R. (eds.) 3rd International Conference on Formal Concept Analysis, Lecture Notes in Computer Science, vol. 3403, pp. 299–314. Springer (2005)
31. Wang, J., He, K.: Towards representing FCA-based ontologies in semantic web rule language. In: CIT '06: Proceedings of the 6th IEEE International Conference on Computer and Information Technology, pp. 41. IEEE Computer Society, Washington (2006)
32. Wille, R.: Preconcept algebras and generalized double boolean algebras. In: Eklund P.W. (ed.) Concept Lattices, 2nd International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, 23–26 Feb 2004. Proceedings, of Lecture Notes in Computer Science, vol. 2961. Springer (2004)