Using Taxonomies on Objects and Attributes to Discover Generalized Patterns

Léonard Kwuida[1], Rokia Missaoui[2]*, Jean Vaillancourt[2]

[1] Bern University of Applied Sciences
kwuida@gmail.com
[2] Université du Québec en Outaouais
{rokia.missaoui,jean.vaillancourt}@uqo.ca

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* patterns in the form of concepts. To that end, we analyze three generalization cases and different scenarios of a simultaneous generalization on both objects and attributes.
We also contrast the number of generalized patterns against the number of simple patterns.

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 analyze some possible ways to abstract or group objects and/or attributes together 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" [6]. One recurrent problem in FCA is the number of concepts that can be exponential in the size of the context. To handle this problem many techniques have been proposed [2] to use or produce a taxonomy on attributes or objects to control the size of the context and the corresponding concept lattice.

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, and discusses different scenarios of generalizing both objects and attributes. In Section 4 we discuss the visualization issue of generalized patterns and provide the real meaning of the three generalization cases. In Section 5 the size of the generalized concept set is compared to the size of the initial (*i.e.*, before generalization) concept set. Finally, existing work about combining FCA with ontology is briefly described in Section 6.

* The second author acknowledges the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

© Laszlo Szathmary, Uta Priss (Eds.): CLA 2012, pp. 327–338, 2012.
ISBN 978–84–695–5252–0, Universidad de Málaga (Dept. Matemática Aplicada)

2 Formal Concept Analysis and Data Mining

2.1 Elementary information systems, contexts and concepts

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

Fig. 1: A formal context

In formal concept analysis, a context is a triple $\mathbb{K} := (G, M, I)$ 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 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 the formal context $\mathbb{K}$, respectively. A subset X is closed if $X'' = X$. Closed subsets of G are exactly extents while closed subsets of M are intents of $\mathbb{K}$. Figure 1 describes items $a, \ldots, h$ that appear in eight transactions (customers) of a *market basket analysis* application. Such a setting defines a binary relation I between the set G of objects/transactions and the set M of properties/items.

The concept hierarchy is formalized with a relation $\leq$ defined on $\mathfrak{B}(\mathbb{K})$ by $A \subseteq C :\Longleftrightarrow (A, B) \leq (C, D) :\Longleftrightarrow B \supseteq D$. This is an order relation, and is also called a *specialization/generalization* relation on concepts. In fact, a concept (A, B) is called 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 $\bigwedge \mathcal{C}$)[3]. 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 $(\gamma g := (g'', g'))_{g \in G}$ and the attribute concepts $(\mu m := (m', m''))_{m \in M}$ 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[4]. Thus, the set $\{\gamma g \mid g \in G\}$ is $\bigvee$-dense and the set $\{\mu m \mid m \in M\}$ is $\bigwedge$-dense in $\mathfrak{B}(G, M, \mathrm{I})$.

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 [6], based on context decomposition or lattice pruning/reduction (atlas decomposition, direct or subdirect decomposition, iceberg concept lattices,

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

[4] For $(A, B) \in \mathfrak{B}(G, M, \mathrm{I})$ we have $\bigvee\limits_{g \in A} \gamma g = (A, B) = \bigwedge\limits_{m \in B} \mu m$.

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 Figure 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 Figure 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. 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 [6].

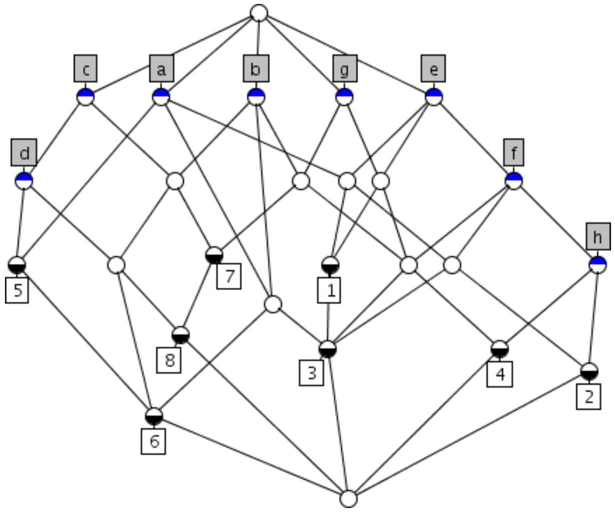

Fig. 2: Concept lattice of the context in Fig 1

Before we move to generalized patterns, let us see how data are transformed into binary contexts, the suitable format for our data.

2.3 Information Systems

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 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 $(g, m, w) \in I$ 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 $\mathbb{S}_m := (G_m, M_m, I_m)$ 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 $\mathbb{K}_m := (G, M_m, I^m)$ with

$gI^m s_m :\iff m(g)I_m s_m$, 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 $\mathbb{K}^S := (G, N, I^S)$ where $N := \bigcup\{M_m \mid m \in M\}$ and $gI^S s_m \iff m(g)I^m s_m$. 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 [9, 12]. 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 $\mathbb{K} := (G, M, I)$ be a context. The attributes of $\mathbb{K}$ can be grouped together to form another set of attributes, namely S, to get a context where the attributes are more general than in $\mathbb{K}$. 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:

($\exists$) $g\,\mathrm{J}\,s :\iff \exists m \in s,\, g\,\mathrm{I}\,m$. 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.

($\forall$) $g\,\mathrm{J}\,s :\iff \forall m \in s,\, g\,\mathrm{I}\,m$. 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.

($\alpha\%$) $g \,\mathrm{J}\, s \;:\iff\; \frac{|\{m \in s \;\mid\; g \,\mathrm{I}\, m\}|}{|s|} \geq \alpha_s$ where α_s is a threshold set by the user for the generalized attribute s. This case generalizes the ($\exists$)-case ($\alpha = \frac{1}{|M|}$) and the ($\forall$)-case ($\alpha = 1$). 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

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

$\mathbb{K}_{\exists-\forall-\alpha}$	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		×	×	×			×		×	×	×			×			×		

Fig. 3: Three generalizations of the context in Fig. 1 (see Subsection 3.1). The $\exists$-generalized attributes are $A := \{e, g\}$, $B := \{b, c\}$, $C := \{a, d\}$ and $D := \{f, h\}$. The $\forall$-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 $\alpha = 60\%$.

The α-case discussed here generalizes the alpha Galois lattices investigated by Ventos *et al* [14, 10]. In fact, if the set S forms a partition of M and all α_s are equal, then the generalization is an alpha Galois lattice.

Attribute generalization reduces the number of attributes. 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.

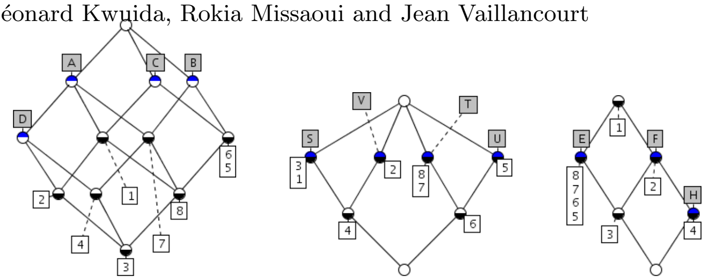

Fig. 4: Lattices of contexts in Fig 3. ∃-generalization (left), ∀-generalization (middle) and α-generalization (right)

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 Figure 4, 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 buys at least one item from the product line A.

From the ∀-case in Figure 4 (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.

To illustrate the α-case, we put the attributes of M in three groups $E := \{a, b, c\}$, $F := \{d, e, f\}$ and $H := \{g, h\}$ and set $\alpha := 60\%$ for all groups. This α-generalization on the attributes of M is presented in Figure 4 (right). Note that if all groups have two elements, then any α-generalization would be either an ∃-generalization ($\alpha \leq 0.5$) or a ∀-generalization ($\alpha > 0.5$). From the lattice in Figure 4 (right) one can see that any customer buying 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 less) 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).

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.

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 [1]), since the objects are subsets of G and attributes are subsets of M. Let $\mathcal{A}$ be a group of objects and $\mathcal{B}$ be a group of attributes related to a context $\mathbb{K}$. Then, the relation J can be defined using one or a combination of the following cases:

1. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\exists a \in \mathcal{A}$, $\exists b \in \mathcal{B}$ such that $a\,\mathrm{I}\,b$, i.e. some objects from the group $\mathcal{A}$ are in relation with some attributes in the group $\mathcal{B}$;

2. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\forall a \in \mathcal{A}$, $\forall b \in \mathcal{B}$ $a\,\mathrm{I}\,b$, i.e. every object in the group $\mathcal{A}$ is in relation with every attribute in the group $\mathcal{B}$;
3. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\forall a \in \mathcal{A}$, $\exists b \in \mathcal{B}$ such that $a\,\mathrm{I}\,b$, i.e. every object in the group $\mathcal{A}$ has at least one attribute from the group $\mathcal{B}$;
4. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\exists b \in \mathcal{B}$ such that $\forall a \in \mathcal{A}$ $a\,\mathrm{I}\,b$, i.e. there is an attribute in the group $\mathcal{B}$ that belongs to all objects of the group $\mathcal{A}$;
5. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\forall b \in \mathcal{B}$, $\exists a \in \mathcal{A}$ such that $a\,\mathrm{I}\,b$, i.e. every property in the group $\mathcal{B}$ is satisfied by at least one object of the group $\mathcal{A}$;
6. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\exists a \in \mathcal{A}$ such that $\forall b \in \mathcal{B}$ $a\,\mathrm{I}\,b$, there is an object in the group $\mathcal{A}$ that has all the attributes in the group $\mathcal{B}$;
7. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\frac{\left|\left\{a\in\mathcal{A} \mid \frac{|\{b\in\mathcal{B} \mid a\,\mathrm{I}\,b\}|}{|\mathcal{B}|} \geq \beta_{\mathcal{B}}\right\}\right|}{|\mathcal{A}|} \geq \alpha_{\mathcal{A}}$, i.e. at least $\alpha_{\mathcal{A}}$ fraction of objects in the group $\mathcal{A}$ have each at least $\beta_{\mathcal{B}}$ fraction of the attributes in the group $\mathcal{B}$;
8. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\frac{\left|\left\{b\in\mathcal{B} \mid \frac{|\{a\in\mathcal{A} \mid a\,\mathrm{I}\,b\}|}{|\mathcal{A}|} \geq \alpha_{\mathcal{A}}\right\}\right|}{|\mathcal{B}|} \geq \beta_{\mathcal{B}}$, i.e. at least $\beta_{\mathcal{B}}\%$ of attributes in the group $\mathcal{B}$ belong altogether to at least $\alpha_{\mathcal{A}}\%$ of objects in the group $\mathcal{A}$;
9. $\mathcal{A}\,\mathrm{J}\,\mathcal{B}$ iff $\frac{|\mathcal{A}\times\mathcal{B}\cap I|}{|\mathcal{A}\times\mathcal{B}|} \geq \alpha$, i.e. the density of the rectangle $\mathcal{A}\times\mathcal{B}$ is at least equal to α.

Remark 1. The cases 7 and 8 generalize Case 1 ($\alpha_{\mathcal{A}} := \frac{1}{|G|}$, $\beta_{\mathcal{B}} := \frac{1}{|M|}$ for all $\mathcal{A}$ and $\mathcal{B}$) and Case 2 ($\alpha_{\mathcal{A}} := 1$, $\beta_{\mathcal{B}} := 1$ for all $\mathcal{A}$ and $\mathcal{B}$). Moreover, Case 7 also generalizes Case 3 ($\alpha_{\mathcal{A}} := 1$, $\beta_{\mathcal{B}} := \frac{1}{|M|}$ for all $\mathcal{A}$ and $\mathcal{B}$) and Case 5 ($\alpha_{\mathcal{A}} := \frac{1}{|G|}$, $\beta_{\mathcal{B}} := 1$ for all $\mathcal{A}$ and $\mathcal{B}$). However, Cases 4 and 6 cannot be captured by Case 7, but are captured by Case 8 ($\alpha_{\mathcal{A}} := 1$, $\beta_{\mathcal{B}} := \frac{1}{|M|}$ for all $\mathcal{A}$ and $\mathcal{B}$ to get Case 4, and $\alpha_{\mathcal{A}} := \frac{1}{|G|}$, $\beta_{\mathcal{B}} := 1$ for all $\mathcal{A}$ and $\mathcal{B}$ to get Case 6).

An example of generalization on both objects and attributes would be one of customers grouped according to some common features and items grouped into product lines. We can also assign to each group all items bought by their members (an $\exists$-generalization) or only their common items (a $\forall$-generalization), or just some of the frequent items among their members (similar to an α-generalization).

4 Visualizing Generalized Patterns on Line diagrams

4.1 Visualization

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 (G, S, J) which contains concepts with generalized attributes (See Fig 4). However, one may be interested, after getting (G, S, J) and constructing a line diagram for $\mathfrak{B}(G, S, J)$, to refine further on the attributes in M or recover the lattice constructed from $\mathbb{K}$.

When storage space is not a constraint, then the attributes in M and the generalized attributes can be kept altogether. This is done using the apposition of $\mathbb{K} := (G, M, \mathrm{I})$ and (G, S, J) to get $(G, M \cup S, I \cup J)$.

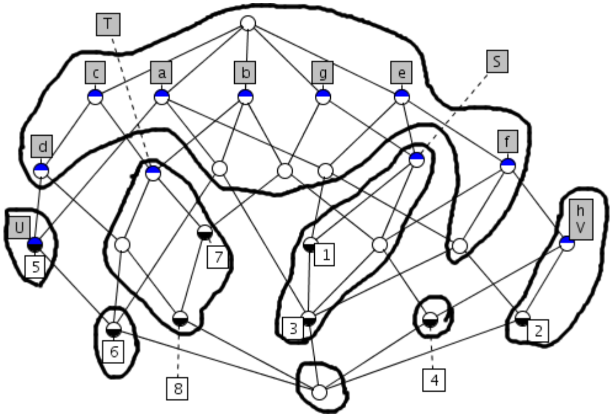

Fig. 5: Projection of the lattice in Figure 2 onto the ∀-generalized attributes.

A nested line diagram [6] 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.*, we construct a line diagram for $\mathfrak{B}(G, S, J)$ with nodes large enough to contain copies of the line diagram of $\mathfrak{B}(\mathbb{K})$. The generalized patterns can then be visualized by conducting a projection (*i.e.*, a restricted view) on generalized attributes, and keeping track of the effects of the projection, i.e, we display the projection of the concept lattice $\mathfrak{B}(G, M \cup S, I \cup J)$ on S by marking the equivalence classes on $\mathfrak{B}(G, M \cup S, I \cup J)$. Note that two concepts (A, B) and (C, D) are equivalent with respect to the projection on S iff $B \cap S = D \cap S$ (*i.e.*, their intents have the same restriction on S [8]). This is illustrated by Figure 5.

4.2 Are generalized attributes really generalizations?

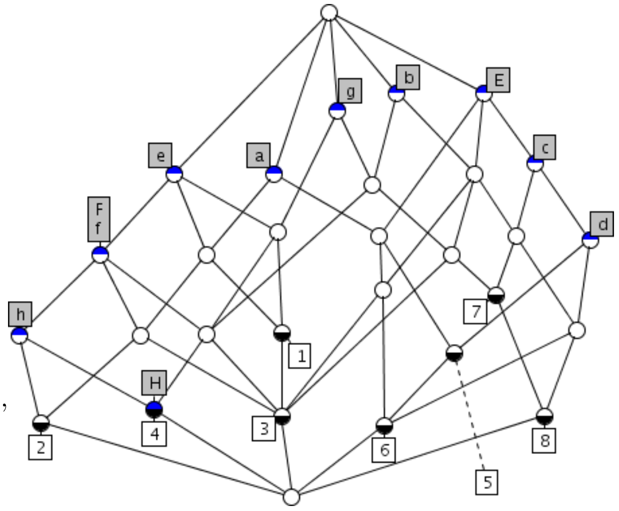

Fig. 6: α-generalization with $\mu E \| \mu b$. $E = \{a, b, c\}$, $F = \{d, e, f\}$, $H = \{g, h\}$, $\alpha = 0.6$.

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 $\mu a \geq \mu b$. 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 is really a *generalization* of the attributes $m \in m_s$.

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 is rather a *specialization* of the attributes $m \in m_s$. For the α-case, $\frac{1}{|M|} < \alpha < 1$, an object $g \in G$ is in relation with an attribute m_s iff $\alpha \leq \frac{|\{m \in m_s \mid g \,\mathrm{I}\, m\}|}{|m_s|}$. The following situations can happen:

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

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

Therefore, there are α-generalized attributes m_s that are neither a *generalization* of the m's nor a *specialization* of the m's. In Figure 6, the element b belongs to the group E, but μE is neither a specialization nor a generalization of μb, since $\mu b \not\leq \mu E$ and $\mu E \not\leq \mu b$. Thus, we should better call the α-case an attribute *approximation*, the $\forall$-case a *specialization* and only the $\exists$-case a *generalization*.

5 Controlling the size of generalized concepts

A generalized concept is a concept whose intent (or extent) contains generalized attributes (or objects). Figure 7 displays an $\exists$-generalization that leads to a larger number of concepts. The two concepts μm_1 and μm_2 will be put together. Although attributes m_1 and m_2 are replaced with m_{12}, the nodes γg_2 and γg_3 will remain since they will be obtained as $\mu m_{12} \wedge \mu m_4$ and $\mu m_{12} \wedge \mu m_3$ respectively. Then we get the configuration on Figure 7 (right) which has one concept more than the initial concept lattice shown in the left of the same figure.

In the following, we analyze the impact of $\exists$ and $\forall$ attribute generalizations on the size of the resulting set of generalized concepts.

5.1 An $\exists$-generalization on attributes

Let (G, M, I) be a context and (G, S, J) a context obtained from an $\exists$-generalization on attributes, i.e the elements of S are groups of attributes from M. We set $S = \{m_s \mid s \in S\}$, with $m_s \subseteq M$. Then, an object $g \in G$ is in relation with a generalized attribute m_s if there is an attribute m in m_s such that $g\,\mathrm{I}\,m$. To compare the size of the corresponding concept lattices, we can 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 between derivations in (G, M, I) and in (G, S, J), we will replace $'$ by the name of the corresponding relation. For example $g^I = \{m \in M \mid g\,\mathrm{I}\,m\}$ and $g^J = \{s \in S \mid g\,\mathrm{J}\,s\}$. Two canonical maps α and β are defined as follows:

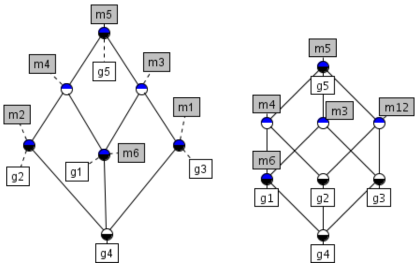

Fig. 7: An $\exists$-generalization (right) increasing the size of the initial lattice (left). $m_{12} = \{m_1, m_2\}$.

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

The maps α and β induce two order preserving maps φ and ψ defined by

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

If φ or ψ is surjective, then the generalized context is of smaller cardinality. As we have seen on Figure 7 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 $\bar{\gamma} g \leq \mu \bar{m}_s$. 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.

Theorem 1. *The $\exists$-generalizations on distributive concept lattices whose contexts are object reduced decrease the size of the concept lattice.*

Proof. Let (G,M,I) be an object reduced context such that $\mathfrak{B}(G,M,I)$ 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,

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

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

$$\gamma g = \gamma m \wedge \bigvee\{\mu m \mid m \in m_s\} = \bigvee\{\gamma g \wedge \mu m \mid m \in m_s\} = \gamma g \wedge \mu m \text{ for some } m \in m_s.$$

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

Remark 2. The above discussed cases are not the only ones where the size does not increase. Therefore, it would be interesting to describe the classes of lattices on which $\exists$-generalizations do not increase the size.

5.2 A $\forall$-generalization on attributes

Let (G,S,J) be a context obtained from (G,M,I) by a $\forall$-generalization. In the context $(G, M \cup S, I \cup J)$, 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, $|\mathfrak{B}(G,S,J)| \leq |\mathfrak{B}(G, M \cup S, I \cup J)| = |\mathfrak{B}(G,M,I)|$.

Theorem 2. *The $\forall$-generalizations on attributes reduce the size of the concept lattice.*

6 Related work

There are a set of studies [2–5, 7, 13, 15] 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 strengths. For example, starting from the observation that both domain ontologies and FCA aim at modeling concepts, [2] show 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 [13], 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. Starting from a set of domain specific texts, [7] proposes a semi-automatic method for ontology extraction and design based on FCA and Horn clause model. [5] 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 [15] 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.

In [4], a method for ontology mapping, called FCA-Mapping, is defined based on FCA and allows the identification of equal and subclass mapping relations. In [3], FCA is also used to propose an ontology mediation method for ontology merging. The resulting ontology includes new concepts not originally found in the input ontologies but excludes some redundant or irrelevant concepts.

In association rule mining, there are many efforts to integrate knowledge in the process of rule extraction to produce generalized patterns [11].

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 ($\exists$, $\forall$, 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 $\forall$-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.

References

1. Claude Berge. *Graphs and hypergraphs*, volume 6 of *North - Holland Mathematical Library*. North-Holland, Elsevier, Amsterdam, repr. of the 2., rev. ed. edition, 1979.
2. Philipp Cimiano, Andreas Hotho, Gerd Stumme, and Julien Tane. Conceptual knowledge processing with formal concept analysis and ontologies. In *ICFCA*, pages 189–207, 2004.
3. Olivier Curé and Robert Jeansoulin. An fca-based solution for ontology mediation. In *ONISW '08: Proceeding of the 2nd int. workshop on Ontologies and nformation systems for the semantic web*, pages 39–46, New York, NY, USA, 2008. ACM.
4. Liya Fan and Tianyuan Xiao. An automatic method for ontology mapping. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 661–669, 2007.
5. Anna Formica. Ontology-based concept similarity in formal concept analysis. *Inf. Sci.*, 176(18):2624–2641, 2006.
6. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., 1999. Translator-C. Franzke.
7. Hele-Mai Haav. A semi-automatic method to ontology design by using fca. In *CLA*, 2004.
8. Léonard Kwuida, Rokia Missaoui, Beligh Ben Amor, Lahcen Boumedjout, and Jean Vaillancourt. Restrictions on concept lattices for pattern management. In *Concept Lattices and Applications (CLA)*, pages 235–246, 2010.
9. Patrick Scheich, Martin Skorsky, Frank Vogt, Cornelia Wachter, and Rudolf Wille. Conceptual data systems. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 72–84. Springer, Berlin-Heidelberg, 1993.
10. Henry Soldano and Véronique Ventos. Abstract concept lattices. In *ICFCA*, pages 235–250, 2011.
11. Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. In *VLDB*, pages 407–419, 1995.
12. Gerd Stumme. Conceptual On-Line Analytical Processing. In K. Tanaka, S. Ghandeharizadeh, and Y. Kambayashi, editors, *Information Organization and Databases*, chapter 14, pages 191–203. Kluwer, 2000.
13. Gerd Stumme and Alexander Maedche. FCA-MERGE: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234, 2001.
14. Véronique Ventos and Henry Soldano. Alpha galois lattices: An overview. In *ICFCA*, pages 299–314, 2005.
15. Jian Wang and Keqing He. Towards representing fca-based ontologies in semantic web rule language. In *CIT '06: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*, page 41, Washington, DC, USA, 2006. IEEE Computer Society.