Pattern Structures and Their Projections

Bernhard Ganter[1] and Sergei Kuznetsov[2]

[1] Insitut für Algebra, TU Dresden
D-01062 Dresden, Germany
ganter@math.tu-dresden.de
[2] All-Russia Institute for Scientific and Technical Information
Moscow, Russia
serge@viniti.msk.ru

Abstract Pattern structures consist of objects with descriptions (called patterns) that allow a semilattice operation on them. Pattern structures arise naturally from ordered data, e.g., from labeled graphs ordered by graph morphisms. It is shown that pattern structures can be reduced to formal contexts, however sometimes processing the former is often more efficient and obvious than processing the latter. Concepts, implications, plausible hypotheses, and classifications are defined for data given by pattern structures. Since computation in pattern structures may be intractable, approximations of patterns by means of projections are introduced. It is shown how concepts, implications, hypotheses, and classifications in projected pattern structures are related to those in original one.

Introduction

Our investigation is motivated by a basic problem in pharmaceutical research. Suppose we are interested which chemical substances cause a certain effect, and which do not. A simple assumption would be that the effect is triggered by the presence of certain molecular substructures, and that the non-occurence of the effect may also depend on such substructures.

Suppose we have a number of observed cases, some in which the effect does occur and some where it does not; we then would like to form hypotheses on which substructures are responsible for the observed results. This seems to be a simple task, but if we allow for combinations of substructures, then this requires an effective strategy.

Molecular graphs are only one example where such an approach is natural. Another, perhaps even more promising domain is that of *Conceptual Graphs* (CGs) in the sense of Sowa [21] and hence, of logical formulas. CGs can be used to represent knowledge in a form that is close to language. It is therefore of interest to study, how hypotheses can be derived from Conceptual Graphs.

A strategy of hypothesis formation has been developed under the name of JSM-method by V. Finn [8] and his co-workers. Recently, the present authors have demonstrated [11] that the approach can neatly be formulated in the language of another method of data analysis: Formal Concept Analysis (FCA) [12].

The theoretical framework provided by FCA does not always suggest the most efficient implementation right away, and there are situations where one would choose other data representation forms. In this paper we show that this can be done in full compliance with FCA theory.

1 Formal Contexts

From every binary relation, a complete lattice can be constructed, using a simple and useful construction. This has been observed by Birkhoff [3] in the 1930s, and is the basis of Formal Concept Analysis, with many applications to data analysis.

The construction can be described as follows: Start with an arbitrary relation between two sets G and M, i.e., let $I \subseteq G \times M$, and define

$$A' := \{m \in M \mid (g,m) \in I \text{ for all } g \in A\} \qquad \text{for } A \subseteq G,$$

$$B' := \{g \in G \mid (g,m) \in I \text{ for all } m \in B\} \qquad \text{for } B \subseteq M.$$

Then the pairs (A, B) satisfying

$$A \subseteq G, B \subseteq M, A' = B, A = B'$$

are called the **formal concepts** of the **formal context** (G, M, I). When ordered by

$$(A_1, B_1) \le (A_2, B_2) :\iff A_1 \subseteq A_2 \quad (\iff B_2 \subseteq B_1),$$

they form a complete lattice, called the **concept lattice** of (G, M, I).

The name "Formal Concept" reflects the standard interpretation, where the elements of G are viewed as "objects", those of M as "attributes", and where $(g,m) \in I$ encodes that object g has attribute m. It has been demonstrated that the concept lattice indeed gives useful insight in the conceptual structure of such data (see [12] and references there).

That data are given in form of a formal context is a particularly simple case. If other kind of data is to be treated, the usual approach is first to bring it in this standard form by a process called "scaling". Recently, another suggestion was discussed by several authors [14], [15] [16] [17]: to generalize the abovementioned lattice construction to contexts with an additional order structure on G and/or M. This seems quite natural, since the mappings $A \mapsto A'$, $B \mapsto B'$ used in the construction above form a Galois connection between the power sets of G and M. It is well known that a complete lattice can be derived more generally from any Galois connection between two complete lattices.

On the other hand, one may argue that there is no need for such a generalization and that no proper generalization will be achieved, since the basic construction already is as general as possible: it can be shown that every complete lattice is isomorphic to some concept lattice.

Nevertheless, such a more general approach may be worthwhile for reasons of efficiency, and it seems natural as well. Several authors [2], [4], [7] have considered the case where instead of having attributes the objects satisfy certain logical formulas. In such a situation, shared attributes are replaced by common subsumers of the respective formulae.

We show how such an approach is linked to the general FCA framework. We discuss some operational and algorithmic aspects and demonstrate our results on an example.

2 Pattern structures

Let G be some set, let $(D, \sqcap)$ be a meet-semilattice and let $\delta : G \to D$ be a mapping. Then $(G, \underline{D}, \delta)$ with $\underline{D} = (D, \sqcap)$ is called a **pattern structure**, provided that the set

$$\delta(G) := \{\delta(g) \mid g \in G\}$$

generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$.[1] Each such complete semilattice has lower and upper bounds, which we denote by **0** and **1**, respectively. The intuitive meaning of a pattern structure is the set of objects with "descriptions" (patterns) with a "similarity operation" $\sqcap$ on them, i.e., an operation that for an arbitrary set of objects gives a "description" that represents the similarity of the objects from the subset. The similarity should be independent of the order in which the objects occur, therefore the operation should be idempotent, commutative, and associative.

The condition on the complete subsemilattice looks unpleasant. But note that there are two situations where this is automatically satisfied: when $(D, \sqcap)$ is complete, and when G is finite.

If $(G, \underline{D}, \delta)$ is a pattern structure, we define the derivation operators as

$$A^\square := \bigsqcap_{g \in A} \delta(g) \qquad \text{for } A \subseteq G$$

and

$$d^\square := \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in D.$$

The elements of D are called **patterns**. The order on them is given, as usual, by

$$c \sqsubseteq d :\iff c \sqcap d = c,$$

and is called the **subsumption** order. The operators $(\cdot)^\square$ obviously make a Galois connection between the power set of G and $(D, \sqsubseteq)$. The pairs (A, d) satisfying

$$A \subseteq G, \quad d \in D, \quad A^\square = d, \quad \text{and} \quad A = d^\square$$

are called the **pattern concepts** of $(G, \underline{D}, \delta)$, with extent A and **pattern intent** d. The above notions are analogues of the corresponding notions in formal contexts. For $a, b \in D$ the **pattern implication** $a \to b$ holds if $a^\square \sqsubseteq b^\square$. A pattern implication says what patterns occur in an object "description" if a certain pattern does. Similarly, for $C, D \subseteq G$ the **object implication** $C \to D$ holds if

[1] By which we mean that every subset X of $\delta(G)$ has an infimum $\sqcap X$ in $(D, \sqcap)$ and that D_δ is the set of these infima.

$C^\Box \sqsubseteq D^\Box$. Informally, this implication says that "all patterns that occur in all objects from the set C occur also in all objects from the set D."

Since $(D_\delta, \sqcap)$ is complete, there is a (unique) operation $\sqcup$ such that $(D_\delta, \sqcap, \sqcup)$ is a complete lattice. It is given by

$$\sqcup X = \bigsqcap \{c \in D_\delta \mid \forall_{x \in X}\ x \sqsubseteq c\}.$$

A subset M of D is **$\sqcup$-dense** for $(D_\delta, \sqcap)$ if every element of D_δ is of the form $\sqcup X$ for some $X \subseteq M$. If this is the case, then with

$$\downarrow d := \{e \in D \mid e \sqsubseteq d\}$$

we get

$$c = \sqcup(\downarrow c \cap M) \qquad \text{for every } c \in D_\delta.$$

Of course, $M := D_\delta$ is always an example of a $\sqcap$-dense set.

If M is $\sqcup$-dense in $(D_\delta, \sqcap)$, the formal context (G, M, I) with I given as $gIm\!: \Leftrightarrow m \sqsubseteq \delta(g)$ is called a **representation context** for $(G, \underline{D}, \delta)$.

Theorem 1. *Let $(G, \underline{D}, \delta)$ be a pattern structure and let (G, M, I) be a representation context of $(G, \underline{D}, \delta)$. Then for any $A \subseteq G$, $B \subseteq M$ and $d \in D$ the following two conditions are equivalent*

1. *(A, d) is a pattern concept of $(G, \underline{D}, \delta)$ and $B = \downarrow d \cap M$.*
2. *(A, B) is a formal concept of (G, M, I) and $d = \bigsqcup B$.*

The **proof** is by a standard application of the basic theorem of Formal Concept Analysis [12].

Thus the pattern concepts of $(G, \underline{D}, \delta)$ are in 1-1-correspondence with the formal concepts of (G, M, I). Corresponding concepts have the same first components (called **extents**). These extents form a closure system on G and thus a complete lattice, which is isomorphic to the concept lattice of (G, M, I).

3 Computing pattern concepts

When a pattern structure is given, then in principle we have all the information necessary to determine its concepts. We might, for example, compute all infima of subsets of D_δ and thereby all pattern concepts. To this end we can, e.g., adapt the Next Concept algorithm [12]. In computation of even finite pattern structures, one should take into account the fact that performing a single closure may be intractable. For example, already the problem of testing the $\sqsubseteq$ relation for labeled graphs from Section 7 is NP-complete, and computing $X \sqcap Y$ is even more difficult. Thus, in designing an algorithm for computing pattern concepts, one needs first to minimize the number of operations $\sqcap$, then the number of $\sqsubseteq$ relation tests, and, in the last turn, the number of operations with Boolean vectors.

After each backtrack of the original version of the Next Concept algorithm, it performs intersection of $|G|$ object intents. To avoid this in case of "expensive"

$\sqcap$ operation, one can introduce a natural tree data structure. Each vertex of the tree corresponds to a concept (A, B) and the children of the tree correspond to concepts of the form $((A \cup \{g\})'', (A \cup \{g\})')$ (actually, only some concepts of this form). An algorithm of this kind was given in [9].

A similar algorithm of this type was described in [16] for computing with sets of graphs. Given a family $\mathcal{F}$ of graph sets and an idempotent, commutative and associative operation $\sqcap$ on them defined as in Section 7, the algorithm constructs the set of "all possible intersections" of sets from $\mathcal{F}$, i.e., the semilattice generated by sets from $\mathcal{F}$, and its line (Hasse) diagram. The time complexity of the algorithm is $O((\alpha + \beta|G|)|G||L|)$ and its space complexity is $O((\gamma|G||L|))$, where α is time needed to perform $\sqcap$ operation and β is time needed to test $\sqsubseteq$ relation and γ is the space needed to store the largest object from D_δ. Computing the line diagram of the set of all concepts, given the tree generated by the previous algorithm, takes $O((\alpha|G| + \beta|G|^2)|L|)$ time and $O((\gamma|G||L|)$ space [16].

A similar approach to computing pattern concepts and implications between objects can be made in lines of a procedure proposed in [2]. This procedure, called the **object exploration**, is the dual of the **attribute exploration** algorithm, which is standard in Formal Concept Analysis. For a given closure operator on G it computes its **stem base**; which is an irredundant system of implications on G that generate the closure operator. Here an implication

$$A \to B, \quad A, B \subseteq G$$

holds, like in case of implications between sets of attributes, if $B \subseteq A''$. The order studied in [2] is given by the hierarchy of descriptions in Description Logic, where the description that is an infimum of two other descriptions (their least common subsumer) can be of exponential size, e.g., for ALE logic [2].

In the beginning of the exploration process one has the empty set of object implications and the set of extents E, consisting at the initialization step of the empty extent. One considers the set of implications of the form $A \to A''$ for $A \in E$ in the lexicographical order and asks an expert whether each particular implication holds. If the expert says yes, then either the set of implications or the set of extents are updated (dependent on the fact whether a set of objects is pseudoclosed or closed), if not, the expert should provide a counterexample that updates the current set of objects.

As in case of formal contexts [12], $A \to B$ is defined for pattern structures as $B^\square \sqsubseteq A^\square$, which is equivalent to $\bigsqcap \delta(A) = \bigsqcap \delta(A \cup B)$.

As the result of object exploration one obtains the context with the same concept lattice as the lattice of description hierarchy (i.e., the lattice of least common subsumers) and the stem base of object implications.

The procedure proposed in [2] also applies to the general setting with an arbitrary semilattice D.

Its worst-case complexity sums up from two parts. The first one is related to computing pseudoclosed (for definition see [12], no good estimate of the number of pseudoclosed of a given context is known so far). The second term of the complexity estimate (due to tests whether pseudoclosed sets are closed, i.e., in case of object exploration, that they are extents) is similar to the upper bound from [16] given above.

4 Structured Attribute Sets

We have introduced pattern structures to replace sets of attributes by a sort of "descriptions". However, this does not exclude the possibility that the patterns are themselves attribute combinations. A natural situation where this occurs is when the attribute set is large, but structured, so that admissible attribute combinations can be described by generating subsets.

Consider the example mentioned in the introduction: there the observed patterns are chemical compounds, described by their molecular graphs. There is no natural $\sqcap$-operation for such graphs, except when we use a little trick: we replace each graph by the set of its subgraphs, including the graph itself. Then the meet is the set of all common substructures. When describing such sets, one will usually restrict to the *maximal* common substructures and tacitly include the sub-substructures of these. This can considerably reduce the computational effort, see the examples in Section 7.

To phrase this situation more abstractly, assume that the attribute set $(P, \leq)$ is finite and (partially) ordered, and that all attribute combinations that can occur must be order ideals (downsets) of this order. Any order ideal O can be described by the set of its maximal elements M as $O := \{x \mid \exists y \in M x \leq y\}$. The maximal elements form an antichain, and conversely each antichain is the set of maximal elements of some order ideal. Thus, in this case, the semilattice $(D, \sqcap)$ of patterns will consist of all antichains of the ordered attribute set, and will be isomorphic to the lattice of all order ideals of the ordered set (and thus isomorphic to the concept lattice of the context $(P, P, \not\geq)$, see [12]). For given antichains C_1 and C_2, the infimum $C_1 \sqcap C_2$ then consists of all maximal elements of the order ideal

$$\{m \mid \exists_{c_1 \in C_1} \exists_{c_2 \in C_2} \quad m \leq c_1 \text{ and } m \leq c_2\}.$$

Computing $C_1 \sqcap C_2$ may however be a problem. Note, e.g., that in the introductory example of chemical compounds, even the $\leq$-relation is difficult to compute, since $x \leq y$ amounts to x *is isomorphic to a subgraph of* y, which is an NP-complete problem [13].

There is a canonical representation context for this pattern structure $(G, \underline{D}, \delta)$. It is easy to see that the set of *principal ideals* $\downarrow p$ is $\sqcap$-dense in the lattice of all order ideals. Thus

$$(G, P, I) \qquad \text{with } (g, p) \in I :\Longleftrightarrow\ p \leq \delta(g)$$

is (isomorphic to) a representation context for $(G, \underline{D}, \delta)$.

Since the set of order ideals is closed under unions, the semilattice $\underline{D}$ of antichains will be a distributive one. The same approach also works in the meet-distributive case, for sets selected from a closure system with the *anti-exchange* property [12]. The anti-exchange property implies that each closed set is the closure of its *extreme points*, as it is known for the example of convex polyhedra. Again we get that each closed set has a canonical generating set, that may be used as a pattern.

5 Projections

It may happen that some of the patterns in a pattern structure are too complex and difficult to handle. In such a situation one is tempted to replace the patterns with weaker, perhaps simpler ones, even if that results in some loss of information.

We formalize this using a mapping $\psi\colon D \to D$ and replacing each pattern $d \in D$ by $\psi(d)$ such that the pattern structure $(G, \underline{D}, \delta)$ is replaced by $(G, \underline{D}, \psi \circ \delta)$[2].

It is natural to require that ψ is a kernel operator (or **projection**), i.e., that ψ is

monotone: if $x \sqsubseteq y$, then $\psi(x) \sqsubseteq \psi(y)$,
contractive: $\psi(x) \sqsubseteq x$ (or $\psi \leq id$, where id denotes the identity mapping), and
idempotent: $\psi(\psi(x)) = \psi(x)$.

In what follows we will use the following fact, well-known in order theory [6]:

Proposition 1. *Any projection of a complete semilattice $(D, \sqcap)$ is $\sqcap$-preserving, i.e., for any $X, Y \in D$*

$$\psi(X \sqcap Y) = \psi(X) \sqcap \psi(Y).$$

It is easy to describe how the lattice of pattern concepts changes when we replace $(G, \underline{D}, \delta)$ by $(G, \underline{D}, \psi \circ \delta)$. First, the folowing statement establishes the invariance of subsumption relation for projected data

Proposition 2. $\psi(d) \sqsubseteq \delta(g) \Leftrightarrow \psi(d) \sqsubseteq \psi \circ \delta(g)$.

Proof. If $\psi(d) \sqsubseteq \delta(g)$ then $\psi(d) = \psi(\psi(d)) \sqsubseteq \psi(\delta(g))$ by the idempotence of ψ. On the other hand, if $\psi(d) \sqsubseteq \psi \circ \delta(g)$ then $\psi(d) \sqsubseteq \delta(g)$, since ψ is contractive.

The following statement establishes the relation between projected pattern structures and their representation contexts: taking a projection is equivalent to taking a subset of attributes of the representation context of the original pattern structure.

Theorem 2. *For pattern structures $(G, \underline{D}, \delta_1)$ and $(G, \underline{D}, \delta_2)$ the following statements are equivalent:*

1. $\delta_2 = \psi \circ \delta_1$ *for some projection ψ of $\underline{D}$.*

[2] In this situation we consider two pattern structures simultaneously. When we use the symbol $^{\Box}$, it always refers to $(G, \underline{D}, \delta)$, not to $(G, \underline{D}, \psi \circ \delta)$.

2. *There is a representation context (G, M, I) of $(G, \underline{D}, \delta_1)$ and some $N \subseteq M$ such that $(G, N, I \cap (G \times N))$ is a representation context of $(G, \underline{D}, \delta_2)$.*

Proof. $1 \Rightarrow 2$. Let $\delta_1, \delta_2 : G \to D$ be mappings and let $\psi : D \to D$ be a projection of $\underline{D}$ such that $\delta_2 = \psi \circ \delta_1$.

Define $M := D$ and $N := \{\psi(m) \mid m \in M\} \subseteq M$. Clearly (G, M, I) with $(g, m) \in I \Leftrightarrow m \sqsubseteq \delta_1(g)$, is a representation context of $(G, \underline{D}, \delta_1)$. Moreover, (G, N, J) with $(g, n) \in J :\Leftrightarrow n \sqsubseteq \delta_2(g)$ is a representation context of $(G, \underline{D}, \delta_2)$. It remains to prove that $J = I \cap (G \times N)$, i.e., that the equation $n \sqsubseteq \delta_1(g) \Leftrightarrow n \sqsubseteq \delta_2(g)$ holds for arbitrary $g \in G, n \in N$. Note that for each $n \in N$ we have $\psi(n) = n$ (since $\psi(n)$ is idempotent). Thus, the equation follows from 2.
$1 \Leftarrow 2$. Having $N \subseteq M$ we define

$$\psi(d) := \{n \in N \mid n \sqsubseteq d\}.$$

This mapping is obviously contractive, monotone, and idempotent. By definition of ψ, we have $n \sqsubseteq d \Leftrightarrow n \sqsubseteq \psi(d)$ for all $n \in N$ and all $d \in D$. Therefore,

$$\psi \circ \delta_1(g) = \sqcup\{n \in N \mid n \sqsubseteq \delta_1(g)\} = \sqcup\{n \in N \mid n \sqsubseteq \psi(\delta_1(g))\} = \delta_2(g).$$

Corollary 1. *Every extent of $(G, \underline{D}, \psi \circ \delta)$ is an extent of $(G, \underline{D}, \delta)$. If d is a pattern intent of $(G, \underline{D}, \delta)$, then $\psi(d)$ is a pattern intent of $(G, \underline{D}, \psi \circ \delta)$, for which $\psi(d)^{\Box\Box} \sqsubseteq d$.*

Pattern structures are naturally ordered by projections: $(G, \underline{D}, \delta_1) \geq (G, \underline{D}, \delta_2)$ if there is a projection ψ such that $\delta_2 = \psi \circ \delta_1$. In this case, representation $(G, \underline{D}, \delta_2)$ can be said to be rougher than $(G, \underline{D}, \delta_1)$ and the latter to be finer than the former.

The following proposition relates implications in comparable pattern structures.

Proposition 3. *Let $a, b \in D$. If $\psi(a) \to \psi(b)$ and $\psi(b) = b$ then $a \to b$.*

Proof. By contractivity of projection we have $\psi(a) \sqsubseteq a$, hence $a^\Box \subseteq (\psi(a))^\Box$ and $a \to \psi(a)$. If $\psi(a) \to \psi(b) = b$, then $a \to b$ follows from the transitivity of the relation $\to$.

Thus, for a certain class of implications, it is sufficient to compute them in projected data (which can be far more efficient than to compute in original pattern structure) to establish their validity for the original pattern structure. Note that this proposition does not require a and b to be subsumed by patterns from $\delta(G)$. In the case where $a \sqsubseteq \delta(g)$ for no $g \in G$, we have $a^\Box = \emptyset$ by definition and all implications in Proposition 3 hold automatically.

Some examples considered in the last section demonstrate that the inverse of the proposition, as well as the generalization of it when $\psi(B) \neq B$, do not hold.

6 Hypothesis Generation from Projected Data

In [11] we considered a learning model from [8] in terms of Formal Concept Analysis. This model assumes that the cause of a **goal property** resides in common attributes of objects that have this property. If our objects are described by some mathematical structures, we may look for common substructures of those objects that have the goal property.

This can be transferred to pattern structures, where it is assumed that the presence and absence of the goal property can be predicted from the patterns associated with the objects. The setting can be formalized as follows. Let $(G, \underline{D}, \delta)$ be a pattern structure together with an external goal property w. The set G of all objects can be partitioned into three disjoint sets: The set G_+ of those objects that are known to have the property w (these are the *positive examples*), the set G_- of those objects of which it is known that they do not have w (the *negative examples*) and the set G_τ of *undetermined examples*, i.e., of those objects, of which it is unknown if they do have property w or not. This gives three pattern substructures of $(G, \underline{D}, \delta)$: $(G_+, \underline{D}, \delta)$, $(G_-, \underline{D}, \delta)$, $(G_\tau, \underline{D}, \delta)$.

A **positive hypothesis** h is defined as a pattern intent of $(G_+, \underline{D}, \delta)$ that is not subsumed by any pattern from $\delta(G_-)$ (for short: not subsumed by any negative example). Formally:

$$h \in D \text{ is a positive hypothesis iff } h^\square \cap G_- = \emptyset \text{ and } \exists A \subseteq G_+ : A^\square = h.$$

A **negative hypothesis** is defined accordingly. These definitions implement the general idea of machine learning in terms of formal concept analysis: "given positive and negative examples of a goal class, find generalizations (generalized descriptions) of positive examples that do not cover any negative examples."

A hypothesis in the sense of [11] is obtained as a special case of this definition when $(D, \sqcap) = (2^M, \cap)$ for some set M. Hypotheses can be used for classification of undetermined examples as introduced in [8] in the following way. If $g \in G_\tau$ is an undetermined example, then a hypothesis h with $h \sqsubseteq \delta(g)$ is **for the positive classification** of g if h is positive and **for the negative classification** of g if it is a negative hypothesis. Example $g \in G_\tau$ is **classified positively** if there is a hypothesis for its positive classification and none for its negative classification. It is **classified negatively** in the converse situation. We have no classification if there is no hypothesis for positive and negative classification or contradictory classification (if there are hypotheses for both positive and negative classification).

Hypotheses have been studied in detail elsewhere [11]. Here we focus our consideration on the following aspect. What happens when we use "weaker" data, approximating the original data? What is the significance of hypotheses obtained from weak data?

On the one hand, we almost always deal with weak data that describe reality approximately. For example, in case of molecular structures, a more adequate representation can be the 3D–geometrical one. But even this geometrical representation of a molecule is already an abstraction of a quantum-mechanical one, etc. On the other hand, having some data for which computation is intractable, we would like to have their tractable reasonable approximation that would allow one to judge about hypotheses and classifications in the original representation by means of results about hypotheses and classification for weak data.

This problem becomes more precise if we describe the data weakening by means of a projection $\psi: \underline{D} \to \underline{D}$. Instead of $(G, \underline{D}, \delta)$ we than work with $(G, \underline{D}, \psi \circ \delta)$ and its three parts $(G_+, \underline{D}, \psi \circ \delta)$, $(G_-, \underline{D}, \psi \circ \delta)$, and $(G_\tau, \underline{D}, \psi \circ \delta)$, as above.

To simplify our language, let us reserve the term "hypothesis" to those obtained from $(G, \underline{D}, \delta)$ and let us refer to those obtained from $(G, \underline{D}, \psi \circ \delta)$ as **ψ-hypotheses**. Now the question to be studied is: How are hypotheses and ψ-hypotheses related? In what follows we shall try to answer this question for positive hypotheses. Results similar to that below hold also for negative hypotheses and classifications.

There is no guarantee that the ψ-image of a hypothesis must be a ψ-hypothesis. In fact, our definition allows that ψ is the „null projection" with $\psi(d) = \mathbf{0}$ for all $d \in D$. This corresponds to total abandoning of the data, and no interesting hypotheses are to be expected in that situation.

However, we have the following

Proposition 4. *If $\psi(d)$ is a (positive) hypothesis, then $\psi(d)$ is also a (positive) ψ-hypothesis.*

Proof. If $\psi(d)$ is a positive hypothesis, then $\psi(d)$ is not subsumed by any negative example. Moreover, $\psi(d)$ is a pattern intent of $(G, \underline{D}, \psi \circ \delta)$ according to Corollary 1. Thus $\psi(d)$ is a ψ-hypothesis.

The classification set does not shrink when we pass from d to $\psi(d)$:

Proposition 5. *If d is a hypothesis for the positive classification of g and $\psi(d)$ is a positive ψ-hypothesis, then $\psi(d)$ is for the positive classification of g.*

Proof. is obvious, since $\psi(d) \sqsubseteq d \sqsubseteq g^{\square}$.

Proposition 6. *If $\psi(d)$ is a (positive) ψ-hypothesis, then $\psi(d)^{\square\square}$ is a (positive) hypothesis.*

Proof. Assume that $\psi(d)$ is a positive ψ-hypothesis. Then the corresponding extent in $(G, \underline{D}, \psi \circ \delta)$,

$$E := \{g \in G \mid \psi(d) \sqsubseteq \psi(\delta(g))\},$$

is contained in G_+ and is also an extent of $(G, \underline{D}, \delta)$ (by the corollary of Theorem 2). Thus $\psi(d)^{\square\square} = E^{\square}$ is an intent of $(G_+, \underline{D}, \delta)$ and cannot subsume any negative example, since it is subsumed by $\psi(d)$.

The propositions show that we may hunt hypotheses starting from ψ-hypotheses. We can shoot only those that can be seen in the projected data, but those can in fact be found, as the following theorem states:

Theorem 3. *For any projection ψ and any positive hypothesis $d \in D$ the following are equivalent:*

1. *$\psi(d)$ is not subsumed by any negative example.*
2. *There is some positive ψ-hypothesis h such that $h^{\Box\Box} \sqsubseteq d$.*

Proof. If $\psi(d)$ is not subsumed by any negative example, then $h := \psi(d)$ is a ψ-hypothesis and $h^{\Box\Box} \sqsubseteq d$ by Corollary 1. If h is a ψ-hypothesis, then $h^{\Box\Box}$ is a hypothesis by Proposition 6 and hence d, which subsumes $h^{\Box\Box}$, is not subsumed by any negative example.

7 An Application to Graphs

As an application we shall consider a situation where the patterns are given as labelled graphs. These may be structure graphs of chemical compounds, as mentioned in the introduction, and the task may be to find the patterns responsible for a pharmaceutical effect under investigation. It is natural to assume (and it is the common assumption in this applied domain) that common biological effect of chemical compounds is caused by their common substructures. The graphs could as well be conceptual graphs in the sense of Sowa [21]. Let P be the set of all graphs under consideration, and let this set be ordered by generalized subgraph isomorphism (a definition is given below). As above, we have a set G (of experiments, observations, or the like) and to each $g \in G$ there is an associated graph from P, denoted by $\delta(g)$.

To be more concrete, let $(\mathcal{L}\ \preceq)$ be some ordered set of "labels". In the examples below, this set will be the one displayed in Figure 1.

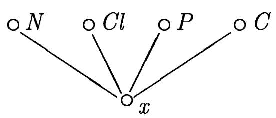

Figure 1. The ordered set of labels for the molecular graphs in our example. x stands for „any element“.

Let P be the set of all finite graphs, vertex-labelled by labels from $(\mathcal{L}, \preceq)$, up to isomorphism. A typical such graph is of the form $\Gamma := ((V, l), E)$, with vertex set V, edge set E and label assignment l. We say that $\Gamma_1 := ((V_1, l_1), E_1)$ **dominates** $\Gamma_2 := ((V_2, l_2), E_2)$, for short $\Gamma_2 \leq \Gamma_1$, if there exists a one-to-one mapping $\varphi : V_2 \to V_1$ that (for all $v, w \in V_2$)

- respects edges: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- fits under labels: $l_2(v) \preceq l_1(\varphi(v))$.

Obviously this is an order relation. It may be called the "generalized subgraph isomorphism relation", since in the unlabelled case it reduces to the subgraph isomorphism order. For conceptual graphs it corresponds to the injective specialization relation [19] or injective morphism [18].

Example 1. Let Γ_1 and Γ_2 be molecular graphs given in Figure 2. They represent

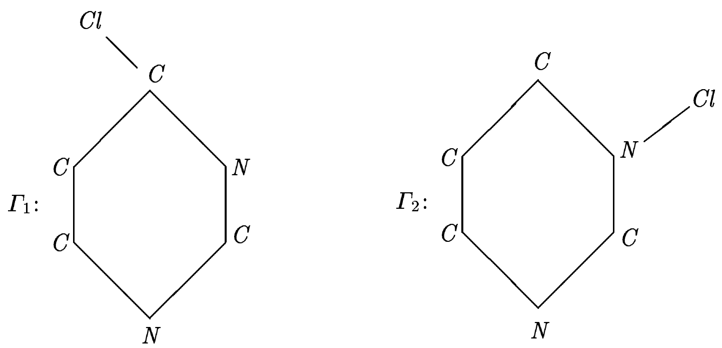

Figure 2. Two molecular graphs representing patterns.

patterns in the described sense. Their meet, with a slight misuse of notation written as $\Gamma_1 \sqcap \Gamma_2$, is given by the set of three graphs depicted in Figure 3. Here,

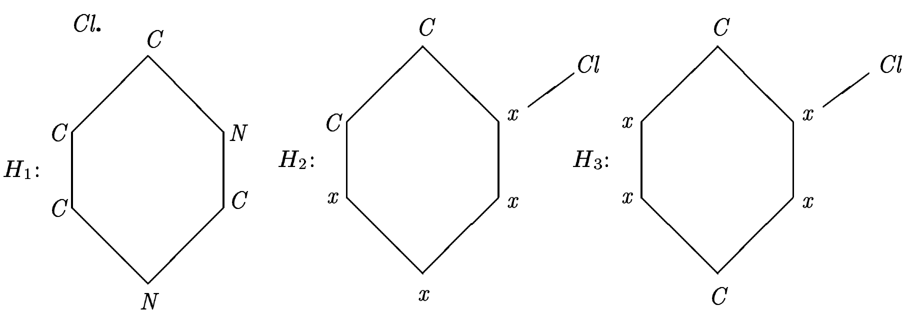

Figure 3. $\Gamma_1 \sqcap \Gamma_2$ given by the maximal graphs in the intersection of the downsets generated by Γ_1 and Γ_2.

the disconnected graph H_1 contains more information about the cyclic structure, whereas H_2 and H_3 contain more information about the connection of the cycle with the vertex labeled by "*Cl*".

Example 2. It is clear from the previous example that computing hypotheses for this pattern structure may be very tedious, since it may require to decide about subgraph isomorphism many times. It is therefore advisable to restrict the considerations to a subclass of the graphs, for which this is easier. This can neatly be described using the notion of a projection: let $Q \subseteq P$ be the set of the "simple" graphs in P, then

$$(G, Q, I \cap (G \times Q))$$

represents, according to Theorem 2, some pattern structure $(G, \underline{D}, \psi \circ \delta)$ obtained from a projection $\psi : \underline{D} \to \underline{D}$.

A natural choice for Q is the class of all paths of a bounded length n. This projection is important for pharmaceutical applications, since biological activities of chemical compounds often reside in linear fragments of their molecular structures. By Theorem 2, this describes a projection ψ_n. The images of Γ_1 and Γ_2 under ψ_3, represented by sets of 3-paths maximal w.r.t. $\sqsubseteq$, are shown in Figure 4.

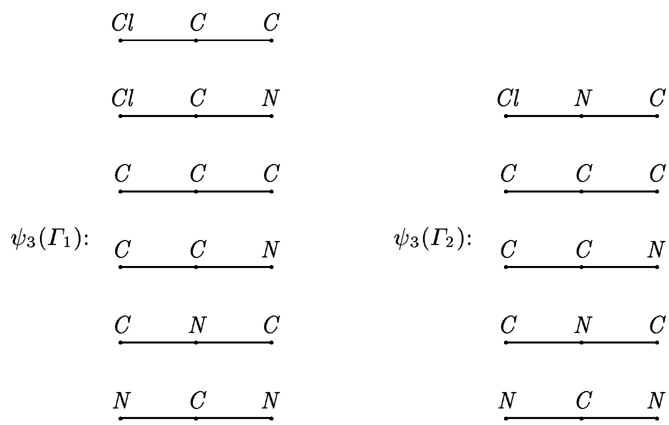

Figure 4. The sets representing the projected graphs $\psi_3(\Gamma_1)$ and $\psi_3(\Gamma_2)$.

The number of chains of fixed length contained in a molecular graph is polynomial in the size of the graph. Therefore, the computation of the meet operation drastically reduces in complexity. The maximal (w.r.t. $\sqsubseteq$) chains representing $\psi_3(\Gamma_1) \sqcap \psi_3(\Gamma_2)$ (which is, by Proposition 1, equal to $\psi_3(\Gamma_1 \sqcap \Gamma_2)$), are shown in Figure 5.

Example 3. We conclude with counterexamples for the inverse of Proposition 6 and its generalization for $\psi(B) \neq B$. Let ψ be ψ_2, which takes each graph in the set of all edges (which are not dominated by other edges). Then for A_1, A_2, B_1, B_2, B_3 given in Fig. 6 and the pattern structure $(\{g_1, g_2\}, (\Omega, \sqcap), \{\delta(g_1) = B_1, \delta(g_2) = B_2\})$ we have the implication $\psi_2(A_1) \to \psi_2(B_3)$ (since $\psi_2(A_1)^{\square} =$

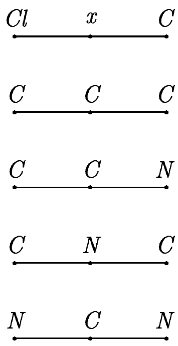

Figure 5. $\psi_3(\Gamma_1) \sqcap \psi_3(\Gamma_2)$.

$\{g_1, g_2\} = \psi_2(B_3)^\Box$), but not the implication $A_1 \to B_3$ (since $(A_1)^\Box = \{g_1, g_2\} \not\subseteq \emptyset = (B_3)^\Box$); we have the implication $A_2 \to B_1$ (since $(A_2)^\Box = \{g_2\} = (B_1)^\Box$), but not the implication $\psi_2(A_2) \to \psi_2(B_1)$ (since $\psi_2(A_2)^\Box = \{g_1, g_2\} \not\subseteq \{g_2\} = \psi_2(B_1)^\Box$).

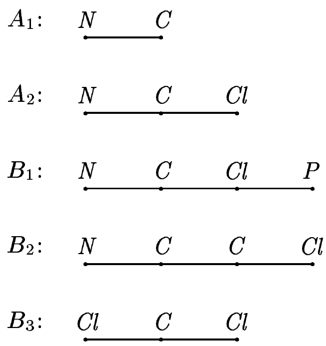

Figure 6. Graphs for Example 3.

8 Conclusion

We considered analogues to formal contexts, but with a (semi)lattice of patterns instead of attributes. Such structures can easily be mapped to formal contexts, and it can be described how hypothesis generation from such pattern structures is to be organized. This becomes more intricated when we allow data weakening. We have described methods to recover those hypotheses that are reflected in the weaker data. The notion of a pattern can be applied to ordered attribute sets. Then the pattern lattice is the lattice of order ideals of the attribute order. Weakening corresponds to a restriction of the attribute set.

Acknowledgments

The authors are grateful to the members of the *Arbeitsgruppe Allgemeine Algebra und Diskrete Mathematik* at the TU Darmstadt and colleagues of *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier* for helpful discussions.

The second author was supported by the Alexander von Humboldt Foundation.

References

1. H.-G. Bartel, Matematische Methoden in der Chemie, Spektrum, Heidelberg, 1996.
2. F. Baader and R. Molitor, Building and Structuring Description Logic Knowledge Spaces Using Least Common Subsumers and Concept Analysis, in *Proc. 8th Int. Conf. on Conceptual Structures, ICCS'2000*, G. Mineau and B. Ganter, Eds., Lecture Notes in Artificial Intelligence, **1867**, 2000, pp. 292-305.
3. G. Birkhoff: *Lattice Theory.* AMS Colloquium Publications XXV, Providence, Rhode Island, 3rd edition 1967.
4. L. Chaudron and N. Maille, Generalized Formal Concept Analysis, in *Proc. 8th Int. Conf. on Conceptual Structures, ICCS'2000*, G. Mineau and B. Ganter, Eds., Lecture Notes in Artificial Intelligence, **1867**, 2000, pp. 357-370.
5. B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, 1990.
6. M. Erné, *Einführung in die Ordnungstheorie*, Mannheim, B.I.-Wissenschaftsverlag, 1982.
7. S. Férré and O. Ridoux, A Logical Generalization of Formal Concept Analysis, in *Proc. 8th Int. Conf. on Conceptual Structures, ICCS'2000*, G. Mineau and B. Ganter, Eds., Lecture Notes in Artificial Intelligence, **1867**, 2000.
8. V. K. Finn, Plausible Reasoning in Systems of JSM Type, *Itogi Nauki i Tekhniki, Seriya Informatika*, **15**, 54-101, 1991 [in Russian].
9. B. Ganter and K. Reuter, Finding all closed sets: a general approach, *Order*, **8**, 283-290, 1991.
10. B. Ganter and S. O. Kuznetsov, Stepwise Construction of the Dedekind-MacNeille Completion, *Proc. 6th Int. Conf. on Conceptual Structures, ICCS'98*, M-L. Mugnier, M. Chein, Eds., Lecture Notes in Artificial Intelligence, **1453**, 1998, pp. 295-302.
11. B. Ganter and S. Kuznetsov, Formalizing Hypotheses with Concepts, *Proc. 8th Int. Conf. on Conceptual Structures, ICCS'2000*, G. Mineau and B. Ganter, Eds., Lecture Notes in Artificial Intelligence, **1867**, 2000, pp. 342-356.
12. B. Ganter and R. Wille, Formal Concept Analysis. Mathematical Foundations, Berlin, Springer, 1999.

13. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York, Freeman, 1979.
14. R. Gugisch, Lattice Contexts: A Generalization in Formal Concept Analysis, unpublished manuscript, 2000.
15. S. O. Kuznetsov, JSM-method as a Machine Learning System, *Itogi Nauki Tekhn., ser. Informatika*, no. 15, 17-54, 1991 [in Russian].
16. S. O. Kuznetsov, Learning of Simple Conceptual Graphs from Positive and Negative Examples, in Proc. *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD'99*, J. Zytkow, J. Rauch, Eds., Lecture Notes in Artificial Intelligence, **1704**, 1999, pp. 384-392.
17. M. Liquiere and J. Sallantin, Structural Machine Learning with Galois Lattice and Graphs, *Proc. Int. Conf. Machine Learning ICML'98*, 1998.
18. M.-L. Mugnier and M. Chein, Représenter des connaissances et raisonner avec des graphes, Revue d'Intelligence Artificielle, **10**(1), 1996, pp. 7-56.
19. M.-L. Mugnier, Knowledge Representation and Reasonings Based on Graph Homomorphisms, in *Proc. 8th Int. Conf. on Conceptual Structures, ICCS'2000*, G. Mineau and B. Ganter, Eds., Lecture Notes in Artificial Intelligence, **1867**, 2000, pp. 172-192.
20. S. Prediger, Simple Concept Graphs: A Logic Approach, in *Proc. 6th Int. Conf. on Conceptual Structures, ICCS'98*, M.-L.Mugnier, M. Chein, Eds., Lecture Notes in Artificial Intelligence, **1453**, 1998, pp. 225-239.
21. J. F. Sowa, *Conceptual Structures - Information Processing in Mind and Machine*, Reading, M.A.: Addison-Wesley, 1984.