Learning of Simple Conceptual Graphs from Positive and Negative Examples

Sergei O. Kuznetsov

All-Russia Institute for Scientific and Technical Information, Moscow, Russia
serge@viniti.msk.su
Institut für Algebra, Technische Universität Dresden, Dresden, Germany
kuznetso@math.tu-dresden.de

Abstract. A learning model is considered in terms of formal concept analysis (FCA). This model is generalized for object represented by sets of graphs with partially ordered labels of vertices and edges (these graphs can be considered as simple conceptual graphs). An algorithm that computes all concepts and the linear (Hasse) diagram of the concept lattice in time linear with respect to the number of concepts is presented. The linear diagram gives the structure of the set of all concepts with respect to the partial order on them and provides a useful tool for browsing or discovery of associations (implications) in data mining.

1 Introduction

In this paper we propose an efficient algorithmic framework for learning simple conceptual graphs and diagrammatic representation of the space of these graphs, which may be used for solving data mining problems. We consider a model of learning from positive and negative examples in terms of formal concept analysis (FCA) [Wille 1982], [Ganter 1999]. FCA proved to be a helpful mathematical framework for various branches of knowledge processing, including conceptual clustering, browsing retrieval [Carpineto 1996], and generation of association rules in data mining [Pasquier 1998]. We show how this model can be extended to data more general than classical contexts. To this end, we give a definition of the closure operation based on an arbitrary semilattice. Classical binary contexts [Wille 1982] are obtained when semilattice is taken to be a Boolean lattice. As an example we consider a semilattice induced by a set of graphs with partial ordered labels of vertices. These graphs can be interpreted, for example, as molecular graphs or as conceptual graphs [Sowa 1984], [Mugnier 1995] without negation and nestedness, i.e., as simple conceptual graphs. We present some results on algorithmic complexity of generating concepts and concept lattice for generalized closure operation $''$. The problem of computing the number of all concepts is #P-complete, however, an algorithm that constructs linear (Hasse) diagram of the concept lattice in time linear with respect to the number of concepts can be proposed. This result improves the quadratic worst-time time bounds for algorithms given in [Guènoche 1990], [Skorsky 1992], [Godin 1995], and [Carpineto 1996].

2 Learning from Examples in Formal Concept Analysis

In general terms, the model proposed in [Finn 1991] is based on the common paradigm of machine learning: given positive and negative examples of a concept, construct a generalization of the positive examples that would not cover any negative example. First, we present a particular case of the model, which can easily be described in terms of FCA. The following definition recalls some well-known notions from [Wille 1982], [Ganter 1999].

Definition 1. Let G be a set of objects, M be a set of attributes, and I be a relation defined on $G \times M$: for $g \in G$, $m \in M$, gIm holds iff the object g has the attribute m, the triple $K = (G, M, I)$ is called a *context*. If $A \subseteq G$, $B \subseteq M$ are arbitrary subsets, then the *Galois connections* $s : G \mapsto M$ and $t : M \mapsto G$ are given in the following way:

$$A' \rightleftharpoons \{m \in M | gIm \text{ for all } g \in A\}, \quad B' \rightleftharpoons \{g \in G | gIM \text{ for all } m \in B\}.$$

The pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *concept (of the context K)* with *extent* A and *intent* B. The set of attributes B *is implied by the set of attributes* A, or *implication* $A \to B$ holds, if all objects from G that have all attributes from the set A also have all attributes from the set B, i.e., $A' \subseteq B'$. $\diamondsuit$

Now assume that W is a *functional* (goal) property of objects from a domain under study. For example, in pharmacological applications [Finn 1991] W can be a biological activity of chemical compounds (like carcinogenicity or, to the contrary, some useful pharmacological activity like sedativity). Thus, W is opposed to the attributes from M, which correspond to structural properties of objects. For example, in pharmacological applications the structural attributes can correspond to particular subgraphs of the molecular graphs of chemical compounds.

Input data for learning can be represented by the sets of positive, negative, and undefined examples. *Positive examples* are objects that are known to have the property W and *negative examples* are objects that are known not to have this property. *Undefined examples* are those that are neither known to have the property nor known not to have the property. The results of learning are supposed to be rules used for the classification of undefined examples (or forecast of property W).

In terms of formal concept analysis, this situation can be described by three contexts: a positive context $K_+ = (G_+, M_+, I_+)$, a negative context $K_- = (G_-, M_-, I_-)$, and an undefined one $K_\tau = (G_\tau, M_\tau, I_\tau)$. Here G_+, G_-, and G_τ are sets of positive, negative, and undefined examples, respectively; M is a set of *structural* attributes; $I_j \subseteq G_j \times M$, $j \in \{+, -, \tau\}$, are relations that specify the structural attributes of positive, negative, and undefined examples. Now, a positive hypothesis from [Finn 1983, 1991] can be defined in the following way.

Definition 2. Consider a positive context $K_+ = (G_+,\ M_+, I_+)$ and a negative context $K_- = (G_-,\ M_-, I_-)$. A pair (e_+, i_-) is a *a positive concept* if it is a concept of the context K_+. If intent i_+ of a positive concept (e_+, i_+) is not contained in the intent of any negative concept (i.e., $\forall g \in G_-$, $i_+ \not\subseteq \{g\}'$) and $|e_+| \geq 2$, then the concept (e_+, i_+) is called a *positive hypothesis with respect to the property* W. Negative hypotheses are defined dually.

Thus, a hypothesis is an implication with a fixed consequent and antecedent equal to the intent of a positive concept. Note that if (e_+, i_+) is a positive hypothesis with respect to the property W, then $i_+ \to W$ is an implication for the context $K_{+-} = (G_+ \cup G_-, M \cup \{W\}, I_+ \cup I_- \cup G_+ \times \{W\})$. Hypotheses can be used for the classification of undefined examples from G_τ (i.e., for forecasting whether they have the property W or not). If an undefined example $g_\tau \in G_\tau$ has all attributes from the intent i_+ of a positive hypothesis (e_+, i_+) (i.e., $\{g_\tau\}' \supseteq i_+$) and does not have all attributes from the intent of any negative hypothesis, then g_τ is *classified positively. Negative classifications* are defined dually. If $\{g_\tau\}'$ does not include an intent of any negative or positive hypothesis, or includes intents of hypotheses of different signs, then no classification is made.

Example 1. Consider the following sets of positive and negative examples: $G_+ = \{X_1,\ X_2,\ X_3,\ X_4\}$, $G_- = \{Y_1,\ Y_2,\ Y_3,\ Y_4\}$, and the undefined example g_τ, where $X_1' = \{A, B, C\}$, $X_2' = \{A, B, D\}$, $X_3' = \{A, E, F\}$, $X_4' = \{A, C, G\}$; $Y_1' = \{A, F, G\}$, $Y_2' = \{A, D, F\}$, $Y_3' = \{B, E, F, G\}$, $Y_4' = \{B, D, F\}$, $g_\tau' = \{A, B, D, E\}$.

The pairs $(\{X_1, X_2\}, \{A, B\})$, $(\{X_1,\ X_4\}, \{A, C\})$ are positive hypotheses. The pair $(\{X_1, X_2, X_3, X_4\}, \{A\})$, which is a positive concept with extent larger than one, is not a positive hypothesis, since $\{A\} \subset Y_1', Y_2'$. The negative hypotheses are $(\{Y_3, Y_4\}, \{B, F\})$, $(\{Y_2, Y_4\}), \{D, F\})$, $(\{Y_1, Y_3\}, \{F, G\})$. Since $\{A, B\}$, the intent of the first positive hypothesis is contained in $\{A, B, D, E\}$, the intent of the undefined example, whereas no negative intent does, the undefined example $\{g_\tau\}$ is classified positively. $\diamondsuit$

3 Extension of the Learning Model

In this section we use a simple construction from [Kuznetsov 1991], where a partial order on graphs with ordered labels of vertices and edges is completed to a semilattice (actually to a distributive lattice).

Let Ω_g be a set of graphs with partially ordered labels of vertices and edges. For molecular graphs this can correspond to some natural hierarchy of classes of chemical elements.

Suppose that two graphs $F = \langle\langle V_F, M_F\rangle, E_F\rangle$ and $G = \langle\langle V_G, M_G\rangle, E_G\rangle$ from Ω_g are given. Here V_F, V_G are sets of vertices, M_F, M_G are sets of vertices labels, E_F, E_G are sets of edges, respectively, $E_F \subseteq V_F \times V_F$; $E_G \subseteq V_G \times V_G$. The sets m_F and m_G belong to a set of labels ordered with respect to some ordering $\leq$. We shall consider only vertex labeling. The case with labeled edges and vertices can be reduced to the case where only vertices are labeled.

Definition 3. F subsumes G or $G \preceq F$ iff there exists one-to-one mapping φ from the set V_G into the set V_F that maps each vertex $v_G \in V_G$ with label m_G to a vertex $v_F \in V_F$ with label m_F such that $m_G(v_G) \leq m_F(v_F)$. The mapping should not violate incidence relation, i.e. if $(a, b) \in E_G$, then $(\varphi(a), \varphi(b)) \in E_F$.$\diamondsuit$

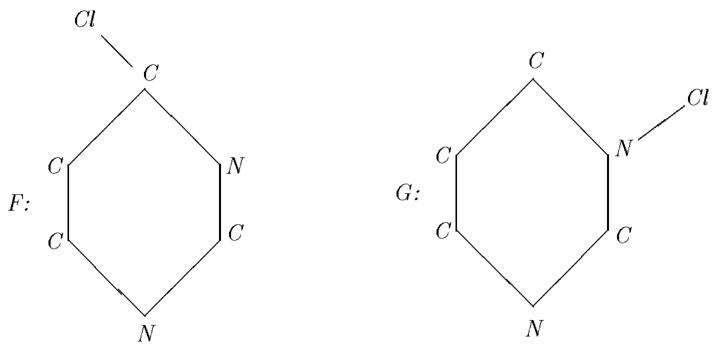
The graphs of the aforementioned form can be interpreted as conceptual graphs [Sowa 1984], [Mugnier 1995] and $\preceq$ as the specialization relation [Mugnier 1995]. The relation $\preceq$ is a generalization of the "subgraph isomorphism" relation and coincides with it when labels are not ordered.

Definition 4. Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ and $G_1, \ldots, G_n \in \Omega_g$. Then $\mathrm{N}(\mathcal{G}) \rightleftharpoons \{G_i | \forall G_j \in \mathcal{G},\, G_i \preceq G_j \rightarrow G_j = G_i\}$.

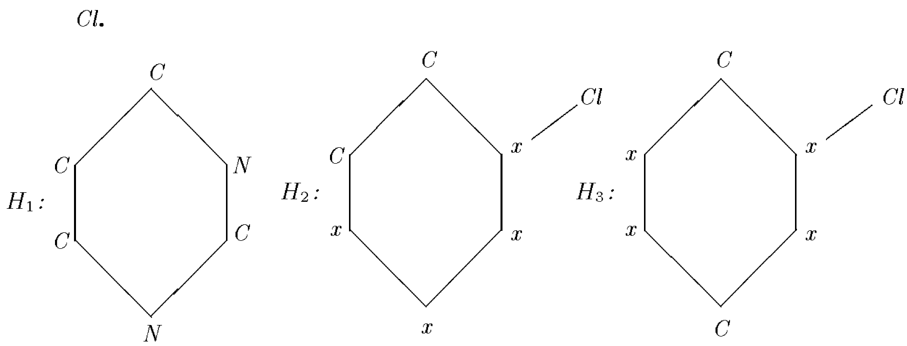
For $i \neq j$ $\{G_i\} \sqcap \{G_j\} \rightleftharpoons \mathrm{N}(\{H : H \preceq G_i, H \preceq G_j\})$ (the set $\{G_i\} \sqcap \{G_j\}$ consists of all graphs maximal by inclusion among those subsumed by both G_i and G_j.

Let $\mathcal{G} = \{G_1, \ldots, G_n\}$ and $\mathcal{F} = \{F_1, \ldots, F_m\}$ be sets of graphs from Ω_g. Then $\mathcal{G} \sqcap \mathcal{F} = \mathrm{N}(\bigcup_{i,j} \{G_i\} \sqcap \{F_j\})$, $\mathcal{G} \sqcup \mathcal{F} = N(\mathcal{G} \cup \mathcal{F})$. $\diamondsuit$

Example 2. Let F and G be molecular chemical graphs of the following form:



where the vertex labels are ordered as follows (x denotes arbitrary element): $x \leq C$, $x \leq N$, $x \leq Cl$. Other pairs of vertex labels are incomparable. Then $\mathcal{F} \sqcap \mathcal{G} = \{H_1, H_2, H_3\}$, where H_1, H_2, and H_3 are as follows:



Here, 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".

It is easily seen that operation $\sqcap$ induces a semilattice (i.e., it is idempotent, commutative, and associative). Thus, the order relation $\sqsubseteq$ can be defined as usual: $X \sqsubseteq Y \rightleftharpoons X \sqcap Y = X$. By Ω_g^N we denote the set of all possible results of application of operation $\sqcap$ to sets of graphs from Ω_g. Now we define operations $'$ and $''$ analogous to those from Section 2.

Definition 5. Let $G_1, \ldots, G_k \in \Omega$, then

$$(\{G_1, \ldots, G_k\})' \rightleftharpoons G_1 \sqcap \ldots \sqcap G_k, (\{G_1, \ldots, G_k\})'' \rightleftharpoons \{G \in \Omega_g : G \sqsupseteq G_1 \sqcap \ldots \sqcap G_k\}.$$

It can be shown that $''$ is a closure operation (i.e., it is extensive, idempotent, and monotone). For arbitrary $X_1, \ldots, X_k \in \Omega_g$ the pair $((\{X_1, \ldots, X_k\})', (\{X_1, \ldots, X_k\})'')$ is called a *concept* with *extent* $(\{X_1, \ldots, X_k\})''$ and *intent* $(\{X_1, \ldots, X_k\})'$.

The closure operator $''$ and concept from Definition 1 can be obtained from Definition 5 when Ω_g is a set of objects G, $\sqcap$ is the set-theoretic intersection $\cap$ defined on subsets of the set of attributes M, all terms for objects from Ω_g in the right-hand sides that occur in expressions with $\sqsubseteq$ and $\sqcap$ are replaced by their intents (in the left hand-sides and in the expression $G \in \Omega_g$, the terms G and $G_1, \ldots, G_n$ will still denote objects). Note that the operation $\sqcap$ and the corresponding $''$ can be defined in lines of Definitions 4-5 for arbitrary partial orders (and thus, data types), not only for those given in Definition 3.

4 Algorithms and Complexity

A crucial problem here is that of the generation of the set of all concepts of a given context. It is difficult not only to generate the set of all concepts, whose size can be exponential in the size of the source context, but also to calculate or even estimate its size, since the problem of computing all concepts is $\#P$-complete [Kuznetosv 1989].

Now we describe an algorithm, which is similar to the algorithm from [Ganter 1986]. This algorithm generates the set of all concepts. We show how this algorithm can be transformed to result in an algorithm generating Hasse diagram in time linear with respect to the size of the source context.

We assume that all objects from G are numbered, and therefore, a set $X \subseteq G$ can be represented by a correspondingly ordered tuple. The numbering of objects from G induces lexicographical ordering of sets from $\mathcal{P}(G)$, the powerset of G.

Definition 6. A *path* is defined inductively as follows:

(1) If $g \in G$ and $\{g\}'' = \{g\} \cup Z$, $g \notin Z$, $Z \subseteq G$, then $[(\emptyset, \{g\})Z]$ is called a *path* and $\{g\}''$ is called the *extent* of the path or $\mathrm{Ext}[(\emptyset, \{g\})Z]$. We also say that $[(\emptyset, \{g\})Z]$ is an *inference* of $\{g\}''$. The inference $[(\emptyset, \{g\})Z]$ of $\{g\}''$ is *canonical* (or the path $[(\emptyset, \{g\})Z]$ is *canonical*) iff the numbers of all objects from Z (in the sense of numbering of objects from G) are greater than the number of g.

(2) If Y is a path, $h \in G$, $(\mathrm{Ext}(Y) \cup \{h\})'' = \mathrm{Ext}(Y) \cup \{h\} \cup Z$, $Z \cap Y = \emptyset$, then $[(Y, \{h\})Z]$ is a path. $\mathrm{Ext}[(Y, \{h\})Z] \rightleftharpoons (\mathrm{Ext}(Y) \cup \{h\})'' = \mathrm{Ext}(Y) \cup \{h\} \cup Z$. We say that $[(Y, \{h\})Z]$ is an inference of $(\mathrm{Ext}(Y) \cup \{h\})''$. The inference $[(Y, \{h\})Z]$

of $(\mathrm{Ext}(Y) \cup \{h\})''$ is called *canonical* iff Y is a canonical path and the numbers of all objects from Z are greater than the number of h. $\diamondsuit$

The following procedure (we call it Close-by-One or CbO Algorithm) is based upon the depth-first strategy, though other strategies are possible as well. Y denotes the path to the current concept.

Algorithm 1.
Step 0. There is only one root vertex where all objects are unlabeled, $Y := \emptyset$.
Step 1. The current vertex corresponds to the concept with the extent Y. The first unlabeled object from G, say X_i, is taken and labeled at Y, $(\mathrm{Ext}(Y) \cup \{X_i\})'$ and $(Y \cup \{X_i\})'' = (\mathrm{Ext}(Y) \cup \{X_i\} \cup Z$ are computed. A new vertex that corresponds to $(\mathrm{Ext}(Y) \cup \{X_i\})''$ is generated and connected to the vertex associated with Y.
Step 2. If Z contains objects with numbers less than i (i.e., the path $[(Y, X_i)Z]$ is *not canonical*), then we label all objects from G at the vertex $(\mathrm{Ext}(Y) \cup \{X_i\})''$ (thus, the branch will not be extended). If Z does not contain objects with numbers less than i (i.e., $[(Y, X_i)Z]$ is *canonical*), then we label all objects from $(\mathrm{Ext}(Y) \cup \{X_i\})'' \cup \{X_1, \ldots, X_{i-1}\}$ at the vertex $(\mathrm{Ext}(Y) \cup \{X_i\})''$.
Step 3. If all elements of G are labeled at $(\mathrm{Ext}(Y) \cup \{X_i\})''$, we go to Step 4. Otherwise, $Y := [(Y, X_i)Z]$, and we return to Step 1.
Step 4. We backtrack the tree upwards to the nearest vertex with unlabeled elements of G. If such a vertex exists and corresponds, say, to the path Z, then $Y := Z$ and we go to Step 1. If such a vertex does not exist, then this means that all concepts have been generated and the algorithm halts. $\diamondsuit$
Example 3. Consider objects X_1, X_2, X_3, X_4 from Example 1. In this case, Algorithm 1 constructs the tree with the following left-most branch: root – $[(X_1)X_2]$ – $[([(X_1)X_2]X_3)X_4]$, which consists of two non-root vertices. Both these vertices are canonical. $\diamondsuit$
Theorem 1. *The tree output by Algorithm 1 has $O(|G||L|)$ vertices. The set of canonical vertices of this tree are in one-to-one correspondence with the set of concepts. The time complexity of Algorithm 1 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 Ω_g^N. When contexts and concepts are given by Definition 1, the time complexity is $(|M||G|^2|L|)$ and the space complexity is $O(|M||G||L|)$.*$\diamondsuit$

The computation of $\sqcap$ and $\sqsubseteq$ may be in general, NP-hard, but in some reasonable cases may be polynomially tractable [Mugnier 1995].

Unlike the order on tree vertices, the corresponding incidence relation with that of the Hasse diagram of the concept lattice. To construct the diagram, we need to connect each pair of vertices in the tree that correspond to adjacent vertices in the diagram. To this end, we run the following algorithm in the depth-first left-most order.

Algorithm 2
Step 0. We are in the root vertex of the tree constructed by Algorithm 1 (CbO tree). $Y := \emptyset$, $\mathrm{Fr}(C) := \emptyset$, and $\mathrm{To}(C) := \emptyset$ for all concepts C. All vertices are unlabeled.

Step 1. The current canonical vertex corresponds to the concept with the extent Y. For each element X_i of G we compute $(Y \cup \{X_i\})'$ and $(Y \cup \{X_i\})''$. Among sets $(Y \cup \{X_i\})''$, $i = 1, \ldots, |G|$, we select those minimal by inclusion. These are extents of concepts adjacent to (Y, Y') from below. We denote the set of these extents by $M(Y)$. $\mathrm{Fr}(Y') := \{\langle (Y, Y'), (Z, Z') \rangle | Z \in M(Y)\}$. Thus, the set of arcs in the Hasse diagram leading from the vertex (Y, Y') to its children is constructed. We label the vertex Y and number the elements of $M(Y)$ using the numbering of G).
Step 2. For every extent $E \in M(Y)$ we take the corresponding concept (E, E') and find its canonical inference. This is equivalent to finding the corresponding canonical path in the tree generated by Algorithm 1. We update the set of arcs leading to (E, E') by letting $\mathrm{To}(E') := \mathrm{To}(E) \cup \{\langle (E, E'), (Y, Y') \rangle\}$.
Step 3. If there are unlabeled canonical vertices corresponding to extents in $M(Y)$, we pass from Y to the first of them (with respect to the numbering on elements of $M(Y)$ given in Step 2), say Z, $Y := Z$ and return to Step 1. If there are no such vertices and Y is not the root of the tree, we backtrack to the parent of Y in the tree, denote it by $R(Y)$, $Y := R(Y)$ and return to Step 3. If Y is the root of the tree and there are no unlabeled vertices corresponding to extents in $M(Y)$, then algorithm halts. $\diamondsuit$

For every concept C Algorithm 2 outputs sets $\mathrm{Fr}(C)$ and $\mathrm{To}(C)$ of arcs that lead to concepts immediately adjacent to C in the Hasse diagram from above and below, respectively. Thus, the CbO tree, together with sets $\mathrm{Fr}(C)$ and $\mathrm{To}(C)$ related to each canonical vertex is a representation of the concept lattice. Unlike the incidence matrix of a lattice, whose size is quadratic in the number of concepts, the size of this structure and time needed to construct it are linear. Given also a negative context at the input, one can generate hypotheses by slightly modifying Algorithm 2: at Step 3 it should be tested whether positive intents are not subsumed by negative examples. The algorithm is also easily extended to include a test for sufficient number of examples supporting a hypothesis, for example, in lines of [Pasquier 1998].

Theorem 2. *Algorithm 2 constructs the Hasse diagram of a concept lattice in* $O((\alpha|G| + \beta|G|^2)|L|)$ *time and* $O((\gamma|G||L|)$ *space, where* $|L|$ *is the number of concepts,* α *is the time needed to perform* $\sqcap$ *operation,* β *is the time needed to test* $\sqsubseteq$ *relation, and* γ *is the space needed to store the largest object from* Ω_g^N. *When the context is given as in Definition 1 the diagram is constructed in* $O(|M||G|^2|L|)$ *time and* $O(|M||G||L|)$ *space.*

5 Conclusion

We presented a learning model in a version of the formal concept analysis that allows processing graph structures. This model can be used for learning implications, e.g., on simple conceptual graphs [Sowa 1984] or molecular graphs. The model can be also extended to arbitrary data structures with partial order. Algorithmic analysis was provided. Though computations on graphs may be hard in general, this does not affect the linear dependence of time and space needed for the computatio on the number of resulting concepts (hypotheses, implications).

6 Acknowledgments

This work was supported by the Russian Foundation for Basic Research, project no. 98-06-80191 and the Alexander von Humboldt Foundation.

7 References

[Carpineto 1996] C. Carpineto and G. Romano, A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval, *Machine Learning*, **24**, 95-122, 1996.

[Finn 1983] V.K. Finn, On Machine-Oriented Formalization of Plausible Reasoning in the Style of F. Backon–J. S. Mill, *Semiotika Informatika,* No. 20, 35-101, 1983 [in Russian].

[Finn 1991] V.K. Finn, Plausible Reasoning in Systems of JSM Type, *Itogi Nauki i Tekhniki, ser. Informatika*, **15**, 54-101, 1991.

[Ganter 1987] B. Ganter, Algorithmen zur Formalen Begriffsanalyse, in: *Beiträge zur Begriffsanalyse,* (B. Ganter, R. Wille, and K. E. Wolff, eds.), B. I. Wissenschaftsverlag, Mannheim, 241-254, 1987.

[Ganter 1999] B. Ganter and R. Wille, Formal Concept Analysis. Mathematical Foundations, Berlin, Springer, 1999.

[Godin 1995] R. Godin, R. Missaoui, and H. Alaoui, Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. Computational Intelligence, **11**(2), 246-267, 1995.

[Guénoche 1990] A. Guénoche, Construction du treillis de Galois d'une relation binaire, *Math. Sci. Hum.*, **95**, 5-18, 1990.

[Kuznetsov 1989] S.O. Kuznetsov, Interpretation on Graphs and Algorithmic Complexity Characteristics of a Search of Specific Patterns, *Automatic Documentation and Mathematical Linguistics*, **23**, no. 1, 37-45, 1989.

[Kuznetsov 1991] S.O. Kuznetsov, JSM-method as a Machine Learning System, *Itogi Nauki Tekhn., ser. Informatika*, no. 15, 17-54, 1991 [in Russian].

[Mugnier 1995] M.L. Mugnier, On Generalization/Specialization for Conceptual Graphs, *J. Exp. Theor. Artif. Intel.*, **7**, pp. 325-344, 1995.

[Pasquier 1998] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal, Pruning Closed Itemset Lattices for Association Rules, *Proc. 14th BDA Conference on Advanced Databases (BDA'98),* pp. 177-196, Hamammet, October 1998.

[Skorsky 1992] M. Skorsky, Endliche Verbände - Diagramme und Eigenschaften. Shaker, 1992.

[Sowa 1984] J.F. Sowa, Conceptual Structures - Information Processing in Mind and Machine, Reading, M.A.: Addison-Wesley, 1984.

[Wille 1982] R. Wille, Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts, in: *Ordered Sets* (I. Rival, ed.), Reidel, Dordrecht–Boston, 445-470, 1982.