Automation and Remote Control, Vol. 62, No. 10, 2001, pp. 1543–1564. Translated from Avtomatika i Telemekhanika, No. 10, 2001, pp. 3–27.

REVIEWS

Machine Learning on the Basis of Formal Concept Analysis

S. O. Kuznetsov

All-Russia Institute for Scientific and Technical Information (VINITI), Moscow, Russia
Received June 1, 2001

Abstract—A model of machine learning from positive and negative examples (JSM-learning) is described in terms of Formal Concept Analysis (FCA). Graph-theoretical and lattice-theoretical interpretations of hypotheses and classifications resulting in the learning are proposed. Hypotheses and classifications are compared with other objects from domains of data analysis and artificial intelligence: implications in FCA, functional dependencies in the theory of relational data bases, abduction models, version spaces, and decision trees. Results about algorithmic complexity of various problems related to the generation of formal concepts, hypotheses, classifications, and implications.

1. INTRODUCTION

Machine Learning in broad sense is the domain related to mathematical models and practical methods of adaptation of machine systems. More precise definitions depend on the type of a learning system, its goal and data types it processes [56]. A typical refinement of the idea of learning is learning from positive and negative examples, when a learning system obtains descriptions of positive and negative examples and constructs a generalization of positive examples that does not "cover" negative examples. In this review we consider a realization of this idea within the framework of the JSM-method of hypothesis generation and Formal Concept Analysis (FCA), as well as its relation with well-known models of data analysis and artificial intelligence.

JSM-method[1] was formulated by V.K. Finn [16, 20, 19] as a theory of plausible inference on the basis of infinite-valued first-order logical theory with quantifiers over tuples of variable length [20]. Within the framework of this theory it is possible to realize and justify inductive and abductive inference of regularities. On the other hand, the realization of the JSM-method and the practice of its application allows one to consider it as a method of machine learning [10, 12]. In contrast to the majority of methods of machine learning where similarity is given as a metric or as a relation [12], the JSM-method uses similarity defined as an algebraic operation with certain properties. The main principle that underlies methods of machine learning, besides finding a decision rule that separates positive and negative examples, is the principle of constructing a minimal cover of positive examples (which does not cover negative examples, due to the principle of separating positive and negative examples, see [53], [56]). Whereas the idea of separating positive and negative examples is retained, the JSM-method use not the principle of minimal coverage, but the principle of finding maximal similarity of these examples. Without underestimating the value of the idea of coverage (which gives the lower bound of the set of all possible learned rules or classifiers in the sense of subsumption relation on classifiers), we underline that the principle of maximal similarity, which is in a certain sense dual to the principle of minimal coverage, indicates another (upper) bound of the set of all possible classifiers ordered by logical strength (subsumption order). The relation between

[1] Called so in honor of English philosopher and logician John Stuart Mill who proposed his schemes of inductive reasoning in the 19th century.

0005-1179/01/6210-1543$25.00 © 2001 MAIK "Nauka/Interperiodica"

these bounds can be characterized in terms of most general and least general generalizations [60, 61], as well as version spaces [54, 55, 44, 45]. The definition of hypotheses and classification by means of similarity operation has a natural graph-theoretical interpretation in terms of tri- and quadro-partite graphs and their bipartite subgraphs with certain properties.

FCA [34] started from the paper of R. Wille [73] and developed further in his school. The fundamental fact underlying FCA is the representability of complete lattices by ordered sets of their meet- and join-irreducibles (indication to this fact for finite lattices was made in [22]). Since ordered sets of irreducibles are naturally represented by binary matrices, this makes it possible to apply certain aspects of the lattice theory to the analysis of data given by object-attribute matrices. Establishing a relation between JSM-hypotheses and formal concepts (i.e., elements of the lattice generated by the object-attribute matrix, see below) resulted in the interaction of the FCA theory and the theory of JSM-method [47, 14, 37, 38]. On the one hand, it became possible to introduce ideas of learning from positive and negative examples and extend some combinatorial and algorithmic-complexity results to FCA and lattice theory. On the other hand, the fundamental fact that formal concepts induce a lattice allows one to consider JSM-hypotheses in terms of lattice theory. Besides mathematical elegance, the algebraic formalization helps solving certain problems related to generation of hypotheses and classifications, hierarchical organization and vizualization of the set of all hypotheses, as well as navigation in it.

The paper has the following structure. In Section 2 we give main definitions from the FCA, order and lattice theory. In Section 3 definitions of hypotheses and classifications are given in terms of FCA, examples are given. In Sections 4 and 5 we consider graph-theoretical and lattice-theoretical interpretations of hypotheses and classifications. In Section 6 we discuss the relation of hypotheses to well-known constructions from data analysis and artificial intelligence: implications in FCA [34], functional dependencies in the theory of relational data bases [15], abduction models [25], version spaces [55] and decision trees [64]. In Section 7 we give a review of results concerning complexity of computing the set of formal concepts, hypotheses, classifications, and implication bases.

2. FORMAL CONCEPT ANALYSIS

First we recall the main definitions from the theories of ordered sets and lattices [1, 3]. An *ordered set* $(P, \leq)$ consists of a nonvoid set P and a binary relation $\leq$ on A that satisfies the following properties for all $x, y, z \in P$:

1. $x \leq x$ (reflexivity);
2. $x \leq y$ and $y \leq x$ imply $x = y$ (antisymmetry);
3. $x \leq y$ and $y \leq z$ imply $x \leq z$ (transitivity).

Ordered sets are often called *partially ordered sets*. The fact that $x \leq y$ and $x \neq y$ is denoted by $x < y$ (x is *strictly less* than y. The element $a \in P$ is said to *cover* $b \in P$ (denoted by $a \succ b$) if $a > b$ and there does not exist an element $x \in P$ such that $a > x > b$. Assume that a system of coordinates with vertical and horizontal axes is given on the plane R^2. The *diagram* of an ordered set is a plane embedding of the covering relation of $(P, \leq)$ such that for the pair $a \geq b$ the vertical coordinate of a is larger than the vertical coordinate of b. Since in general case the covering graph is not planar, an embedding allows for overlapping of arcs. In the literature the terms *Hasse diagram* and *line diagram* [34] are also often used.[2] In this paper we shall use the term "diagram" following [3] and [23].

The *upper bound* of a subset X in the ordered set P is an element $a \in P$ such that $a \geq x$ for all $x \in P$. The *least upper bound* of the set X (denoted by sup$X$) is an upper bound a such that $a \leq b$

[2] R. Wille notes that the term *Hasse diagram* is incorrect since objects of this type were considered by Birkhoff earlier than by Hasse.

for any upper bound b of X. The notion of the *greatest lower bound* (denoted by inf$X$) of the set X is defined dually by replacing $\geq$ with $\leq$. A *lattice* is an ordered set L where any two elements x and y have a greatest lower bound or *meet* (denoted by $x \wedge y$) and a lowest upper bound or *join* (denoted by $x \vee y$). The lattice L is called *complete* if any subset $X \subseteq L$ has join and meet in L. A *semilattice* is an ordered set where each two elements have a meet (this is called a lower semilattice) or any two elements have an upper bound (upper semilattice). A *sublattice* of the lattice L is a subset $X \subset L$ such that $a \in X$ and $b \in X$ imply $a \wedge b \in X$ and $a \wedge b \in X$. The *interval* $[a, b]$ consists of all elements $x \in X$ that satisfy inequalities $a \leq x \leq b$. An *order filter* (*order ideal*) of a lattice L is a subset $X \subset L$ such that $a \in X$ and $b \geq a$ imply $b \in X$ ($a \in X$ and $b \leq a$ imply $b \in X$, respectively).

Binary operations $\wedge$ and $\vee$ in lattices have the following properties:

1. $x \wedge x = x$, $x \vee x = x$ (idempotency);
2. $x \wedge y = y \wedge x$, $x \vee y = y \vee x$ (commutativity);
3. $x \wedge (y \wedge z) = (x \wedge y) \wedge z$, $x \vee (y \vee z) = (x \vee y) \vee z$ (associativity);
4. $x \wedge (x \vee y) = x \vee (x \wedge y) = x$ (absorption).

In semilattices the properties 1–3 hold for the corresponding operations, i.e., the operation $\wedge$ is idempotent, associative, and commutative in a *lower semilattice* and the operation $\vee$ is idempotent, associative, and commutative in the *upper semilattice*.

The following notions play the key role for the relation of the lattice theory and FCA. The element of a lattice a is called $\wedge$-*irreducible* (*meet-irreducible*) if $a = b \wedge c$ implies $a = b$ or $a = c$. $\vee$-*irreducibility* (or *join-irreducibility*) of the element a takes place when $a = b \vee c$ implies $a = b$ or $a = c$. The sets of all join- and meet-irreducible elements of the lattice L are denoted by $J(L)$ and $M(L)$, respectively.

A subset D of a complete lattice L is called *infimum-dense* (*supremum-dense*) if $L = \{\wedge_{x \in X} x | X \subseteq D\}$ ($L = \{\vee_{x \in X} x | X \subseteq D\}$, respectively).

Now we proceed to definitions from FCA.

Let G and M be sets that are called the sets of objects and attributes, respectively, and let $I \subseteq G \times M$ be a relation. This relation is interpreted as follows: for $g \in G$ and $m \in M$ we have gIm if the object g has attribute m. The triple $K = (G, M, I)$ is called a *formal context. The Galois connection* between the ordered sets $(2^G, \subseteq)$ and $(2^M, \subseteq)$ is given by the following mappings called derivation operators: for any $A \subseteq G$ and $B \subseteq M$

$$A' := \{m \in M \mid gIm \text{ for all } g \in A\}, \quad B' := \{g \in G \mid gIm \text{ for all } m \in B\}.$$

The pair of sets (A, B) such that $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$ is called a *formal concept of the context* K with *(formal) extent* A and *(formal) intent* B. Note that the formal definition of a concept as a pair of this form complies with the logical-philosophical tradition, e.g., it is close to the notion of a concept in *Logique de Port-Royale* (A. Arnauld, P. Nicole, France, XVII century).

As it was shown in [73, 34] (for finite sets it was shown in [22]), the set of all formal concepts of a formal context K forms a complete lattice with the following meet and join operations:

$$\bigwedge_{j \in J} (A_j, B_j) = \left(\bigcap_{j \in J} A_j, \left(\bigcup_{j \in J} B_j \right)'' \right), \quad \bigvee_{j \in J} (A_j, B_j) = \left(\left(\bigcup_{j \in J} A_j \right)'', \bigcap_{j \in J} B_j \right).$$

This lattice called a formal concept lattice (or *Galois lattice* in French and Canadian literature) is usually denoted by $\underline{\mathfrak{B}}(K)$ and the set of its elements is denoted by $\mathfrak{B}(K)$. Moreover, the following statement holds. If L is a complete lattice, then $L \cong \underline{\mathfrak{B}}(K)$ iff there exist mappings $\gamma \colon G \mapsto L$ and

$\mu\colon M \mapsto L$ such that the set γG is join-dense in L, μM is meet-dense in L, and gIm is equivalent to $\gamma g \le \mu m$ for all $g \in G$ and $m \in M$. In particular, $L \cong \underline{\mathfrak{B}}(L, L, \le)$, $L \cong \underline{\mathfrak{B}}(J(L), M(L), \le)$.

The fact that in the formal context $K = (G, M, I)$ the object g has attribute m (i.e., gIm holds) is equivalent to $(g'', g') \le (m', m'')$. Moreover, it is also equivalent to the fact that the diagram of the lattice $\underline{\mathfrak{B}}(K)$ has an upward-directed path from the vertex corresponding to the formal concept (g'', g') to the vertex corresponding to the formal concept (m', m'').

The definitions below are related to implications on attributes that are certain dependencies between subsets of attributes. The relation $A \to B$ is called an *implication (on attributes)* if $A' \subseteq B'$ (or $B \supseteq A''$), i.e., all objects from G that have the set of attributes A also have the set of attributes B. Implication on objects are defined similarly. Unless explicitly specified, by an implication we shall mean implication on attributes.

The implication $A \to B$ in the formal context K corresponds to the fact that the formal concept (A', A'') lies below than the formal concept (B', B'') in the diagram of the concept lattice $\underline{\mathfrak{B}}(K)$.

Implications of a formal context satisfy the following Armstrong axioms [34] (they were mentioned first in [21], see also [15]) for arbitrary X, Y, Z:

1. $X \to X$;
2. $X \to Y$ implies $X \cup Z \to Y$;
3. $X \to Y$ and $Y \cup Z \to W$ implies $X \cup Z \to W$.

Besides formal contexts defined above (two-valued contexts), so-called *many-valued contexts* are studied in FCA. Here attributes can take values from a set of values. Formally, a many-valued context is a quadruple (G, M, W, I), where G, M W are sets (of objects, attributes, and attribute values, respectively) and I is a ternary relation $I \subseteq G \times M \times W$ giving the value w of the attribute m, where

$$(g, m, w) \in I \text{ and } (g, m, v) \in I \text{ implies } w = v.$$

Representation of a many-valued context by a two-valued one is called *scaling*. Possible types of scaling are considered in [34].

3. JSM-METHOD

The first version of the JSM-method of automated hypotheses generation was described in [16]. Later versions can be found in [19], see also survey [10]. Plausible reasoning in the JSM-method is formalized by means of an infinite-valued first order logical theory with quantifiers over tuples of variable length. Logical aspects of the JSM-method are given in detail in [20, 19], the problems of definability of plausible JSM-reasoning are studied in [2]. As a method of data analysis, JSM-method is a system of machine learning from positive and negative examples [10, 14, 37] based on the general principle of learning from positive and negative examples [53, 62, 56]: for given positive and negative examples w.r.t. a "goal concept," one needs to construct a "generalization" of positive examples that does not "cover" any negative examples. Generalization of examples in the JSM-method is defined by means of a semilattical *similarity* operation [4, 17, 10].

Now we present the learning model in the JSM-method [19] in terms of FCA.

Let $K = (G, M, I)$ be a formal context and $w \notin M$ is a certain *goal* attribute of objects, different from attributes from the set M (the latter are called *structural*). For example, in pharmacological applications structural attributes may correspond to certain subgraphs of molecular graphs and w may correspond to their biological property.

The training sample is given by sets of positive, negative, and undetermined examples. *Positive examples* (or (+)-*examples*) are objects that are known to have the property w, *negative examples* (or $(-)$-*examples*) are objects that are known not to have the property. *Undetermined examples* (or (τ)-*examples*) are objects that are neither known to have this property nor not to have this property. It assumed that results of learning from positive and negative examples are used for the classification of undetermined examples, i.e., forecast whether the undetermined examples really have or do not have the goal property w.

In terms of FCA this situation can be described by means of three contexts: positive context $K_+ = (G_+, M, I_+)$, negative context $K_- = (G_-, M, I_-)$, and undetermined context $K_\tau = (G_\tau, M, I_\tau)$. Here G_+, G_-, and G_τ are sets of positive, negative, and undetermined examples, respectively; M is a set of *structural* attributes; $I_\varepsilon \subseteq G_\varepsilon \times M$, $\varepsilon \in \{+, -, \tau\}$ are relations that define structural attributes of positive, negative, and undetermined examples, respectively. Derivation operators in these contexts are denoted by superscripts $+, -, \tau$, respectively, e.g. A^+, A^- denote a single application of derivation operators in contexts K_+, K_-, respectively; $A^{\tau\tau}$ denotes the application of a composition of derivation operators in the context K_τ.

Now a positive hypothesis by Finn (called a "counterexample forbidding hypothesis" in [16, 19]) can be defined in the following way.

Let a positive context $K_+ = (G_+,\ M, I_+)$, negative context $K_- = (G_-,\ M, I_-)$, and undetermined context $K_\tau = (G_\tau, M, I_\tau)$ be given. The context $K_\pm = (G_+ \cup G_-, M \cup \{w\}, I_+ \cup I_- \cup (G_+ \times \{w\}))$ is called a *learning* context. The context $K_c = (G_+ \cup G_- \cup G_\tau, M \cup \{w\}, I_+ \cup I_- \cup I_\tau \cup G_+ \times \{w\})$ is called a *classification context*. Derivation operators in these two contexts are denoted by superscripts $\pm$ and c, respectively. The pair (e_+, h_+) is called a *positive concept* if it is a formal concept of the context K_+. The set $A \subseteq G_+$ $(B \subseteq M)$ is called a positive formal extent (intent) if (A, A^+) $((B^+, B)$, respectively) is a positive concept. A positive formal intent h_+ is called a *positive or* $(+)$-prehypothesis w.r.t. the property[3] w if it is not a formal intent of any negative concept (i.e., $\forall (e_-, i_-) \in \mathfrak{B}(G_-)$, $h_+ \neq i_-$). If a positive formal intent h_+ is not contained in intent of any negative example (i.e., $\forall g_- \in G_-$, $h_+ \not\subseteq \{g_-\}^-$)), then it is called a *positive or* $(+)$-hypothesis w.r.t. the property[4] w. The set e_+ is called a *formal extent* of the positive hypothesis h_+. A *falsified* $(+)$-hypothesis is a positive formal intent h such that $h \subseteq \{g_-\}^-$ for some negative example g_-. *A negative (or* $(-)$-*) hypothesis* is defined similarly.

It follows directly from the definition that hypothesis is a prehypothesis. Hypotheses are used for classification of undetermined examples from the set G_τ (i.e., for forecasting if they actually have the goal attribute w or not).

If an undetermined example $g_\tau \in G_\tau$ contains a positive hypothesis h_+ (i.e., $\{g_\tau\}^\tau \supseteq h_+$), then it is said that h_+ is a *hypothesis in favor of a positive classification* of the undetermined example g_τ. A *hypothesis in favor of a negative classification* g_τ is defined in a similar way. If there is a hypothesis in favor of a positive classification of g_τ and there is no hypothesis in favor of negative classification of g_τ, then g_τ is *classified positively.*[5] A *negative classification* of g_τ takes place in the case where there are hypotheses in favor of negative classification and there are neither hypothesis in favor of positive nor in favor of negative classification. If $\{g_\tau\}^\tau$ does not contain any negative or positive hypotheses as subsets, the classification does not take place. If $\{g_\tau\}^\tau$ contains both positive and negative hypotheses, then it is said that the classification is *contradictory.*

One can indicate a useful subset of hypotheses equivalent to the set of all hypotheses w.r.t. all possible classifications. Formally, a positive hypothesis h_+ is a *minimal positive hypothesis* if no subset $h \subset h_+$ of it is a positive hypothesis [11].

---

[3] In [16, 19] this object is called a *simple hypothesis of the first kind.*

[4] In [16, 19] it is required that $|e_+| \geq 2$, but we shall neglect this inessential condition to retain mathematical homogeneity.

[5] In papers [16], [19] the undetermined example g_τ is called a "(+)-hypothesis of the second kind."

The following definitions are related to the *generalized JSM-method* [18] that uses a more general idea of causality: a hypothetical cause of a goal property may have specific "hindrances" that inhibit the manifestation of the goal property. The pair $(h_+, \mathcal{X})$ is called a *positive generalized hypothesis with a set of hindrances* $\mathcal{X}$ if h_+ is a positive formal intent and

$$\mathcal{X} := \min\{(h_+ \cup \{m\})^{--} \mid m \in M \setminus h_+\},$$

where $\min(Y)$ takes inclusion-minimal subsets from the set Y.

The intuitive meaning of a generalized hypothesis is that h_+ is a hypothetical cause of the property w only in the absence of elements of the set $\mathcal{X}$ called *hindrances*. Note that a hypothesis is a generalized hypothesis with empty set of hindrances.

The definition of a classification by means of generalized hypotheses looks as follows. If h_+ is a subset of the intent of an undetermined example $g_\tau \in G_\tau$ and $(h_+, \mathcal{X})$ is a positive generalized hypothesis such that g_τ^τ does not contain any hindrance from $\mathcal{X}$ (i.e., $\{g_\tau\}^\tau \supseteq h_+$ and $\forall X \in \mathcal{X}$ $\{g_\tau\}^\tau \supseteq X$), then it is said that $(h_+, \mathcal{X})$ is a *generalized hypothesis in favor of a positive classification* of the undetermined example g_τ. A *generalized hypothesis in favor of a negative classification* of g_τ is defined similarly. Now the rule of classification of the undetermined example g_τ by means of generalized hypotheses may be described in the same way as above. For example, if there is a generalized hypothesis in favor of a positive classification of g_τ and there is no generalized hypothesis in favor of a negative classification of g_τ, then g_τ *is classified positively*, etc.

Another important element of the JSM-method is the *principle of sufficient ground for acceptance of hypotheses* [19], according to which the generated hypotheses with extent greater or equal than two should correctly classify initial positive and negative examples (regarded as undetermined during the classification). If this condition is not satisfied, then it is considered that the generation of hypotheses in the learning context is not justified and the set of initial examples should be updated with new ones.

The construction of the JSM-method allows one to iteratively update the sets of positive and negative examples with results of positive and negative classifications of undetermined examples. This process goes until *stabilization*, i.e., the iteration step when no new classifications can be made [19]. In practice of the JSM-method the stabilization takes place in few steps.

Example 3.1. Consider a context with results of expert analysis of 17 winter chains (information is taken from ADAC Magazin, 1999, no. 11). Hypotheses and implications of this context were considered in detail in [37].

Different chains are considered here as objects (examples), their consumer properties and features of constructions are taken as attributes.

As a goal property we took "high cost of a chain" (thus, positive examples correspond to expensive chains and negative examples correspond to cheap chains). In this paper we consider only a subset of the set of examples (four positive and three negative examples) from initial seventeen and a simplified scaling (i.e., a means of transforming a many-valued context to a two-valued one, see [34] for details) to illustrate the definition of a hypothesis and their graph-theoretic interpretation by a simple example. The values of the attribute **system** define the type of a chain: R—a rope chain, S—a steal chain, Q—a quick mounting chain. The attribute **mount** takes the values F and B that mean the ability of a chain to be mounted either on front wheels or both on front and rear wheels. All values of these attributes are incomparable.

The initial values of other attributes were numerical. The attribute **con** correspond to the average expert estimate of the chain convenience; the values **snow** and **ice** correspond to the average expert estimate of the maneuverability of a car with the chain on snow and ice, respectively; the attribute

Table 3.1. Positive context

chain	system	mount	con	snow	ice	dur	grade
2	S	B	+	+		+	+
5	Q	B	+		+	+	+
8	Q	B	+			+	+
14	Q	B	+			+	

Table 3.2. Negative context

chain	system	mount	con	snow	ice	dur	grade
1	R	F	+	+	+	+	+
3	R	F	+	+	+		+
17	R	F				+	

Table 3.3. Undetermined context

chain	system	mount	con	snow	ice	dur	grade
18	R	B	+	+	+	+	
19	R	F	+	+	+		+
20	Q	F	+			+	

dur corresponds to the average expert estimate of durability of a chain; **grade** corresponds to the average expert estimate of the chain quality. In the initial statement the smaller values of the attributes **con**, **snow**, **ice**, **dur** correspond **grade** to better estimates of the corresponding chain properties. In [37] we used some scales for turning numerical values into Boolean ones. Here we use a simpler scaling: these attributes take true values (denoted by + in the corresponding table entries) if the initial values were less than a certain threshold. The corresponding positive and negative contexts are given in Tables 3.1, 3.2. The undetermined context is given in Table 3.3.

Here {**B, con, dur**} is a minimal positive hypothesis. It is unique since the intersection of formal intents of all positive examples is not empty and does not lie in formal intent of any negative example.

Other positive hypotheses are

$$\{\mathbf{Q}, \mathbf{B},\ \mathbf{con},\ \mathbf{dur}\}, \quad \{\mathbf{B},\ \mathbf{con},\ \mathbf{dur},\ \mathbf{grade}\}.$$

These hypotheses may be useful from the standpoint of taxonomy of positive examples. For example, the first nonminimal hypothesis describes the class of chains of a certain type, same mounting possibilities, which are convenient but behave bad on snow.

A minimal negative hypothesis here is $\{\mathbf{R}, \mathbf{F}\}$. It is also unique since the intersection of formal intents of all negative examples is not empty and does not lie in formal intent of any positive example.

Another negative hypothesis is

$$\{\mathbf{R}, \mathbf{F},\ \mathbf{con},\ \mathbf{ice},\ \mathbf{grade}\}.$$

Note that there exist subsets of the minimal positive hypothesis (such as $\{\mathbf{Q}\}$ or $\{\mathbf{B}\}$) that give smaller sufficient conditions for the occurrence of the goal property, but the minimal hypothesis gives a more detailed description of an "expensive chain." The minimal hypothesis takes into account the set of all attributes that occured with the goal one. One can say that this set corresponds to the "notion of expensive chain."

As for the classification of the undetermined examples, example 18 is classified positively, since there is a hypothesis {**B, con, dur**} in favor of positive classification and there is no hypothesis in favor of the negative classification. Example 19 is classified negatively since there is a hypothesis $\{\mathbf{R}, \mathbf{F}\}$ in favor of this and there is no hypothesis in favor of positive classification. Example 20 is not classified since there is neither hypotheses for its positive classification nor for the negative one.

4. LATTICE-THEORETIC INTERPRETATION OF HYPOTHESES AND CLASSIFICATIONS

The definition of a prehypothesis implies that a positive prehypothesis h_+ corresponds to an element of the lattice $\underline{\mathfrak{B}}(K_+)$ for which there does not exist an element of the lattice $\underline{\mathfrak{B}}(K_-)$ with the same formal intent. Thus, the problem of existence of a prehypotheses is equivalent to the problem of isomorphism of a lattice to a sublattice of another lattice. Formally, we have the following

Assertion 4.1. *Let there be a positive context $K_+ = (G_+,\ M, I_+)$ and a negative context $K_- = (G_-,\ M, I_-)$. There is no positive prehypotheses iff there is an isomorphism of the lattice $\underline{\mathfrak{B}}(K_+)$ into the lattice $\underline{\mathfrak{B}}(K_-)$ that takes each formal concept of the context K_+ (element of the lattice $\underline{\mathfrak{B}}(K_+)$) into the formal concept of the context K_- (element of the lattice $\underline{\mathfrak{B}}(K_-)$) with the same formal intent.*

Hypotheses have the following interpretation. Each negative example “cuts off” an order filter consisting of falsified hypotheses from the lattice of positive concepts $\underline{\mathfrak{B}}(K_+)$. The set of all positive hypotheses is a complementation of the set of falsified hypotheses to the set of positive concepts and is closed with respect to the join operation of the lattice of positive concepts. The same statement (where + and − are interchanged) holds for the lattice $\underline{\mathfrak{B}}(K_-)$).

The situation looks different in the concept lattice $\underline{\mathfrak{B}}(K_\pm)$. One can distinguish the following three types of formal concepts: first, concepts of the form $(A, B \cup \{w\})$, where B is a formal intent of the context K_+. Second, there are concepts of the form (A, B), where B is a formal intent of the context K_-. Third, there are formal concepts with intents that are neither formal intents of the context K_+ nor formal intents of the context K_-. These are concepts of the form (A, B), for which $A = E_+ \cup E_-$, $E_+ \subseteq G_+$, $E_- \subseteq G_-$, $B \neq E_+^+$, $B \neq E_-^-$.

Assertion 4.2. *A positive hypothesis corresponds to a formal concept of the context $K_\pm$ that has the form $(A, B \cup \{w\})$ and for which there is no formal concept of the context $K_\pm$ with intent B.*

A negative hypothesis corresponds to a formal concepts of the context $K_\pm$ of the form (A, B), $w \notin B$ such that there is no concept of the context $K_\pm$ with formal intent containing $B \cup \{w\}$ (i.e., lying below the corresponding vertex in the lattice diagram).

In the concept lattice $\underline{\mathfrak{B}}(K_\pm)$ the formal concepts that correspond to positive and negative hypotheses lie below (in the sense of order $\leq$ on formal concepts from $K_\pm$) formal concepts whose intents are not hypotheses.

In terms of this lattice the problem of classification of an undetermined example $g_\tau \in G_\tau$ looks as follows. Consider an order filter of the lattice $\underline{\mathfrak{B}}(K_\pm)$ given by maximal subsets of the set $\{g_\tau\}^\tau \cup \{w\}$ that are formal intents of the context $K_\pm$. If there exists a formal concept $(A, B \cup \{w\})$ lying below in this order filter so that the concept (A, B) does not lie in the same order filter (hence, not in the lattice $\underline{\mathfrak{B}}(K_+)$) and $w \notin B$, then B is a hypothesis in favor of the positive classification of the undetermined example g_τ. The absence of a hypothesis in favor of the negative classification of the example g_τ means that for any formal concept (A, B) of the lattice $\underline{\mathfrak{B}}(K_\pm)$ such that $w \notin B$ the concept (A, B) lies in the order filter of the lattice $\underline{\mathfrak{B}}(K_\pm)$ corresponding to the undetermined example g_τ there exists a formal concept lying below $(((B \cup \{w\})'(B \cup \{w\})'')$.

In the lattice of the classification context K_c the situation looks as follows.

Assertion 4.3. *Given a formal context K_c and an undetermined example g_τ, a hypothesis in favor of a positive classification of the undetermined example g_τ exists iff there exist certain A, B, A_τ for which the following conditions hold: $A, B \subseteq M$, $g_\tau \in A_\tau \subseteq G_\tau$:*

1. $(A, B \cup \{w\}) \in \mathfrak{B}(K_c)$,

2. $(A \cup A_\tau, B) \in \mathfrak{B}(K_c)$.

Moreover, there does not exist a hypothesis in favor of negative classification of example $g_\tau \in G_\tau$ iff the relation $C \cap G_+ \neq \varnothing$ (C lies in the order filter of the lattice $\underline{\mathfrak{B}}(K_c)$ given by at least one positive example) holds for any formal concept $(C, D) \in \underline{\mathfrak{B}}(K_c)$ such that the following conditions hold.

3. $w \notin D$, i.e., the formal concept (C, D) does not lie in the order ideal of the lattice $\underline{\mathfrak{B}}(K_c)$ given by the formal concept (w'', w');

4. $D \subseteq \{g_\tau\}^c$, i.e., the formal concept (C, D) lies in the order ideal of the lattice $\underline{\mathfrak{B}}(K_c)$ given by the formal concept $(\{g_\tau\}^{cc}, \{g_\tau\}^c)$.

At least one of conditions 3–4 is violated for the contradictory classification, whereas Statement 4.3 always holds. For the undetermined classification the condition of Statement 4.3 does not hold.

Lattice-theoretic interpretation of generalized hypotheses has the following form. Each generalized hypothesis $(h_+, \mathcal{X})$ corresponds to an element of the lattice $\underline{\mathfrak{B}}(K_+)$ and the set of elements of the lattice $\underline{\mathfrak{B}}(K_-)$ each of which is maximal (in the sense of order relation in $\underline{\mathfrak{B}}(K_-)$) among negative concepts that are less than (h_+^-, h_+^{--}). The set of elements of the lattice $\underline{\mathfrak{B}}(K_+)$ that correspond to generalized hypotheses with nonempty set of hindrances is the complementation (to the set of all elements of the lattice $\underline{\mathfrak{B}}(K_+)$) of the set of all elements of the lattice $\underline{\mathfrak{B}}(K_+)$) that correspond to positive hypotheses.

5. GRAPH-THEORETIC INTERPRETATION OF HYPOTHESES AND CLASSIFICATIONS

We shall need the following auxiliary definitions. In the bipartite graph $B = (X \cup Y, Z)$, where X and Y are sets of vertices of different parts ($X \cap Y = \varnothing$), and Z is the set of edges, the set of vertices $X_1 \subseteq X$ *dominates* the set of vertices $Y_1 \subseteq Y$ if each vertex from Y_1 is adjacent to a vertex from X_1. The *common shadow* of the set of vertices $X_1 \subseteq X$ is defined as the set $Y_2 \subseteq Y$ of all vertices adjacent to all vertices of the set X_1.

By the learning context $K_\pm = (G_+ \cup G_-, M \cup \{w\}, I_+ \cup I_- \cup (G_+ \times \{w\}))$ one can construct a tripartite graph $T = (W_1 \cup W_2 \cup W_3, E_1)$ such that the vertices of the set W_1 are in one-to-one correspondence to (+)-examples, the vertices of the set W_3 are in one-to-one correspondence to (−)-examples, the vertices of the set W_2 are in one-to-one correspondence to the elements of M, the subset of edges $E_1 \cap W_1 \times W_2$ is given by the relation I_+, and the subset of edges $E_1 \cap W_2 \times W_3$ is given by the complementation of the relation I_-, i.e., by the relation $W_2 \times W_3 \setminus I_-$. A positive hypothesis h of the context $K_\pm$ corresponds to the subset $V_2 \subseteq W_2$ that dominates all the vertices from the set W_3 and induces inclusion maximal bipartite subgraph on the vertices $W_1 \cup W_2$ (i.e., the common shadow V_2 on the set W_1 is the set $V_1 \subseteq W_1$, and the common shadow of the set V_1 on the set W_2 is $V_2 \subseteq W_2$).

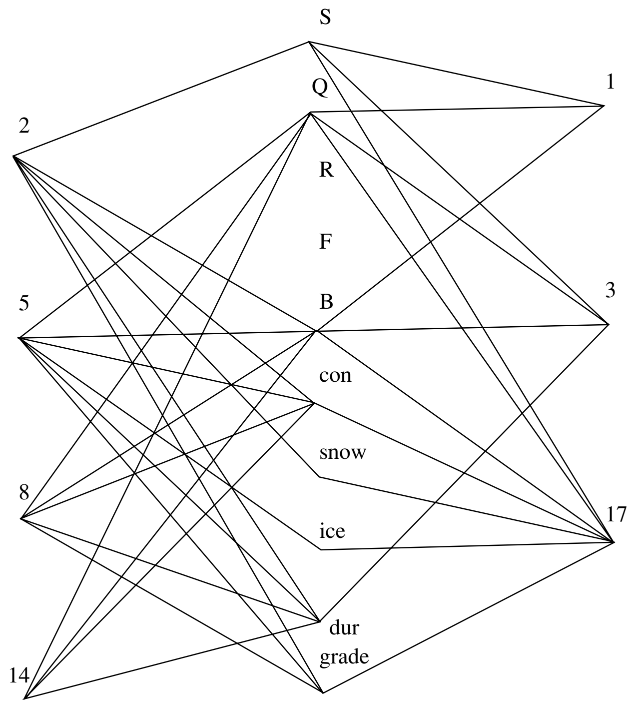

Fig. 1.

Example 5.1. The graph T in Fig. 1 corresponds to the learning context from Example 3.1.

Note that the inverse also holds: for each tripartite graph the problem about the existence of a subgraph of the type indicated above is reduced to the problem of existence of a hypothesis for the learning context constructed by the tripartite graph. Prehypotheses and generalized hypotheses also have natural graph-theoretic interpretation on tripartite graph. We define the sets of vertices $W_1 \cup W_2 \cup W_3$ (where the sets W_1, W_2, and W_3 are called the left, middle, and right parts, respectively) of this graph in the same way as in the case of interpretation of hypotheses: the set W_1 corresponds to the set of positive examples G_+, the set W_2 corresponds to the set of attributes M, and the set W_3 corresponds to the set of negative examples G_-.

The set of edges $E_1 \subseteq W_1 \times W_2$, as in the case of hypotheses, is defined as $E_1 := I_+$. The set of edges $E_2 \subseteq W_2 \times W_3$ is defined similarly, i.e., $E_2 := I_-$, i.e., complementary w.r.t. definition in case of hypotheses. Then a positive prehypothesis corresponds to a complete bipartite subgraph $(V_1 \cup V_2, V_1 \times V_2)$ (here $V_1 \subseteq W_1$, $V_2 \subseteq W_2$) of the graph $(W_1 \cup W_2, E_1)$, for which there does not exist a complete bipartite subgraph of the graph $(W_2 \cup W_3, E_2)$ with the same set of vertices in the middle part W_2. A positive generalized hypothesis corresponds to a complete bipartite subgraph $(V_1 \cup V_2, V_1 \times V_2)$ (here $V_1 \subseteq W_1$, $V_2 \subseteq W_2$) of the graph $(W_1 \cup W_2, E_1)$, for which there exists a set of complete subgraphs of the graph $(W_2 \cup W_3, E_2)$, where the set of vertices in the middle part of each graph from this set contains the set V_2.

It is worth while to consider also a graph-theoretic interpretation of the problem of classification of undetermined examples. As we show below, this can be used for the study of algorithmic complexity of classification of undetermined examples. An obvious strategy of computing classifications is computing the set of all (minimal) hypotheses with their subsequent use for the classification. The problem concerning the possibility of classification without precomputing all (minimal) hypotheses seems to be interesting both from theoretical and practical standpoints.

By definition, the problem about classification of an undetermined example, say about positive classification, consists of two subproblems. First, it is required to test the existence of a hypothesis

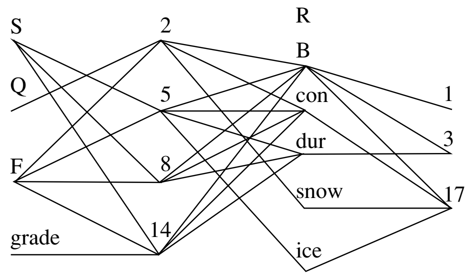

Fig. 2.

in favor of a positive classification and second, it is required to test the absence of a hypothesis in favor of negative classification.

The problem of existence of a hypothesis in favor of the positive classification of an undetermined example g_τ may be formulated in the form of the following decision problem:

INSTANCE. A classification context $K_c = (G_+ \cup G_- \cup G_\tau, M \cup \{w\}, I_+ \cup I_- \cup I_\tau \cup (G_+ \times \{w\}))$, where $G_\tau = \{g_\tau\}$.

QUESTION. Does there exist a hypothesis h in favor of positive classification of the undetermined example g_τ, i.e. such that $h \subseteq \{g_\tau\}^\tau$?

The problem about the absence of a hypothesis in favor of the negative classification is formulated complementary.

It can be shown that the problem about existence of a hypothesis in favor of positive classification is equivalent to the following problem about the domination over parts in a quadripartite graph.

INSTANCE. Given a quadripartite graph $Q = (V_1 \cup V_2 \cup V_3 \cup V_4, E)$, $E \subseteq (V_1 \times V_2) \cup (V_2 \times V_3) \cup (V_3 \times V_4)$. The graphs B_1, B_2, B_3 that are subgraphs of the graph Q induced by the sets of vertices $(V_1 \cup V_2)$, $(V_2 \cup V_3)$, $(V_3 \cup V_4)$, respectively.

QUESTION. Does there exist a complete bipartite subgraph $(W_2 \cup W_3, W_2 \times W_3)$ of the graph B_2 such that it is inclusion maximal, $W_2 \subseteq V_2$, $W_3 \subseteq V_3$, the set of vertices W_2 dominates V_1, and the set of vertices W_3 dominates V_4?

We show that the problem about domination over parts in a quadripartite graph is reduced to a problem about the existence of a hypothesis in favor of positive classification, i.e., in the same way as the problem about hypothesis is constructed from the problem about tripartite graph.

Classification context K_c with a single undetermined example $g_\tau \in G_\tau$ is given by the problem about domination over parts in the following way: $M = (V_1 \cup V_3)$, $G_+ = \{g_i \mid i = 1, \ldots, |V_2|\}$, $G_- = \{f_l \mid l = 1, \ldots, |V_4|\}$. The relation I_+ is given by example intents in the following way: $\{g_i\}^+$ consists of all vertices from the set V_1 that are not adjacent to the vertex $v_i^2 \in V_2$ and all vertices from the set V_3 that are not adjacent to the vertex $v_i^2 \in V_2$. Undetermined context is given by one-element set of examples $G_\tau = \{g_\tau\}$, and the relation I_τ is given by the relation $\{g_\tau\}^\tau = V_3$. The relation I_- is defined as follows: $\{f_l\}^- = V_3 \backslash \{w_1^3, \ldots, w_q^3\}$, where $\{w_1^3, \ldots, w_q^3\}$ is the set of all vertices from the set V_3 adjacent to the vertex $v_l^4 \in V_4$.

Example 5.2. Consider positive, negative, and undetermined contexts from Example 3.1 and the problem of classification of the undetermined example 18 with intent {**R, B, con, snow, ice, dur**}.

The undetermined example is classified positively, since the minimal positive hypothesis is contained in the set $\{g_\tau\}^\tau$ (note that nonminimal hypotheses are not its subsets) and no negative hypothesis is contained in the set $\{g_\tau\}^\tau$. In terms of the corresponding quadripartite graph in Fig. 2 the set of vertices $\{2, 5, 8, 14\} \cup \{\mathbf{B}, \mathbf{con}, \mathbf{dur}\}$ induces a complete bipartite subgraph, the set of vertices $\{2, 5, 8, 14\}$ dominates all vertices of the first part and the set of vertices $\{\mathbf{B}, \mathbf{con}, \mathbf{dur}\}$ dominates all vertices of the fourth part.

6. HYPOTHESES AND RELATED NOTIONS FROM INTELLIGENT DATA ANALYSIS

In this section we show how hypotheses are related to certain well-known notions from data analysis and artificial intelligence, such as implications in FCA [34], functional dependencies [15, 52], and version spaces [55, 45].

6.1. Hypotheses and Implications

Let K_+ and K_-, as before, denote positive and negative formal contexts w.r.t. a goal attribute w and

$$K_\pm := (G_+ \cup G_-, M \cup \{w\}, I_+ \cup I_- \cup (G_+ \times \{w\}))$$

is a learning context. In this context derivation and closure operators are denoted by superscripts $^\pm$ and $^{\pm\pm}$, respectively. In particular, for an arbitrary set of attributes $P \subseteq M$ we write $P^{\pm\pm}$ to denote the least formal intent of the formal context $K_\pm$ that contains P. Derivation and closure operators for contexts K_+ and K_- are denoted by superscripts $^+$, $^-$ and $^{++}$, $^{--}$, respectively.

Assertion 6.1. *Let K_+ and K_- be positive and negative contexts and let h_+ be a positive hypothesis. Then $h_+ \to \{w\}$ is an implication in the learning context*

$$K_\pm := (G_+ \cup G_-, M \cup \{w\}, I_+ \cup I_- \cup (G_+ \times \{w\})).$$

Similarly, a negative hypothesis h_- is an implication in the context

$$K_\pm := (G_+ \cup G_-, M \cup \{\neg w\}, I_+ \cup I_- \cup (G_- \times \{\neg w\})),$$

where an object has attribute $\neg w$ iff it does not have the attribute w.

An implication basis is a subset of the set of all implications from which all implications can be inferred by means of Armstrong axioms applied as inference rules. A minimal (in the number of implications) base was characterized in [30] by minimal nodes of irredundancy. A more elegant recursive definition was given in [33] (see also [34]) by means of pseudointents.

A *pseudointent* [34] of a formal context (G, M, I) is a set $P \subseteq M$ such that

1. $P \neq P''$ and
2. for any pseudointent $Q \subseteq P$, $Q \neq P$ the relation $Q'' \subseteq P$ does not hold.

Pseudointents are related to minimal hypotheses in the following way [37]:

Assertion 6.2. *Let H be a minimal positive hypothesis w.r.t. the goal attribute w (so, $w \notin H$ holds). Then*

1. *if $P \subseteq H$ is a pseudointent for which $w \in P^{\pm\pm}$, then $P^{\pm\pm} = H \cup \{w\}$;*
2. *there exists a pseudointent P of the context $K_\pm$ such that $P \subseteq H$ and $P^{\pm\pm} = H \cup \{w\}$;*
3. *if P is a pseudointent such that $w \notin P$, but $w \in P^{\pm\pm}$, then $P^{\pm\pm} \setminus \{w\}$ is a positive hypothesis.*[6]

6.2. Implications and Dependencies

We show equivalence (two-direction reducibility by Karp [39]) of implications and functional dependencies in relational databases. Recall definitions of *relation* [27] and *functional dependency* [21] (see [15, 52, 29]).

[6] Not necessarily minimal.

Let M be a finite set of attributes. The mapping *dom* takes each attribute $m \in M$ in its domain $dom(m)$. *Relation* R over M is a subset of Cartesian product $\prod_{m \in M} dom(m)$. The relation R over M is given by a set of tuples of size n: $R = \{g_1, \dots, g_n\}$,

$$g_i\colon\ M \to \bigcup_{m \in M} dom(m), \quad g_i(m) \in dom(m), \quad i = 1, \dots, n.$$

A *functional dependency* $X \to Y$ for $X, Y \subseteq M$ holds for relation $R = \{g_1, \dots, g_n\}$ if $g_i(m) = g_j(m)$ for all $m \in X$ implies $g_i(m) = g_j(m)$ for all $m \in Y$, $\forall g_i, g_j \in R$, $i \neq j$.

In terms of FCA it is natural to consider relation as a many-valued context whose tuples correspond to object intents.

A lot of literature is devoted to the relation of functional dependencies with lattices (see [29] and papers cited there) where results on characterization of lattices that correspond to given sets of functional dependencies on the set of attributes M are given.

For an arbitrary relation R given by a many-valued context $K = (G, M, W, I)$ there exists a context $K_N = (P_2, M, I_N)$, where $P_2\colon = \{\{g, h\} \mid g, h \in G, g \neq h\}$ and $\{g, h\} I_N m$, iff $m(g) = m(h)$, so that $X \to Y$ is an implication of the context K iff it is a functional dependency for the relation R (the proof is given in [34], Section 2.4).

Below we present the inverse reduction.

Assertion 6.3. *For each context $K = (G, M, I)$ there exists a relation R such that $X \to Y$ is an implication of the context K iff $X \to Y$ is a functional dependency for the relation R.*

Proof. To each object $g \in G$ we assign a function $g\colon M \mapsto \{0, 1\}$ in a natural way: $g(m) := 1$ if $(g, m) \in I$ and $g(m) := 0$ otherwise. We construct the required relation R by the context K as a many-valued context $K_m = (G \cup \{g_0\}, M, W_m, I_m)$. Attributes of the set M take values from the set $W_m = \{w_1, \dots, w_{|G|}\}$. The relation I_m is defined as follows: $g_0(m) = 1$ for all $m \in M$. For $g \in G$

$$(g, m, 1) \in I_m, \text{ if } (g, m) \in I \text{ and } (g, m, m_g) \in I_m \text{ otherwise.}$$

Let $X \to Y$ be an implication of the context K. By the definition of the implication, $X' \subseteq Y'$. It means that $(g, m) \in I$ for all $m \in X$ implies $(g, m) \in I$ for all $m \in Y$. Then, in the context K, $g(m) = 1$ for all $m \in X$ implies $g(m) = 1$ for all $m \in Y$.

Let $g_i(m) = g_j(m)$ for all $m \in X$. By the construction of the many-valued context K_m, all nonunit attribute values are different for different objects and hence $g_i(m) = g_j(m)$ iff $g_i(m) = g_j(m) = 1$. Therefore, by the above condition we have $g_i(m) = g_j(m)$ for all $m \in Y$, which means that the dependency $X \to Y$ holds for relation R.

In the other direction. Let the dependency $X \to Y$ hold for the relation R and let $g(m) = 1$ for some $g \in G$ and all $m \in X$. By the construction of K_m, $g_0(m) = 1$ for all $m \in M$. Hence, $g_0(m) = g(m)$ for all $m \in X$. By the definition of dependency $X \to Y$ for the relation R, we have $g_0(m) = g(m)$ for all $m \in Y$, but since $g_0(m) = 1$ for all $m \in M$, we have $g(m) = 1$ for all $m \in Y$. Therefore, $(g, m) \in I$ for all $m \in X$ implies $(g, m) \in I$ for all $m \in Y$ in the context K, which means that $X' \subseteq Y'$ and $X \to Y$ is an implication in the context K $\diamond$.

Thus, a hypothesis is also a functional dependency for a suitable relation.

Recently, the search for dependencies attracts great attention in relation with the search for information in the world wide web and very large databases. The related domain was called data mining and knowledge discovery. Some researchers [68, 67, 71] in this domain use intensively concept lattices and constructions related to them: implications, dependencies, partial implications and dependencies. The set of attributes A partially implies the set of attributes B if "many" objects (the "majority") that have the set of attributes A, have also the set of attributes B (see [51, 71] for exact definitions).

6.3. The Version Space between Minimal Hypotheses and Proper Premises

The term "version space" was proposed by T. Mitchell in [54], [55] and became the name of a Machine Learning domain. Version spaces can be defined differently, e.g., Mitchell defined them in terms of sets of maximal and minimal elements, in [66] they are defined in terms of minimal elements and sets of negative examples. Version spaces can also be defined in terms of some matching predicate. These representations are equivalent, however transformations from one into another are not always polynomially tractable. We will use the representation with matching predicates. According to [54, 66] the basic notions of the version spaces are as follows:

Instance language Li by means of which *instances* are described. Concept language Lc by means of which *concepts* are described. Concepts may be considered extensionally (as in [43]) as sets of instances or by means of a *matching predicate* $M(c, i)$ that defines that a concept c *covers* or not covers an instance i: $M(c, i)$ iff i is an instance of concept c. The set of concepts is partially ordered by *more general or equal* relation $\leq$: for $c_1, c_2 \in Lc$, $c_1 \leq c_2$, i.e., the concept c_1 is more general than the concept c_2 or coincides with it, if every instance of concept c_2 is an instance of concept c_1.

Sets I^+ and I^- of *positive* and *negative* instances of a *target concept*. Each instance is described in terms of the language Li, $I^+ \cap I^- = \varnothing$; the target concept is not given explicitly.

Consistency relation $\text{cons}(c, I^+, I^-)$: $\text{cons}(c, I^+, I^-)$ holds iff for every $i \in I^+$ $M(c, i)$ and for every $i \in I^-$ $\neg M(c, i)$. A concept for which consistency relation holds is called *consistent*. The set of all consistent concepts in Lc is called a *version space* or VS.

The learning problem in a version space is defined as follows:

Given $Lc, Li, M(c, i), I^+, I^-$.
Find Version space $VS(Lc, Li, M(c, i), I^+, I^-)$.

Version spaces are often considered in terms of *boundary sets* proposed in [55]. They can be defined if the language Lc is *admissible*, i.e., if every chain in it has a minimal and a maximal element. In this case,

$$GVS = MAX(VS) = \{c \in VS | \neg\exists c_1 \in VS \ c \leq c_1\},$$
$$SVS = MIN(VS) = \{c \in VS | \neg\exists c_1 \in VS \ c_1 \leq c\}.$$

The following result from [44] about *merging version spaces* provides a convenient means for updating and incremental construction of version spaces:

For version spaces $\text{VS1}(I_1^+, I_1^-)$ and $\text{VS2}(I_2^+, I_2^-)$ based on the sets of examples $(I_1^+, I_1^-))$ and (I_2^+, I_2^-) the version space $\text{VS}(I_1^+ \cup I_2^+, I_1^- \cup I_2^-)$ based on the union of examples of both version spaces consists of common elements of VS1 and VS2, or

$$VS(I_1^+ \cup I_2^+, I_1^- \cup I_2^-) = VS(I_1^+, I_1^-) \cap VS(I_2^+, I_2^-).$$

Below we relate version spaces with implications and hypotheses.

Recall that a set Q is a *proper premise* of a context $K = (G, M, I)$ if $Q \subset M$, $Q'' \neq Q$, and $(Q \setminus \{n\})'' \neq Q''$ for every $n \in Q$ [34]. We introduce the notion of a relative proper premise similar to that of a relative pseudointent. For a learning context $K_\pm$ w.r.t. a goal attribute w a set $Q \subset M$ is called a *proper premise relative to the goal attribute* if it is a proper premise of $K_\pm$, $w \notin Q$, and

$w \in Q''$. For a learning context $K_\pm$ a set $Q \subset M$ is called a *positive* or (+)-*classifier* if $Q \subset g'_+$ for a (+)-example g_+ and $Q \not\subset g'_-$ for any (-)-example g_-.

Consider a learning context $K_\pm$ and the following specification of the version space. Let the instance language Li be $\mathcal{P}(M)$, the power set of M, the concept language Lc coincide with Li, and the matching predicate $M(c, b)$ for $c \in Lc$, $b \in Li$ be defined as follows: $M(c, b)$ iff $c \subseteq b$. The sets of positive and negative instances of a target concept are the sets of positive and negative examples of the context $K_\pm$.

To avoid collision of notations (e.g., I denotes a set of instances in version space models and the relation of a formal context in the FCA), instead of the sets I_+ and I_- of positive and negative instances, we shall rather speak of sets G_+ and set G_- of positive and negative examples from the contexts K_+ and K_-, respectively. Moreover, since the word "concept" is used differently in FCA and version spaces, we shall use the word "classifier" (as, e.g., in [44]) instead of "concept" when speaking about version spaces. The lower and upper boundary of a version space will be denoted by SVS and GVS instead of S and G, respectively.

Thus, the version space that corresponds to the learning context $K_\pm$ can be represented by the tuple

$$\langle \mathcal{P}(M), \mathcal{P}(M), \subseteq, K_+, K_- \rangle.$$

Assertion 6.4. *The version space* $VS = \langle \mathcal{P}(M), \mathcal{P}(M), \subseteq, K_+, K_- \rangle$ *is not empty iff there is a single minimal (+)-hypothesis of the learning context* $K_\pm$.

Assertion 6.5. *If the version space* $VS = \langle \mathcal{P}(M), \mathcal{P}(M), \subseteq, K_+, K_- \rangle$ *is not empty, then the corresponding boundary sets GVS and SVS are given as follows: SVS is the set consisting of a single minimal (+)-hypothesis and GVS is the set of all proper premises w.r.t. the goal attribute.*

Assertion 6.6. *The boundary sets SVS and GVS of the nonempty version space* $VS = \left\langle \mathcal{P}(M), \mathcal{P}(M), \subseteq, \frac{K_+^1}{K_+^2}, \frac{K_-^1}{K_-^2} \right\rangle$, *resulting from merging two version spaces* $VS_1 = \langle \mathcal{P}(M), \mathcal{P}(M), \subseteq, K_+^1, K_-^1 \rangle$ *and* $VS_2 = \langle \mathcal{P}(M), \mathcal{P}(M), \subseteq, K_+^2, K_-^2 \rangle$, *are given as follows:*

$$SVS = \{s_1 \cap s_2\},$$
$$GVS = MIN \left(\bigcup_{i,j} \{q_i^1 \cup q_j^2\} \right),$$

where $SVS_1 = \{s_1\}$, $SVS_2 = \{s_2\}$, $q_i^1 \in GVS_1$, $q_j^2 \in GVS_2$, *and the operator MIN(X) takes all inclusion minimal sets from the family of sets* $\subseteq$.

The algorithmic aspects of computing the result of merging two version spaces are discussed in general in [44]. In our particular case the complexity bounds are somewhat better. The obvious worst-case bounds that follow from Statement 6.6 are as follows: the computation of the resulting SVS takes $O(|M|)$ time and the computation of the resulting GVS takes $O(|M| \times (|GVS_1| \times |GVS_2|)^2)$ time.

Consider Example 3.1 from the point of view of version spaces.

The minimal positive hypothesis

$$\{\mathbf{B}, \mathbf{con}, \mathbf{dur}\}$$

w.r.t the goal attribute "high chain cost" is unique and, thus, covers all positive examples. Hence, the minimal hypothesis constitutes the one-element set SVS. The most general positive classifier for this problem (in this case it is also unique) has the form $\{\mathbf{B}\}$.

This classifier is a relative proper premise w.r.t. a proper premise and at the same time a relative pseudointent.

All classifiers w.r.t. the goal attribute range between the minimal hypothesis and proper premises, so they are more general or equal to the former and less general or equal to one of the latter.

The correspondence between version spaces and minimal hypotheses is valid only if there exists a unique minimal hypothesis. If minimal hypothesis is not unique, then the version space in the sense of definition above does not exist. For the description of this situation the construction called disjunctive version space is used [66].

6.4. Hypotheses, Concept Lattices, and Decision Trees

Methods for constructing decision trees or classification trees were used by various researches starting from 1960s and become widely known due to ID3 system [64] (see also review [12]). In terms of formal contexts, before the construction of a decision tree the system gets a positive and a negative context. The root of the tree corresponds to the beginning of the process and is not labeled. Other vertices of the classification tree are labeled by attributes from M and edges are labeled by values of the attributes (e.g., 0 or 1 in case of binary contexts), each leaf is additionally labeled by a class + or −. The construction of the tree proceeds as follows. Starting from the root, for each vertex and each attribute not studied above, the value of the information (antientropy) functional is computed. The attribute that "most strongly" (w.r.t. the functional value) "separates" objects from classes + and − is chosen. The process proceeds until the moment when all objects having a given attribute and all attributes above it in the tree relate only to a single class (e.g., +) and all attributes that do not have a given attribute (but have all attributes above in the tree) belong to the opposite class (i.e., − in case of example above). Thus, the set of all attributes that occur along the way from the root to a leaf corresponds to a sufficient condition that an object belongs to a class + or −.

Consider the process of constructing a decision tree in the lattice of the learning context $K_\pm = (G_+ \cup G_-,\ M,\ I_+ \times \{w\} \cup I_-)$ more formally. The decision tree is given by a set of tuples $\{(m_i,\ m_j,\ p_j,\ n_j,\ \varepsilon)\}$, where (m_i, m_j) corresponds to an arc in the tree between the vertex with attribute m_i and the vertex with the attribute m_j; $\varepsilon = 1$ if m_j was chosen for the continuation of the tree and $\varepsilon = 0$ otherwise; p_j corresponds to the number of positive examples with attributes m_j, m_i and all attributes lying in the tree above m_i; n_j corresponds to the number of negative examples with attributes m_j, m_i and all attributes lying in the tree above m_i.

For an arbitrary set of attributes $B \subseteq M$ by $P(B)$ $(N(B))$ we denote the number of positive (negative) examples that have the set of attributes B, i.e.,

$$P(B) = \{g \in G_+ \mid B \subseteq g^+\},$$
$$N(B) = \{g \in G_- \mid B \subseteq g^+\}.$$

The set of attributes B_1 *separates the classes not worse than* the set of attributes B_2 if either $P(B_1) \geq P(B_2)$ and $N(B_1) \leq N(B_2)$ or $P(B_1) \leq P(B_2)$ and $N(B_1) \geq N(B_2)$.

Assertion 6.7. *If $P(B) \geq N(B)$ holds for an arbitrary set $B \subseteq M$ then B^{++} separates the classes not worse than B.*

Assertion 6.8. *The path* $(m_{i1}, \ldots, m_{ik})$ *from the root of the decision tree to a positive leaf corresponds to the sequence* $(m_{i1}^{\pm\pm}, m_{i1}^{\pm}) \geq \ldots \geq (\{m_{i1}, \ldots, m_{ik}\}^{\pm\pm}, \{m_{i1}, \ldots, m_{ik}\}^{\pm})$ *of lattice elements, where* $\{m_{i1}, \ldots, m_{ik}\}^{\pm\pm} \setminus \{w\}$ *is a positive hypothesis.*

Note that the members of the sequence of lattice elements from the Statement are not necessarily neighbors in the lattice diagram. For data from Example 3.1 the best separating attribute is **mount**, which strictly separates positive examples (that all have value B) and negative examples (that all have value F). Thus, in this case the path from the root of the tree to the leaf has length 1.

7. ALGORITHMIC PROBLEMS OF GENERATING CONCEPTS, HYPOTHESES AND CLASSIFICATIONS

In this section we review the main results about algorithmic complexity of generating sets of formal concepts, hypotheses, and classifications. First note that the number of formal concepts of a given formal context may depend exponentially on the size of context as, e.g., in case of the context $K = (G, G, \neq)$, which gives rise to a Boolean concept lattice. Moreover, as shown in [9, 49], the problem of determining the number of all formal concepts of a given formal context is #P-complete. Recall that the class #P of enumeration problems is defined by the counting nondeterministic Turing machine and #P-completeness is defined w.r.t. Turing reducibility [69, 70], see also [7]. By the basic theorem of FCA, #P-completeness of counting the number of all formal concepts implies that the problem of determining the size of a finite lattice given by an ordered set of its meet- and join-irreducibles is also #P-complete. An obvious corollary of this result is also #P-completeness of the problem of counting all hypotheses of a learning context. Moreover, one can show that the problem of counting all minimal hypotheses is also #P-complete. The proof of this fact that uses graph-theoretic interpretation of hypotheses is given in [11].

Algorithmic complexity of some decision problems about existence of a concepts with certain size constraints are presented in Table 7.1, where P denotes the existence of a polynomial algorithm, NP denotes NP-completeness. Thus, the left upper entry of the table denotes that the problem "does there exist a formal concept (e, i) with $|e| < k$" has a polynomial decision algorithm.

Complexity results concerning decision problems related to existence of a hypothesis h with similar size constraints are given in Table 7.2. As in Table 7.1 the entry P denotes here the existence of a polynomial algorithm and NP denotes NP-completeness.

The above $\#P$-complete and NP-complete problems about positive hypotheses become polynomially solvable, e.g., in the case where formal intents of positive examples are bounded from above by a constant k (i.e., $|\{g\}^+| \leq k$ for all $g \in G_+$). In this case the number of all positive concepts is $O(|M|^k)$. Therefore, an algorithm is possible that can generate all positive concepts and test whether their intents are hypotheses in time polynomial in $|G_+|$, $|G_-|$ and $|M|$. Similar situation takes place in the case where the sizes of formal extents of attributes are bounded from above by a constant.

As for the algorithmic complexity of classification, it should be noted first that the problem about the existence of a hypothesis in favor of classification of an undetermined example is NP-complete.

Table 7.1

	$\leq$	$=$	$\geq$
$\|i\|$	P	NP	P
$\|e\|$	P	NP	P
$\|e\| + \|i\|$	NP	NP	P

Table 7.2

	$\leq$	$=$	$\geq$
$\|h\|$	NP	NP	P
$\|h^+\|$	P	NP	NP
$\|h^+\| + \|h\|$	NP	NP	NP

The proof of this result is based on the reduction of the problem of domination over parts in a quadripartite graph to the problem of classification. By the symmetry of $(-)$- and $(+)$-hypotheses, this implies that the problem of determining that "there does not exist a $(-)$-hypothesis in favor of negative classification of g_τ" is coNP-complete (i.e., complete in the class dual to NP, see [7]) and hence, the problem "whether the undetermined example $g_\tau \in G_\tau$" is classified positively is Δ_p-complete (the class Δ_p was defined in [59]).

Note that in one of the following cases, where

- $V_1 = \varnothing$ $(M = \{g_\tau\}^\tau)$ or
- $V_2 = V_3$ $(G_+ = \{g_\tau\}^\tau)$ or
- $V_1 = \varnothing$ ($G_- = \varnothing$: the quadripartite graph becomes a tripartite),

there exists a polynomial algorithm for the classification of an undetermined example in a given classification context$K_c = (G_+ \cup G_- \cup G_\tau, M \cup \{w\}, I_+ \cup I_- \cup I_\tau \cup G_+ \times \{w\})$.

A polynomial algorithm for the solution of the problem is also possible when the size of the set $\{g_\tau\}^\tau$ is bounded from above by a constant. A constraint of this kind is justified in various practical applications, e.g., in the "Structure–Activity Relation" problem, where the goal attribute w corresponds to a certain biological activity, and classification denotes prediction whether a chemical compound has this activity [6]. It can be considered that the size of a compound is bounded by a constant when a series of classifications of a single compound is made and the sets of examples and attributes (i.e., elements of the set M) are growing. General algorithms for hypotheses generation are given in [8, 13, 48]. In [5] we gave an algorithm for the generation of generalized hypotheses.

A trivial algorithm for solving a classification problem in polynomial time by sequentially testing all subsets of the set $\{g_\tau\}^\tau$, by computing their closures (i.e., applying the operator $^{++}$ to them) and testing the containment of these closures in intents of negative examples. When a positive formal intent that is not a subset of any negative example is found, the algorithms proceeds similarly on negative examples. The time complexity of this algorithm is $O(2^{|\{g_\tau\}^\tau|} \times (|G_+| + |G_-|) \times |M|)$. A more efficient algorithm based on the Close-by-One algorithm is described in [13, 48].

Now we consider algorithms for generation of formal concepts, hypotheses, and implication bases.

Over dozen of algorithms for generation of the set of all formal concepts is known. One of the first of them is found in [26, 57]. The first two reviews and experimental comparisons are found in [42, 40], the last and most complete review is found in [50].

Due to the fact that the number of formal concepts can be exponential in the size of a formal context, the worst-case time complexity of algorithms for generation of formal concepts is exponential in the input size (the size of context), it is reasonable to estimate the time complexity of algorithms as a function of the output size (i.e., of the number of formal concepts) or in terms of cumulative polynomial delay. Recall that an algorithm listing a family of objects has a *delay* d if it satisfies the following conditions whenever it is run with an input of size p [46]:

1. It executes at most $d(p)$ computation steps before either outputting the first structure or halting;

2. After any output it executes at most $d(p)$ computation steps before either outputting the next structure or halting. An algorithm whose delay is bounded from above by a polynomial in the length of the input is called a *polynomial delay algorithm* [46].

A weaker notion of efficiency of listing algorithms was proposed in [41]. An algorithm is said to have a *cumulative delay* d if it is the case that at any point of time in any execution of the algorithm with any input p the total number of computation steps that have been executed is at most $d(p)$ plus the product of $d(p)$ and the number of structures that have been output so far. If

$d(p)$ can be bounded by a polynomial of p, the algorithm is said to have a *polynomial cumulative delay.*

Batch algorithms presented in [24, 33, 35, 13, 31] are algorithms with polynomial delay that does not exceed $(|G|^3 \times |M|)$. Their worst-case time complexity is $O(|\underline{\mathfrak{B}}(K)| \times |G|^2 \times |M|)$.

Algorithms described in [57, 58] are incremental. They cannot have polynomial delay since the arrival of a next object brings about the update of the set of formal concepts generated before. However these are algorithms with polynomial cumulative delay. The algorithm from [58] has the best worst-case estimate as a function of the number of formal concepts and input size: $O(|\underline{\mathfrak{B}}(K)| \times |G| \times (|M| + |G|))$ (recall that $|\underline{\mathfrak{B}}(K)|$ denotes the size of the lattice of formal concepts $\underline{\mathfrak{B}}(K)$, i.e., the number of all formal concepts of the formal context K).

Algorithms from [26, 8, 40] are not algorithms with polynomial delay and their worst-case complexity is a function quadratic in the number of formal concepts (within a factor polynomial in $|G|$ and $|M|$).

The algorithms generating formal concepts are easily adapted for the generation of hypotheses: it suffices to add the test for containment of each new formal intent in intents of negative examples. The arising change in the worst-case complexity is inessential. For example, the complexity of the algorithm for generating hypotheses from [48] is $O(|\underline{\mathfrak{B}}(K_+)| \times (|G_+| + |G_-|) \times |G_+| \times |M|)$.

The adaptation of algorithms for computing classifications is also easy: one needs an additional test for containment of hypotheses in the intent of the undetermined example to be classified.

As for the generation of the minimal implication basis, its algorithm was described in [32, 34]. Its complexity is exponential in the input size even in the case of empty output, i.e., when there is no nontrivial implications, as for example, in the case of the formal context $(G, G, \neq)$, which gives rise to the Boolean concept lattice. A more efficient algorithm, as well as a good estimate of the base size and the algorithmic complexity of computing this number are not known.

8. CONCLUSION

We considered mathematical models of machine learning and data analysis in terms of Formal Concept Analysis. The main attention was drawn to the JSM-method of automated hypotheses generation, its lattice-theoretic and graph-theoretic interpretations. We showed the relation of the JSM-method to the well-known domains in data analysis and artificial intelligence: implications in FCA, search for functional dependencies in data bases, version spaces, decision trees, data mining and knowledge discovery. We also studied the problems of algorithmic complexity of generating concept lattices, implication bases, hypotheses, and classifications.

REFERENCES

1. Birkhoff, G.D., *Lattice Theory*, Providence: AMS, 1979. Translated under the title *Teoriya reshetok*, Moscow: Nauka, 1984.

2. Vinogradov, D.V., Formalization of Plausible Reasoning in Predicate Logic, *Nauch. Tekh. Inf., Ser. 2*, 2000, no. 3, pp. 17–20.

3. Grätzer, G., *General Lattice Theory*, Basel: Brikhäuser, 1978. Translated under the title *Obshchaya teoriya reshetok*, Moscow: Mir, 1982.

4. Gusakova, S.M. and Finn, V.K., On New Means of Formalizing the Notion of Similarity, *Nauch. Tekh. Inf., Ser. 2*, 1987, no. 10, pp. 14–22.

5. Gusakova, S.M. and Kuznetsov, S.O., Similarity in the Generalized JSM-Method and Algorithms for Its Generation, *Nauch. Tekh. Inf., Ser. 2*, 1995, no. 6.

6. Gusakova, S.M. and Pankratova, E.S., Principles of Construction of an Intelligent System of JSM Type for the Forecast of Carcinogenicity of Chemical Compounds, *Nauch. Tekh. Inf., Ser. 2*, 1996, no. 11, pp. 16–20.

7. Garey, M. and Johnson, D., Computers and Intractability (A Guide to the Theory of NP-Completeness), San Francisco: Freeman, 1979. Translated under the title *Vychislitel'nye mashiny i trudnoreshaemye zadachi*, Moscow: Mir, 1982.

8. Zabezhailo, M.I., Ivashko, V.G., Kuznetsov, S.O., Mikheenkova, M.A., Khazanovskii, K.P., and Anshakov, O.M., Algorithmic and Software Means of the JSM-Method of Automated Hypotheses Generation, *Nauch. Tekh. Inf., Ser. 2*, 1987, no. 10, pp. 1–14.

9. Kuznetsov, S.O., Interpretation on Graphs and Complexity Characteristics of the Search for Dependences of a Certain Type, *Nauch. Tekh. Inf., Ser. 2*, 1989, no. 1, pp. 23–28.

10. Kuznetsov, S.O., JSM-Method as a System of Machine Learning, *Itogi Nauki Tekh., Ser. Inf.*, 1991, vol. 15, pp. 17–54.

11. Kuznetsov, S.O., Complexity of Learning and Classification Algorithms Based on the Search for Set Intersections, *Nauch. Tekh. Inf., Ser. 2*, 1991, no. 9, pp. 8–15.

12. Kuznetsov, S.O., Models and Methods of Machine Learning, *Itogi Nauki Tekh., Ser. Vychisl. Nauki*, 1991, vol. 7, pp. 89–137.

13. Kuznetsov, S.O., A Fast Algorithm for Construction of All Intersections of Objects from a Finite Semilattice, *Nauch. Tekh. Inf., Ser. 2*, 1993, no. 1, pp. 17–20.

14. Kuznetsov, S.O. and Finn, V.K., On Models of Learning Based on Operation of Similarity, *Obozrenie Prikl. Promyshl. Mat.*, 1996, vol. 3, no. 1, pp. 66–90.

15. Maier, D., The Theory of Relational Databases, Rockville: Computer Science Press, 1983. Translated under the title *Teoriya relyatsionnykh baz dannykh*, Moscow: Mir, 1987.

16. Finn, V.K., On Computer-Oriented Formalization of Plausible Reasoning in F.Bacon–J.S.Mill Style, *Semiotika Inf.*, 1983, vol. 20, pp. 35–101.

17. Finn, V.K., Plausible Inference and Plasuible Reasoning, *Itogi Nauki Tekh., Ser. Teor. Veroyatn. Mat. Statist. Teor. Kibern.*, 1988, vol. 28, pp. 3–84.

18. Finn, V.K., On Generalized JSM-Method of Automated Hypothesis Generation, *Semiotika Inf.*, 1989, vol. 29, pp. 93–123.

19. Finn, V.K., Plausible Reasoning in Intelligent Systems of JSM-type, *Itogi Nauki Tekh., Ser. Inf.*, 1991, vol. 15, pp. 54–101.

20. Anshakov, O.M., Finn, V.K., and Skvortsov, D.P., On Axiomatization of Many-Valued Logics Associated with the Formalization of Plausible Reasonings, *Stud. Log.*, 1989, vol. 25, no. 4, pp. 23–47.

21. Armstrong, W.W., Dependency Structure of Data Base Relationships, *IFIP Congress*, Geneva, 1974, pp. 580–583.

22. Barbut, M. and Monjardet, B., *Ordre et classification, II*, Paris: Hachette, 1970.

23. Birkhoff, G.D., *Lattice Theory*, Providence: AMS, 1979.

24. Bordat, J.P., Calcul pratique du treillis de Galois d'une correspondance, *Math. Sci. Hum.*, 1986, no. 96, pp. 31–47.

25. Bylander, T., Allemang, D., Tanner, M.C., and Josephson, J.R., The Computational Complexity of Abduction, *Artif. Intell.*, 1991, vol. 49, no. 1, pp. 25–60.

26. Chein, M., Algorithme de recherche des sous-matrices premières d'une matrice, *Bull. Math. R.S. Roumanie*, 1969, vol. 13, no. 1, pp. 21–25.

27. Codd, E.F., A Relational Model for Large Shared Data Banks, *Comm. ACM.*, 1970, vol. 13, pp. 377–387.

28. Davey, B.A. and Priestley, H.A., *Introduction to Lattices and Order*, Cambridge: Cambridge: Univ. Press, 1990.

29. Demetrovics, J., Libkin, L., and Muchnik, I., Functional Dependencies in Relational Databases: A Lattice Point of View, *Discrete Appl. Math.*, 1992, vol. 40, pp. 155–185.

30. Duquenne, V. and Guigues, J.-L., Familles minimales d'implications informatives resultant d'un tableau de données binaires, *Math. Sci. Humaines*, 1986, vol. 95, pp. 5–18.

31. Freese, R., Ježek, J., and Nation, J.B., *Free Lattices*, Providence: AMS, 1995.

32. Ganter, B., Two Basic Algorithms in Concept Analysis, *FB4-Preprint no. 831*, TH Darmstadt, 1984.

33. Ganter, B., Algorithmen zur Formalen Begriffsanalyse, in: *Beiträge zur Begriffsanalyse*, Ganter, B., Wille, R., and Wolf, K.E., Eds., Hrsg., Mannheim: B.-I. Wissenschaftsverlag, 1987.

34. Ganter, B. and Wille, R., *Formal Concept Analysis: Mathematical Foundations*, Berlin: Springer, 1999.

35. Ganter, B. and Reuter, K., Finding All Closed Sets: A General Approach, *Order*, 1991, vol. 8, pp. 283–290.

36. Ganter, B. and Kuznetsov, S.O., Stepwise Construction of the Dedekind-MacNeille Completion, *6th Int. Conf. on Conceptual Structures, ICCS'98*, 1998, vol. 1453, pp. 295–302.

37. Ganter, B. and Kuznetsov, S.O., Formalizing Hypotheses with Concepts, *8th Int. Conf. on Conceptual Structures, ICCS'98*, 2000, vol. 1867, pp. 342–356.

38. Ganter, B. and Kuznetsov, S.O., Pattern Structures and Their Projections, *9th Int. Conf. on Conceptual Structures, ICCS'98*.

39. Garey, M. and Johnson, D., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: Freeman, 1979.

40. Godin, R., Missaoui, R., and Allaoui, H., Incremental Concept Formation Algorithms Based on Galois Lattices, *Comput. Intell.*, 1995.

41. Goldberg, L.A., *Efficient Algorithms for Listing Combinatorial Structures*, Cambridge: Cambridge Univ. Press, 1993.

42. Guénoche, A., Construction du treillis de Galois d'une relation binaire, *Math. Inf. Sci. Hum.*, 1990, no. 109, pp. 41–53.

43. Gunter, C.A., Ngair, T.-H., and Subramanian, D., The Common Order-Theoretic Structure of Version Spaces and ATMSs, *Artif. Intell.*, 1997, vol. 95, pp. 357–407.

44. Hirsh, H., Generalizing Version Spaces, *Machine Learning*, 1994, vol. 17, pp. 5–46.

45. Hirsh, H., Mishra, N., and Pitt, L., Version Spaces without Boundary Sets, *14th National Conference on Artificial Intelligence (AAAI97)*, 1997.

46. Johnson, D.S., Yannakakis, M., and Papadimitriou, C.H., On Generating All Maximal Independent Sets, *Inf. Process. Lett.*, 1988, vol. 27, pp. 119–123.

47. Kuznetsov, S.O., Mathematical Aspects of Concept Analysis, *J. Math. Sci., Ser. Contemp. Math. Appl.*, 1996, vol. 18, pp. 1654–1698.

48. Kuznetsov, S.O., Learning of Simple Conceptual Graphs from Positive and Negative Examples, in: Zytkow, J. and Rauch, J., Eds., *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD'99*, Lecture Notes in Artificial Intelligence, 1999, vol. 1704, pp. 384–392.

49. Kuznetsov, S.O., Some Counting and Decision Problems in Formal Concept Analysis, *Preprint of the Technische Universität Dresden*, 1999, MATH-Al-14-1999.

50. Kuznetsov, S.O. and Obiedkov, S.A., Algorithms for the Construction of the Set of All Concepts and Their Line Diagram, *Preprint of the Technische Universität Dresden*, 2000, MATH-Al-05-2000.

51. Luxenburger, M., Implications partielles dans un contexte, *Math. Inf. Sci. Hum.*, 1991, vol. 29, no. 113, pp. 35–55.

52. Mannila, H. and Räihä, K.J., *The Design of Relational Databases*, Reading: Addison-Wesley, 1992.

53. Michalski, R.S. and Stepp, R.E., Learning from Observation: Conceptual Clustering, in *Machine Learning: An Artificial Intelligence Approach*, Michalski, R.S., Carbonell, J.G., and Mitchell T.M., Eds., Palo Alto: Tioga, 1983, pp. 41–81.

54. Mitchell, T., Version Space: An Approach to Concept Learning, *PhD Thesis*, Stanford Univ., 1978.

55. Mitchell, T., Generalization as Search, *Artif. Intell.*, 1982, vol. 18, no. 2.

56. Mitchell, T., *Machine Learning*, New York: McGraw-Hill, 1997.

57. Norris, E.M., An Algorithm for Computing the Maximal Rectangles in a Binary Relation, *Rev. Roum. Math. Pures Appl.*, 1978, vol. 23, no. 2, pp. 243–250.

58. Nourine, L. and Raynaud, O., A Fast Algorithm for Building Lattices, *Inf. Process. Lett.*, 1999, vol. 71, pp. 199–204.

59. Papadimitriou, C.H. and Yannakakis, M., The Complexity of Facets (and Some Facets of Complexity), *J. Comp. Sys. Sci.*, 1984, vol. 28, pp. 244–259.

60. Plotkin, G.D., A Note on Inductive Generalization, *Machine Intell.*, 1970, no. 3, pp. 153–163.

61. Plotkin, G.D., A Further Note on Inductive Generalization, *Machine Intell.*, 1971, no. 6, pp. 101–124.

62. Pudlak, P. and Springsteel, F., Complexity in Mechanized Hypothesis Formation, *Theor. Comp. Sci.*, 1979, vol. 8, no. 2, pp. 203–225.

63. Quilllian, M.R., Semantic Memory, in: *Semantic Information Processing*, Minsky, M., Ed., Cambridge: MIT Press, 1968, pp. 227–270.

64. Quinlan, J.R., Induction on Decision Trees, *Mach. Learn.*, 1986, vol. 1, no. 1, pp. 81–106.

65. Skorsky, M., Endliche Verbände—Diagramme und Eigenschaften, *Dissertation*, TH Darmstadt, 1992.

66. Smirnov, E.N. and Braspenning, P.J., Version Space Learning with Instance-Based Boundary Sets, in: Prade, H., Ed., *Proceedings 13th European Conference on Artificial Intelligence*, Chichester: Wiley, 1998, pp. 460–464.

67. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., and Lakhal, L., Fast Computation of Concept Lattices Using Data Mining Techniques, *7th International Workshop on Knowledge Representation Meets Databases*, Berlin, 2000, pp. 129–139.

68. Stumme, G., Wille, R., and Wille, U., Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods, *2nd European Symposium on Principles of Data Mining and Knowledge Discovery*, Nantes, 1998.

69. Valiant, L.G., The Complexity of Computing the Permanent, *Theor. Comp. Sci.*, 1979, vol. 8, no. 2, pp. 189–201.

70. Valiant, L.G., The Complexity of Enumeration and Reliability Problems, *SIAM J. Comput.*, 1979, vol. 8, no. 3, pp. 410–421.

71. Waiyamai, K. and Lakhal, L., Knowledge Discovery from Very Large Databases Using Frequent Concept Lattices, *11th European Conference on Machine Learning* (ECML 2000), 2000, pp. 437–445.

72. Wild, M., Implicational Bases for Finite Closure Systems, in: *Arbeitstagung, Begriffsanalyse und Künstliche Intelligenz*, Lex, W., Ed., 1991, pp. 147–169.

73. Wille, R., Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts, in: *Ordered Sets*, Rival, I., Ed., Dordrecht: Reidel, 1982, pp. 445–470.

74. Wille, R., Concept Lattices and Conceptual Knowledge Systems, *Comput. Math. Appl.*, 1992, vol. 23, no. 6–9, pp. 493–515.

This paper was recommended for publication by O.P. Kuznetsov, a member of the Editorial Board