Symbolic Objects in Formal Concept Analysis

Susanne Prediger

Technische Hochschule Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D–64289 Darmstadt, prediger@mathematik.th-darmstadt.de

Abstract. Symbolic objects are the basic elements for knowledge representation in symbolic data analysis. This paper aims to integrate symbolic objects into formal concept analysis in order to compare and tie together both approaches.

1 Introduction

Symbolic objects are the basic elements of a formal language which has been developed since 1987 in symbolic data analysis. The general aim was to extend the field of application, methods and algorithms of classic data analysis to more complex data. Meanwhile, the formalism of symbolic objects is not only used in a broad field of data analysis, but also in knowledge representation and knowledge processing.

From the point of view of formal concept analysis, the most interesting parts of symbolic data analysis are those which are concerned with knowledge processing and conceptual classifications. These parts of symbolic data analysis and formal concept analysis both emphasize the intensional view. Hence, there are various points of common interest which have been manifested by common conferences on ordinal and symbolic data analysis in the last few years. Nevertheless, the communication is still limited, mostly because of the different formalisms.

This paper tries to integrate the language of symbolic objects in the wider range of logically scaled many-valued contexts in formal concept analysis in order to harmonize both approaches. For that, the basic notions of symbolic data analysis are described in the language of formal concept analysis and some differences are discussed.

A certain familiarity with at least one of both languages could surely be helpful to appreciate this integration. That is because it is obviously not possible to explain and motivate all notions of both approaches in detail. For a basic introduction, please refer to [11] and [6] for formal concept analysis and to [2] for symbolic data analysis. Nevertheless, the basic ideas of this article can be understood without every formal detail.

2 Formalization of Data in many-valued Contexts

Knowledge processing in formal concept analysis and symbolic data analysis usually starts with object-attribute-value relationships, a frequently used data structure to code real-world problems. They can be represented in *many-valued contexts* which are formally defined as quadruples (Ω, Y, O, I) where Ω, Y, and O are sets whose elements are called *(individual) objects, attributes*, and *attribute values*, respectively, and I is a ternary relation with $I \subseteq \Omega \times Y \times O$ such that $(o, y, v) \in I$ and $(o, y, w) \in I$ implies $v = w$. Thus, an attribute y of a many-valued context (Ω, Y, O, I) may be considered as a partial map of Ω to O which suggests writing $y(o) = v$ whenever $(o, y, v) \in I$. In the language of symbolic objects, the attributes are often called *variables*. The *observation set* O_y of the variable y is a set satisfying $y(\Omega) \subseteq O_y \subseteq O$.

It is often emphasized that this formalization and the corresponding methods allow the full freedom of choice for the observation sets. They can be intervals of $\mathbb{R}$ in the case of quantitative variables as well as finite sets or even power sets for so-called qualitative variables (then, it would not be called obeservation set anymore).

$\mathbb{K}$	Fabricator F	Temperature T	Weight W	Price P	Material M
One Kilo Bag	Wolfskin	7° C	940 g	149,-	Liteloft
Sund	Kodiak	3° C	1880 g	139,-	Hollow fiber
Kompakt Basic	Ajungilak	0° C	1280 g	249,-	MTI Loft
Finmark Tour	Finmark	0° C	1750 g	179,-	Hollow fiber
Interlight Lyx	Caravan	0° C	1900 g	239,-	Thermolite
Kompakt	Ajungilak	-3° C	1490 g	299,-	MTI Loft
Touch the Cloud	Wolfskin	-3° C	1550 g	299,-	Liteloft
Cat's Meow	The North Face	-7° C	1450 g	339,-	Polarguard
Igloo Super	Ajungilak	-7° C	2060 g	279,-	Terraloft
Donna	Ajungilak	-7° C	1850 g	349,-	MTI Loft
Tyin	Ajungilak	-15° C	2100 g	399,-	Ultraloft
Travellers Dream	Yeti	3° C	970 g	379,-	Goose-downs
Yeti light	Yeti	3° C	800 g	349,-	Goose-downs
Climber	Finmark	-3° C	1690 g	329,-	Duck-downs
Viking	Warmpeace	-3° C	1200 g	369,-	Goose-downs
Eiger	Yeti	-3° C	1500 g	419,-	Goose-downs
Climber light	Finmark	-7° C	1380 g	349,-	Goose-downs
Cobra	Ajungilak	-7° C	1460 g	449,-	Duck-downs
Cobra Comfort	Ajungilak	-10° C	1820 g	549,-	Duck-downs
Foxfire	The North Face	-10° C	1390 g	669,-	Goose-downs
Mont Blanc	Yeti	-15° C	1800 g	549,-	Goose-downs

Fig. 1. Many-valued context *Sleeping Bags*

For illustration, we will consider a little example which is presented in detail in [9]. It deals with a data set extracted from a katalogue about outdoor-equipment. The objects are sleeping bags with the attributes fabricator F, minimal temperature T, weight W, price P and material M. Their attribute values are represented in the table shown in figure 1.

Additionally given structures on the observation sets O_y can be formalized by relations. Then the formalization has to be extended. We formally define a *relational context* as a tuple $(\Omega, Y, (O, \mathcal{R}), \mathcal{I})$ where (Ω, Y, O, I) is a many-valued context and $\mathcal{R}$ a set of relations on O. In particular, the observation sets $O_y \subseteq O$ are often ordered, like the observation sets O_T, O_W, O_P in our example, which can be considered as subsets of the linearly ordered set $(\mathbb{Z}, \leq)$.

The basic approach of symbolic data analysis consists of extending the data array by unary predicates which are constructed with attributes and attribute values of the many-valued context. In our example, we could consider the symbolic objects like

$$[M = \{\text{goose-downs, duck-downs}\}] \wedge [P = [250, 400]]$$

This predicate is satisfied by all sleeping bags whose material is goose-down or duck-down and which has a price between 200,- and 400,- DM. Using predicates like these, one can describe objects and classes of objects in a more complex way than by only using the attributes of the many-valued context.

For that, the terminology of symbolic objects is defined. This terminology determines a derived context which is considered to be the extended data array. In formal concept analysis, this procedure of defining a terminology and deriving the context is called *logical scaling*. We shall give a short introduction to this general procedure before we describe the specific terminology of symbolic objects:

3 Logical Scaling in Formal Concept Analysis

Formal concept analysis provides different so-called *scaling methods* to transform a many-valued context (Ω, Y, O, I) into a formal context (Ω, P, E) whose extents can be thought of as the "meaningful" subsets of Ω. From this formal context, conceptual hierachies can be explored and represented by line diagrams based on concept lattices (cf. [5]). It must be emphasized that the transformation itself can never be conducted automatically because it depends on the research questions. Hence, scaling is a first, purpose-oriented interpretation of the data.

The approach of *logical scaling* itself was developed when we tried to integrate the language of symbolic objects into formal concept analysis (cf. [9]) and is described in detail in [10]. The basic idea is to use a formal language to generate unary predicates from attributes and attribute values of the many-valued context. These predicates form a terminology.
Formally, we define a *terminology* as a tuple $(\mathcal{P}, \mathcal{N}, \nu) =: \mathcal{T}$ where $\mathcal{P}$ is a set of unary predicates, N a set of names of attributes and ν a surjective *naming function* $\nu: N \to \mathcal{P}$. If the naming function is ν equals $id: \mathcal{P} \to \mathcal{P}$, i. e. the predicates are not named, we can write $\mathcal{P}$ for the terminology.

$$
\begin{array}{lll}
\nu\colon N & \to \mathcal{P} & \\
\quad \mathsf{cheap} & \mapsto (\mathsf{Price} \leq 250) & \\
\quad \mathsf{not\ expensive} & \mapsto (\mathsf{Price} > 250 \wedge \leq 400) & \\
\quad \mathsf{expensive} & \mapsto (\mathsf{Price} > 400) & \\
\quad \mathsf{down\ fibres} & \mapsto (\mathsf{Material} = \text{goose-downs} \vee \text{duck-downs}) & \\
\quad \mathsf{synthetic\ fibre} & \mapsto (\mathsf{Material} \neq \text{goose-downs} \vee \text{duck-downs}) & \\
\quad \mathsf{good} & \mapsto ((T > 0 \wedge \leq 7) \wedge (W \leq 1000)) \vee & \\
 & \quad ((T > -7 \wedge \leq 0) \wedge (W \leq 1400)) \vee & \\
 & \quad ((T > -15 \wedge \leq -7) \wedge (W \leq 1700)) \vee & \\
 & \quad\ (T \leq -15) \wedge (W \leq 2000)) & \\
\quad \mathsf{acceptable} & \mapsto ((T > 0 \wedge \leq 7) \wedge (W \leq 1400)) \vee & \\
 & \quad ((T > -7 \wedge \leq 0) \wedge (W \leq 1700)) \vee & \\
 & \quad ((T > -15 \wedge \leq -7) \wedge (W \leq 2000)) \vee & \\
 & \quad\ (T, \leq -15) & \\
\quad \mathsf{bad} & \mapsto ((T > 0 \wedge \leq 7) \wedge (W > 1400)) \vee & \\
 & \quad ((T > -7 \wedge \leq 0) \wedge (W > 1700)) \vee & \\
 & \quad ((T > -15 \wedge \leq -7) \wedge (W > 2000)) &
\end{array}
$$

Fig. 2. Example for a terminology

Let us consider a terminology for our example. Therefore, we specify a set of attribute names like

$$N := \{\text{cheap, not expensive, expensive, down fibres,}\\ \text{synthetic fibres, good, acceptable, bad}\}.$$

Every attribute name is assigned to a predicate which describes its special meaning in this terminology. For the attributes cheap, not expensive and expensive, we have specified certain intervals of prices. The attributes down fibres and synthetic fibres are decribed by the material, and the attributes good, bad and acceptable refer to the relation between the minimal temperature and the weight: If a sleeping bag does not stand low temperature although it is quite heavy, we want to call it bad.

Obviously, all these attributes and their descriptions can only be justified by the question we have for our data analysis. Nevertheless, the decisions which are taken here can always be discussed because they are explictly specified in the terminology.

When we *scale logically* the many-valued context $\mathbb{K} := (\Omega, Y, O, I)$ by the terminology $\mathcal{T} := (\mathcal{P}, \mathcal{N}, \nu)$, we obtain the one-valued *derived context* $\mathbb{K}^{\mathcal{T}} := (\Omega, N, E)$ where the relation $E \subseteq \Omega \times N$ is given by the semantics of the describing predicates. For all objects $o \in \Omega$ and names of attributes $m \in N$ we define

$$o\, E\, m :\iff o \text{ satisfies } \nu(m).$$

$\mathbb{K}^{\mathcal{T}}$	cheap	not expensive	expensive	down fibres	synthetic fibres	good	acceptable	bad
One Kilo Bag	×				×	×	×	
Sund	×				×			×
Kompakt Basic	×				×	×	×	
Finmark Tour	×				×			×
Interlight Lyx	×				×			×
Kompakt		×			×		×	
Touch the Cloud		×			×		×	
Cat's Meow		×			×	×	×	
Igloo Super		×			×			×
Donna		×			×		×	
Tyin		×			×		×	
Travellers Dream		×		×		×	×	
Yeti light		×		×		×	×	
Climber		×		×			×	
Viking		×		×		×	×	
Eiger			×	×			×	
Climber light		×		×		×	×	
Cobra			×	×		×	×	
Cobra Comfort			×	×			×	
Foxfire			×	×		×	×	
Mont Blanc			×	×		×	×	

Fig. 3. Derived context *Sleeping Bags*

When we derive the many-valued context *Sleeping Bags* by the terminology given in figure 2, we obtain the derived context shown in figure 3. Its objects are the objects of the original context, and its attributes are the attribute names of the terminology.

From the derived context, we can explore the conceptual structure by means of formal concept analysis. We briefly recall the basic definitions; for more details, please refer to [12] or [6]. For a *formal context* (Ω, N, E) where Ω and N are sets and E is a relation between Ω and N, a *formal concept* is defined as a pair (A, B) with $A \subseteq \Omega$, $B \subseteq N$, $A^E := \{m \in N \mid o\,E\,m \text{ for all } o \in A\} = B$ and $B^E := \{o \in \Omega \mid o\,E\,m \text{ for all } m \in B\} = A$. A and B are called the *extent* and the *intent* of the concept (A, B), respectively. The *hierarchical order* of concepts is defined by $(A_1, B_1) \leq (A_2, B_2) :\Longleftrightarrow A_1 \subseteq A_2$. The set of all concepts of the formal context (Ω, N, E) with the hierarchical order is a complete lattice. It is called *concept lattice* of (Ω, N, E) and is denoted by $\underline{\mathfrak{B}}(\Omega, N, E)$.

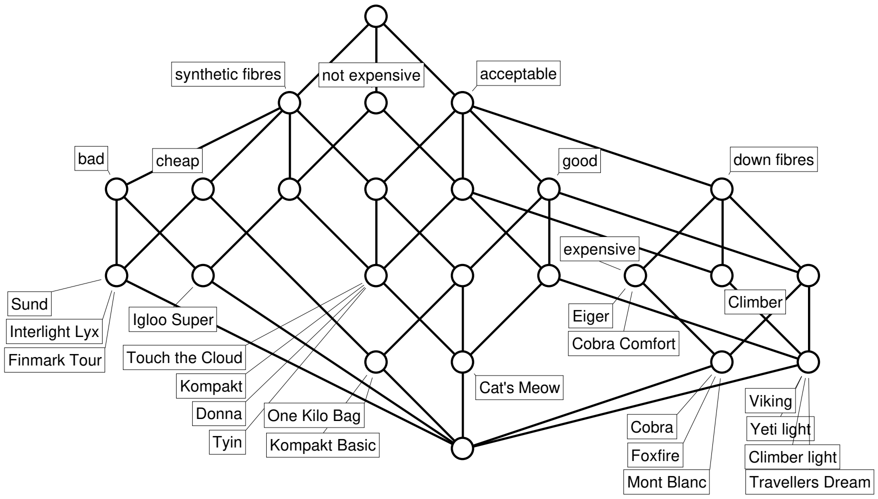

Fig. 4. Concept lattice of the logically scaled context *Sleeping Bags*

The concept lattice of the derived context *Sleeping Bags* is shown in figure 4. From this concept lattice, we can read off interesting conceptual patterns. For example, there is no sleeping bag out of down fibres which is cheap. On the other hand, all expensive sleeping bags are made out of down fibres. All bad sleeping bags are out of synthetic fibres whereas god sleeping bags are out of down or synthetic fibres. The only good and cheap sleeping bags are "One Kilo Bag" and "Kompakt Basic" which are necessarily out of synthetic fibres.

4 Symbolic Objects

For symbolic objects, Diday gives a general description which will be substantiated later: A symbolic object is "a description which is expressed by means of a conjunction of predicates, in terms of the values taken by the variables" ([2], p. 4). There are different types of symbolic objects called *events, assertion objects, horde objects, synthesis objects* etc., all of which are constructed by a specific combination of basic symbolic objects: the events. In [2], it is proved that horde objects and synthesis objects can be expressed by assertion objects of a suitable many-valued context. For this reason, we only consider events and assertion objects in the following paragraphs.

Events. Let (Ω, Y, O, I) be a many-valued context. A unary predicate p is called an *event* if there exists an attribute $y \in Y$ and a subset V of the observation set O_y such that for all $o \in \Omega$ it holds

$$o \text{ satisfies } p \quad \Longleftrightarrow \quad y(o) \in V.$$

In the formal language of symbolic objects, this predicate is denoted by the expression $[y = V]$. Note that, as well as the name "event" itself, the formal language which is used for the description of predicates is derived from the language of statistics. Generally, the formalism used in symbolic data analysis was strongly influenced by statistics.

Every event is a disjunction of atomic predicates, independent of the formal language we chose. The event p can also be expressed in a formal language used for logical scaling in formal concept analysis. Here, the event $[y = V]$ is represented by the expression

$$\bigvee_{v \in V} (y = v).$$

Assertion Objects. The assertion objects are conjunctions of events. A unary predicate p is called an *assertion object* if there exists a set $X \subseteq Y$ of attributes and subsets V_y of the observation sets O_y for all $y \in X$ such that, for all $o \in \Omega$, it holds

$$o \text{ satisfies } p \quad \Longleftrightarrow \quad y(o) \in V_y \text{ for all } y \in X.$$

Such an assertion object is represented by the expression $\bigwedge_{y \in X} [y = V_y]$ or, in the language of logical scaling, by

$$\bigwedge_{y \in X} \big(\bigvee_{v \in V_y} (y = v) \big).$$

The set of all predicates of this conjunctive normal form is called the *terminology $\mathcal{T}$ of the symbolic objects* where the naming function is just the identity. (From now on, we will write symbolic object meaning assertion object.)

The extended data array that symbolic data analysis is working with, can be considered as the derived many-valued context $\mathbb{K}^{\mathcal{T}} := (\Omega, \mathcal{T}, \mathcal{E})$ whose objects are the individual objects of $\mathbb{K}$ and whose attributes are the symbolic objects, i. e. the assertion objects.

Obviously, the derived context is very large. For one single attribute y, there are $2^{|O_y|}$ events. Thus, in practice, only a subset of symbolic objects will determine our terminology. For example, in the case of a quantitative variable $y \in Y$ with observation set $\mathbb{R}$, not all subsets of $\mathbb{R}$ are taken into account but only a subset of the intervals. Further restrictions of the terminology are usually necessary. Usually, the derived context is not explicitly calculated, but it provides the formalism for the knowledge representation.

5 Basic Notions for Symbolic Objects

For the presentation of the basic notions of the language used in symbolic data analysis, let us start with a many-valued context $\mathbb{K} := (\Omega, Y, O, I)$. We construct the set $\mathcal{S}$ of symbolic objects which consists of all predicates of the form

$$\bigwedge_{y \in Y} [y = V_y] \qquad \text{where } V_y \subseteq O_y \text{ for all } y \in Y.$$

We shall see that most of the basic notions used in symbolic data analysis have direct correspondents in the language of formal concept analysis if we consider the derived context $\mathbb{K}^{\mathcal{S}} := (\Omega, \mathcal{S}, \mathcal{E})$ with

$$o\, E\, p :\Longleftrightarrow o \text{ satisfies } p \qquad \text{for all } o \in \Omega, p \in \mathcal{S}.$$

Extension. In symbolic data analysis, the *extension* of the assertion object

$$p := \bigwedge_{y \in X} [y = V_{(p,y)}]$$

is defined as the set of all individual objects which satisfy p, that means $Ext(p) := \{o \in \Omega \mid \forall y \in X \;\; y(o) \in V_{(p,y)}\}$. This definition of extension corresponds directly to the *extents* defined in formal concept analysis. The attribute extent of an attribute p in a formal context is specified as the extent of the attribute concept $(p^E, \, p^{EE})$, i. e. as $p^E := \{o \in \Omega \mid o \, E \, p\} = \{o \in \Omega \mid o \text{ satisfies } p\} = Ext(p)$. In the same way, the equivalence and symbolic order of symbolic objects can easily be explained in the derived context and its concept lattice.

Equivalence. Two symbolic objects $p, q \in \mathcal{S}$ can be called *equivalent* (written $p \sim q$) if their extensions are identical. In the derived context, two equivalent symbolic objects have identical attribute extents and thus they have the same attribute concept.

Symbolic order. The *symbolic order* of symbolic objects is a preorder defined by the inclusion of extensions. We have

$$p \leq q :\Longleftrightarrow \; Ext(p) \subseteq Ext(q) \quad \text{ for all } p, q \in \mathcal{S}.$$

That means, we can read off the symbolic order in the concept lattice because it reflects the hierarchical order defined on formal concepts of the derived context. A symbolic object p is less than another symbolic object q for the symbolic order if the attribute concept (p^E, p^{EE}) is less than the attribute concept (q^E, q^{EE}) in the concept lattice $\underline{\mathfrak{B}}(\mathbb{K}^{\mathcal{S}})$. In other words, the ordered set of equivalence classes of symbolic objects is isomorphic to the ordered set of attribute concepts of the derived context.

We conclude that if we consider symbolic objects to be attributes of a logically derived context, the basic notions of the language of symbolic objects can easily be integrated because they correspond to those of formal concept analysis. This correspondence justifies the interpretation of symbolic objects as more complex attributes and not as more complex objects.

Intension. In contrast to extension, the notion of intension is rather different in the two aproaches. In symbolic data analysis, the *intension* of a set $A \subseteq \Omega$ of objects is defined as a single symbolic object:

$$Int(A) := \bigwedge_{y \in Y} [y = y(A)]$$

Thus, it is the conjunction of all symbolic objects whose extensions contain A. In formal concept analysis, the *intent* of $A \subseteq \Omega$ is defined as the set of all attributes whose extents contain A, i. e. as the set A^E of attributes in the derived context $\mathbb{K}^{\mathcal{S}}$. That means, the two notions are not equal, but yet similar; they have identical extensions. For all $A \subseteq \Omega$, we have

$$Ext(Int(A)) = Int(A)^E = A^{EE} = (A^E)^E.$$

Union and Intersection. The definition of union and intersection given in the literature of symbolic data analysis depends on the considered subset of symbolic objects. A general definition can be given for arbitrary subsets U of the set $\mathcal{S}$. (cf. [1]).

The *union* in $U \subseteq \mathcal{S}$ of two symbolic objects p and q is defined as the conjunction of all symbolic objects of U whose extension contains the extension of p and q:

$$p \sqcup_U q := \bigwedge\{s \in U \mid Ext(p) \cup Ext(q) \subseteq Ext(s)\}$$

Accordingly for the intersection: the *intersection* in $U \subseteq \mathcal{S}$ of two symbolic objects p and q is defined as the conjunction of all symbolic objects in U whose extension contains all objects which are in the extension of p and q:

$$p \sqcap_U q := \bigwedge\{s \in U \mid Ext(p) \cap Ext(q) \subseteq Ext(s)\}$$

For the complete set $\mathcal{S}$, union and intersection are defined as the intension of $Ext(p) \cup Ext(q)$ and $Ext(p) \cap Ext(q)$ respectively. They can be specified by

$$\bigwedge_{y \in V} [y = V_{(p,y)}] \sqcup_{\mathcal{S}} \bigwedge_{y \in V} [y = V_{(q,y)}] \quad = \quad \bigwedge_{y \in V} [y = V_{(p,y)} \cup V_{(q,y)}]$$

and

$$\bigwedge_{y \in V} [y = V_{(p,y)}] \sqcap_{\mathcal{S}} \bigwedge_{y \in V} [y = V_{(q,y)}] \quad = \quad \bigwedge_{y \in V} [y = V_{(p,y)} \cap V_{(q,y)}]$$

The extension of the union (intersection) of two symbolic objects is identical to the extension of their disjunction (conjunction) which is not an element of the terminology $\mathcal{S}$. That is why union and intersection are defined. Using these notions, conjunction and disjunction can be constructed without abandoning the given syntax.

6 The Lattice of Symbolic Objects

Let $<p>:= \{q \in \mathcal{S} \mid \text{II} \sim \surd\}$ be the equivalence class of the symbolic object $p \in \mathcal{S}$, and $\mathcal{S}_\sim$ the set of equivalence classes, i. e.

$$\mathcal{S}_\sim := \{\, <\surd>| \surd \in \mathcal{S}\}.$$

Obviously, the *symbolic* (pre-)*order* $\leq$ on $\mathcal{S}$ induces an order on $\mathcal{S}_\sim$. It can be proved, that the supremum and the infimum of two equivalence classes $<p>$ and $<q>$ in $\mathcal{S}_\sim$ are equal to the equivalence class of the union and the intersection of two representatives, respectively:

$$<p> \vee <q>=<p \sqcup_{\mathcal{S}} q> \quad \text{and} \quad <p> \wedge <q>=<p \sqcap_{\mathcal{S}} q>$$

Infimum and supremum always exist and are commutative and associative. Thus, it is proved in [2] that $\mathcal{S}_\sim$ is a lattice for the induced symbolic order. We add that the lattice is complete. The following theorem explains the coherence between this lattice and the concept lattice of the logically derived context:

Theorem 1. *Let $\mathbb{K} := (\Omega, \mathbb{Y}, \mathbb{O}, \mathbb{I})$ be a many-valued context and let $\mathcal{S} := \{\bigwedge_{y \in Y} [y = V_y] \mid y \in Y, V_y \subseteq O_y\}$ be the terminology of the symbolic objects.*

Then the lattice $\mathcal{S}_\sim$ of equivalence classes of symbolic objects with the induced symbolic order is isomorphic to the concept lattice of the logically derived context $\mathbb{K}^\mathcal{S} := (\Omega, \mathcal{S}, \mathcal{E})$.

Proof. In the concept lattice $\underline{\mathfrak{B}}(\mathbb{K}^\mathcal{S})$, every concept is an attribute concept, i. e. for every $(A, B) \in \underline{\mathfrak{B}}(\mathbb{K}^\mathcal{S})$ there exists a predicate $a \in \mathcal{S}$ with $(a^E, a^{EE}) = (A, B)$. That is because for every concept extent $A \subseteq \Omega$ we have $A = A^{EE} = Int(A)^E$. Thus, the concept (A, A^E) is identical to the attribute concept of the attribute $Int(A)$. Consequently, the assignment $\varphi : \underline{\mathfrak{B}}(\mathbb{K}^\mathcal{S}) \to \mathcal{S}_\sim$ with $\varphi(p^E, p^{EE}) = <p>$ is a bijective. In fact, it is an order-isomorphism. □

This theorem is important for the understanding of the formalism based on symbolic objects. It points out that all sets of objects which are specified by a set of common attributes (i. e. all monothetic classes of objects) can be described by a single attribute in the extended context.

If all concepts of the concept lattice are attribute concepts, the attribute order describes the whole lattice. This is the reason for the language in symbolic data analysis being so much concentrated on symbolic objects: The complete conceptual structure of the data is given by the order of symbolic objects. In consequence, the duality of intension and extension which is constitutive for the approach is neglected.

Complete Symbolic Objects. Symbolic objects do not always specify explicitly all common attributes of the objects in their extensions. In other words, these symbolic objects do not correspond to the intension of their extension. That is only true for complete symbolic objects. A symbolic object $p \in \mathcal{S}$ is called *complete* if it holds $p = Int(Ext(p))$.

In [1, p. 15], it is pointed out, that "This notion agrees with the notion of 'concepts' used by, for instance, Wille (1981) and Ganter (1984) and 'Galois closure' used by Guénoche (1989) for binary tables [i. e. one-valued contexts]." This is because concept intents always satisfy $B = B^{II}$.

We can find further correspondences if we consider the structure of the set $\mathcal{C}$ of all complete symbolic objects. It can be proved that the mapping $f : p \mapsto Int(Ext(p))$ which assigns an equivalent complete symbolic object to every symbolic object, is a closure operator on the preordered set $(\mathcal{S}, \leq)$. Consequently, the set $\mathcal{C}$ of all complete symbolic objects is a lattice with the symbolic order.

Lemma 2. *The lattice $\mathcal{C}$ of all complete symbolic objects is isomorphic to the lattice $\mathcal{S}_\sim$ of equivalence classes of symbolic objects.*

Proof. We can find a complete symbolic object in every equivalence class $<p>$ by the closure operator f. Conversely, there is only one complete symbolic object in every equivalence class because for $p, q \in \mathcal{C}$ with $<p> = <q>$, we obtain $Ext(p) = Ext(q)$ which implies $p = Int(Ext(p)) = Int(Ext(q)) = q$. □

Although this theorem is not explicitly stated in symbolic data analysis, it is important. By means of this isomorphism, we can guarantee that all subsets of objects which can be described by symbolic objects can already be described by a complete symbolic object. In other words, the set $\mathcal{C}$ of complete symbolic objects builds a system of representatives of the equivalence classes of $\mathcal{S}$. That means, in the derived context $\mathbb{K}^{\mathcal{S}} := (\Omega, \mathcal{S}, \mathcal{E})$, for every attribute there exists a complete symbolic object with identical attribute extent. Thus, the subcontext $\mathbb{K}^{\mathcal{C}} := (\Omega, \mathcal{C}, \mathcal{E} \cap (\Omega \times \mathcal{C}))$ has an isomorphic concept lattice. Together with theorem 1 we conclude the following corollary:

Corollary 3.

$$\mathcal{S}_{\sim} \;\cong\; \mathcal{C} \;\cong\; \underline{\mathfrak{B}}(\mathbb{K}^{\mathcal{S}}) \;\cong\; \underline{\mathfrak{B}}(\mathbb{K}^{\mathcal{C}})$$

In other words, if we eliminate all incomplete symbolic objects of the context $\mathbb{K}^{\mathcal{S}} := (\Omega, \mathcal{S}, \mathcal{E})$, we obtain the subcontext $\mathbb{K}^{\mathcal{C}} := (\Omega, \mathcal{C}, \mathcal{E} \cap (\Omega \times \mathcal{C}))$ whose extents are identical to those of $\mathbb{K}^{\mathcal{S}}$. Thus, they have an isomorphic concept lattice. In the language of formal concept analysis, $\mathbb{K}^{\mathcal{C}}$ is called a *clarified context* of $\mathbb{K}^{\mathcal{S}}$. This clarified context is the extended data array which symbolic data analysis usually treats instead of the context $\mathbb{K}^{\mathcal{S}}$.

7 Classification and Knowledge Representation in Symbolic Data Analysis

The language of symbolic objects we have presented, is used for various methods of numerical and symbolic data analyis (cf. [3] for a survey). Here, we only give a short survey on those domains which are concerned with conceptual classification and knowledge processing.

7.1 Identification Problems

Symbolic objects are often used to process knowledge bases in order to solve identification problems. There are two main problems: either, a classification of a set of individual objects is to be found, or, an existing classification is to be represented in such a way that the assignment of individual objects to their classes is as simple as possible. Both approaches are pursued for example by Lebbe and Vignes (cf. e. g. [7]) who deal with various methods of identification in biology and medicine. Instead of trying to find complete descriptions for all monothetic classes, they usually want to find so-called discriminants. These are minimal sets of attributes to identifiy a class.

In describing existing classes (e. g. given by taxonomies), one frequently runs into the problem of having to deal with non-monothetic classes. Thus, they usually cannot be represented by assertion objects. In such cases, more complex symbolic objects such as horde objects, synthesis objects, or rule objects need to be used.

7.2 Classification

In formal concept analysis, we determine all monothetic classes of the derived context and deduce the concept lattice in order to explore the conceptual structure. Methods and algorithms have been developed to find dependencies and conceptual patterns in the structure.

But as we have seen, the set of symbolic objects and even the restricted set of complete symbolic objects is very large. This is why specifying all monothetic classes is often considered as unfeasible in symbolic data analysis. Thus, criteria and procedures must be developed to select the interesting classes.

Often, these classes are built by means of classical methods like clustering. In a second step, these classes are described by symbolic objects. Then, these symbolic objects are examined in order to find dependencies and regularities (cf. e. g.[4], [1]). Evidently, the regularities which can be explored, depend on the preceding method of classification. This raises the question how far the methods of classification allow an appropriate interpretation with regard to the research questions.

In general, the approach of formal concept analysis is different. A preliminary decision has to be made to select the interesting attributes. Then, all monothetic classes are considered which are described by these selected attributes. Following this approach, the necessary reduction can be made with regard to the subject matter more easily because the criteria for the selection of attributes can be deduced from the research question more directly than the selection of certain classes.

7.3 Classification of Classes

In symbolic data analysis, it is often emphasized that the symbolic objects themselves can be the objects of data analysis and can be treated with various methods. Not the individual objects, but classes of individual objects which are represented by symbolic objects are then analysed and classified (cf. [1]). Therefore, the notions *elementary symbolic object* and *symbolic extension* are defined.

Elementary Symbolic Objects. An assertion object p is called *elementary* if it exists an object $o \in \Omega$ where

$$p = \bigwedge_{y \in Y} [y = y(o)].$$

We can assign an elementary symbolic object to every individual object by the mapping

$$\varepsilon : \Omega \to \mathcal{S} \qquad \text{with } \varepsilon(\wr) = \bigwedge_{\dagger \in \mathcal{Y}} [\dagger = \dagger(\wr)] =: \wr^{\mathcal{S}}.$$

Assuming that the original many-valued context is object-clarified (i. e. that there are no two different objects $o, u \in \Omega$ with $y(o) = y(u)$ for all $y \in Y$), the mapping ε is injective. Thus, ε is an embedding of the set Ω of individual objects into the set $\mathcal{S}$ of symbolic objects. It identifies the objects with their describing symbolic objects. The identification is suggested by the notations $o^{\mathcal{S}} := \varepsilon(o)$ and $\varepsilon(\Omega) := \Omega'$. In the language of formal concept analysis, the embedding ε assigns the object concept (o^{EE}, o^{E}) to every object $o \in \Omega$. The same concept can be represented as the attribute concept of $o^{\mathcal{S}}$. The general concentration on attributes suggests that we consider only the predicate $o^{\mathcal{S}}$, i. e. the elementary symbolic object of the object o, and not its attribute concept.

Having substituted the individual objects by their describing elementary symbolic object, the basic notions can be defined by means of elementary symbolic objects:

Symbolic Extension. For a symbolic object $p \in \mathcal{S}$, the *symbolic extension* is defined by

$$Ext_{\Omega'}(p) := \{o^{\mathcal{S}} \in \Omega' \mid o \in Ext(p)\}$$

In other words, the symbolic extension of p consists of the embedded individual objects of the extension $Ext(p)$. Obviously, the symbolic order, union and intersection of symbolic objects can also be defined by the symbolic extension instead of the extension. We obtain the same lattice $\mathcal{S}_\sim$ which is again isomorphic to $\mathcal{C}$. Using these notions, the individual objects have completely disappeared.

General Objects. A class $C \subseteq \Omega$ of individual objects which is described by a symbolic object is called *general object.* For the transition from individual to general objects, we use the embedding of the set Ω of individual objects into the set $\mathcal{S}$ of symbolic objects and we describe a set $\mathcal{G}$ of general objects as a subset of $\mathcal{S}$. Using this notion, it is not necessary to know the whole context. In particular, we do not need to specify the attribute value of every individual object. Only the attribute values of every set of individual objects merged to one general object $C \in \mathcal{G} \subseteq \mathcal{S}$ need to be known.

In the derived context, the transition from individual objects to general objects can only be realized by substituting the set Ω of individual objects by a subset G of the power set $\mathfrak{P}(\Omega)$. If G is the set of all concept extents of $\mathbb{K}^{\mathcal{S}}$ it can be proved that the new context has a concept lattice isomorphic to $\underline{\mathfrak{B}}(\mathbb{K}^{\mathcal{S}})$. For $G' \subset G$, the corresponding concept lattice is a $\bigwedge$-subsemilattice of $\underline{\mathfrak{B}}(\mathbb{K}^{\mathcal{S}})$.

Further research on the transition from contexts to so-called class-contexts is needed. A first approach to a formal concept analysis of general objects is made in [9].

Classes of General Objects. In [2], different approaches are presented to describe classes of symbolic objects. The simplest way is to assign again a symbolic object to every class of symbolic objects, for example the intersection or the union of its elements. Then, a preordered set of classes of symbolic objects is obtained where equivalence and order of classes are defined by means of the symbolic extension.

But if a class $C \subseteq \mathcal{G}$ of symbolic objects is described by the union $\bigsqcup_{s \in C} s$ of all class-elements, the symbolic extension $Ext_{\Omega'}(\bigsqcup_{s \in C} s)$ of this union sometimes is bigger than the union $\bigcup_{s \in C} Ext_{\Omega'}(s)$ of each symbolic extension. That is why the extension of a class $C \subseteq \mathcal{S}$ is defined as the union of the symbolic extensions.

This notion raises some questions about the describability of class extensions by means of symbolic objects.

This feature is described by the notion *stability*. It measures to what extent a class can be described by a conjunction of events and thus, by its common attributes. The most stable classes are the monothetic classes.

Given a class $C \subseteq \mathcal{G}$, it can be interesting to know the cardinality of the smallest subset of $\mathcal{G}$ which generates C. This feature is formalized by the notion crumbling: the *crumbling* of a class $C \subseteq \mathcal{G}$ of symbolic objects is the smallest number of symbolic objects of $\mathcal{G}$ whose union of extensions contain the extension of C.

This notion measures to what extent the class $C \subseteq \mathcal{G}$ is represented by assertion objects. In particular, it is equal to one if the class is most stable. Stability and crumbling are often important criteria for the quality of a classification of symbolic objects.

7.4 Modal Symbolic Objects

Modal symbolic objects are considered to be an important extension for the language of symbolic objects. They have been introduced to allow the symbolic objects to be applied to knowledge representation of uncertain knowledge. They are used to formalize restrictions like "it is possible that", "surely", "he does not know if" or temporal restrictions.

Therefore, symbolic objects are not considered as predicates anymore but as mappings from the set Ω of individual objects to an ordered set, usually the interval $[0,1] \subseteq \mathbb{R}$. Then, the data array of symbolic data analysis is no more a one-valued context (Ω, P, E), but a many-valued context $(\Omega, P, [0,1], I)$ where P is a set of modal symbolic objects and I is defined by $(o,p,x) \in I :\Longleftrightarrow p(o) = x$.

This context corresponds exactly to the *fuzzy-context* by which fuzzy knowledge is formalized in formal concept analysis. (It is treated in detail in the book [8].) Nevertheless, the methods of knowledge processing for these contexts are very different in symbolic data analysis and formal concept analysis, but we will not be able to discuss that issue in this paper (cf. [3] and [8] for a survey).

8 Discussion

Providing a formal language for attributes of many-valued contexts and by the approach of logical scaling, we could tie together the general approaches of symbolic objects and formal concept analysis in order to extend the scope of knowledge representation. The introduction of logical scaling allows us to integrate the most important notions for the language of symbolic objects into formal concept analysis. This is a connection both approaches can gain from.

Methods and experiences from formal concept analysis now are directly available for research in symbolic data analysis: Computation methods for dependencies and implications of attributes and a relational language of attributes and predicates, to name just two features.

Conversely, formal concept analysis has already gained insights which led to results like the new method logical scaling. Among the remaining material for future research are the analogy of modal symbolic objects and fuzzy concepts a well as a formal concept analysis of general objects.

References

1. Diday, E. / Brito, P. (1989): Symbolic cluster analysis, in: O. Opitz (ed.): Proceeding der 13. Konferenz der Gesellschaft für Klassifikation, Universität Augsburg
2. Diday, E. (1987): Introduction à l'approche symbolique en analyse des données, in: Actes des journées symboliques-numériques pour l'apprentissage de connaissances à partir des données, Paris
3. Diday, E. (1993): From data to knowledge. Boolean, probabilist, possibilist and belief objects for Symbolic Data Analysis. Une introduction à l'analyse des données symboliques, INRIA–Rocquencourt
4. Diday, E. / Roy, L. (1988): Generating rules by symbolic data analysis and application to soil feature recognition, in: Actes des 8èmes Journées Internationales: Les systèmes experts et leurs applications, Avignon
5. Ganter, B. / Wille, R. (1989): Conceptual scaling, in: F. Roberts (ed.): Applications of combinatorics and graph theory to the biological and social sciences, Springer–Verlag, New York, 139 – 167
6. Ganter, B. / Wille, R. (1996): Formale Begriffsanalyse. Mathematische Grundlagen, Springer–Verlag, Berlin – Heidelberg
7. Lebbe, J. / Vignes, R. / Diday, E. (1987): Generate identification graphs and rules by features selection from set of symbolic objects, in: Actes des journées symboliques-numériques pour l'apprentissage de connaissances à partir des données, Paris
8. Pollandt, S. (1996): Fuzzy-Begriffe. Formale Begriffsanalyse unscharfer Daten, Springer–Verlag, Berlin – Heidelberg
9. Prediger, S. (1996): Symbolische Datenanalyse und ihre begriffsanalytische Einordnung, Masters Thesis, FB Mathematik, Technische Hochschule Darmstadt
10. Prediger, S. (1997): Logical Scaling in Formal Concept Analysis, in: Dickson Lukose et.al. (eds): Conceptual Structures: Fulfilling Peirce's Dream. Proceedings of the Fifth International Conference on Conceptual Structures (ICCS '97), Lecture Notes in Artificial Intelligence No. 1257, Springer – Verlag, Berlin
11. Wille, R. (1982): Restructuring lattice theory: an appraoch based on hierarchies of concepts, in: I. Rival (ed.): Ordered sets. Reidel, Dordrecht–Boston, 445 – 470
12. Wille, R. (1992): Concept Lattices and Conceptual Knowledge Systems, in: Computers & Math. Applications, vol. 23, no. 5

This article was processed using the LaTeX macro package with LLNCS style