

Exploration Tools in Formal Concept Analysis

Gerd Stumme

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

© Springer-Verlag Berlin–Heidelberg 1995

Summary: The development of conceptual knowledge systems specifically requests knowledge acquisition tools within the framework of formal concept analysis. In this paper, the existing tools are presented, and further developments are discussed.

1 Introduction

In the last years *conceptual data systems* have been established as tools in data analysis (cf. Vogt and Wille (1995)). Based on the mathematical theory of formal concept analysis (cf. Wille (1982), Ganter and Wille (1995)), they allow to browse through the data and to explore interactively their conceptual structure. Although conceptual data systems are already used within a broad field of applications, they have to be extended in order to make them more flexible. The extension of conceptual data systems to *conceptual knowledge systems* specifically includes the treatment of incomplete data and updates as well as techniques for knowledge inference and knowledge acquisition.

In this paper we focus on techniques for knowledge acquisition. As they are used for exploring incompletely known contexts we refer to them as exploration tools. Specifically, we discuss *Attribute Exploration*, *Object Exploration*, *Concept Exploration* and *Distributive Concept Exploration*. The problem of efficient knowledge acquisition appears quite natural in formal concept analysis, and so the work on these tools began soon after the introduction of formal concept analysis by Wille (1980). Already in Wille (1982) the idea of Concept Exploration is discussed.

All exploration tools presented in this paper are interactive procedures. This means that knowledge is not “learned automatically” by the machine, but has to be asked from an expert of the domain. This means that, concerning complexity, we are not interested in minimizing the calculation that has to be done. What is more important is to minimize the number and complexity of questions the expert has to be asked.

Formal concept analysis is a set-theoretical model for concepts that reflects the philosophical understanding of a concept as a unit of thoughts

consisting of two parts: the extent which contains all objects that belong to the concept, and the intent which contains all attributes common to all the objects (cf. Wagner (1973)). The basic notion which models the knowledge about a specific domain is the *(one-valued) formal context*. A formal context consists of a set of objects, a set of attributes, and a relation that indicates whether an object has an attribute or not. The *formal concepts* can be derived from this context as described below. There is a natural order on the formal concepts that reflects the subconcept-superconcept-relation. The adjective “formal” indicates that we do not talk about concepts of the real world, but only about their counterparts in our model.

In many applications the notion of one-valued contexts is too rigid to model the data adequately. Conceptual knowledge systems are often based on *many-valued contexts*, in which the objects may have different values for an attribute. However the exploration tools described in this paper apply only to one-valued contexts; exploration tools for many-valued contexts have still to be developed. Later in this paper we briefly discuss the conception of *Scale Exploration* for exploring dependencies in many-valued contexts.

We give now the mathematical definitions of (one-valued) contexts and concepts: A *(formal) context* (G, M, I) consists of two sets G and M and a relation I between them. The elements of G and M are called *objects* resp. *attributes*, and $(g, m) \in I$ (or gIm) is read as “the object g has the attribute m ”.

For every set of objects $A \subseteq G$ we define the set $A' := \{m \in M \mid gIm \text{ for all } g \in A\}$ of all attributes shared by all objects in A . Dually the set $B' := \{g \in G \mid gIm \text{ for all } m \in B\}$ is the set of all objects having all attributes in $B \subseteq M$.

A *(formal) concept* of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The set A is called the *extent* of the concept, the set B the *intent*. The hierarchical *subconcept-superconcept-relation* is formalized by $(A_1, B_1) \leq (A_2, B_2) : \iff A_1 \subseteq A_2 (\iff B_1 \supseteq B_2)$.

The set of all concepts of a context (G, M, I) together with this order relation is a complete lattice which is called the *concept lattice* of (G, M, I) (cf. Wille (1982)). This means that for every set of concepts there exist a unique largest subconcept (the *infimum*) and a unique smallest superconcept (the *supremum*).

The following example shows how the line diagram of a concept lattice unfolds the information contained in the underlying data context.

Example. The context in Fig. 1 classifies some pure chemical substances (cf. Mortimer (1983)). The objects in this example are the substances Cl, He, NaCl, H₂O, HCl, and NaOH, and the attributes are “compound”, “element”, “(Brönsted-) acid”, “(Brönsted-) base”, “salt”, “inert gas”, and “halogen”.

	compound	element	acid	base	salt	inert gas	halogen
Cl		x					x
He		x				x	
NaCl	x				x		
H ₂ O	x		x	x			
HCl	x		x				
NaOH	x			x			

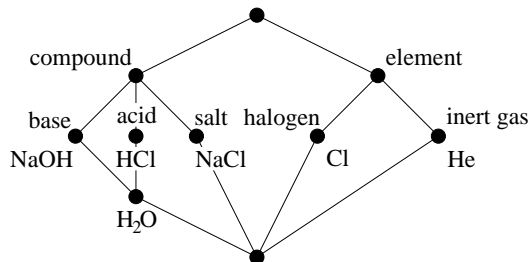


Figure 1: Context and concept lattice of pure chemical substances

The small circles in the line diagram of the concept lattice stand for the concepts. A circle labeled by an object g represents the concept with the smallest extent containing g . A circle labeled with an attribute m represents the concept with the smallest intent containing m . In general, the circle of a concept is linked by a descending path to all the circles that are labeled by objects belonging to the extent of the concept, and by an ascending path to all circles that are labeled by attributes belonging to the intent of the concept. For example, the leftmost concept in the diagram has the objects NaOH and H₂O in its extent and the attributes “base” and “compound” in its intent. The whole information given by the context can though be read from the line diagram of its concept lattice: An object g has an attribute m if and only if there is an ascending path from g to m in the line diagram.

2 Exploration Tools

If a context is completely given, then there is no problem to calculate the concept lattice (cf. Ganter (1987), Burmeister (1987)). In many applications however we either do not have the complete knowledge of the domain in which we are interested or the context is too large to be completely given. As a context consists of the three components G , M and I , there are mainly the following possibilities of incomplete contexts (Wille (1980)):

1. The sets G and M are completely given, but not the relation I .
2. G is incomplete, M is complete.
3. G is complete, M is incomplete.
4. Neither G nor M are completely given.

In the first case, there is no exploration tool available up to now. Such a tool depends on the way how the knowledge asked from the expert is represented (e. g. implications, constraints, ...). In order to get an efficient tool, dependencies between single entries in the relation I have to be considered, so that every answer in the interactive exploration dialogue determines more than one entry.

In the second case we are not interested in completing the whole set G ; it is sufficient to obtain information on “typical objects” which generate the whole concept lattice. *Attribute Exploration* suggests implications between the attributes to the expert. Every rejection of an implication has to be justified with a “typical object” as a counterexample.

The third case is dual to the second one. As all definitions in formal concept analysis are (at least up to now) dually symmetric in G and M , we can just interchange them and then apply Attribute Exploration. So we obtain the *Object Exploration*, in which implications between the objects are suggested to the expert, who — in case of rejection — has to give a “typical attribute” that separates the objects in premise and conclusion. Both attribute and object exploration are discussed in the next section, where also *Rule Exploration* as an extension of Attribute Exploration to horn clauses in first order predicate logic is described.

The fourth case (where neither G nor M are completely known) is more difficult to treat, as we do not have an obvious start as in cases 2 and 3. *Concept Exploration* starts from some “basic concepts” that are assumed to be concepts of the (unknown) context. The expert has to answer questions of the form: “Is s a subconcept of t ?”, where s and t are lattice terms built with the basic concepts. Negative answers have to be answered by an object and an attribute that separate the concepts. The result is the largest lattice that can be generated by the basic concepts with respect to the given answers. Principal and technical difficulties suggested an alternative approach named *Distributive Concept Exploration*. These two exploration tools are described in Section 4.

3 Attribute Exploration, Object Exploration and Rule Exploration

Attribute Exploration can be used to determine the concept lattice of a context, where the set G of objects is either not completely known or too large to be listed completely (G may even be infinite). The implementation in Burmeister (1987) is based on the Next-Closure-Algorithm of B. Ganter (1987) which calculates a minimal set of implications, from which all other implications can be derived (Duquenne and Guigues (1986)). The program starts by asking for the attributes (and eventually some objects) of the context. Then it suggests implications between the attributes to the expert. Either the expert accepts an implication or he gives another object that has all attributes in the premise of the implication, but not all attributes in its conclusion. These objects are enough to determine the structure of the concept lattice. The implications that are valid for these separating objects are exactly those which are valid for all objects in the explored universe.

P. Burmeister (1991) discusses how incomplete knowledge can be handled by the algorithm and gives two examples. In Stumme (1995b) is discussed how the calculation of a minimal set of implications can be done with respect to *background implications* which represent previously given knowledge. There is also an example of an exploration given, where *exceptions* are allowed. Other examples of Attribute Explorations are given in Ganter (1987), Ganter and Zickwolff (1994). A larger exploration is e. g. done by S. Reeg and W. Weiß (1990), who determined the dependencies between 50 attributes of finite lattices with Attribute Exploration. The context in Fig. 1 results from an Attribute Exploration that started with the attributes “compound”, “element”, . . . , “halogen”. Here we obtain for example the implication “base” \rightarrow “compound”, which in fact is valid for all pure chemical substances.

Rule Exploration is a generalization of Attribute Exploration by M. Zickwolff that is able to treat relations between the objects of the field of interest (Zickwolff (1991), Ganter and Zickwolff (1994)). The attributes of the exploration context are positive literals of a first order language, and the objects added by the expert during the exploration are variable assignments to the objects of the field of interest. The application of Attribute Exploration to this specific context is called Rule Exploration. The resulting implications are then Horn clauses of the first order language describing the dependencies between the relations on the objects of the field of interest.

Object Exploration is used, when all objects of the context are given, but not all attributes. It determines “typical attributes” that distinguish the objects. This “structured brainstorming” (Wille) uses the same algorithm as Attribute Exploration, only objects and attributes interchange their roles.

Object Exploration can for example be used for determining key words for books. This has been done in a research project concerning literature of interdisciplinary technology research at the Technische Hochschule Darmstadt. The exploration started with 20 books, for which distinguishing attributes were determined. The following is part of the exploration dialogue:

[. . .]
 “Is *Mayer-Abich* \rightarrow *IZ* valid?”
 “No! *Mayer-Abich* is about “Philosophie”, *IZ* is not.”
 “Is *Meyer-Abich*, *Zwick* \rightarrow *Böhme*, *Bühl*, *IZ*, *Kepplinger*, *Luthe* valid?”
 “No! *Meyer-Abich*, *Zwick*, *Böhme*, *IZ*, and *Kepplinger* are about “Verantwortungsethik”, *Bühl* and *Luthe* are not.”
 “Is *Meyer-Abich*, *Zwick* \rightarrow *Böhme*, *IZ*, *Kepplinger* valid?”
 “Yes!”
 [. . .]

An implication $A \rightarrow B$ is valid if every attribute that is common to all books in A is also common to all books in B . E. g. the last answer expresses that the experts did not find any attribute common to the two books written by *Meyer-Abich* resp. *Zwick*, which is not common to all three books of *Böhme*, *IZ*, resp. *Kepplinger*.

This example already shows that an exploration may request some effort even for a small number of objects. On the other hand one is often only interested in attributes which distinguish the objects on a rather coarse level — at least for the generation of keywords for a library. In this case one could do an “Object Exploration with exceptions”, where implications may be accepted, even if there are contradicting attributes (cf. Stumme (1995b)). It is also promising to generalize Attribute Exploration (resp. Object Exploration) for partial implications (i. e. implications that are only valid for a certain percentage of objects, Luxenburger (1993)).

4 Concept Exploration and Distributive Concept Exploration

In the last section Attribute and Object Exploration were discussed. These knowledge acquisition tools can only be applied if either the set of attributes or the set of objects is completely given. But what can we do if both sets are only partially given?

P. Burmeister suggested to start with an attribute exploration with the given attributes, which extends the set of objects. A succeeding Object Exploration on these objects extends the set of attributes. Both tools are applied alternately until no more objects and attributes are added. Although it is a natural approach, there are no experimental results so far.

Another approach is the *Concept Exploration*. It is based on the observation that “in many situations one has only a vague knowledge of a context although many of its concepts are fairly clear” (Wille (1982)). Concept Exploration starts with these “fairly clear concepts” (called *basic concepts*) and follows the free generating process of lattices. It produces new concepts by taking greatest common subconcepts and least common superconcept. The questions that have to be answered are essentially of the form: “Is s a subconcept of t ?”, where s and t are lattice terms built with the basic concepts. Different concepts are separated by an object and an attribute given by the expert. This conception is already mentioned in Wille (1982). It is elaborated in Wille (1987) and Wille (1989a), where also examples can be found. U. Klotz and A. Mann (1988) further elaborated and implemented the method.

There are some principal problems in using Concept Exploration, which led to the development of another tool named *Distributive Concept Exploration*: Concept Exploration assumes that the exploration takes place in a — potentially given — context (which is sometimes called *conceptual universe* to emphasise that the exploration is restricted to this context). One has to be sure that the basic concepts are really concepts of the conceptual universe before starting the exploration. As the example of a concept exploration in number-theory in Wille (1987) shows, this is not always obvious. Another problem is the interpretation of suprema in the resulting concept lattice. It depends on the set of attributes that is assigned to the conceptual universe. In the number-theory exploration this set is well specified, but in non-mathematical examples often “all possible attributes of the objects” are allowed. The suprema vary depending on how restrictive this “definition” is understood. The interpretation of infima on the other hand is generally no problem, as it depends on the set of objects of the conceptual universe, which is usually already circumscribed by the name of the exploration (e. g. “an exploration of pure chemical substances” in the following example).

Another problem is the fact that it is (at least theoretically) possible that a concept exploration does not end.¹ For usual applications however (where the basic concepts are sufficiently related) this does not seem to be a serious problem.

These difficulties led to the development of *distributive concept exploration* (cf. Stumme (1995a)). Here we understand the conceptual universe in a more liberal way: The decision whether an object or attribute belongs to the universe or not is left to the expert. He is responsible for choosing appropriate objects and attributes. The questions appearing in the dialogue again are essentially of the form: “Is s a subconcept of t ?” with lattice terms s and t ; and if a question is denied, then again an object and an attribute have to be given which separate the two concepts.

All we assume about the universe is that its attributes are closed under disjunction. The infimum of concepts corresponds to the *conjunction* of the attributes in their intents — and therefore it is natural to let the supremum correspond to the *disjunction* of the attributes. Then an object belongs to the supremum of some concepts, if and only if it belongs to at least one of them (as it belongs to their infimum, if and only if it belongs to all of them).

This definition implies that the resulting concept lattices are always completely distributive, and we can benefit from the richer mathematical structure of completely distributive complete lattices (instead of arbitrary (complete) lattices). Specifically we use tensor products and congruence re-

¹The lattice $\text{FL}(3)$ that is freely generated by three elements is infinite, and so a concept exploration in the conceptual universe $(\text{FL}(3), \text{FL}(3), \leq)$ with the three generators as basic concepts cannot end.

lations on completely distributive complete lattices, which can efficiently be described by operations on contexts. This allows a more efficient determination of the questions for the exploration dialogue.

Another advantage is the fact that the bottom element of the concept lattice can now (as supremum of the empty set) be identified with the concept “nothing”, while in the general case only the top element (as infimum of the empty set) represents the concept “everything (in our field of interest)”. The following example shows how the existence of the concept “nothing” supports the interpretation of the result.

Distributive concept exploration subsequently adds the basic concepts. When the concept lattice generated by the first n basic concepts is determined, the $(n+1)$ th basic concept is added. Then the relationships of the already given objects and attributes to the new basic concept are asked from the expert, before he has to answer to questions of the form “Is s a subconcept of t ?”, where s and t are lattice terms built with the basic concepts. If the expert replies “No!”, then he has to justify his answer by a separating pair, i. e. an object g belonging to s and an attribute m belonging to t , such that g does not have the attribute m .

Example. The concept lattice in Fig. 1 is the result of an Attribute Exploration. If we are interested in more details about pure chemical substances, then we can perform a Distributive Concept Exploration instead — now with “compound”, “element”, “acid”, “base”, “inert gas”, and “halogen” as basic concepts. Here is that part of the dialogue which determines the concept lattice generated by the first four basic concepts, when the concept lattice generated by the first three basic concepts “compound”, “element”, and “acid” is already calculated:

[. . .]
 “Is H_2O a base?”
 “Yes!”
 “Is NaOH a base?”
 “Yes!”
 “Is He a base?”
 “No!”
 “Has every base the attribute ‘consists of two or more elements’?”
 “Yes!”
 “Is the infimum of ‘element’ and ‘base’ a subconcept of the supremum of ‘compound’ and ‘acid’?”
 “No! A separating pair is HCl and ‘cannot be split further in substances or can receive protons’.”
 “Is ‘compound’ a subconcept of the supremum of ‘element’, ‘acid’, and ‘base’?”
 “No! A separating pair is NaCl and ‘cannot be split in further

substances or can exchange protons’.”
[...]

The concept lattice generated by all seven basic concepts is shown in Fig. 2. It is drawn as a nested line diagram (Wille (1989b)), in which the 55 small circles represent the concepts. Every thick line between two ellipses has to be replaced by five lines between the corresponding concepts. The separating pairs are indicated by similar symbols (e. g. NaCl and ‘cannot be split in further substances or can exchange protons’, which is a separating pair for “salt” (with symbol \otimes) and the supremum of “base”, “acid”, and “halogen” (with symbol \oplus)).

The concept lattice resulting from Attribute Exploration is infimum-embedded in this lattice. Additionally we obtain concepts which can only be reached from the basic concepts by taking suprema. In general there may also be an additional top and a bottom element.

We can now see in the diagram that there exists no “base” that is also a “salt” (because the infimum of these two concepts is the concept “nothing”) — unlike in the concept lattice resulting from Attribute Exploration, where it is possible that the lowest concept has a non-empty extent. We can also see, if some concepts cover a common superconcept or not: For example, there are other compounds than bases, acids, and salts (e. g. CO₂), because the supremum of “base”, “acid”, and “salt” is unequal to “everything”.

From the fact that the supremum of “compound” and “element” is the top element (the concept “everything”) and the infimum is the bottom element (the concept “nothing”) of the lattice, we can deduce that the pure chemical substances can be divided in two disjunctive classes “compound” and “element”. The whole lattice is thus the direct product of the five-element concept lattice which is generated by the concept “element” and its subconcepts “inert gas” and “halogen” (with “everything”:=“element”) and the eleven-element concept lattice which is generated by the concept “compound” and its subconcepts “base”, “acid”, and “salt” (with “everything”:=“compound”). These two concept lattices are shown in Fig. 3.

The concept lattices resulting from Distributive Concept Explorations are often “blown up at the top”, e. g. the basic concepts are usually placed in the lower half of the lattice. This results from the fact that all possible unions of concept extents are built. The advantage is that one can see if a concept is covered by some of its subconcepts or not. However a lot of concepts may be generated, which are not really interesting. In this case one should consider to split up the exploration in two or more explorations on suitable subsets of basic concepts as, e. g., in Fig. 3.

In the existing examples of Concept Explorations, where the set of attributes is only vaguely fixed (“all possible attributes of the objects”), the results are mainly the same when distributive concept exploration is applied.

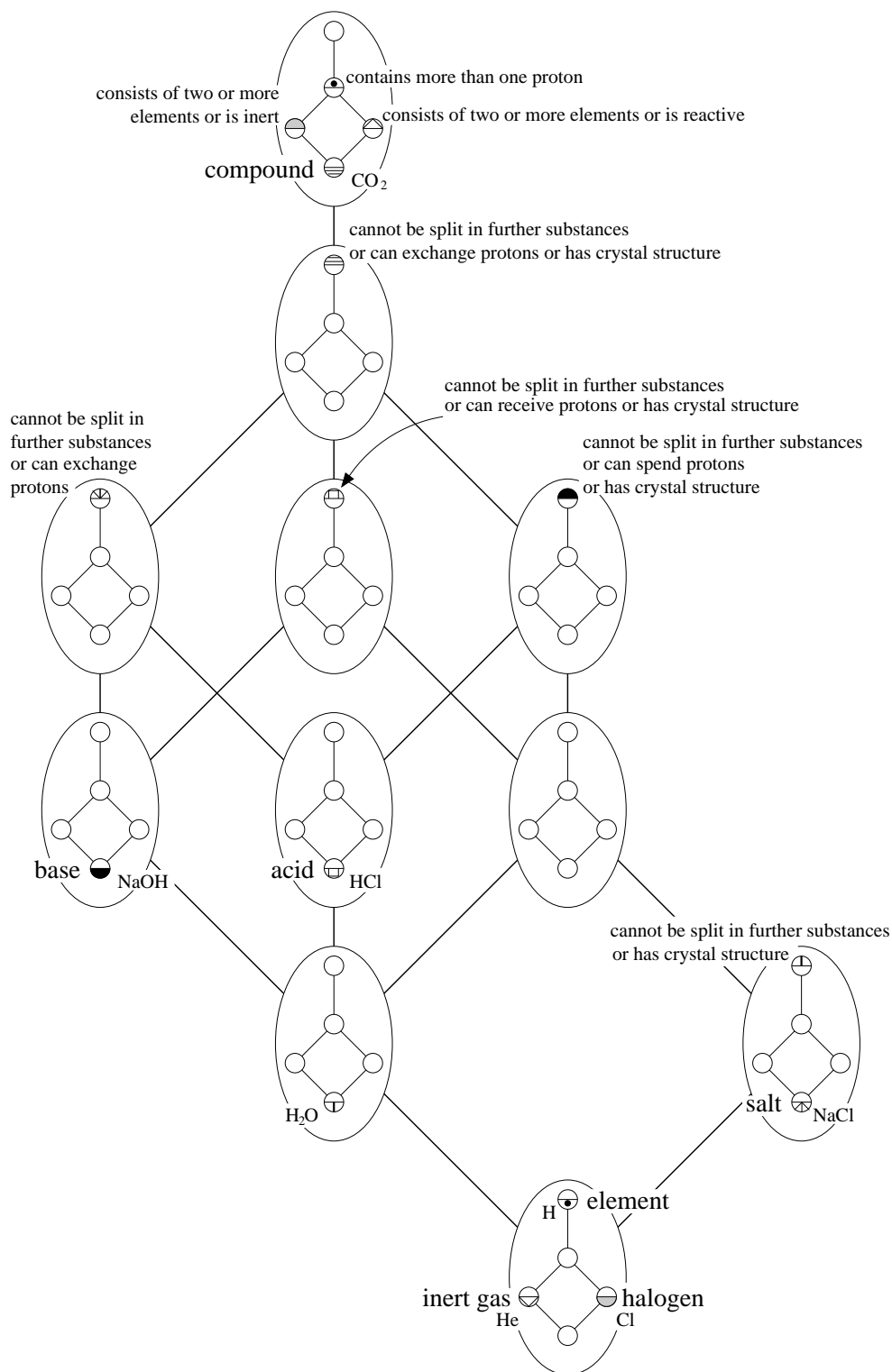


Figure 2: Result of the distributive concept exploration of pure chemical substances

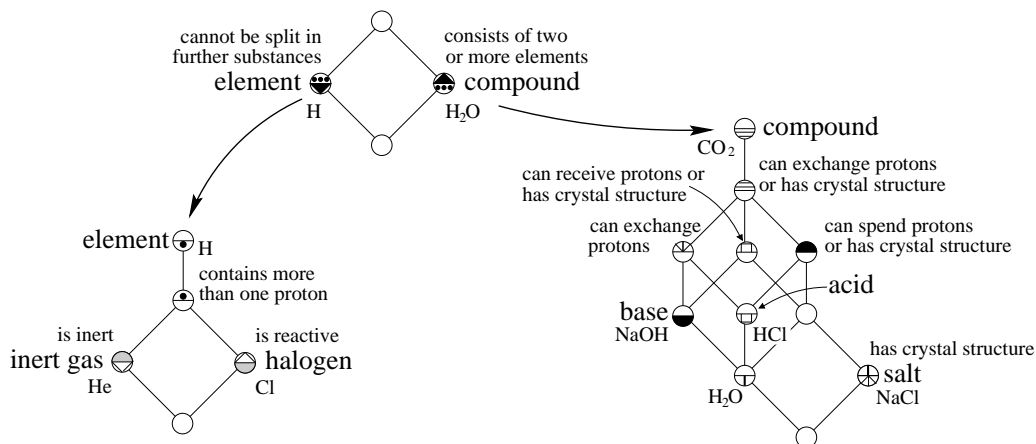


Figure 3: Results of three partial Distributive Concept Explorations of pure chemical substances

In both cases the use of attributes that are built with disjunctions can be observed. We conclude that if there is no canonical set of attributes, Distributive Concept Exploration may be preferred, because the interpretation of suprema is assured and the dialogue is less complex. If there is a canonical set of attributes, and it is assured that the basic concepts are concepts of the conceptual universe, then the general Concept Exploration may yield useful results.

5 Outlook

The exploration tools described in this paper have two major restrictions: They are not able to treat negations and — what is more important for the development of conceptual knowledge systems — they are limited to one-valued contexts. There are two exploration tools in development which bypass these restrictions, *Boolean Concept Exploration* and *Scale Exploration*.

Boolean Concept Exploration includes negated concepts and works similar as Distributive Concept Exploration. It also starts with basic concepts, but it determines the Boolean lattice that is freely generated by them with respect to the answers given by the user. The implementation of the algorithm should not be too difficult, because the contexts of Boolean concept lattices have a very clear structure. The complexity is probably very similar to the one of distributive concept exploration — theoretically the resulting lattices can grow super-exponentially, but in real applications (where the basic concepts are sufficiently related) there should be no problem.

Scale Exploration determines dependencies between the attributes of a many-valued context. *Scales* are used to interpret the data given by the

context. Usually there is one scale for every many-valued attribute. A scale is a one-valued context whose objects are the values of the many-valued attribute. Its attributes determine how the values are interpreted.

Scales can be combined to larger scales. The most natural combination is done by the semi-product². It produces the scale that respects all dependencies in the single scales, but no dependencies between them. The latter ones have then to be asked from an expert in an efficient way. This approach is analogous to Distributive Concept Exploration, where we also first calculate the lattice that respects all dependencies between the first n basic concepts, but no dependencies between them and the $(n+1)$ th one. The latter ones are again determined by the answers of the expert.

For the concept lattices of the scales the exploration basically corresponds to calculating their direct product on which then a (meet-semilattice) congruence relation is determined. In this view Attribute Exploration is just a special case of Scale Exploration.

Exploration tools in formal concept analysis are able to treat incomplete knowledge and they yield criteria for the completeness of the acquired knowledge. The fact that explorations of a larger domain request some effort from the experts indicates the complexity of the explored domain. Often however complete knowledge about *all* dependencies in the domain is not required. It would be sufficient to obtain complete knowledge only about parts of the domain. Therefore it is interesting to extend the exploration tools such that they support the splitting of the exploration in suitable parts, as we did for the chemical substance exploration in Fig. 3. The possibility of combining different tools (both from inside and outside of formal concept analysis) should also be supported.

References

- Burmeister, P. (1987):** Programm zur formalen Begriffsanalyse einwertiger Kontexte. Technische Hochschule Darmstadt (Latest version 1995 for Atari ST and MS DOS)
- Burmeister, P. (1991):** Merkmalsimplikationen bei unvollständigem Wissen. In: W. Lex (ed.): *Arbeitstagung Begriffsanalyse und Künstliche Intelligenz*. Informatik-Bericht 89/3, TU Clausthal, 15-46
- Duquenne, V. and Guigues, J.-L. (1986):** Familles minimales d' implications informatives resultant d' un tableau de données binaires. In: *Math. Sci. Humaines* **95**, 5–18

²Which “is a direct product in the category of all contexts (scales) with the scale measures as morphisms” (Ganter et al. (1986)).

- Ganter, B. (1987):** Algorithmen zur Begriffsanalyse. In: B. Ganter, R. Wille, K. E. Wolff (eds.): *Beiträge zur Begriffsanalyse*. B. I.-Wissenschaftsverlag, Mannheim, Wien, Zürich, 241–254
- Ganter, B. and Stahl, J. and Wille, R. (1986):** Conceptual measurement and many-valued contexts. In: W. Gaul and M. Schader (eds.): *Classification as a tool for research*. North-Holland, Amsterdam, 169–176
- Ganter, B. and Wille, R. (1995):** Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Berlin, Heidelberg (to appear)
- Ganter, B. and Zickwolff, M. (1994):** A tool to acquire conceptual knowledge. Forschungsbericht, Universität Karlsruhe (to appear in the proceedings of the ECML workshop 1993)
- Klotz, U. and Mann, A. (1988):** Begriffexploration. Diplomarbeit, TH Darmstadt
- Luxenburger, M. (1993):** Implikationen, Abhängigkeiten und Galois Abbildungen. Beiträge zur Formalen Begriffsanalyse. PhD thesis, TH Darmstadt, Shaker Verlag, Aachen
- Mortimer, Ch. E. (1983):** Chemie. Das Basiswissen der Chemie in Schwerpunkten. 4th Ed., Georg Thieme Verlag, Stuttgart, New York
- Reeg, S. and Weiß, W. (1990):** Properties of finite lattices. Diplomarbeit, TH Darmstadt
- Stumme, G. (1994):** Boolesche Begriffe. Diplomarbeit, TH Darmstadt
- Stumme, G. (1995a):** Knowledge acquisition by distributive concept exploration. In: Proceedings Supplement of the Third International Conference on Conceptual Structures: Applications, Implementation and Theory. Santa Cruz, CA
- Stumme, G. (1995b):** Attribute exploration with background implications and exceptions. In: H. H. Bock, W. Polasek (eds.): *Data analysis and information systems. Statistical and conceptual approaches*. Studies in Classification, Data Analysis, and Knowledge Organization **7**, Springer, Heidelberg
- Vogt, F. and Wille, R. (1995):** TOSCANA — A graphical tool for analyzing and exploring data. In: R. Tamassia, I. G. Tollis (eds.): *Graph Drawing '94*, Lecture Notes in Computer Sciences **894**, Springer, 226–233
- Wagner, H. (1973):** Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe*. Kösel Verlag, München, 191–209
- Wille, R. (1980):** Lattice Theory as Algebra of Concepts. Colloquium Lecture, University of Calgary, 13.3.1980

- Wille, R. (1982):** Restructuring lattice theory: An approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets*. Reidel, Dordrecht–Boston, 445–470
- Wille, R. (1987):** Bedeutungen von Begriffsverbänden. In: B. Ganter, R. Wille, K. E. Wolff (eds.): *Beiträge zur Begriffsanalyse*. B.I.–Wissenschaftsverlag, Mannheim, 161–211
- Wille, R. (1989a):** Knowledge acquisition by methods of formal concept analysis. In: E. Diday (ed.): *Data analysis, learning symbolic and numeric knowledge*. Nova Science Publisher, New York, Budapest, 365–380
- Wille, R. (1989b):** Lattices in data analysis: how to draw them with a computer In: I. Rival (ed.): *Algorithms and order*. Kluwer, Dordrecht–Boston, 33–58
- Wille, R. (1992):** Concept lattices and conceptual knowledge systems. In: *Computers & mathematics with applications* **23**, 493–515
- Zickwolff, M. (1991):** Rule exploration: First order logic in formal concept analysis. PhD thesis, TH Darmstadt