Knowledge Acquisition by Distributive Concept Exploration

Gerd Stumme

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

1 Introduction

A variety of quite different approaches to the concept *"concept"* can be found in philosophy and psychology. Formal Concept Analysis (cf. [12], [3]) is a mathematical approach which reflects the philosophical understanding that a concept can be understood as a unit of thoughts consisting of two parts: the extent containing all objects which belong to the concept and the intent containing the attributes which are shared by all those objects (cf. [11]). In Formal Concept Analysis this is modeled by *formal concepts* that are derived from a *formal context.* The subconcept-superconcept-relation is modeled naturally yielding a complete lattice (called *concept lattice*), in which greatest common subconcepts and lowest common superconcepts can be calculated. This approach has become a successful tool in data analysis (cf. [7], [14]).

Knowledge acquisition tools of Formal Concept Analysis can be used for exploring type hierarchies for conceptual graphs. They are interactive procedures which acquire knowledge from an expert in an exploration dialogue. These tools determine the concept lattice that is generated by some formal concepts according to the answers given by the expert.

For exploring type hierarchies for conceptual graphs with tools of Formal Concept Analysis we identify types with formal concepts, even if there are some differences: Formal Concept Analysis starts with the basic notion of a formal context (sometimes called *conceptual universe*) which models the underlying domain. If the domain changes, the conceptual universe which models it has to change, too. The restriction to one universe establishes the link between extension and intension; it assures a one-to-one-relation between them. Conceptual graphs on the other side are open for other "possible worlds". This provides more flexibility, but it also weakens the link between extension and intension (cf. [9, Theor. 3.2.6]). If one accepts the restriction to one fixed universe then techniques developed in Formal Concept Analysis provide also useful tools for conceptual graphs.

Attribute Exploration (cf. [2]) determines the meet-semilattice which is generated by the starting attribute concepts. The expert has to answer whether suggested implications between attributes are valid for all objects or not. When he denies an implication he has to give an object as counterexample. These counterexamples then span the resulting lattice. Attribute Exploration corresponds to Aristotelian type hierarchies, where subtypes can be derived by a join (followed by a type contraction) (cf. [9, Theor. 3.6.11]). This approach does not involve the join-operation, i. e., common supertypes are not considered.

Concept Exploration treats joins and meets equally. It determines the lattice that is generated by the starting concepts (which are also called the *basic concepts*). The basic idea of Concept Exploration is already mentioned in [12]. In [14] and [15] examples are given. U. Klotz and A. Mann worked out the method and implemented it as an interactive procedure ([6]).

Distributive Concept Exploration is a knowledge acquisition tool in Formal Concept Analysis similar to the more general Concept Exploration. The remaining part of this paper is devoted to the discussion of this tool. The main difference between Concept Exploration and Distributive Concept Exploration lies in the treatment of joins. While the meet corresponds in Formal Concept Analysis to the intersection of the concept extents, the extent of the join may contain additional objects. Depending on "how close one looks", the join may be more or less general. The join of CAT and DOG may for example be CARNIVORE ([9, pg. 81]), but it could also be FOUR-LEG-CARNIVORE or something even more specific. However the smallest concept that can be obtained in this way is the one that has as extent exactly the union of the extents. E. g., the smallest concept that can be understood as CAT $\vee$ DOG has exactly all cats and all dogs in its extent.

In Formal Concept Analysis the join in a concept lattice is exactly this "canonical join" if the set of attributes is closed under disjunction. Concept (resp. type) lattices with this join are distributive. This has (beside an easier understanding of the join operation) the advantage of a richer mathematical theory. Distributive lattices are more regularly structured than arbitrary ones. The knowledge of this structure reduces the complexity of the exploration dialogue.

Distributive Concept Exploration allows the user to examine interactively a list of concepts (called the *basic concepts*) which are only given by their names. The result of the exploration is a concept lattice that reflects the hierarchical subconcept–superconcept–relation between these concepts and a list of objects and attributes which separate the concepts. The algorithm of Distributive Concept Exploration follows the free generating process of distributive lattices by using the tensor product of lattices. During the exploration the user is asked questions in order to determine congruence relations describing the subconcept–superconcept–relation. These questions are of the form "Is s a subconcept of t?", here s and t are lattice terms built with the basic concepts. If the user replies "No", he must justify his answer by an object and an attribute which separate these concepts.

In the next section the basic notions of Formal Concept Analysis are introduced. The algorithm of Distributive Concept Exploration will be presented in Sect. 3 and explained by an example in Sect. 4.

2 Formal Concept Analysis

Both congruence relations on lattices and tensor products of lattices can adequately be described in terms of Formal Concept Analysis. We briefly recall the basic definitions of Formal Concept Analysis:

A *(formal) context* is a triple $\mathbb{K} := (G, M, I)$ where G and M are sets and I is a relation between G and M. The elements of G and M are called *objects* and *attributes*, respectively, and gIm is read *"the object g has the attribute m"*. For $A \subseteq G$ and $B \subseteq M$ we define $A' := \{m{\in}M \mid \forall g{\in}A : gIm\}$ and dually $B' := \{g{\in}G \mid \forall m{\in}B : gIm\}$. Now a *(formal) concept* is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The set A is called the *extent* and the set B the *intent* of the concept.

The hierarchical subconcept–superconcept–relation of concepts is formalized by $(A, B) \leq (C, D) :\iff A \subseteq C$ ($\iff B \supseteq D$). The set of all concepts of the context $\mathbb{K}$ together with this order relation is a complete lattice that is called the *concept lattice* of $\mathbb{K}$ and is denoted by $\underline{\mathfrak{B}}(\mathbb{K})$. In the concept lattice, suprema and infima are calculated as follows:

$$\bigwedge_{t \in T} (A_t, B_t) = (\bigcap_{t \in T} A_t, (\bigcup_{t \in T} B_t)'')$$

$$\bigvee_{t \in T} (A_t, B_t) = ((\bigcup_{t \in T} A_t)'', \bigcap_{t \in T} B_t)$$

Every complete lattice can be viewed as a concept lattice: The Basic Theorem of Formal Concept Analysis (cf. [12]) shows that a complete lattice L is isomorphic to the concept lattice $\underline{\mathfrak{B}}(L, L, \leq)$.

Example. In Fig. 1 a formal context of the National State Park Areas in California is given. Its objects are the National State Park Areas that are situated in California. The attributes are activities, and the relation I indicates if an activity is possible in a park ([1]).

The corresponding concept lattice is shown in Fig. 2. In the line diagram we label for every object $g \in G$ its *object concept* $\gamma g := (\{g\}'', \{g\}')$ with the name of the object and for every attribute $m \in M$ its *attribute concept* $\mu m := (\{m\}', \{m\}'')$ with the name of the attribute. This labeling allows to determine for every concept its extent and its intent: The extent (resp. intent) of a concept contains all objects (resp. attributes) whose object concepts (resp. attribute concepts) are linked to the concept with a descending (resp. ascending) path of straight line segments.

If we are e. g. interested in riding bike and swimming then we take the infimum of μ(*Bicycle Trail*) and μ(*Swimming*); this concept is represented by the circle labeled by "Death Valley". The extent of this concept contains Death Valley Natl. Mon., Golden Gate Natl. Recreation Area, Point Reyes Natl. Seashore, and Yosemite Natl. Park — so these are exactly the National Park Areas in California in which *Bicycle Trails*

	NPS Guided Tours	*Hiking*	*Horseback Riding*	*Swimming*	*Boating*	*Fishing*	*Bicycle Trail*	*Cross Country Trail*
Cabrillo Natl. Mon.						×	×	
Channel Islands Natl. Park		×		×		×		
Death Valley Natl. Mon.	×	×	×	×			×	
Devils Postpile Natl. Mon.	×	×	×	×		×		
Fort Point Natl. Historic Site	×					×		
Golden Gate Natl. Recreation Area	×	×	×	×		×	×	
John Muir Natl. Historic Site	×							
Joshua Tree Natl. Mon.	×	×	×					
Kings Canyon Natl. Park	×	×	×			×		×
Lassen Volcanic Natl. Park	×	×	×	×	×	×		×
Lava Beds Natl. Mon.	×	×						
Muir Woods Natl. Mon.		×						
Pinnacles Natl. Mon.		×						
Point Reyes Natl. Seashore	×	×	×	×		×	×	
Redwood Natl. Park	×	×	×	×		×		
Santa Monica Mts. Natl. Recr. Area	×	×	×	×	×	×		
Sequoia Natl. Park	×	×	×			×		×
Whiskeytown-Shasta-Trinity Natl. Recr. Area	×	×	×	×	×	×		
Yosemite Natl. Park	×	×	×	×	×	×	×	×

Fig. 1. A formal context of the National Park Areas in California

exist and *Swimming* is possible. The intent of this concept contains — beside *Bicycle Trail* and *Swimming* — the activities *Horseback Riding, Hiking* and *NPS Guided Tours*, so these activities are possible as well in the mentioned parks.

A context is called *reduced* if every object concept is $\bigvee$-irreducible and every attribute concept is $\bigwedge$-irreducible. It is called *distributive* if its concept lattice is distributive. For a $\bigvee$–irreducible (resp. $\bigwedge$–irreducible) element x of a finite lattice we write x_* (resp. x^*) for its unique lower (resp. upper) cover.

The *tensor product* of two complete lattices L_1 and L_2 is defined to be the concept lattice $L_1 \otimes L_2 := \underline{\mathfrak{B}}(L_1 \times L_2, L_1 \times L_2, \nabla)$ with $(x_1, x_2)\nabla(y_1, y_2) :\iff x_1 \leq y_1$ or $x_2 \leq y_2$. R. Wille showed in [13] that the tensor product is the free product in the category of completely distributive complete lattices with complete homomorphisms. There exist natural complete embeddings of L_1 and L_2 in $L_1 \otimes L_2$:

$$\varepsilon_1 \colon L_1 \to L_1 \otimes L_2, \quad x_1 \mapsto ([0, x_1] \times L_2 \cup L_1 \times \{0\}, [x_1, 1] \times L_2 \cup L_1 \times \{1\})$$
$$\varepsilon_2 \colon L_2 \to L_1 \otimes L_2, \quad x_2 \mapsto (L_1 \times [0, x_2] \cup \{0\} \times L_2, L_1 \times [x_2, 1] \cup \{1\} \times L_2)$$

For two contexts (G_1, M_1, I_1) and (G_2, M_2, I_2) and objects $g \in G_1$ and $h \in G_2$ the equality $\varepsilon_1(\gamma_1 g) \wedge \varepsilon_2(\gamma_2 h) = \gamma_\otimes(g, h)$ holds in $\underline{\mathfrak{B}}(G_1, M_1, I_1) \otimes \underline{\mathfrak{B}}(G_2, M_2, I_2)$. Dually, we have $\varepsilon_1(\mu_1 m) \vee \varepsilon_2(\mu_2 n) = \mu_\otimes(m, n)$ for $m \in M_1$ and $n \in M_2$.

We define the *direct product* of two contexts $\mathbb{K}_1 := (G_1, M_1, I_1)$ and $\mathbb{K}_2 := (G_2, M_2, I_2)$ to be the context $\mathbb{K}_1 \times \mathbb{K}_2 := (G_1 \times G_2, M_1 \times M_2, \nabla)$ with the incidence $(g_1, g_2)\nabla(m_1, m_2) :\iff (g_1, m_1) \in I_1$ or $(g_2, m_2) \in I_2$. The tensor product of two concept lattices is (up to isomorphism) just the concept lattice of the direct

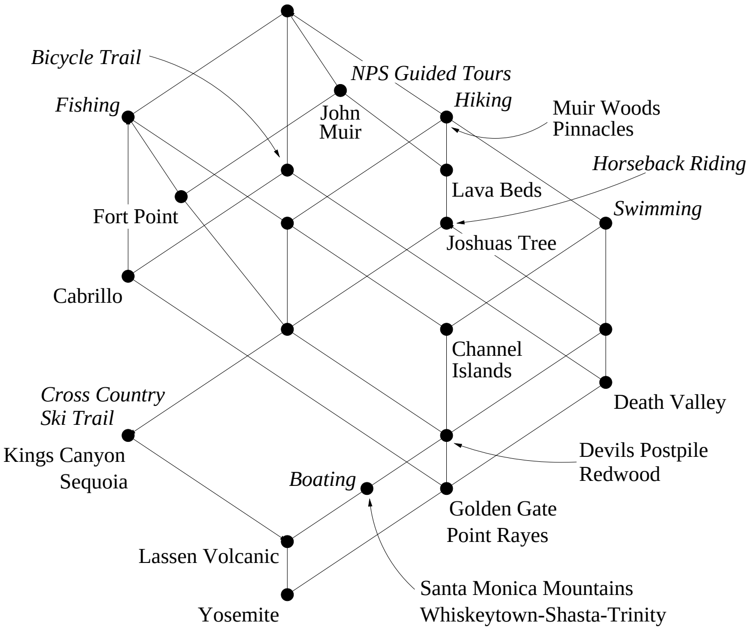


Fig. 2. Concept lattice of the formal context in Fig. 1

product of their contexts: We have $\underline{\mathfrak{B}}(\mathbb{K}_1) \otimes \underline{\mathfrak{B}}(\mathbb{K}_2) \cong \underline{\mathfrak{B}}(\mathbb{K}_1 \times \mathbb{K}_2)$. If $\mathbb{K}_1$ and $\mathbb{K}_2$ are reduced, then $\mathbb{K}_1 \times \mathbb{K}_2$ is also reduced (cf. [13]).

Congruence relations of complete lattices appear in a quite natural way in Formal Concept Analysis. For finite concept lattices they can always be described by *compatible subcontexts*: A context (H, N, J) is called a *subcontext* of a context (G, M, I) if $H \subseteq G$, $N \subseteq M$ and $J = I \cap (H \times N)$. It is called *compatible* if for every concept (A, B) of (G, M, I) the pair $(A \cap H, B \cap N)$ is also a concept of the subcontext.

The subcontext (H, N, J) of (G, M, I) is compatible if and only if the mapping $\Pi_{H,N}\colon (A, B) \mapsto (A \cap H, B \cap N)$ is a surjective complete homomorphism from $\underline{\mathfrak{B}}(G, M, I)$ to $\underline{\mathfrak{B}}(H, N, J)$. If G and M are finite, then for every complete congruence relation on $\underline{\mathfrak{B}}(G, M, I)$ exists a compatible subcontext, such that the congruence relation is just the kernel of this homomorphism. If (G, M, I) is reduced, this subcontext is unique. Every subcontext of a reduced context is also reduced (cf. [3]).

So factorizing a concept lattice is equivalent to deleting suitable rows and columns in the context. The rows and columns that have to be deleted can be described by using the two relations $\swarrow$ and $\nearrow$: For $g \in G$ and $m \in M$ we define $g \swarrow m$ if $g \not I m$ and if $g' \subset h'$ implies hIm for all $h \in G$. We define $g \nearrow m$ if $g \not I m$ and if $m' \subset n'$ implies gIn for all $n \in M$. We write $g \mathrel{\swarrow\!\!\!\!\nearrow} m$ if $g \swarrow m$ and $g \nearrow m$.

In a distributive reduced finite context[1] the $\mathrel{\swarrow\!\!\!\!\nearrow}$-relation is a bijection between the set of objects and the set of attributes, because the following implications are valid:

$$g \nearrow m \Rightarrow g \mathrel{\swarrow\!\!\!\!\nearrow} m, \qquad g \mathrel{\swarrow\!\!\!\!\nearrow} m,\ g \mathrel{\swarrow\!\!\!\!\nearrow} n \Rightarrow m = n,$$
$$g \swarrow m \Rightarrow g \mathrel{\swarrow\!\!\!\!\nearrow} m, \qquad g \mathrel{\swarrow\!\!\!\!\nearrow} m,\ h \mathrel{\swarrow\!\!\!\!\nearrow} m \Rightarrow g = h\ \ .$$

In [3] is shown, that a subcontext (H, N, J) of a context (G, M, I) is compatible if and only if $h \nearrow m$, $h \in H$ implies $m \in N$ and $g \swarrow n$, $n \in N$ implies $g \in H$. Hence, in a distributive reduced finite context the compatible subcontexts are exactly those of the form $(H, N, I \cap (H \times N))$ with $N = \{m \in M \mid \exists g \in H\colon g \mathrel{\swarrow\!\!\!\!\nearrow} m\}$. The following theorem describes the correspondence between compatible subcontexts and congruence relations.

[1] All the contexts needed for Distributive Concept Exploration are of this type

Theorem 1. *Let (G,M,I) be a distributive reduced finite context, $g \in G$ and $m \in M$ with $g \swarrow\!\!\!\!\nearrow m$. Then* $\ker \Pi_{G\setminus\{g\},M\setminus\{m\}}$ *is the congruence relation on $\underline{\mathfrak{B}}(G,M,I)$ that is generated by forcing $\gamma g \leq \mu m$.*

Proof. The subcontext $(G\setminus\{g\}, M\setminus\{m\}, I\cap(G\setminus\{g\}\times M\setminus\{m\}))$ is compatible. The compatible subcontexts $(H,N,I\cap(H\times N))$ with $g\notin H$ and $m \notin N$ are exactly those with $(\gamma g, \gamma g \wedge \mu m) \in \ker \Pi_{H,N}$, because the equation $\Pi_{H,N}(\gamma g \wedge \mu m) = \Pi_{H,N}((\gamma g)_*) = (g'' \cap H, (g''\cap H)') = \Pi_{H,N}(\gamma g)$ holds if and only if $g \notin H$. Among these subcontexts, $(G\setminus\{g\}, M\setminus\{m\}, I\cap(G\setminus\{g\}\times M\setminus\{m\}))$ is the largest and induces therefore the smallest congruence relation containing $(\gamma g, \gamma g\wedge \mu m)$. □

3 Distributive Concept Exploration

In this section the algorithm of Distributive Concept Exploration is introduced; in the next it is explained by an example. For an easier understanding we suggest to switch between the two sections as appropriate.

Let $\mathfrak{b}_1, \mathfrak{b}_2, \ldots, \mathfrak{b}_n$ be the list of concepts the user wants to explore. All we assume is that the exploration takes place in a (somehow fixed) conceptual universe where the set of attributes is closed under disjunction. This yields the "canonical join" as described in Section 1. Distributive Concept Exploration determines a concept lattice reflecting the hierarchical relationship between the basic concepts together with a list of objects and attributes which are separating different concepts:

Definition. For two concepts $\mathfrak{a}$ and $\mathfrak{b}$ with $\mathfrak{a}$ not being a subconcept of $\mathfrak{b}$, a pair (g,m) is called a *separating pair* if g is an object of the concept $\mathfrak{a}$ and m is an attribute of the concept $\mathfrak{b}$ such that g does not have the attribute m. (The existence of such an object and such an attribute is a counterexample against the hypothesis that $\mathfrak{a}$ is a subconcept of $\mathfrak{b}$.)

The lattice that we want to determine can be seen as a quotient lattice $\mathrm{FBD}(\mathfrak{b}_1,\ldots,\mathfrak{b}_n)/\Theta$ of the free bounded distributive lattice generated by the basic concepts, where Θ is the congruence relation that reflects the answers given by the user. We use the fact that $\mathrm{FBD}(\mathfrak{b}_1,\ldots,\mathfrak{b}_n) \cong \mathrm{FBD}(\mathfrak{b}_1)\otimes\ldots\otimes\mathrm{FBD}(\mathfrak{b}_n)$ for splitting the determination of Θ into smaller parts: For $i = 0,\ldots,n$, the exploration algorithm subsequently determines the lattice L_i that is completely generated by the basic concepts $\mathfrak{b}_1,\ldots,\mathfrak{b}_i$ with respect to their hierarchical relationships. The lattice L_i is obtained from L_{i-1} by $L_i \cong (L_{i-1}\otimes\mathrm{FBD}(\mathfrak{b}_i))/\Theta_i$, where Θ_i reflects the hierarchical relationship between $\mathfrak{b}_i$ and the elements of L_{i-1}. The result of the exploration is then given by the lattice L_n.

For every $i \in \{0,\ldots,n\}$, the lattice L_i will be determined in two steps: First the tensor product $\widetilde{L}_i$ of L_{i-1} with $\mathrm{FBD}(\mathfrak{b}_i)$ (which is the three element chain $\bot < \mathfrak{b}_i < \top$) is calculated. Then the user is asked questions of the kind "Is s a subconcept of t?" with s and t being lattice terms built with $\mathfrak{b}_1,\ldots,\mathfrak{b}_i$. A congruence relation on $\widetilde{L}_i$ is deduced from the answers given by the user. The factorization of $\widetilde{L}_i$ by the deduced congruence relation yields the lattice L_i. The lattice $\widetilde{L}_i$ is only used as intermediate step, it is not needed any longer.

The algorithm starts with the determination of L_0 out of $\widetilde{L}_0$, which is the complete lattice freely generated by the empty set. $\widetilde{L}_0$ is the two element chain $\bot < \top$, and we understand its elements as representations of the concepts *nothing* and *everything (in our field of interest)*, respectively. Any serious user will agree that these concepts are distinct in a non-trivial field of interest. However (as we will see below) he has to be asked the question "Is *everything* a subconcept of *nothing*?" in order to get the first separating pair.

In the algorithm the lattice L_i will be represented[2] by a reduced context $\mathbb{K}_i := (G_i, M_i, I_i)$. As this context is the result of a repeated use of the tensor product, its objects and attributes are tuples. They are of the form $\boldsymbol{x} := (x_0,\ldots,x_i) \in G_i$ with $x_0 = \top$ and $x_k \in \{\top, \mathfrak{b}_k\}$ for $k = 1,\ldots,n$ and $\boldsymbol{y} := (y_0,\ldots,y_i) \in M_i$ with $y_0 = \bot$ and $y_k \in \{\bot, \mathfrak{b}_k\}$ for $k = 1,\ldots,n$. According to the remarks about the tensor product in Sect. 2, the incidence $\boldsymbol{x} I_i \boldsymbol{y}$ represents the inequality $\bigwedge \boldsymbol{x} \leq \bigvee \boldsymbol{y}$ with $\bigwedge \boldsymbol{x} := \bigwedge_{k=0}^{i} x_i$ and $\bigvee \boldsymbol{y} := \bigvee_{k=0}^{i} y_i$.

[2] We say that a complete lattice L is represented by a context $\mathbb{K}$ if L is isomorphic to $\underline{\mathfrak{B}}(\mathbb{K})$.

As mentioned above, the lattice $\widetilde{L}_i$ has to be calculated as intermediate step in the determination of the lattice L_i. This tensor product of L_{i-1} with the chain $\bot < \mathfrak{b}_i < \top$ will be represented by the context $\widetilde{\mathbb{K}}_i := (\widetilde{G}_i, \widetilde{M}_i, \widetilde{I}_i)$ being the direct product of $\mathbb{K}_{i-1}$ with the context $(\{\mathfrak{b}_i, \top\}, \{\bot, \mathfrak{b}_i\}, \{(\mathfrak{b}_i, \mathfrak{b}_i)\})$. The context $\mathbb{K}_i$ will then be derived from $\widetilde{\mathbb{K}}_i$ by deleting suitable rows and columns. This corresponds to finding a suitable congruence relation on the tensor product. Theorem 1 indicates the questions needed for determining these rows and columns: For all $\boldsymbol{x} \in \widetilde{G}_i$ and $\boldsymbol{y} \in \widetilde{M}_i$ with $\boldsymbol{x} \swarrow\!\!\!\nearrow \boldsymbol{y}$ the user is asked: "Is the infimum of $\boldsymbol{x}$ a subconcept of the supremum of $\boldsymbol{y}$?" As we have up to now no (concept) lattice in which supremum and infimum can be calculated, we have to explain how the infimum of $\boldsymbol{x}$ resp. the supremum of $\boldsymbol{y}$ shall be understood: According to the Basic Theorem of Formal Concept Analysis, an object belongs to the infimum of a set of concepts if and only if it belongs to each of them. As the set of attributes of the conceptual universe is assumed to be closed under disjunctions, an object belongs to the supremum of a set of concepts if and only if it belongs to at least one of them. This definition makes all lattices appearing in the exploration distributive. If the user agrees to the question, the object $\boldsymbol{x}$ and the attribute $\boldsymbol{y}$ will be deleted, otherwise they will be kept in G_i and in M_i, respectively.

Observe that the $\swarrow\!\!\!\nearrow$–relation is inherited and can thus easily be calculated: For $\boldsymbol{x} \swarrow\!\!\!\nearrow \boldsymbol{y}$ in $\mathbb{K}_{i-1}$ we have $(\boldsymbol{x}, \top) \swarrow\!\!\!\nearrow (\boldsymbol{y}, \mathfrak{b}_i)$ and $(\boldsymbol{x}, \mathfrak{b}_i) \swarrow\!\!\!\nearrow (\boldsymbol{y}, \bot)$ in $\widetilde{\mathbb{K}}_i$. Deleting corresponding rows and columns does not change the $\swarrow\!\!\!\nearrow$-relation.

The algorithm starts with the determination of the context $\mathbb{K}_0$ out of the context $\widetilde{\mathbb{K}}_0 := (\{\top\}, \{\bot\}, \emptyset)$. As we have $\top \swarrow\!\!\!\nearrow \bot$ in $\widetilde{\mathbb{K}}_0$, the first question in every exploration is "Is $\top$ (*everything*) a subconcept of $\bot$ (*nothing*)?" Usually, this will be denied (and a separating pair will be given). If however the user agrees, the exploration is terminated because he obtains $\mathbb{K}_0 = (\emptyset, \emptyset, \emptyset)$ which is the absorbing element for the direct product of contexts.[3]

Separating pairs are used for separating different concepts. This algorithm computes for every L_i with $i = 0, \ldots, n$ a minimal list of pairs of objects and attributes, such that for two concepts $\mathfrak{a}$ and $\mathfrak{b}$ of L_i with $\mathfrak{a} \not\leq \mathfrak{b}$ there is at least one pair in this list which is a separating pair for $\mathfrak{a}$ and $\mathfrak{b}$.

Definition. For two elements u and v of a lattice L, we write $u \swarrow\!\!\!\nearrow v$, if u is maximal in $L \setminus [v)$ and v is minimal in $L \setminus (u]$.

In a finite lattice this implies that u is $\bigvee$–irreducible and v is $\bigwedge$–irreducible and that $u \not\leq v$, $u_* \leq v$, and $u \leq v^*$ hold. It should not be confusing that we use $\swarrow\!\!\!\nearrow$ at the same time as a relation between elements of a lattice and between objects and attributes of a context because $g \swarrow\!\!\!\nearrow m$ in $\mathbb{K}$ is equivalent to $\gamma g \swarrow\!\!\!\nearrow \mu m$ in $\underline{\mathfrak{B}}(\mathbb{K})$.

It is sufficient to have a list of separating pairs for elements $\mathfrak{c}$ and $\mathfrak{d}$ of L_i with $\mathfrak{c} \swarrow\!\!\!\nearrow \mathfrak{d}$, as for two elements $\mathfrak{a}$ and $\mathfrak{b}$ of L_i with $\mathfrak{a} \not\leq \mathfrak{b}$ there always exist such $\mathfrak{c}$ and $\mathfrak{d}$ with $\mathfrak{a} \geq \mathfrak{c}$ and $\mathfrak{d} \geq \mathfrak{b}$, because L_i is finite and distributive. The separating pair for $\mathfrak{c}$ and $\mathfrak{d}$ is also a separating pair for $\mathfrak{a}$ and $\mathfrak{b}$. On the other hand there must be different separating pairs for different $\mathfrak{c} \swarrow\!\!\!\nearrow \mathfrak{d}$, so that in fact this list is minimal.

During the exploration, the user is asked for separating pairs: Whenever he denies the question "Is the infimum of $\boldsymbol{x}$ a subconcept of the supremum of $\boldsymbol{y}$?", he is prompted for a separating pair for $\bigwedge \boldsymbol{x}$ and $\bigvee \boldsymbol{y}$. The pair will be denoted by $(\mathbf{g}_i(\boldsymbol{x}), \mathbf{m}_i(\boldsymbol{y}))$. Thus we obtain two mappings: $\mathbf{g}_i$ maps from G_i to the set of objects of the conceptual universe, and $\mathbf{m}_i$ maps from M_i to its set of attributes. These mappings indicate that the object $\mathbf{g}_i(\boldsymbol{x})$ belongs to the concept $\bigwedge \boldsymbol{x}$, and that the attribute $\mathbf{m}_i(\boldsymbol{y})$ belongs to the concept $\bigvee \boldsymbol{y}$. Because of $\bigwedge \boldsymbol{x} \swarrow\!\!\!\nearrow \bigvee \boldsymbol{y}$ we know that $\mathbf{g}_i(\boldsymbol{x})$ and $\mathbf{m}_i(\boldsymbol{y})$ form a separating pair. The mappings $\mathbf{g}_i$ and $\mathbf{m}_i$ do not indicate, if an object or attribute does *not* belong to a concept. This information cannot be deduced from the answers given by the expert during the exploration dialogue. That is, because the expert is not asked how different separating pairs are related. Of course, this could be done, too, but it would lengthen the dialogue. At the end of the example in the next section is given an object which has an attribute, even though, in the line diagram, the object does not lay below the attribute.

[3] $\underline{\mathfrak{B}}(\emptyset, \emptyset, \emptyset)$ is the one element lattice which is the absorbing element for the tensor product of lattices.

Unfortunately, $\bigwedge \boldsymbol{x} \swarrow\!\!\!\nearrow \bigvee \boldsymbol{y}$ in L_i does not imply $\bigwedge \boldsymbol{x} \swarrow\!\!\!\nearrow \bigvee \boldsymbol{y}$ in L_{i+1}. This means that the separating pair $(\mathrm{g}_i(\boldsymbol{x}), \mathbf{m}_i(\boldsymbol{y}))$ will in general not remain in the minimal list for L_{i+1}: If neither $\mathrm{g}_i(\boldsymbol{x})$ nor $\mathbf{m}_i(\boldsymbol{y})$ belong to $\mathfrak{b}_{i+1}$, then there is no $\mathfrak{c} \swarrow\!\!\!\nearrow \mathfrak{d}$ in L_{i+1} separated by this pair. However it can be used to find new separating pairs for the minimal list: $\mathrm{g}_i(\boldsymbol{x})$ might appear in a separating pair for $\bigwedge(\boldsymbol{x}, \top)$ and $\bigvee(\boldsymbol{y}, \mathfrak{b}_{i+1})$ and $\mathbf{m}_i(\boldsymbol{y})$ might appear in a separating pair for $\bigwedge(\boldsymbol{x}, \mathfrak{b}_{i+1})$ and $\bigvee(\boldsymbol{y}, \bot)$. It is a separating pair for $\bigwedge(\boldsymbol{x}, \mathfrak{b}_{i+1}) \swarrow\!\!\!\nearrow \bigvee(\boldsymbol{y}, \bot)$ in L_{i+1} and remains therefore in the minimal list if the object $\mathrm{g}_i(\boldsymbol{x})$ belongs to the concept $\mathfrak{b}_{i+1}$ and the attribute $\mathbf{m}_i(\boldsymbol{y})$ does not. It is a separating pair for $\bigwedge(\boldsymbol{x}, \top) \swarrow\!\!\!\nearrow \bigvee(\boldsymbol{y}, \mathfrak{b}_{i+1})$ in L_{i+1} and remains in the list if the object $\mathrm{g}_i(\boldsymbol{x})$ does not belong to the concept $\mathfrak{b}_{i+1}$ and the attribute $\mathbf{m}_i(\boldsymbol{y})$ does. Because the object $\mathrm{g}_i(\boldsymbol{x})$ does not have the attribute $\mathbf{m}_i(\boldsymbol{y})$, it is not possible that both belong to the concept $\mathfrak{b}_{i+1}$. This justifies the following definition:

$$
\begin{aligned}
\tilde{\mathrm{g}}_{i+1}(\boldsymbol{x}, \mathfrak{b}_{i+1}) &:= \begin{cases} \mathrm{g}_i(\boldsymbol{x}) & \text{if } \mathrm{g}_i(\boldsymbol{x}) \text{ belongs to } \mathfrak{b}_{i+1} \\ \text{undefined} & \text{else} \end{cases} \\
\tilde{\mathrm{g}}_{i+1}(\boldsymbol{x}, \top) &:= \begin{cases} \mathrm{g}_i(\boldsymbol{x}) & \text{if } \mathrm{g}_i(\boldsymbol{x}) \text{ does not belong to } \mathfrak{b}_{i+1} \\ \text{undefined} & \text{else} \end{cases} \\
\tilde{\mathbf{m}}_{i+1}(\boldsymbol{y}, \mathfrak{b}_{i+1}) &:= \begin{cases} \mathbf{m}_i(\boldsymbol{y}) & \text{if } \mathbf{m}_i(\boldsymbol{y}) \text{ belongs to } \mathfrak{b}_{i+1} \\ \text{undefined} & \text{else} \end{cases} \\
\tilde{\mathbf{m}}_{i+1}(\boldsymbol{y}, \bot) &:= \begin{cases} \mathbf{m}_i(\boldsymbol{y}) & \text{if } \mathbf{m}_i(\boldsymbol{y}) \text{ does not belong to } \mathfrak{b}_{i+1} \\ \text{undefined} & \text{else} \end{cases}
\end{aligned}
$$

Thus, for every separating pair $(\mathrm{g}_i(\boldsymbol{x}), \mathbf{m}_i(\boldsymbol{y}))$ in L_i, the user has to answer the two following questions: "Does the object $\mathrm{g}_i(\boldsymbol{x})$ belong to the concept $\mathfrak{b}_{i+1}$?" and "Does the attribute $\mathbf{m}_i(\boldsymbol{y})$ belong the concept $\mathfrak{b}_{i+1}$?". The algorithm uses the fact that the answer "Yes" to one of the questions implies the answer "No" to the other one.

The problem of finding the rows and columns in $\widetilde{\mathbb{K}}_i$ that have to be deleted, now turns out to be equivalent to completing the partial mappings $\tilde{\mathrm{g}}_i$ and $\tilde{\mathbf{m}}_i$: If, for $\boldsymbol{x} \in \widetilde{G}_i$ and $\boldsymbol{y} \in \widetilde{M}_i$ with $\boldsymbol{x} \swarrow\!\!\!\nearrow \boldsymbol{y}$, at least one of $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ and $\tilde{\mathbf{m}}_i(\boldsymbol{y})$ is undefined and the user is not able to find an object or attribute for completing the separating pair, then the row $\boldsymbol{x}$ and the column $\boldsymbol{y}$ have to be deleted. In two cases we can benefit from the already given knowledge:

1. If $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ is undefined, $\tilde{\mathbf{m}}_i(\boldsymbol{y})$ is defined and $\boldsymbol{x} = (\top, \ldots, \top, \mathfrak{b}_i)$, then we already know that there must exist an object that belongs to $\mathfrak{b}_i$ and that does not have the attribute $\tilde{\mathbf{m}}_i(\boldsymbol{y})$. The user is then asked for such an object.
2. If $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ is defined and $\tilde{\mathbf{m}}_i(\boldsymbol{y})$ is undefined then there must exist an attribute of $\bigvee \boldsymbol{y}$ that $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ does not have. The user is then asked for such an attribute.

We are now ready to list the algorithm of Distributive Concept Exploration:

Algorithm: Given is the list $\mathfrak{b}_1, \mathfrak{b}_2, \ldots, \mathfrak{b}_n$ of basic concepts.

```
1. $i := 0$, $\widetilde{\mathbb{K}}_0 := (\{\top\}, \{\bot\}, \emptyset)$, $\tilde{\mathrm{g}}_0(\top) :=$ undefined, $\tilde{\mathbf{m}}_0(\bot) :=$ undefined.
2. For every $(\boldsymbol{x}, \boldsymbol{y}) \in \widetilde{G}_i \times \widetilde{M}_i$ with $\boldsymbol{x} \swarrow\!\!\!\nearrow \boldsymbol{y}$,
   where $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ or $\tilde{\mathbf{m}}_i(\boldsymbol{y})$ are undefined, do:
     • If $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ is undefined:
        • If $\tilde{\mathbf{m}}_i(\boldsymbol{y})$ is defined and $\boldsymbol{x} = (\top, \ldots, \top, \mathfrak{b}_i)$:
           Prompt: "Name an object belonging to $\mathfrak{b}_i$ and not having
           the attribute $\tilde{\mathbf{m}}_i(\boldsymbol{y})$!" Set $\tilde{\mathrm{g}}_i(\boldsymbol{x})$ according to the answer.
        • Else do:
           Ask the user: "Is the infimum of $\boldsymbol{x}$ a subconcept
                          of the supremum of $\boldsymbol{y}$?"
           "Yes": Delete $\boldsymbol{x}$ in $\widetilde{G}_i$, $\boldsymbol{y}$ in $\widetilde{M}_i$,
                  and the corresponding row and column in $\widetilde{I}_i$.
           "No":  Prompt: "Give a separating pair for $\bigwedge \boldsymbol{x}$ and $\bigvee \boldsymbol{y}$!"
```

```
                    If $\widetilde{\mathbf{m}}_i(\boldsymbol{y})$ is defined, add:
                       “Eventually you can use $\widetilde{\mathbf{m}}_i(\boldsymbol{y})$ as attribute.”
                    Set $\widetilde{\mathbf{g}}_i(\boldsymbol{x})$ and $\widetilde{\mathbf{m}}_i(\boldsymbol{y})$ according to the answer.
        • Else (i.e. $\widetilde{\mathbf{g}}_i(\boldsymbol{x})$ is defined and $\widetilde{\mathbf{m}}_i(\boldsymbol{y})$ is undefined) do:
             Prompt: “Name an attribute of $\bigvee \boldsymbol{y}$ that $\widetilde{\mathbf{g}}_i(\boldsymbol{x})$ does not have!”
             Set $\widetilde{\mathbf{m}}_i(\boldsymbol{y})$ according to the answer.
 3. Set $\mathbb{K}_i := \widetilde{\mathbb{K}}_i$, $\mathbf{g}_i := \widetilde{\mathbf{g}}_{i\,|G_i}$, $\mathbf{m}_i := \widetilde{\mathbf{m}}_{i\,|M_i}$.
 4. If $i{=}n$, then STOP.
 5. Set $\mathbb{K}_{i+1} := \mathbb{K}_i \times (\{\mathfrak{b}_{i+1}, \top\}, \{\bot, \mathfrak{b}_{i+1}\}, \{(\mathfrak{b}_{i+1}, \mathfrak{b}_{i+1})\})$.
 6. For every $(\boldsymbol{x}, \boldsymbol{y}) \in G_i \times M_i$ with $\boldsymbol{x} \mathrel{\swarrow\!\!\!\!\nearrow} \boldsymbol{y}$:
      – Ask the user: “Does the object $\mathbf{g}_i(\boldsymbol{x})$ belong to the concept $\mathfrak{b}_{i+1}$?”
      – If “No”, ask “Does the attribute $\mathbf{m}_i(\boldsymbol{y})$ belong to the concept $\mathfrak{b}_{i+1}$?”
      – Set $\widetilde{\mathbf{g}}_{i+1}(\boldsymbol{x}, \mathfrak{b}_{i+1})$ and $\widetilde{\mathbf{g}}_{i+1}(\boldsymbol{x}, \top)$ as defined above.
      – Set $\widetilde{\mathbf{m}}_{i+1}(\boldsymbol{y}, \mathfrak{b}_{i+1})$ and $\widetilde{\mathbf{m}}_{i+1}(\boldsymbol{y}, \bot)$ as defined above.
 7. Set $i := i + 1$.
 8. Goto step 2.
```

The result of the algorithm can be shown by a line diagram of $\underline{\mathfrak{B}}(\mathbb{K}_n)$. It is not necessary to label all the object and attribute concepts in the diagram. Only the concepts $\bigvee\{\gamma\boldsymbol{x} \mid \boldsymbol{x}{\in}G_n,\, x_i{=}\mathfrak{b}_i\}$ $(= \bigwedge\{\mu\boldsymbol{y} \mid \boldsymbol{y}{\in}M_n,\, y_i{=}\mathfrak{b}_i\})$ of $\underline{\mathfrak{B}}(\mathbb{K}_n)$ have to be labeled by $\mathfrak{b}_i$, as they correspond to the basic concepts which completely generate the whole lattice. The resulting list of separating pairs can be displayed in the same diagram: For every pair $\boldsymbol{x} \mathrel{\swarrow\!\!\!\!\nearrow} \boldsymbol{y}$ in $\mathbb{K}_n$, there is exactly one separating pair $(\mathbf{g}_n(\boldsymbol{x}), \mathbf{m}_n(\boldsymbol{y}))$. We label the concept $\gamma\boldsymbol{x}$ by $\mathbf{g}_n(\boldsymbol{x})$ and the concept $\mu\boldsymbol{y}$ by $\mathbf{m}_n(\boldsymbol{y})$ and mark $\gamma\boldsymbol{x}$ and $\mu\boldsymbol{y}$ with the same symbol. An example can be seen in the next section.

Here are some remarks about the complexity of the algorithm. If the user denies all dependencies between the basic concepts, then the resulting lattice is isomorphic to the bounded distributive lattice FBD$(\mathfrak{b}_1, \ldots, \mathfrak{b}_n)$ which is freely generated by the basic concepts. Its cardinality is exactly known only for $n \leq 8$, but the fact that $2^n \leq |\mathrm{FBD}(\mathfrak{b}_1, \ldots, \mathfrak{b}_n)| \leq 2^{2^n}$ shows that it grows very fast. So “in the worst case” there is no chance to do the exploration in a reasonable time. If however the basic concepts are sufficiently related, then the presented algorithm is effective, because it works only on the “logarithmic” level of the formal contexts. For basic concepts that are only weakly related, the whole lattice generated by them is usually not requested. We suggest to divide them in stronger related classes and to explore these classes separately.

Up to now, the user is assumed to reply to every question during the exploration either with “Yes” or “No”. The algorithm as described above is not able to treat incomplete knowledge. With a little change we can allow the answer “I don’t know” to the question “Is the infimum of $\boldsymbol{x}$ a subconcept of the supremum of $\boldsymbol{y}$?”: In this case the row $\boldsymbol{x}$ and the column $\boldsymbol{y}$ will not be deleted in $\widetilde{\mathbb{K}}_i$ and $\widetilde{\mathbf{g}}_i(\boldsymbol{x})$ and $\widetilde{\mathbf{m}}_i(\boldsymbol{y})$ will be set to the default value *?*. In step 6 of the algorithm all $\widetilde{\mathbf{g}}_{i+1}(\boldsymbol{x}, \mathfrak{b}_{i+1})$, $\widetilde{\mathbf{g}}_{i+1}(\boldsymbol{x}, \top)$, $\widetilde{\mathbf{m}}_{i+1}(\boldsymbol{y}, \mathfrak{b}_{i+1})$ and $\widetilde{\mathbf{m}}_{i+1}(\boldsymbol{y}, \bot)$ will then automatically be set to *?*. These *?* play the role of “possible separating pairs”. During and after the exploration procedure the user can either replace them by a real separating pair or he can delete the corresponding row and column (if he is then sure that the inequality $\bigwedge \boldsymbol{x} \leq \bigvee \boldsymbol{y}$ holds). The result of the exploration can be shown by a list of line diagrams — one for every possibility of deleting corresponding rows and columns that are not confirmed by a real separating pair.

4 An Exploration of Zinks

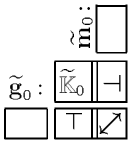
As an example, we want to explore a family of musical instruments: Zinks are wind instruments with a conical wide–bored tube, a shortening hole–system and a mouth piece played like a trumpet. The exploration starts with the following basic concepts:

$\mathfrak{b}_1 =$ *straight zink* [*gerader Zink*]	$\mathfrak{b}_4 =$ *cornettino*
$\mathfrak{b}_2 =$ *silent zink* [*stiller Zink*]	$\mathfrak{b}_5 =$ *cornetto*
$\mathfrak{b}_3 =$ *curved zink* [*krummer Zink*]	

The result is a concept lattice that can be used as a type lattice for conceptual graphs about musical instruments. The knowledge about the objects and attributes serving as separating pairs is an additional information that can be used if specific instances of the types are needed for some conceptual graph.

The exploration is based on information given by the catalogue of the museum of musical instruments of the University of Leipzig ([5], cf. also [15]). Specifically the zinks used for separating pairs will be taken out of this catalogue. They are named by their catalogue number. Of course, we could also use any other existing zink for a separating pair. The contexts $\widetilde{\mathbb{K}}_i$ and $\mathbb{K}_i$ are displayed below the dialogue in which they are determined. Usually they are not shown to the user. The lattices $\widetilde{L}_i$ and L_i are shown at the left of the contexts $\widetilde{\mathbb{K}}_i$ and $\mathbb{K}_i$, respectively. For $\mathfrak{c} \swarrow\!\!\!\nearrow \mathfrak{d}$ the concepts are marked with the same symbol (e. g. $\mathfrak{c}$ with ☮ and $\mathfrak{d}$ with ⊛). The dialogue is abbreviated in the way that a "No" to the question "Is s a subconcept of t?" is justified by the user with a separating pair without waiting to be prompted to do so.

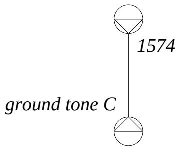
We start the algorithm with the context $\widetilde{\mathbb{K}}_0$:

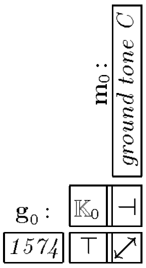


$\widetilde{\mathfrak{m}}_0$:
$\widetilde{\mathfrak{g}}_0$:
$\widetilde{\mathbb{K}}_0$
⊣
⊤
$\swarrow\!\!\!\nearrow$

"Is *everything* a subconcept of *nothing*?"
"No! A dividing pair is *Zink 1574* and *ground tone C*."

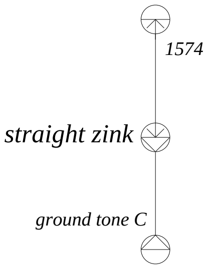
This answer yields the context $\mathbb{K}_0$ which describes the lattice that is generated by no basic concepts:





"Is *Zink 1574* a *straight zink*?"
"No!"
"Has every *straight zink* the attribute *ground tone C*?"
"No!"

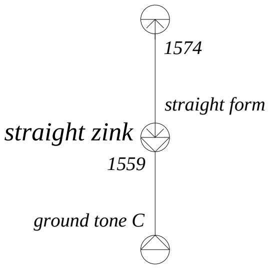
Here is the first example of a separating pair that is split. The object *Zink 1574* and the attribute *ground tone C* however will be used for (two different) separating pairs in the next step again. The last two answers determine the mappings $\widetilde{g}_1$ and $\widetilde{m}_1$:



	$\widetilde{\mathrm{m}}_1$:	*ground tone C*	
$\widetilde{\mathrm{g}}_1$:	$\widetilde{\mathbb{K}}_1$	⊣	$\mathfrak{b}_1$
	$\mathfrak{b}_1$	⤢	×
1574	⊤		⤢

"Name a *straight zink* not having *ground tone C*!"
"*Zink 1559.*"
"Name an attribute of *straight zinks* that *Zink 1574* does not have!"
"*Straight form.*"

This yields the context $\mathbb{K}_1$ which describes the lattice that is generated by the first basic concept *straight zink*:



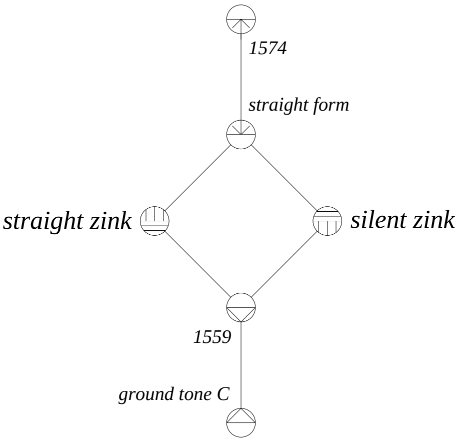
	m_1:	*ground tone C*	*straight form*
g_1:	$\mathbb{K}_1$	⊣	$\mathfrak{b}_1$
1559	$\mathfrak{b}_1$	⤢	×
1574	⊤		⤢

"Is *Zink 1559* a *silent zink*?"
"Yes!"

This implies that "Has every *silent zink* the attribute *ground tone C*?" has to be answered by "No!", because *Zink 1559* and *ground tone C* were a dividing pair.

"Is *Zink 1574* a *silent zink*?"
"No!"
"Has every *silent zink* the attribute *straight form*?"
"Yes!"

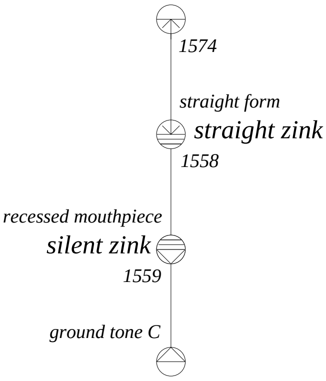
Now $\widetilde{g}_2$ and $\widetilde{m}_2$ are determined. The context $\widetilde{\mathbb{K}}_2$ represents the tensor product of two three element chains:



	$\widetilde{\mathbf{m}}_2$:	*ground tone C*			*straight form*
$\widetilde{\mathbf{g}}_2$:	$\widetilde{\mathbb{K}}_2$	$\bot$	$\mathfrak{b}_1$	$\mathfrak{b}_2$	$\mathfrak{b}_1 \vee \mathfrak{b}_2$
1559	$\mathfrak{b}_1 \wedge \mathfrak{b}_2$	⤢	×	×	×
	$\mathfrak{b}_2$		⤢	×	×
	$\mathfrak{b}_1$		×	⤢	×
1574	$\top$				⤢

"Is *silent zink* a subconcept of *straight zink*?"
"Yes!"
"Is *straight zink* a subconcept of *silent zink*?"
"No! A dividing pair is *Zink 1558* and *recessed mouthpiece*."

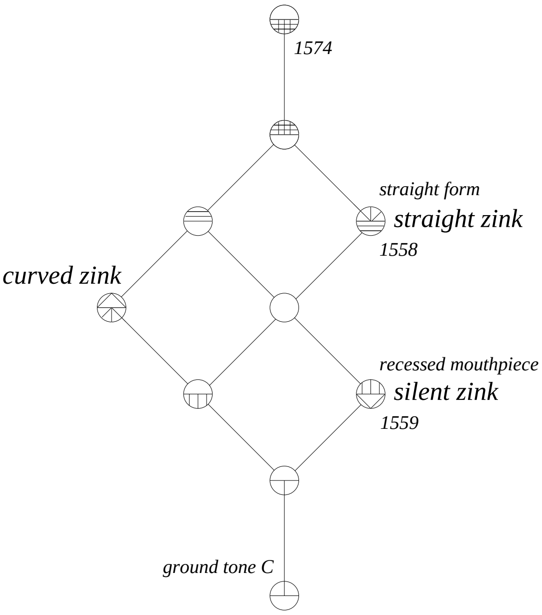
The context $\mathbb{K}_2$ is now determined which is generated by the two first basic concepts *straight zink* and *silent zink*. In particular, one can see in the diagram that *silent zink* is a subconcept of *straight zink*:



	$\mathbf{m}_2$:	*ground tone C*	*recessed mouthpiece*	*straight form*
$\mathbf{g}_2$:	$\mathbb{K}_2$	$\bot$	$\mathfrak{b}_2$	$\mathfrak{b}_1 \vee \mathfrak{b}_2$
1559	$\mathfrak{b}_1 \wedge \mathfrak{b}_2$	⤢	×	×
1558	$\mathfrak{b}_1$		⤢	×
1574	$\top$			⤢

"Is *Zink 1559* a *curved zink*?"
"No!"
"Has every *curved zink* the attribute *ground tone C*?"
"No!"
"Is *Zink 1558* a *curved zink*?"
"No!"
"Has every *curved zink* the attribute *recessed mouthpiece*?"
"No!"
"Is *Zink 1574* a *curved zink*?"
"No!"
"Has every *curved zink* the attribute *straight form*?"
"No!"

The following figure shows the context $\widetilde{\mathbb{K}}_3$ and the mappings $\widetilde{g}_3$ and $\widetilde{m}_3$:



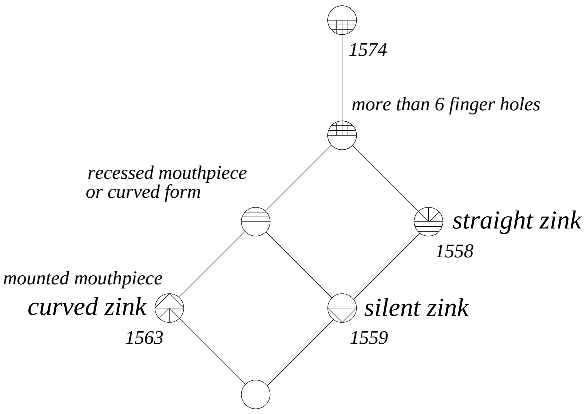
	$\widetilde{\mathfrak{m}}_3$:	*ground tone C*	*recessed mouthpiece*	*straight form*			
$\widetilde{\mathfrak{g}}_3$:	$\widetilde{\mathbb{K}}_3$	$\bot$	$\mathfrak{b}_2$	$\mathfrak{b}_1 \vee \mathfrak{b}_2$	$\mathfrak{b}_3$	$\mathfrak{b}_2 \vee \mathfrak{b}_3$	$\mathfrak{b}_1 \vee \mathfrak{b}_2 \vee \mathfrak{b}_3$
	$\mathfrak{b}_1 \wedge \mathfrak{b}_2 \wedge \mathfrak{b}_3$	⤢	×	×	×	×	×
	$\mathfrak{b}_1 \wedge \mathfrak{b}_3$		⤢	×	×	×	×
	$\mathfrak{b}_3$			⤢	×	×	×
1559	$\mathfrak{b}_1 \wedge \mathfrak{b}_2$		×	×	⤢	×	×
1558	$\mathfrak{b}_1$			×		⤢	×
1574	$\top$						⤢

"Is the infimum of *straight zink*, *silent zink* and *curved zink* a subconcept of *nothing*?"
"Yes!"
"Is the infimum of *straight zink* and *curved zink* a subconcept of *silent zink*?"
"Yes!"
"Name a *curved zink* not having *straight form*!"
"*Zink 1563.*"
"Name an attribute of *curved zinks* that *Zink 1559* does not have!"
"*Attached mouthpiece.*"
"Name an attribute of the supremum of *silent zink* and *curved zink* that *Zink 1558* does not have!"
"*Recessed mouthpiece or curved form.*"
"Name an attribute of the supremum of *straight zink*, *silent zink* and *curved zink* that *Zink 1574* does not have!"
"*More than 6 finger holes.*"

Up to now we have determined the complete lattice generated by the first three basic concepts *straight zink*, *silent zink*, and *curved zink*. It is shown in Fig. 3.

We continue the exploration in the same way with the remaining two basic concepts *cornetto* and *cornettino*. Finally, we get the context $\mathbb{K}_5$ as shown in Fig. 4. Its line diagram shows all information about the hierarchical relationship between the five basic concepts. For example, we can deduce from it that there are no *silent zinks* that are also *cornettos*, because the infimum of *silent zink* and *cornetto* is *nothing*. We can further deduce that there are other zinks than those we chose for the exploration, because the supremum of all basic concepts is different from *everything*. The observation that the supremum of *cornetto* and *cornettino* is *curved zink* and their infimum is *nothing* reflects the fact that the *curved zinks* can be divided in two disjunct classes: *cornettos* and *cornettinos*.

Remark that in the diagram *Zink 1558* is not laying below *attached mouthpiece*, even though Zink 1558 has an attached mouthpiece! *Zink 1558* and *attached mouthpiece* belong to different separating pairs, and so their relationship has not been asked from the expert.



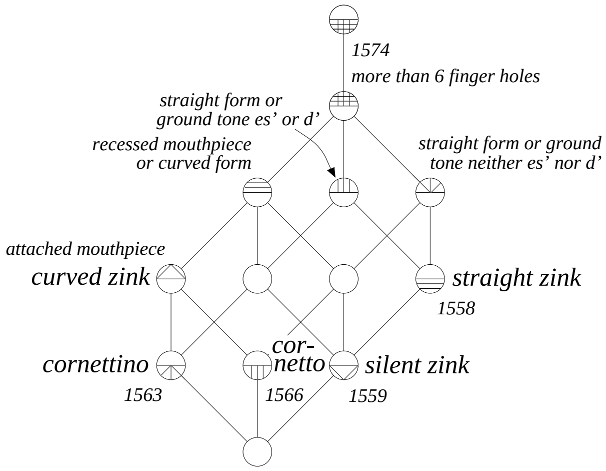
$\mathfrak{g}_3$:	$\mathfrak{m}_3$:	*straight form*	*attached mouthpiece*	*recessed mouthpiece or curved form*	*more than 6 finger holes*
	$\mathbb{K}_3$	$\mathfrak{b}_1 \vee \mathfrak{b}_2$	$\mathfrak{b}_3$	$\mathfrak{b}_2 \vee \mathfrak{b}_3$	$\mathfrak{b}_1 \vee \mathfrak{b}_2 \vee \mathfrak{b}_3$
1563	$\mathfrak{b}_3$	⤢	×	×	×
1559	$\mathfrak{b}_1 \wedge \mathfrak{b}_2$	×	⤢	×	×
1558	$\mathfrak{b}_1$	×		⤢	×
1574	$\top$				⤢

Fig. 3. Result of the exploration of the first three basic concepts

If there are other subconcepts of *zink* we are interested in (for example *tenor zink*, *serpent* or *violoncel serpent*) we can continue the exploration by starting with the context $\mathbb{K}_5$ and adding the new basic concepts. This serial approach allows also to extend already given knowledge at a later time.

References

1. California Road Map, 1:1 200 000, Hallwag 1994
2. B. Ganter: Algorithmen zur Begriffsanalyse. In: B. Ganter,R. Wille, K. E. Wolff (eds.): *Beiträge zur Begriffsanalyse.* B. I.-Wissenschaftsverlag, Mannheim, Wien, Zürich 1987. 241–254
3. B. Ganter, R. Wille: Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Berlin, Heidelberg (to appear)
4. A. W. Hales: On the non-existence of free complete Boolean algebras. Fund. Math. **54**, 1964, 45–66
5. H. Heyde: Hörner und Zinken. Musikinstrumenten–Museum der Universität Leipzig. Katalog Bd. 5. VEB Deutscher Verlag für Musik, Leipzig 1982
6. U. Klotz, A. Mann: Begriffexploration. Diplomarbeit, TH Darmstadt 1988
7. W. Kollewe, M. Skorsky, F. Vogt, R. Wille: TOSCANA — ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille, M. Zickwolff (eds.): *Begriffliche Wissensverarbeitung — Grundfragen und Aufgaben.* B. I.–Wissenschaftsverlag, Mannheim 1994
8. P. Luksch, R. Wille: A mathematical model for conceptual knowledge systems. In: H.–H. Bock, P. Ihm (eds.): *Classification, data analysis and knowledge organization.* Springer, Berlin 1991, 156–162
9. J. F. Sowa: Conceptual Structures. Addison-Wesley, Reading 1983
10. G. Stumme: Attribute exploration with background implications and exceptions. (in preparation)
11. H. Wagner: Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe.* Kösel Verlag, München 1973, 191–209
12. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets.* Reidel, Dordrecht–Boston 1982, 445–470
13. R. Wille: Tensorial decomposition of concept lattices. In: *Order* **2**, 1985, 81–95



$\mathbf{g}_5$:		$\mathbf{m}_5$:				
		straight form or ground tone e' or d'	*straight form or ground tone neither e' nor d'*	*attached mouthpiece*	*recessed mouthpiece or curved form*	*more than 6 finger holes*
	$\mathbb{K}_5$	$\mathfrak{b}_1 \vee \mathfrak{b}_2 \vee \mathfrak{b}_4$	$\mathfrak{b}_1 \vee \mathfrak{b}_2 \vee \mathfrak{b}_5$	$\mathfrak{b}_3 \vee \mathfrak{b}_4 \vee \mathfrak{b}_5$	$\mathfrak{b}_2 \vee \mathfrak{b}_3 \vee \mathfrak{b}_4 \vee \mathfrak{b}_5$	$\mathfrak{b}_1 \vee \mathfrak{b}_2 \vee \mathfrak{b}_3 \vee \mathfrak{b}_4 \vee \mathfrak{b}_5$
1566	$\mathfrak{b}_3 \wedge \mathfrak{b}_5$	⤢	×	×	×	×
1563	$\mathfrak{b}_3 \wedge \mathfrak{b}_4$	×	⤢	×	×	×
1559	$\mathfrak{b}_1 \wedge \mathfrak{b}_2$	×	×	⤢	×	×
1558	$\mathfrak{b}_1$	×	×		⤢	×
1574	$\top$					⤢

Fig. 4. Result of the Distributive Concept Exploration of zinks

14. R. Wille: Bedeutungen von Begriffsverbänden. In: B. Ganter, R. Wille, K. E. Wolff (eds.): *Beiträge zur Begriffsanalyse.* B. I.–Wissenschaftsverlag, Mannheim 1987, 161–211
15. R. Wille: 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 1989, 365–380

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