Local Scaling in Conceptual Data Systems

Gerd Stumme

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

1 Introduction

The management system TOSCANA ([7], [11]) for conceptual data systems ([10], [8]) provides a flexible tool for conceptually structuring large data tables and for dynamicly browsing through the data. Among the applications are a retrieval system for documents in a library ([6]), an evaluation tool for questionaries, and a navigation tool for laws and regulations in civil engineering ([7]).

Conceptual data systems are based on the theory of *formal concept analysis* ([13], [4]) which has been developed mainly at the Technische Hochschule Darmstadt during the last 15 years. The basic idea of conceptual data systems is to provide so-called *conceptual scales* which reflect each a different aspect of the data. In TOSCANA, these scales are displayed on the screen as a line diagram. For focussing on more than one aspect at the same time, the program combines these scales online and shows them as a nested line diagram. Therefore it "blows up" the points of the line diagram of the first scale and inserts the line diagram of the second scale into every point. This nesting process can be repeated. One can also examine only one concept of the first scale by zooming in the point of the scale and displaying a differentiation of that concept by a (nested) line diagram.

The conception of TOSCANA allows a dynamic navigation through the data. There are no restrictions to the user which scale to choose next or where to zoom in. In the implemented version, the scaling process is global: Once the next scale is chosen, it is inserted into every point of the first scale – even if the second scale is irrelevant to the corresponding facet of the first scale.

In this paper we present the technique of *local scaling*, where only concepts of the outer scale, which are differentiated further by the inner scale, are refined. This is also called *dynamic scaling*, since only at run-time the result of the request to the underlying database determines where to scale and where not.

2 Conceptual Data Systems

Conceptual data systems are developed within the theory of formal concept analysis, which is based on a mathematical model of the philosophical understanding of a concept as a unit of thoughts consisting of two parts: The extension which is containing all objects belonging to the concept, and the intension which is containing all attributes common to all those objects (cf. [12]).
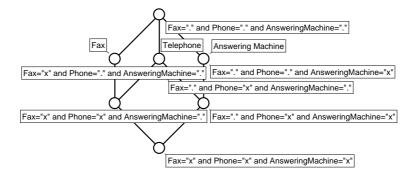
Fig. 1. The concrete scale "Functionality"

In formal concept analysis, the data are given by a *(one-valued) (formal) context* (G, M, I) which consists of a set G of elements called *objects*, a set M of elements called *attributes*, and a relation $I \subseteq G \times M$ where $(g, m) \in I$ is read as "the object g has the attribute m". Now, the *(formal) concepts* of the context (G, M, I) are all pairs (A, B) with $A \subseteq G$ and $B \subseteq M$ such that (A, B) is maximal with the property $A \times B \subseteq I$. The set A is called the *extent* and the set B is called the *intent* of the concept (A, B). The set $\underline{\mathfrak{B}}(G, M, I)$ of all concepts of a context with the order $(A_1, B_1) \leq (A_2, B_2) : \Longleftrightarrow A_1 \subseteq A_2$ is always a complete lattice (cf. [13]) which is called the *concept lattice* of the context (G, M, I). The order relation reflects the natural subconcept-superconcept-relation.

In many applications, objects may not just have attributes or not, but different values for the attributes. This is modeled by a *many-valued context* (G, M, W, I) where G, M, and W are sets whose elements are called *objects, attributes*, and *values*, respectively, and $I \subseteq G \times M \times W$ is a relation such that $(g, m, w_1) \in I$ and $(g, m, w_2) \in I$ implies $w_1 = w_2$. Hence one can understand the attributes m as partial functions from G to W. There is a priori no concept lattice of a many-valued context. It has first to be "translated" into a "one-valued" context. This translation process is called *conceptual scaling*.

A *conceptual scale* for a set $Y \subseteq M$ is a one-valued context $\mathbb{S} := (G_{\mathbb{S}}, M_{\mathbb{S}}, I_{\mathbb{S}})$ with $\bigtimes_{m \in Y} m(G) \subseteq G_{\mathbb{S}}$. The *derived relation* $J_{\mathbb{S}} \subseteq G \times M_{\mathbb{S}}$ is defined by $(g, n) \in J_{\mathbb{S}} : \Longleftrightarrow ((w_m)_{m \in Y}, n) \in I_{\mathbb{S}}$ with $(g, m, w_m) \in I$ for all $m \in Y$. A *conceptual data system* (cf. [10], [8]) consists of a (relational) database in which the many-valued context is stored and of a *conceptual scheme* which contains the scales. In TOSCANA ([7], [11]) one can choose a list $(\mathbb{S}_t)_{t \in T}$ of scales, and the line diagram of the concept lattice of the context $(G, \bigcup_{t \in T} M_{\mathbb{S}_t}, \bigcup_{t \in T} J_{\mathbb{S}_t})$ will be displayed on the screen. Since there is up to now no satisfying algorithm for an automatic drawing of lattices (cf. [9], [14]), it is important that the concept lattice can be embedded in the direct product of the concept lattices of the scales. The diagrams of the scales are provided in the conceptual scheme as well, as so-called *abstract scales. Concrete scales* combine the abstract scales with the
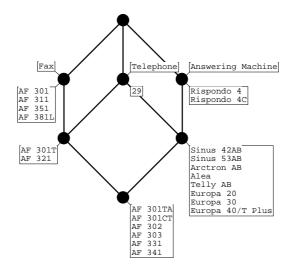
Fig. 2. The realized scale "Functionality"

correct names of the attributes and the database queries. Hence the same line diagram can be used for different concrete scales having the same structure. The result of the scaling process is then shown in a nested line diagram of the concept lattices of the scales.

Let us illuminate this by an example: The conceptual data system "Fax' n' Phone" contains information about all telephones, fax and answering machines sold by the Deutsche Telekom ([2]). The concrete scale "Functionality" in Fig. 1, for example, is used for focussing on the functional aspect of the telephones, fax and answering machines in the database. It is labeled with the three attributes "Fax", "Telephone", and "Answering Machine", and contains furthermore seven SQL statements as objects. The relation $I_\S$ describes which objects have which attributes. If an object has an attribute then, in the line diagram, there is an ascending path from the object to the attribute .

When the user has loaded the conceptual data system "Fax' n' Phone" in TOSCANA, then he can choose among all scales provided in the conceptual scheme. When he chooses for instance the scale "Functionality", then the database is queried with the seven SQL statements. This produces the *realized scale* "Functionality" and displays it on the screen (see Fig. 2). The seven points represent the concepts, and the lines represent the subconcept-superconcept-relation. A concept has all the objects in its extent which are listed below the concept in the line diagram, and it has all the attributes in its intent which are listed above the concept. The lower left concept for example has the objects AF 301T, AF 321, AF 301TA, . . . , AF 341 in its extent, and the attributes "Fax" and "Telephone" in its intent. There are two fax-telephone combinations without
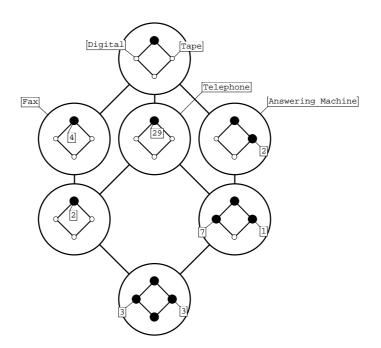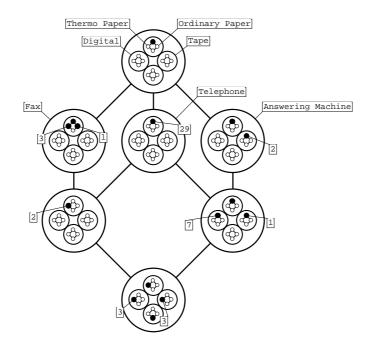
Fig. 3. A nested diagram of the scales "Functionality" and "Registration"

answering machine, namely AF 301T and AF 321, the other six include additionally an answering machine. The number "29" under "Telephone" indicates that there are 29 telephones without fax and answering machine. With a mouse click one can change between the display of this number and the display of the names of the objects.

Scales can be combined on the screen. Figure 3 shows the combination of the scale "Functionality" with the scale "Registration" which has the attributes "Tape" and "Digital" and shows if an answering machine registers on a tape or digitally. The resulting concept lattice is embedded in the direct product of the scales as a join-semilattice. In the diagram, it is marked by filled circles. Every thick line of the outer scale has to be interpreted as four lines each linking the corresponding concepts of the inner scales. For example, the two standalone answering machines have tape registration, and there are seven telephone-answering machines without fax registering digitally and one registering on tape.

On principle, the number of scales nested one into another is unlimited. Figure 4 shows the nested line diagram of the three scales "Functionality", "Registration", and "Paper" (which shows if a fax machine needs ordinary or thermo paper). Since these nested diagrams get soon very large, and one is often not interested in the whole diagram, TOSCANA provides zooming into the scales. E. g., one could zoom in the lower right concept in Fig. 3 and obtain the scale

Fig. 4. A nested diagram of the scales "Functionality", "Registration", and "Paper"

"Registration", but restricted to the telephones with answering machines. Then one can continue the navigation with a new scale, for example one about the number of short dial memories.

However, sometimes it is interesting to see the whole outer scale on the screen – but it would be enough to "blow up" certain circles of the outer scale. For instance, it makes sense to insert the scale "Registration" only into the three concepts of the outer scale having "Answering Machine" in their intent. In the next section, the conception of *local scaling* as an approach to this idea is described.

3   Local Scaling

In the example, one can see that the scale "Registration" is only meaningful for answering machines, hence it would be sufficient to insert it into the bottom and the two right elements of the outer scale. For the scale "Paper" it would be sufficient to insert it into the bottom and the two left elements of the outer scale. We investigate now, for which concepts of an outer scale it is relevant to insert the second scale, and how this fits into the theory. First we look at the bottom elements of the scales.

Scaling a many-valued context (G, M, I) with two scales $\mathbb{S}_1 := (G_1, M_1, I_1)$ and $\mathbb{S}_2 := (G_2, M_2, I_2)$ is equivalent to scaling it with the *semi-product* $\mathbb{S}_1 \boxtimes \mathbb{S}_2 := (G_1 \times G_2, M_1 \dot\cup M_2, \nabla)$ with $(g_1, g_2) \nabla m : \iff (g_i, m) \in I_i$ for $m \in M_i$. The concept lattice of the semi-product is basically the direct product of the concept lattices of the two scales, but with the following modification: From $\underline{\mathfrak{B}}(\mathbb{S}_1)$ and $\underline{\mathfrak{B}}(\mathbb{S}_2)$ we delete the bottom elements, if their extents are empty. Then we build the direct product of the remaining ordered sets, and add a new bottom element, if in the first step at least one bottom element was deleted (cf. [4]). If there is a new bottom element, then we draw it together with a horizontal bracket to indicate that there are lines from all elements not having a lower neighbor to the new bottom element. Figure 5 shows, how the nested diagram of the scales "Functionality", "Registration", and "Paper" in Fig. 4 can be simplified in this way. If there were no models combining telephone, fax and answering machine in the data base, then the lowest circle of the outer scale would be omitted, too.

With the semi-product we have already reduced the complexity of the diagram, since some unrealized concepts are no longer displayed. Figure 6 shows that we can still do better by using *local semi-products*.

For their definition we need the two derivations $A' := \{m \in M \mid (g, m) \in I$ for all $g \in A\}$ for $A \subseteq G$, and $B' := \{g \in G \mid (g, m) \in I$ for all $m \in B\}$ for $B \subseteq M$. The fact that (A, B) with $A \subseteq G$ and $B \subseteq M$ is a concept is equivalent to $A' = B$ and $A = B'$. The smallest concept having an object g in its extent is $\gamma g := (\{g\}'', \{g\}')$.

Definition. Let $\mathbb{S}_1 := (G_1, M_1, I_1)$ be a scale and let C be a convex subset of $\underline{\mathfrak{B}}(\mathbb{S}_1)$ (i.e., $\mathfrak{b}_1, \mathfrak{b}_3 \in C$ and $\mathfrak{b}_1 \leq \mathfrak{b}_2 \leq \mathfrak{b}_3$ imply $\mathfrak{b}_2 \in C$) with $\min(C) \subseteq \gamma(G_1)$. Let $\mathbb{S}_2 := (G_2, M_2, I_2)$ be a scale such that there is an object $\overline{g} \in G_2$ with $\gamma_2 \overline{g} = 1_{\underline{\mathfrak{B}}(\mathbb{S}_2)}$. We define the *local semi-product of* $\mathbb{S}_1$ *and* $\mathbb{S}_2$ *in* C by $\mathbb{S}_1 \boxtimes_C \mathbb{S}_2 := (G, M, \diamond)$ with $G := \{(g_1, g_2) \in G_1 \times G_2 \mid \gamma g_1 \in C$ or $g_2 = \overline{g}\}$, $M := M_1 \dot\cup (\mathfrak{I}_{\max} \times M_2)$ with $\mathfrak{I}_{\max} := \{B \mid (A, B)$ maximal in $C\}$, and $(g_1, g_2) \diamond m_1 : \iff g_1 I_1 m_1$ for $m_1 \in M_1$ and $(g_1, g_2) \diamond (B, m_2) : \iff (g_1 \in B'$ and $g_2 I_2 m_2)$ for $(B, m_2) \in \mathfrak{I}_{\max} \times M_2$.[1]

The following theorem describes how the concept lattice of the local semi-product of two scales looks like. We use the notation $0_1 := 0_{\underline{\mathfrak{B}}(\mathbb{S}_1)}$, $0_2 := 0_{\underline{\mathfrak{B}}(\mathbb{S}_2)}$, $1_1 := 1_{\underline{\mathfrak{B}}(\mathbb{S}_1)}$, $1_2 := 1_{\underline{\mathfrak{B}}(\mathbb{S}_2)}$.

Theorem 1. *For $i = 1, 2$, let $V_i := \underline{\mathfrak{B}}(\mathbb{S}_i)$ if the extent of 0_i is nonempty, and $V_i := \underline{\mathfrak{B}}(\mathbb{S}_i) \setminus \{0_1\}$ otherwise. Then the concept lattice $\underline{\mathfrak{B}}(\mathbb{S}_1 \boxtimes_C \mathbb{S}_2)$ is isomorphic to $V := (V_1 \setminus C) \cup (C \times V_2)$, if $V_2 = \underline{\mathfrak{B}}(\mathbb{S}_2)$ and $0_1 \in C$, with the order $\leq$ given by*

$$\mathfrak{b}_1 \leq \mathfrak{d}_1 : \iff \mathfrak{b}_1 \leq_1 \mathfrak{d}_1,$$
$$\mathfrak{b}_1 \leq (\mathfrak{c}_1, \mathfrak{c}_2) : \iff \mathfrak{b}_1 \leq_1 \mathfrak{c}_1 \text{ and } \mathfrak{c}_2 = 1_{\underline{\mathfrak{B}}(\mathbb{S}_2)}, \qquad (*)$$
$$(\mathfrak{c}_1, \mathfrak{c}_2) \leq \mathfrak{b}_1 : \iff \mathfrak{c}_1 \leq_1 \mathfrak{b}_1,$$
$$(\mathfrak{c}_1, \mathfrak{c}_2) \leq (\mathfrak{e}_1, \mathfrak{e}_2) : \iff \mathfrak{c}_1 \leq_1 \mathfrak{e}_1 \text{ and } \mathfrak{c}_2 \leq_2 \mathfrak{e}_2$$

---

[1] The conditions $\min(C) \subseteq \gamma(G_1)$ and $\gamma\overline{g} = 1_{\underline{\mathfrak{B}}(\mathbb{S}_2)}$ can be avoided when we add additional objects to G, but for our purpose this is not necessary (see Theorem 2).
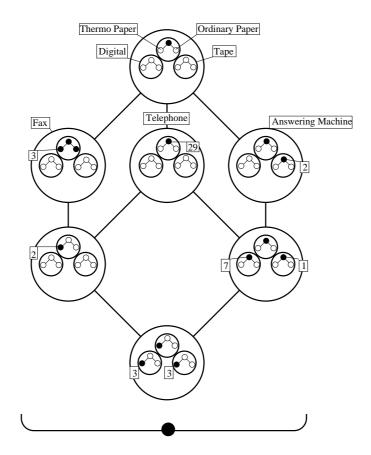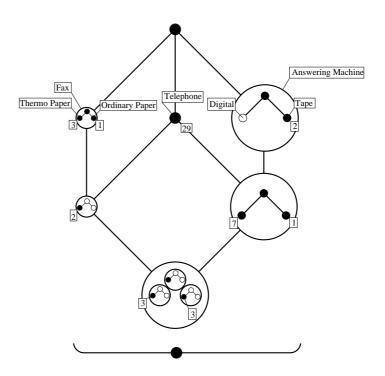
Fig. 5. The semi-product of the scales "Functionality", "Registration", and "Paper"

for $\mathfrak{b}_1, \mathfrak{d}_1 \in \underline{\mathfrak{B}}(\mathbb{S}_1) \setminus C$, $(\mathfrak{c}_1, \mathfrak{c}_2), (\mathfrak{e}_1, \mathfrak{e}_2) \in C \times \underline{\mathfrak{B}}(\mathbb{S}_2)$.
Otherwise $\underline{\mathfrak{B}}(\mathbb{S}_1 \underline{\boxtimes}_C \mathbb{S}_2)$ is isomorphic to $V \dot\cup \{0_V\}$ with $0_V < x$ for all $x \in V$.

Proof. Since $0_1 \in C \Rightarrow 0_1 \in \gamma(G_1) \Rightarrow V_1 = \underline{\mathfrak{B}}(\mathbb{S}_1)$, V or $V \dot\cup \{0_V\}$, resp., is isomorphic to $(V_1 \times \{1_2\}) \cup (C \times V_2) \cup \{(0_1, 0_2)\}$ with the component-wise order. For $(A, B) \in \underline{\mathfrak{B}}(\mathbb{S}_1 \underline{\boxtimes}_C \mathbb{S}_2)$ we write $(A, B_1 \dot\cup B_2)$ with $B_1 := B \cap M_1$ and $B_2 := B \cap (\mathfrak{I}_{\max} \times M_2)$. We define $\iota : \underline{\mathfrak{B}}(\mathbb{S}_1 \underline{\boxtimes}_C \mathbb{S}_2) \to (V_1 \times \{1_2\}) \cup (C \times V_2) \cup \{(0_1, 0_2)\}$ by $\iota(A, B_1 \dot\cup B_2) := ((B_1', B_1), ((\pi_2 B_2)', \pi_2 B_2))$ where π_2 is the projection on the second component of a tuple.

The mapping ι is well defined: B_1 is a concept intent of $\mathbb{S}_1$ because of $(\pi_1 A)' = B_1$, and $\pi_2 B_2$ is a concept intent of $\mathbb{S}_2$ because of $(\pi_2 A)' = \pi_2 B_2$. If $\overline{g} \in (\pi_2 B_2)'$ then $\iota(A, B_1 \dot\cup B_2) = ((B_1', B_1), 1_2)$. If $\overline{g} \notin (\pi_2 B_2)'$ then $A \subseteq \{g_1 \in G_1 \mid \gamma g_1 \in C\} \times (G_2 \setminus \{\overline{g}\})$. Because of $(B_1', B_1) = \bigvee_{g_1 \in \pi_1 A} \gamma g_1$ there is $A = \emptyset$ or there exists $\gamma g_1 \in C$ with $\gamma g_1 \leq (B_1', B_1)$. In the first case we have $\iota(\emptyset, M) = (0_1, 0_2)$. In the second case it remains to show that (B_1', B_1) is laying below a maximal element

Fig. 6. The local semi-product of the scales "Functionality", "Registration", and "Paper"

of C. We have $B_2 \neq \emptyset$, since otherwise $\pi_2 B_2 = \emptyset$ and $\overline{g} \in \emptyset' = (\pi_2 B_2)'$, which is a contradiction to $\overline{g} \notin (\pi_2 B_2)'$. Hence there is $(B, m_2) \in B_2$, which implies $(B_1', B_1) \leq (B', B) \in C$. Hence $(B_1', B_1) \in C$.

The mapping ι is injective: Let $\iota(A, B_1 \dot{\cup} B_2) = \iota(\tilde{A}, \tilde{B}_1 \dot{\cup} \tilde{B}_2)$. Then $B_1 = \tilde{B}_1$ which implies $\pi_1 B_2 = \pi_1 \tilde{B}_2$. With $\pi_2 B_2 = \pi_2 \tilde{B}_2$, it follows $B_1 \dot{\cup} B_2 = \tilde{B}_1 \dot{\cup} \tilde{B}_2$.

The mapping ι is surjective: For $((A, B), (\tilde{A}, \tilde{B})) \in (V_1 \times \{1_2\}) \cup (C \times V_2) \cup \{(0_1, 0_2)\}$, we have $\iota((A \times \tilde{A}) \cap G, B \cup (\{I \in \mathfrak{I}_{\max} \mid I \subseteq B\} \times \tilde{B})) = ((A, B), (\tilde{A}, \tilde{B}))$.

Suprema are preserved by ι, because in concept lattices the supremum is given by $\bigvee_{t \in T} (A_t, B_t) = ((\bigcap_{t \in T} B_t)', \bigcap_{t \in T} B_t)$. Hence infima are preserved, too. □

In [5], W. Geyer examines similar constructions, namely *local direct products*, as generalizations of A. Day's interval doubling construction ([1]). But we cannot use his results, since in local direct products the condition (∗) in Theorem 1 is replaced by $\mathfrak{b}_1 \leq_1 \mathfrak{c}_1$.

The next theorem describes how the concept lattice of the scaled many-valued context is embedded in the nested diagram produced with Theorem 1.
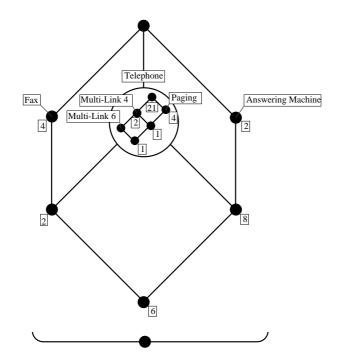
Fig. 7. The local semi-product of the scales "Functionality" and "Wireless Telephone Properties"

Theorem 2. *Let H be a subset of $G_1 \times G_2$, and let C be the convex closure of*
$\{(A_1, B_1) \in \underline{\mathfrak{B}}(\mathbb{S}_1) \mid \exists (A_2, B_2) \in \underline{\mathfrak{B}}(\mathbb{S}_2) \setminus \{1_2\} : ((B_1 \dot\cup B_2)', B_1 \dot\cup B_2) \in \underline{\mathfrak{B}}(H, M_1 \dot\cup M_2, \nabla \cap H \times (M_1 \dot\cup M_2))\}$. *Then*

$$\varepsilon : \underline{\mathfrak{B}}(H, M_1 \dot\cup M_2, \nabla \cap H \times (M_1 \dot\cup M_2)) \to \mathbb{S}_1 \boxtimes_C \mathbb{S}_2$$

with $(A, B_1 \dot\cup B_2) \mapsto (A, B_1 \dot\cup(\{I \in \mathfrak{I}_{\max} \mid I \subseteq B_1\} \times B_2))$ *is a complete* $\bigvee$-*preserving embedding.*

Proof. First we show that ε is well defined: Let $(g_1, g_2) \in A$. For $g_2 = \overline{g}$ we have $(g_1, g_2) \in G$. If $(g_1, g_2) \notin A$, then $\gamma_2 g_2 \neq 1_2$ and $\gamma(g_1, g_2) \in \underline{\mathfrak{B}}(H, M_1 \dot\cup M_2, \nabla \cap H \times (M_1 \dot\cup M_2))$. Hence $\gamma_1 g_1 \in C$, and $(g_1, g_2) \in G$ by definition. It is easy to prove that $A' = B_1 \cup (\{I \in \mathfrak{I}_{\max} \mid I \subseteq B_1\} \times B_2)$ and $A = (B_1 \cup (\{I \in \mathfrak{I}_{\max} \mid I \subseteq B_1\} \times B_2))'$. Because of
$\varepsilon\left(\bigvee_{t \in T}(A_t, B_{t1} \dot\cup B_{t2})\right) = \varepsilon\left(\ldots, (\bigcap_{t \in T} B_{t1}) \dot\cup (\bigcap_{t \in T} B_{t2})\right)$
$= \left(\ldots, (\bigcap_{t \in T} B_{t1}) \dot\cup (\{I \in \mathfrak{I}_{\max} \mid I \subseteq \bigcap_{t \in T} B_{t1}\} \times (\bigcap_{t \in T} B_{t2}))\right)$
$= \bigvee_{t \in T}(A_t, B_{t1} \dot\cup \{I \in \mathfrak{I}_{\max} \mid I \subseteq B_{t1}\} \times B_{t2}) = \bigvee_{t \in T} \varepsilon(A_t, B_{t1} \dot\cup B_{t2})$
the mapping is a $\bigvee$-semilattice-homomorphism. The injectivity follows from $\varepsilon(A, \ldots) = (A, \ldots)$. $\qquad\square$

For applying this theorem, we let the set H be the set of all data base queries yielding a positive result. The theorem states that it is sufficient to display $\mathbb{S}_1 \underset{C}{\times} \mathbb{S}_2$ instead of $\mathbb{S}_1 \times \mathbb{S}_2$, and that the concept lattice can still be embedded in that scale. Figure 6 shows how Fig. 5 can be reduced in this way by applying the local semi-product twice. Instead of 112 potential concepts as in Fig. 4 we only need 24 concepts to show the same information.

More important than the fact that the embedding preserves suprema is that, in general, infima are *not* preserved. It can easily be seen in the line diagram, where an infimum is not preserved. This always indicates an implication between the attributes of the inner and the outer scale. In Fig. 6, for example, one can see that the infimum of the concepts labeled with "Telephone" and "Ordinary Paper" is not realized. The first realized concept below is the bottom element; and this indicates that the combination "Telephone" and "Ordinary Paper" implies all other attributes. In this case the implication is true, because there are no objects in the database fulfilling the premise.

In Fig. 7 we see an example, where the convex set is not including the bottom element of the outer scale. The inner scale is only relevant for wireless telephones, and in fact the $4 + 2 + 1 + 1 = 8$ telephones in the inner scale not noted at its top element are exactly the wireless telephones in the data base. Observe that, according to (∗) in Theorem 1, the two concepts at the lower left and right, labeled with "2" and "8", resp., are lower neighbors only of the top element of the inner scale. The only lower neighbor of the bottom concept of the inner scale is the bottom element of the local semi-product!

In Fig. 8, the scale "Functionality" is combined with a scale about the number of short dial memories. There is a copy of the set M_2 of attributes of the inner scale for every maximal element of the outer scale. This is the (desired) effect of $M := M_1 \dot{\cup} (\mathfrak{I}_{\max} \times M_2)$ in the definition of the local semi-product. Here one can also see that there is not always an additional bottom element to be added, since in this combination the lowest concepts of both scales are realized.

4 Conclusion

We conclude, that there should be four options in the management system for conceptual data systems: The first is the one that is actually implemented, namely to display the complete direct product of the scales, as in Fig. 3 and 4. The second is using the semi-direct product (as in Fig. 5), and the third is using the local semi-direct product as described in the last section.

The fourth option provides more interactivity to the user: It allows him to blow up or to collapse concepts of the outer scale by mouse click. The concepts contained in the convex set C (which are the interesting ones to blow up) are marked by a double circle. Figure 9 shows how this option can be used as a "magnifying glass". It gives the same information as zooming in the lower right concept, but it states additionally the location of the magnifying glass in the outer scale. Hence it combines both the local and the global aspect of the concept. Here one should mention that it is possible not only to display the number (or
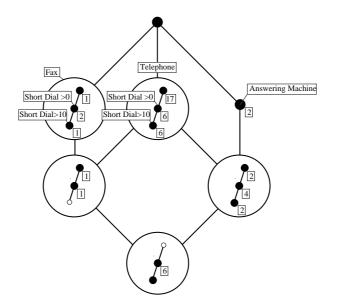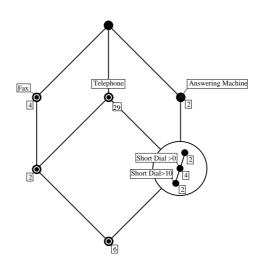
Fig. 8. The local semi-product of the scales "Functionality" and "Short Dial Memory"

names) of the objects labeled at a concept, but also to show the whole extent. Thus in Fig. 9 one could also see how the six objects of the bottom concept behave with respect to the inner scale.
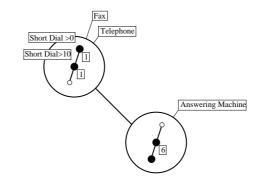
At last we briefly discuss another desirable tool: Often one is interested to see only parts of the outer scale. For example, we want to know more about telephone-fax-combinations. Corresponding to the last paragraph, we can zoom in the lower left concept of the scale "Functionality", but then we lose the information of which combinations include an answering machine. Here *parallel zooming* is a possibility: We "cut out" a convex subset of the outer scale, and only this subset is displayed on the screen (see Fig. 10).

References

1. A. Day: Doubling constructions in lattice theory. Canad. J. Math. **44**, 1992, 252–269
2. Deutsche Telekom: Der Katalog. Herbst/Winter 95/96
3. 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
4. B. Ganter, R. Wille: Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Heidelberg 1996
5. W. Geyer: Lokale direkte Produkte von Begriffsverbänden. Diplomarbeit, TH Darmstadt 1988

Fig. 9. The scale "Functionality" with the scale "Short Dial Memory" as a magnifying glass

Fig. 10. Parallel zooming into the scale "Functionality" with the scale "Short Dial Memory"

6. W. Kollewe, C. Sander, R. Schmiede, R. Wille: TOSCANA als Instrument der bibliothekarischen Sacherschließung. In: H. Havekost, H.-J. Wätjen (eds.): *Aufbau und Erschließung begrifflicher Datenbanken.* (BIS)-Verlag, Oldenburg 1995, 95–114

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. Scheich, M. Skorsky, F. Vogt, C. Wachter, R. Wille: Conceptual data systems. In: O. Opitz, B. Lausen, R. Klar (eds.): *Information and classification.* Springer,

Heidelberg 1993, 72–84

9. G. Stumme, R. Wille: A geometrical heuristic for drawing concept lattices. In: R. Tamassia, I. G. Tollis (eds.): *Graph Drawing*. Springer, Heidelberg 1995, 452–459

10. F. Vogt, C. Wachter, R. Wille: Data analysis based on a conceptual file. In: H.-H. Bock, P. Ihm (eds.): *Classification, data analysis, and knowledge organization*. Springer, Heidelberg 1991, 131–140

11. F. Vogt, R. Wille: 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, Heidelberg 1995, 226–233

12. H. Wagner: Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe*. Kösel Verlag, München 1973, 191–209

13. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets*. Reidel, Dordrecht–Boston 1982, 445–470

14. R. Wille: Lattices in data analysis: how to draw them with a computer In: I. Rival (ed.): *Algorithms and order*. Kluwer, Dordrecht–Boston 1989, 33–58