Dual Retrieval in Conceptual Information Systems

Gerd Stumme

Technische Universität Darmstadt, Fachbereich Mathematik, Schloßgartenstr. 7, D–64289 Darmstadt; stumme@mathematik.tu-darmstadt.de

Abstract. Conceptual Information Systems provide a multi-dimensional conceptually structured view on data stored in relational databases. On restricting the expressiveness of the retrieval language, they allow the visualization of sets of related queries in conceptual hierarchies, hence supporting the search of something one does not have a precise description, but only a vague idea of.
Information Retrieval is considered as the process of finding specific objects (documents etc.) out of a large set of objects which fit to some description. In some data analysis and knowledge discovery applications, the dual task is of interest: The analyst needs to determine, for a subset of objects, a description for this subset. In this paper we discuss how Conceptual Information Systems can be extended to support also the second task.

1 Introduction

Information Retrieval is a widely used term and has many definitions. Essential to all of them is that Information Retrieval is considered as a process of finding specific objects (documents etc.) out of a large set of objects which fit to some description. In some data analysis and knowledge discovery applications, the dual task is of interest: We call *dual retrieval* the process of finding, for a subset of objects, a description for this subset. In this paper we discuss how Conceptual Information Systems support both kinds of retrieval.

In the following we restrict ourselves to a limited subset of Boolean Retrieval, namely on the retrieval of objects (documents, etc.) which are described solely by *conjunctions* of attributes and attribute-value-pairs. This is certainly a strong restriction on the expressiveness of the queries, but allows a powerful visualization of *sets of related queries*, which is based on strong mathematical semantics.

Conceptual Information Systems ([8], [17]) rely on the mathematical theory of *Formal Concept Analysis* ([19], [5]) which provides a formalization of the concept of 'concept'. It reflects an understanding of 'concept' which is first mentioned explicitly in the Logic of Port Royal in 1668 ([1]) and has been established in the German standards DIN 2330 and DIN 2331. There, a concept is understood as a unit of thought consisting of two parts: its extension and its intension. The extension consists of all objects belonging to the concept, and the intension of all attributes common to all these objects ([18]). In Formal Concept Analysis, this leads to a symmetric definition of *formal concepts*. This symmetry allows to describe both retrieval tasks mentioned above in a unified way.

Conceptual Information Systems provide a multi-dimensional conceptually structured view on data stored in relational databases. Conceptual Information Systems are similar to On-Line Analytical Processing (OLAP) tools, but focus on qualitative (i.e. non-numerical) data ([12]). The analog to OLAP dimensions are *conceptual scales*. Each conceptual scale represents a conceptual hierarchy describing the semantics of the range of values for one or more attributes of the database scheme. The conceptual scales are visualized by so-called *Hasse diagrams* which indicate the subconcept-superconcept hierarchy on the concepts.

The drawback that only conjunctions (and not arbitrary Boolean search terms) are allowed in queries, is compensated by the fact that a Hasse diagram does not cover only one concept, but also all related concepts. This makes Conceptual Information Systems powerful for Information Retrieval: Retrieval usually means the search of something one does not have a precise description, but only a vague idea of. For instance, in retrieving literature, the first catchword a user provides is often not the one which describes the requested documents in an optimal way. In a conceptual scale, also all related catchwords are offered to a user; and by visualizing the distribution of the documents over all these catchwords and all their conjunctions, it provides him with an overview over related queries. He can then choose the combination which fits best his needs, and limit the number of hits as much as desired. By zooming into one concept of a conceptual scale with another scale, he can recursively refine the result ([9]).

Based on Formal Concept Analysis, TOSCANA, a management system for Conceptual Information Systems has been developed ([8], [17]). It has been applied in more than 30 scientific and commercial implementations. Up to now, TOSCANA supports only the first task (retrieving objects for given attributes) explicitly. This paper discusses how the dual task can be integrated.

In the next section, we present the basics of Formal Concept Analysis, and describe how the two tasks naturally appear in this theory. In that section, we also introduce Conceptual Information Systems and describe how they support the first retrieval task. The dual retrieval task is the subject of Section 3. In Section 4, we discuss how a preprocessed encoding of the data increases the efficiency of the data access for both tasks. Finally we present some ideas for further research.

2 The Mathematical Background of Conceptual Information Systems: Formal Concept Analysis

Concepts are necessary for expressing human knowledge. Therefore, the retrieval process benefits from a comprehensive formalization of concepts which can be activated to represent knowledge coded in databases. *Formal Concept Analysis* ([19], [5]) offers such a formalization by mathematizing concepts which are understood as units of thought constituted by their extension and intension.

For allowing a mathematical description of extensions and intensions, Formal Concept Analysis always starts with a *formal context.*

2.1 Formal Contexts and their Concept Lattices

Definition. 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 $(g, m) \in I$ is read *"the object g has the attribute m"*.

For $A \subseteq G$, we define $A' := \{m \in M \mid \forall g \in A\colon (g, m) \in I\}$. For $B \subseteq M$, we define dually $B' := \{g \in G \mid \forall m \in B\colon (g, m) \in I\}$.

Now a *(formal) concept* is a pair (A, B) such that $A \subseteq G$, $B \subseteq M$ and $A' = B$, $B' = A$. (This is equivalent to A and B being maximal with $A \times B \subseteq I$.) The set A is called the *extent* and the set B the *intent* of the concept (A, B). The *subconcept-superconcept-relation* is formalized by

$$(A_1, B_1) \leq (A_2, B_2) :\Longleftrightarrow A_1 \subseteq A_2 \quad (\Longleftrightarrow B_1 \supseteq B_2) \ .$$

The set of all concepts of a context (G, M, I) together with the order relation $\leq$ is always a complete lattice, called the *concept lattice* of (G, M, I) and denoted by $\underline{\mathfrak{B}}(G, M, I)$.

Example: The following example is taken from a Conceptual Information System about IT security ([10]). In the 'IT–Grundschutzhandbuch' of the Bundesamt für Sicherheit in der Informationstechnik ([2]), perils to certain objects, such as, e. g., Infrastructure, Telecommunication, Human Resources, are listed, and counter-measures are proposed. The information system presented here is for demonstration purpose only, but a similar, more praxis oriented system with a higher level of detail is offered by NaviCon Gesellschaft für Begriffliche Wissensverarbeitung mbH.

The table for 'Personal' (Human Resources) from [2] is given in Figure 1. It can be understood as a formal context, where the perils 'Personalausfall' (Staff drop out), ..., 'Social Engineering' are the attributes, and the counter-measures M 3.1, ..., M 3.8 are the objects. The relation assigns to each peril possible counter-measures. The context has 13 formal concepts. For instance, there is one concept having M 3.2, M 3.5, M 3.7, and M 3.8 in its extent, and 'Fahrlässige Zerstörung von Gerät oder Daten' (Negligent destruction of machines or data), 'Manipulation/Zerstörung von IT-Geräten oder Zubehör' (Manipulation of IT tools or accessories), and 'Manipulation an Daten oder Software' (Manipulation on data or software) in its intent.

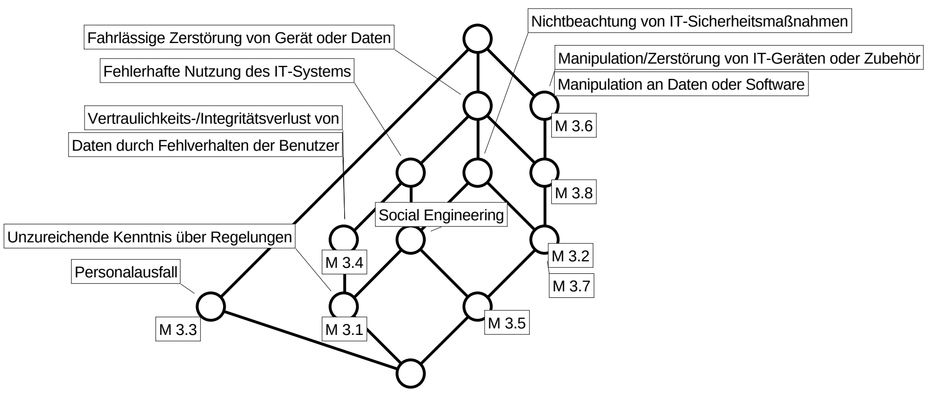
The concept lattice of that formal context is shown in Figure 2. Each circle stands for a formal concept, and the subconcept-superconcept hierarchy can be read by following ascending paths of straight line segments. The intent [extent] of each concept is given by all labels reachable from that context by ascending [descending] paths of straight line segments. For instance, the concept mentioned above is the one labeled by M 3.8.

	Personalausfall	Unzureichende Kenntnis über Regelungen	Vertraulichkeits-/Integritätsverlust von Daten durch Fehlverhalten der IT-Benutzer	Fahrlässige Zerstörung von Gerät oder Daten	Nichtbeachtung von IT-Sicherheitsmaßnahmen	Fehlerhafte Nutzung des IT-Systems	Manipulation/Zerstörung von IT-Geräten oder Zubehör	Manipulation von Daten oder Software	Social Engineering
M 3.1		×	×	×	×	×			×
M 3.2				×	×		×	×	
M 3.3	×								
M 3.4			×	×		×			
M 3.5				×	×	×	×	×	×
M 3.6							×	×	
M 3.7				×	×		×	×	
M 3.8				×			×	×	

M 3.1: Geregelte Einarbeitung/Einweisung neuer Mitarbeiter
M 3.2: Verpflichtung der Mitarbeiter auf Einhaltung einschlägiger Gesetze, Vorschriften und Regelungen
M 3.3: Vertretungsregelungen
M 3.4: Schulung vor Programmnutzung
M 3.5: Schulung zu IT-Sicherheitsmaßnahmen
M 3.6: Geregelte Verfahrensweise beim Ausscheiden von Mitarbeitern
M 3.7: Anlaufstelle bei persönlichen Problemen
M 3.8: Vermeidung von Störungen des Betriebsklimas

Fig. 1. Formal context about perils and counter-measures concerning IT security in Human Resources

In this simple data model, the first retrieval task — finding objects for a given set B of attributes — is equivalent to determining the set B'. For instance, we may want to know which counter-measures can be applied against both the perils 'Nichtbeachtung von IT-Sicherheitsmaßnahmen' (Ignoring of IT security measures) and 'Manipulation an Daten oder Software'. The answer can also be determined in the cross-table in Fig. 1, but for large data sets, the visualization representation by Hasse diagrams presents the information in a more structured way. In the diagram, the user determines the greatest common subconcept of the concepts which are labeled by the two attributes, i. e. the concept where "the attributes first meet at descending the diagram". In our example,



Fahrlässige Zerstörung von Gerät oder Daten
Nichtbeachtung von IT-Sicherheitsmaßnahmen
Fehlerhafte Nutzung des IT-Systems
Manipulation/Zerstörung von IT-Geräten oder Zubehör
Manipulation an Daten oder Software
Vertraulichkeits-/Integritätsverlust von
Daten durch Fehlverhalten der Benutzer
M 3.6
M 3.8
Social Engineering
Unzureichende Kenntnis über Regelungen
M 3.4
M 3.2
M 3.7
Personalausfall
M 3.5
M 3.3
M 3.1

Fig. 2. Hasse diagram of the formal context in Fig. 1

this is the concept labeled by M 3.2 and M 3.7. Below this concept we find additionally the label M 3.5. Hence the answer is the set { Nichtbeachtung von IT-Sicherheitsmaßnahmen, Manipulation an Daten oder Software }$'$ = { M 3.2, M 3.5, M 3.7 }. By using TOSCANA, this answer is usually read directly from the line diagram. TOSCANA provides additionally the possibility to attach reports that are generated by the database management system to the concepts, allowing drill-down to the original data.

For illustrating the dual task — retrieving all attributes common to a given set of objects — let us determine which perils are commonly encountered by the counter-measures M 3.4 and M 3.5. Dually to the query above, we have now to determine the least common superconcept of the two concepts labeled by M 3.4 and M 3.5. This is the concept labeled by 'Fehlerhafte Nutzung des IT-Systems' (Misuse of the IT system). Its intent additionally contains the attribute 'Fahrlässige Zerstörung von Gerät oder Daten' (Negligent destruction of hardware or data). Hence the answer is the set { M 3.4, M 3.5 }$'$ = { Fehlerhafte Nutzung des IT-Systems, Fahrlässige Zerstörung von Gerät oder Daten }. Let us remark that in this example one may also consider all perils that are encountered by *at least one* of the counter-measures M 3.4 and M 3.5. This information is just the set union { M 3.4 }$' \cup$ { M 3.5 }$'$ (i. e, all attributes but 'Personalausfall' and 'Unzureichende Kenntnis über Regelungen') which can also be determined directly in the diagram by taking all perils which are above M 3.4 *or* M 3.5. However, these perils are not grouped together in one single concept, which would support the generation of reports and zooming into the data with other conceptual scales. The handling of disjunctions of attributes und objects is discussed in more detail in [6] and [11]. In this paper, we restrict ourselves to the handling of conjunction (which is logical operator most often applied in Information Retrieval).

In most applications, data tables do not only consist of Boolean attributes as in Fig. 1, but also of many-valued attributes (attribute-value-pairs). Next, we explain how such *many-valued contexts* are handled by formal concept analysis.

2.2 Many-valued Contexts and Conceptual Scales

In database terms, a many-valued context is a relation of a relational database with one key attribute whose domain is the set G of objects. In order to obtain a concept lattice, a many-valued context has to be translated into a one-valued (formal) context. The translation process is described by *conceptual scales* ([16]):

Definition. A *many-valued context* is a tuple $(G, M, (W_m)_{m\in M}, I)$ where G and M are sets of *objects* and *attributes*, resp., W_m is a set of *values* for each $m \in M$, and $I \subseteq G \times \bigcup_{m\in M}(\{m\} \times W_m)$ is a relation such that $(g, m, w_1) \in I$ and $(g, m, w_2) \in I$ imply $w_1 = w_2$.

A *conceptual scale* for an attribute $m \in M$ is a context $\mathbb{S}_m := (G_m, M_m, I_m)$ with $W_m \subseteq G_m$. The context $\mathbb{R}_m := (G, M_m, J_m)$ with $gJ_m n :\iff \exists w{\in}W_m\colon (g, m, w){\in}I \wedge (w, n){\in}I_m$ is called the *realized scale* for the attribute $m \in M$.

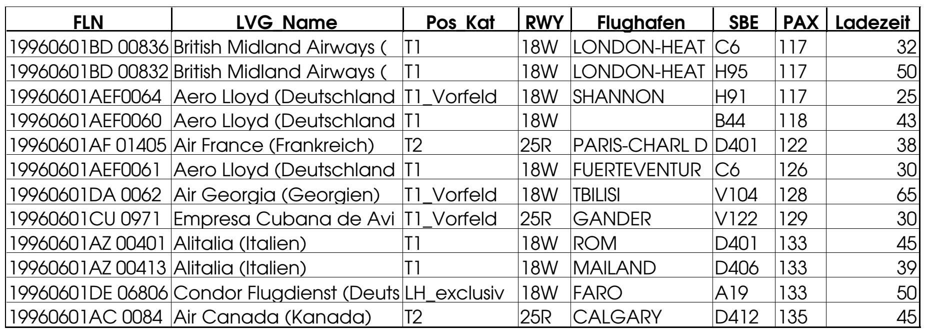
The *apposition* $\mathbb{R}_{m_1}|\mathbb{R}_{m_2}|\ldots|\mathbb{R}_{m_n}$ of realized scales $\mathbb{R}_{m_i}$, $i \in \{1, \ldots, n\}$ is defined as the one-valued context $\left(G, \bigcup_{i\in\{1,\ldots,n\}} M_i, \bigcup_{i\in\{1,\ldots,n\}} J_i\right)$.

Hence, the set M consists of all attributes of the database scheme, while the sets M_m contain the attributes which are shown in the Hasse diagrams.

By apposition of all realized scales one obtains one huge one-valued context. In this context, the derivation operators $\cdot'$ are defined as above. Thus, the two retrieval tasks remain basically the same. Because of the size of the apposition however, this context is never actually computed. It exists only on the conceptual level.

For analyzing the data, the user is usually only interested in considering a small subset of the scales at a time. Hence, it is sufficient to visualize only concept lattices $\underline{\mathfrak{B}}(\mathbb{R}_{m_1}|\mathbb{R}_{m_2}|\ldots|\mathbb{R}_{m_n})$ for small n (i.e. $n \leq 3$). This is an effect also observed in OLAP applications, where the number of dimensions actually displayed is quite small. If more scales (OLAP dimensions, resp.) are involved, then they are usually used for restricting the range of one or more attributes. This corresponds to slicing the data cube in OLAP. TOSCANA supports this feature by allowing to zoom into one concept of a conceptual scale with another scale. Analogies between Conceptual Information Systems and OLAP tools are discussed in [12].

Because of the lack of satisfying lay-out algorithms for lattices, the line diagrams of the different conceptual scales $\underline{\mathfrak{B}}(\mathbb{S}_{m_i})$ have to be precomputed. Their layout is described by the representation language ConScript ([15]) and stored in a so-called *conceptual scheme* ([16]). The concept lattice $\underline{\mathfrak{B}}(\mathbb{R}_{m_1}|\mathbb{R}_{m_2}|\ldots|\mathbb{R}_{m_n})$ can always be embedded in the *direct product* of the concept lattices $\underline{\mathfrak{B}}(\mathbb{S}_{m_1})$, $\ldots$, $\underline{\mathfrak{B}}(\mathbb{S}_{m_n})$. The direct product is visualized by a *nested line diagram* which is explained in the following example.



FLN	LVG_Name	Pos_Kat	RWY	Flughafen	SBE	PAX	Ladezeit
19960601BD 00836	British Midland Airways (	T1	18W	LONDON-HEAT	C6	117	32
19960601BD 00832	British Midland Airways (	T1	18W	LONDON-HEAT	H95	117	50
19960601AEF0064	Aero Lloyd (Deutschland	T1_Vorfeld	18W	SHANNON	H91	117	25
19960601AEF0060	Aero Lloyd (Deutschland	T1	18W		B44	118	43
19960601AF 01405	Air France (Frankreich)	T2	25R	PARIS-CHARL D	D401	122	38
19960601AEF0061	Aero Lloyd (Deutschland	T1	18W	FUERTEVENTUR	C6	126	30
19960601DA 0062	Air Georgia (Georgien)	T1_Vorfeld	18W	TBILISI	V104	128	65
19960601CU 0971	Empresa Cubana de Avi	T1_Vorfeld	25R	GANDER	V122	129	30
19960601AZ 00401	Alitalia (Italien)	T1	18W	ROM	D401	133	45
19960601AZ 00413	Alitalia (Italien)	T1	18W	MAILAND	D406	133	39
19960601DE 06806	Condor Flugdienst (Deuts	LH_exclusiv	18W	FARO	A19	133	50
19960601AC 0084	Air Canada (Kanada)	T2	25R	CALGARY	D412	135	45

Fig. 3. Part of the database of INFO-80 at Frankfurt Airport

Example: As illustrating example, we use a Conceptual Information System established by U. Kaufmann [7] for exploring data of the information system INFO-80 of the "Flughafen Frankfurt Main AG". INFO-80 supports planning, realization, and control of business transactions related to flight movements at Frankfurt Airport. The reason for establishing this Conceptual Information System was that ad-hoc queries (such as "To what extent are the terminal positions reserved for Lufthansa occupied by other carriers?" or "How do take-offs on Runway 18W influence the noise pollution?") could not be generated on the fly but had to be programmed manually.

Here we consider 18389 take-offs effectuated at Frankfurt Airport during one month. For each of these flight movements 110 attributes are registered automatically and stored in a database. Figure 3 shows a small fraction of this database. The column FLN (Flight number) provides a unique identifier for each flight movement. These identifiers were chosen as set G of objects. 77 conceptual scales have been designed (some of the 110 database attributes were of minor importance for data analysis). The resulting apposition of all realized scales has the 18389 flight numbers as objects, and 731 Boolean attributes in total.

Figure 4 shows the conceptual scale *Position of baggage conveyor*. The objects are replaced by strings which can be composed to WHERE-parts of SELECT-clauses in order to query the database. The result of the query is the realized scale in Figure 5. Because of the large number of flight movements, TOSCANA displays only the number of flight movements assigned to each concept. If desired, the user can drill-down to the flight number and to more detailed information.

From the diagram, we can for instance read that there are 5316 flight movements to which a baggage conveyor at Terminal 2 was assigned. By switching to the scale *Position of Aircraft (detailed version)* (cf. Fig. 6), we see that for only 3778+1414=5192 movements the aircraft position was at Terminal 2.

For analyzing how these two different numbers fit together, we may combine the two conceptual scales. The result is displayed in the nested line diagram in

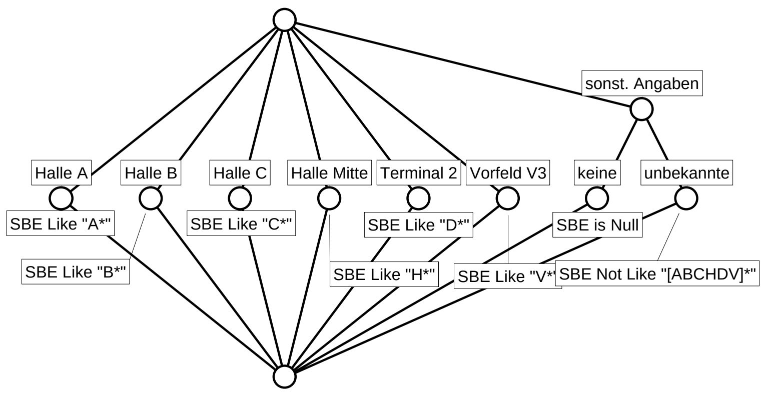


Fig. 4. Conceptual Scale Position of baggage conveyor

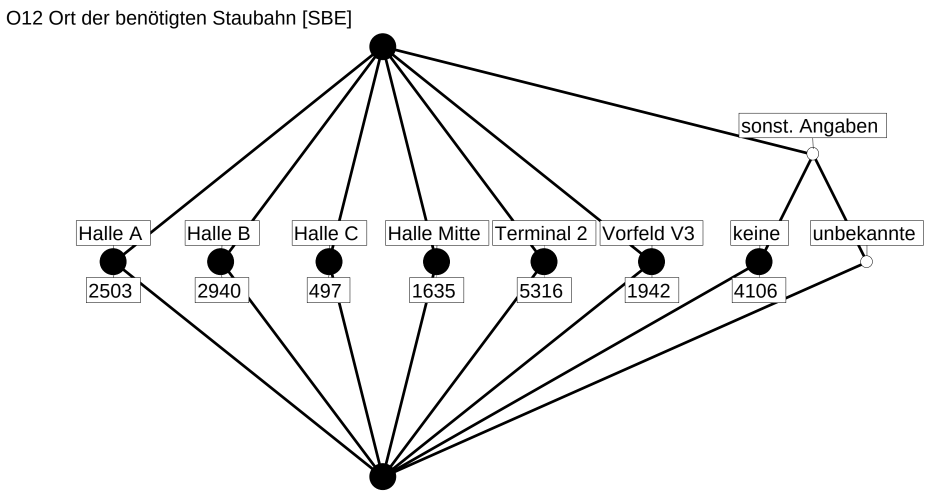


Fig. 5. Realized Scale *Position of baggage conveyor*

Figure 7. The diagram visualizes the direct product

$$\underline{\mathfrak{B}}(\mathbb{S}_{\text{Position of baggage conveyor}}) \times \underline{\mathfrak{B}}(\mathbb{S}_{\text{Position of Aircraft (coarse version)}}) \ .$$

Each of the 17 lines of the outer scale represents seven parallel lines linking corresponding concepts of the inner scale. The embedding of the concept lattice

$$\underline{\mathfrak{B}}(\mathbb{R}_{\text{Position of baggage conveyor}}|\mathbb{R}_{\text{Position of Aircraft (coarse version)}})$$

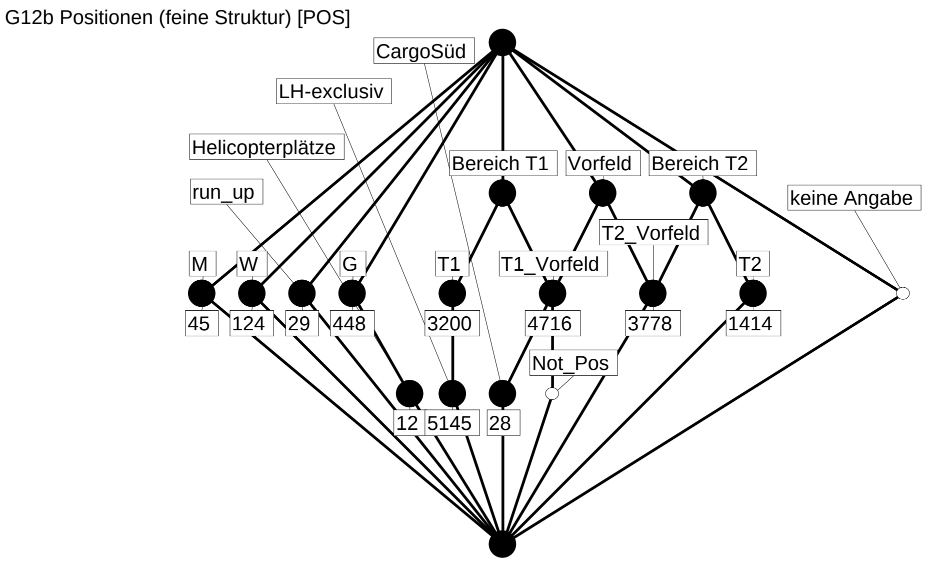


Fig. 6. Realized Scale *Position of Aircraft (detailed version)*

— this is the concept lattice we are interested in — is indicated by the bold circles. Additionally we added the option to sum-up (aggregate) along the hierarchy (i.e., to display the cardinality of the *extent* rather than the one of the *contingent* as in the previous diagrams).

In the diagram, we see that there are 17 movements having the attribute 'Vorfeld V3' (Apron V3) of the outer scale and the attribute T1 of the inner scale. This means that although the aircraft dock at Terminal 1, the luggage to be loaded will leave the baggage transportation system on the apron. In the diagram we can detect some more unusual combinations. For instance, there are four aircraft that docked at Terminal 2, while their assigned baggage conveyors are in Terminal 1. Vice versa, 180 aircraft at Terminal 1 were assigned conveyors in Terminal 2. The seven cases in which the aircraft dock at Terminal 2, while the assigned conveyors are on the apron, should also be considered. In all these cases, one can drill down to the original data by clicking on the number to obtain the flight movement numbers, which in turn lead to the data set stored in the INFO-80 system. Another way to analyze these data is to apply the dual retrieval task: Find all attributes that are common to all these flight movements. This is the subject of the next section.

For concluding this section, we summarize how the first retrieval task is already supported by the management system TOSCANA: For a given set B of attributes, the user obtains the set of all objects having all these attributes (and eventually some more) in common — i.,e., the set B' — by determining

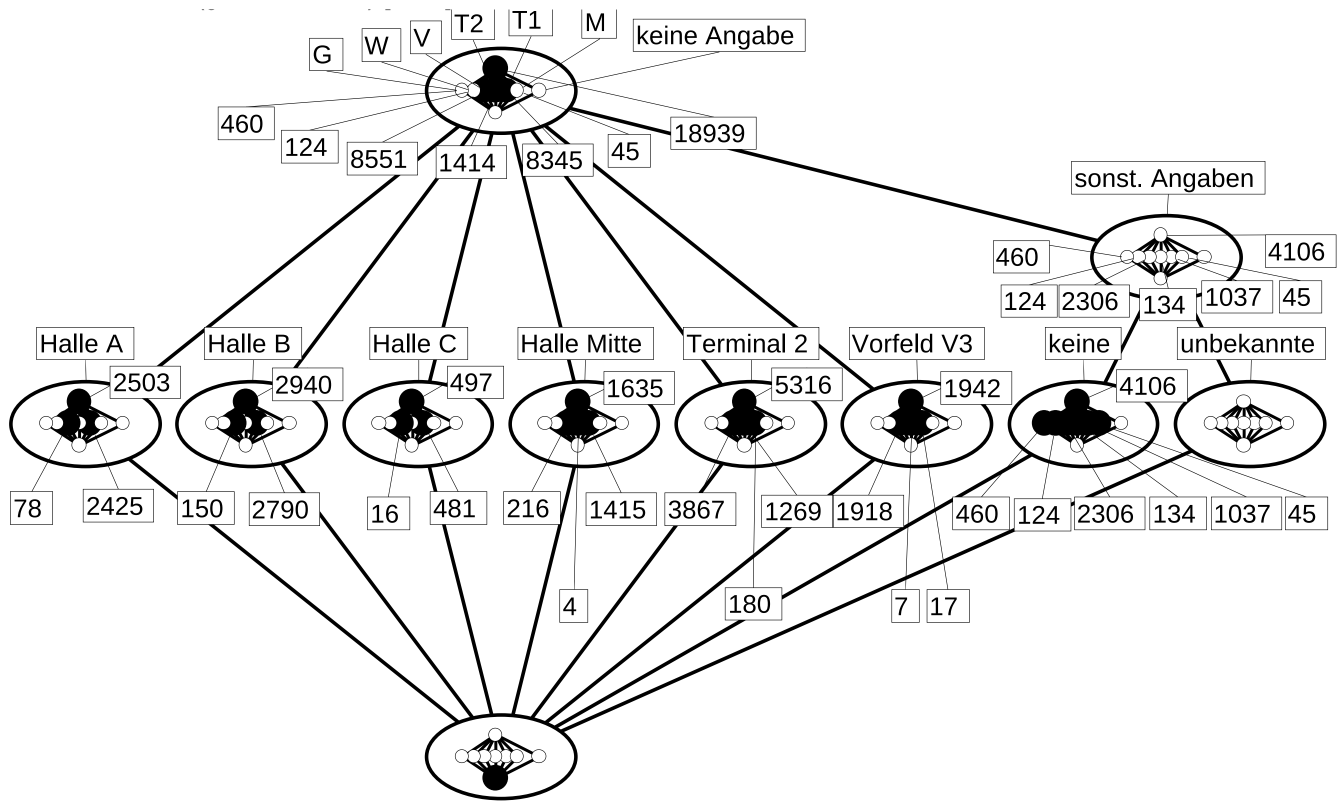


Fig. 7. Nested line diagram of the scales *Position of baggage conveyor* and *Position of Aircraft (coarse version)*

the largest common subconcept of the concepts labeled by the attributes in the nested line diagram of the associated scales. If there are too many scales involved, then zooming into the outer scale reduces the size of the displayed diagram. For instance, by zooming into the concept labeled by 'Terminal 2' in Fig. 7, we obtain the diagram of the conceptual scale *Position of Aircraft (coarse version)*, but with the objects being restricted to those flight movements where the assigned baggage conveyor is at Terminal 2.

3 Retrieving Attributes for a Given Set of Objects

Let us return to the motivating example: What are the reasons for the mismatches between the aircraft positions and the assigned baggage conveyors for certain flight movements? One approach is to examine all attributes that these flight movements have in common in the apposition of all realized scales. For instance, for the 180 flight movements with aircraft positions at Terminal 1 and baggage conveyors at Terminal 2, we obtain 27 out of 731 attributes.

To our dismay, we had to recognize that the mismatch could not be explained by the data stored in the database — neither with the proposed method nor with a detailed manual analysis. The problem is actually under examination by the Airport, and we hope that our work supports this investigation. However, the task of retrieving attributes for a given set of objects is an interesting general problem.

The required functionality can be stated in general terms: Let A be a subset of the set G of all objects stored in the database. *i*) How to compute A' efficiently? *ii*) How to present the result in a suitable way? The first question will be discussed in Section 4. Let us first have a look at the representation of the result.

The example of the 180 flight movements shows us that the number of resulting attributes can be large, so that they should be returned in a structured way. Since the conceptual scales are designed such that related attributes are grouped together, it is natural to group the attributes of A' by these scales. The scales are then arranged in such a way that those which provide the most specific information are listed first. For an automatic listing, 'most specific' has to be formalized: Scales that have many attributes fitting to the objects in A describe more specific features of these objects. Hence, the number of attributes common to all objects in A should mainly determine the position of the scale in the list. For scales with an equal number of hits, the location of the most specific concept having all objects from A in its extent determines the position of the scale in the list. We obtain the two following rules for listing the scales:

1. List the scales $\mathbb{S}_m$ in descending cardinality of the set $A' \cap M_m$.
2. For scales with the same cardinality, list those first where the relative degree of the smallest concept having A in its extent is the highest. The *relative degree* of an element x in a lattice is defined (for this purpose) as the quotient $\frac{a}{b}$ where b is the maximal length (counted in numbers of traversed concepts)

of a path from the top to the bottom element via the element x, and a is the maximal length of a path from the top element to x.

For the 180 flight movements with aircraft positions at Terminal 1 and baggage conveyors at Terminal 2, this list is given in Figure 8.

For teh suggested extension of TOSCANA, the names of the scales should be hyperlinks to the corresponding Hasse diagram. The diagram should display the distribution of the objects in the specific set A together with the distribution of all objects in the database in order to facilitate comparison.

4 Using Bitmap Encoding for Efficiently Supporting both Retrieval Tasks

In Conceptual Information Systems where the database consists of only Boolean attributes (e. g., in the IT Security Information System), bitmap encoding is already in use in order to improve response time. The encoding is done by DOKUANA, a preparation tool for Conceptual Information Systems based on one-valued contexts ([4], [14]). It has for instance been applied in a retrieval system for the library of the Center of Interdisciplinary Studies in Technology at the TU Darmstadt ([9]) and to medical data ([3]).

We suggest to apply the encoding also to many-valued contexts: For each conceptual scale $\mathbb{S}_m$, one adds an attribute m`_bitmap` of type `integer` to the database scheme. To each object in the database one assigns a bitmap which represents the combination of attributes the object has in this particular scale: Each attribute of the scale is represented by a single bit in the bitmap. The bitmap is stored in m`_bitmap` as an integer.

For computing (nested) Hasse diagrams, this encoding increases efficiency, since all information needed from the database can be obtained by *one* query by using GROUP BY. In the actual implementation, this is used only for scales with relatively simple WHERE-clauses. For instance, in the Airport Information System, each concept where objects can be attached to gives rise to one query. 56 queries (and scans!) of the database were needed to obtain Figure 7.

The dual retrieval task is also supported by bitmap encoding. For determining, for a given subset A of objects, which attributes of a conceptual scale $\mathbb{S}_m$ they have in common (i. e., $A' \cap M_m$), we project the selection of these objects on m`_bitmap`. By applying bitwise AND, we obtain all attributes of the scale which are common to all objects in A. Based on these results, one can list the conceptual scales according to the rules given in the previous section.

Combining both retrieval tasks leads to the investigation of *implications*. The set A is normally obtained by an intensional description, i. e. by a (small) set B of attributes with $B' = A$. Hence by computing A', we compute at the same time B''. This results in detecting the implication $B \to B''$ which means that each object having all attributes in B necessarily also have all attributes in B''. Implications are structurally the same as functional dependencies. The difference is that functional dependencies are used as constraints at the design stage

Scale/Attributes	card $(A' \cap M_m)$	rel. degree
Off-Blockzeit in 6 Blöcken zu je 4 Stunden	3	$\frac{2}{7}$
$\rightarrow$ von 4:00 Uhr		
$\rightarrow$ von 0:00 Uhr		
$\rightarrow$ bis 23:59 Uhr		
Anzahl der Post-Paletten	2	$\frac{3}{4}$
$\rightarrow$ keine		
$\rightarrow$ sonstige Angaben		
Passagiere Umsteiger	2	$\frac{3}{4}$
$\rightarrow$ keine		
$\rightarrow$ sonstige Angaben		
Positionen (feine Struktur)	2	$\frac{3}{5}$
$\rightarrow$ Bereich T1		
$\rightarrow$ T1		
Flugabfertigungsart	2	$\frac{3}{6}$
$\rightarrow$ Flüge mit PAX		
$\rightarrow$ GFA-relevant		
Ort der benötigten Staubahn	1	$\frac{2}{3}$
$\rightarrow$ Terminal 2		
Positionen (grobe Struktur)	1	$\frac{2}{3}$
$\rightarrow$ T1		
Folgeflugnummer	1	$\frac{2}{3}$
$\rightarrow$ =0		
Zielflughafen: Nordeuropa	1	$\frac{2}{3}$
$\rightarrow$ nicht Nordeuropa		
Zielflughafen: Australien/Oceanien	1	$\frac{2}{3}$
$\rightarrow$ nicht aus diesem Gebiet		
Flugzeug: Motorenart des Fluggerätes	1	$\frac{2}{3}$
$\rightarrow$ Turbo-Jet		
Flugzeug: Art des Fluggerätes	1	$\frac{2}{3}$
$\rightarrow$ landplane		
Anzahl der benutzten Gates	1	$\frac{1}{2}$
$\rightarrow$ 1 Gate benötigt		
Rollzeit im Outbound	1	$\frac{2}{6}$
$\rightarrow$ 1–. . .		
Flugfunktionen	1	$\frac{2}{5}$
$\rightarrow$ Luftbew.		
Handlingspartner Passage	1	$\frac{2}{5}$
$\rightarrow$ Passage-Abfertigung		
Handlingspartner Operations	1	$\frac{2}{5}$
$\rightarrow$ operative Abfertigung		
Beladezeit	1	$\frac{2}{6}$
$\rightarrow$ > 0		
Dauer der Gateabfertigung	1	$\frac{2}{6}$
$\rightarrow$ > 0		
Dauer der Gepäckbeförderung	1	$\frac{2}{6}$
$\rightarrow$ > 0		
Alle Informationen aus [PCG]	1	$\frac{2}{7}$
$\rightarrow$ letzter Flughafen im EU-Gebiet		

Fig. 8. The attributes in A' grouped according to the scales

of database systems. Functional dependencies are used for designing conceptual scales. Implications are not given apriori, but are results of a data analysis process. They are only valid for the actual data and may become invalid after updates of the database. In a (nested) Hasse diagram, implications which are not functional dependencies are indicated by empty circles. For instance, in Fig. 4, the fact that the concept labeled with 'sonst. Angaben' (Other entries) is not realized indicates the implication {sonst. Angaben}→{keine} because the concept labeled by 'keine' (none) is the largest realized concept below.

5 Outlook

For the dual retrieval task we always required that *all* objects must have the attributes to be returned. Attributes that pertain to all but one of the objects are not shown to the analyst. Further research how to meet this problem is needed. One approach that is actually studied is to measure how much the distribution of the sample set of objects in a scale differs from the distribution of all objects. Actually we investigate different statistical measures derived from the χ^2 distance function ([13]).

References

1. A. Arnauld, P. Nicole: La logique ou l'art de penser — contenant, outre les règles communes, plusieurs observations nouvelles, propres à former le jugement. 3^o édit. reveüe & augm. P., Ch. Saveux, 1668
2. Bundesamt für Sicherheit in der Informationstechnik: *IT–Grundschutzhandbuch 1996. Maßnahmenempfehlungen für den mittleren Schutzbedarf.* Bundesanzeiger, Köln 1996
3. R. Cole, P. Eklund, B. Groh: Scalability in Formal Concept Analysis. *J. on Computational Intelligence***15**(1), Blackwell Publishers, Oxford 1999
4. D. Eschenfelder, W. Kollewe, M. Skorsky, R. Wille: Ein Erkundungssystem zum Baurecht: Methoden der Entwicklung eines TOSCANA-Systems. In: G. Stumme, R. Wille (eds.): *Begriffliche Wissensverarbeitung: Methoden und Anwendungen.* Springer, Heidelberg 1998 (in preparation)
5. B. Ganter, R. Wille: Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Heidelberg 1996
6. B. Groh, S. Strahringer, R. Wille: TOSCANA-Systems Based on Thesauri. In: M.-L. Mugnier, M. Chein (eds.): *Conceptual Structures: Theory, Tools and Applications.* Proc. of the 6th International Conference on Conceptual Structures. LNCS **1453**, Springer, Heidelberg 1998, 127–138
7. U. Kaufmann: *Begriffliche Analyse von Daten über Flugereignisse — Implementierung eines Erkundungs- und Analysesystems mit TOSCANA.* Diplomarbeit, TU Darmstadt, 1996
8. 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
9. T. Rock, R. Wille: Ein TOSCANA-System zur Literatursuche. In: G. Stumme and R. Wille (eds.): *Begriffliche Wissensverarbeitung: Methoden und Anwendungen.* Springer, Berlin-Heidelberg (in preparation)
10. H. Söll: *Begriffliche Analyse triadischer Daten: Das IT-Grundschutzhandbuch des Bundesamtes für Sicherheit in der Informationstechnik.* Diplomarbeit, TU Darmstadt 1998

11. G. Stumme: *Boolesche Begriffe.* Diplomarbeit, TH Darmstadt 1994
12. G. Stumme: On-Line Analytical Processing with Conceptual Information Systems. *Proc. of the 5th Intl. Conf. of Foundations of Data Organization.* Kobe, November 12-14, Kluwer 1999 (to appear)
13. G. Stumme: Exploring Conceptual Similarities of Objects for Analyzing Inconsistencies in Relational Databases. *Proc. 5th Pacific Rim Intl. Conf. on Artificial Intelligence.* Singapore, Nov. 22–27, 1998, LNCS, Springer 1999 (to appear)
14. G. Stumme: Acquiring Expert Knowledge for the Design of Conceptual Scales. *Proc. of the 11th European Workshop on Knowledge Acquisition, Modeling, and Management.* Dagstuhl, May 26–29, 1999 (submitted)
15. F. Vogt: *Datenstrukturen und Algorithmen zur Formalen Begriffsanalyse: Eine C++-Klassenbibliothek.* Springer, Heidelberg 1996
16. 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
17. 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
18. H. Wagner: Begriff. In: H. M. Baumgartner, C. Wild (eds.): *Handbuch philosophischer Grundbegriffe.* Kösel Verlag, München 1973, 191–209
19. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets.* Reidel, Dordrecht–Boston 1982, 445–470