Numerical Aspects in the Data Model of Conceptual Information Systems

Gerd Stumme[1] and Karl Erich Wolff[2]

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

[2] Fachhochschule Darmstadt, Fachbereich Mathematik und Naturwissenschaften,
Schöfferstr. 3, D–64295 Darmstadt; wolff@mathematik.tu-darmstadt.de

Abstract. While most data analysis and decision support tools use numerical aspects of the data, Conceptual Information Systems focus on their conceptual structure. This paper discusses how both approaches can be combined.

1 Introduction

The data model of Conceptual Information Systems relies on the insight that concepts are basic units of human thinking, and should hence be activated in data analysis and decision support. The data model is founded on the mathematical theory of *Formal Concept Analysis.* Conceptual Information Systems provide a multi-dimensional conceptually structured view on data stored in relational databases. They are similar to On-Line Analytical Processing (OLAP) tools, but focus on qualitative (i. e. non-numerical) data. The management system TOSCANA visualizes arbitrary combinations of conceptual hierarchies and allows on-line interaction with the database to analyze and explore data conceptually.

Data tables are usually equipped with different types of structures. While most data analysis tools use their numerical structure, Conceptual Information Systems are designed for conceptually structuring data. As concepts are the basic units of human thought, the resulting data model is quite universal — and is also able to cover numerical aspects of the data. However, up to now, the model does not have any features which support techniques specific to numerical data.

Many applications indicate the need for not only using tools which operate only on numerical or only on conceptual aspects, but to provide an integrative approach combining both numerical and conceptual structures for data analysis and decision support in one tool. In this paper we discuss how the data model of Conceptual Information Systems can be extended by numerical aspects. The developments discussed in the sequel arose mostly from scientific and commercial applications, but for sake of simplicity, we start with a small demonstration application: a Conceptual Information System for a private bank account. But first, we provide some basics about Formal Concept Analysis.

	greenhouseeffect	ozone depletion potential	acidification	nutrification	human toxicity potential	carcinogenity/toxicity	ecotoxicity
NOx	☒		☒	☒	☒		☒
N2O	☒	☒			☒		
NH3			☒	☒		☒	☒
SO2	☒		☒		☒		☒
CO	☒				☒		☒
CO2	☒						

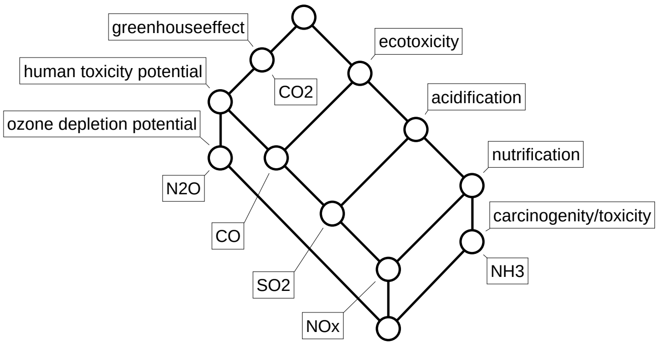

Fig. 1. Formal context and concept lattice of gaseous pollutants

2 The Mathematical Background: Formal Concept Analysis

Concepts are necessary for expressing human knowledge. Therefore, the process of knowledge discovery in databases benefits from a comprehensive formalization of concepts which can be activated to represent knowledge coded in databases. *Formal Concept Analysis* ([10], [1], [13]) 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.*

Definition. A *formal context* is a triple (G, M, I) where G is a set whose elements are called (*formal*) *objects*, M is a set whose elements are called (*formal*) *attributes*, and I is a binary relation between G and M (i.e. $I \subseteq G \times M$); in general, $(g, m) \in I$ is read: "the object g *has* the attribute m".

A *formal concept* of a formal context (G, M, I) is defined as a pair (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 sets A and B are called the *extent* and the *intent* of the formal 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 $\mathfrak{B}(G, M, I)$.

Example. Figure 1 shows a formal context about the potential of gaseous pollutants. The six gases NO_x, …, CO_2 are the objects, and the seven listed perils are the attributes of the formal context. In the line diagram of the concept lattice, we label, for each object $g \in G$, the smallest concept having g in its extent with the name of the object and, for each attribute $m \in M$, the largest concept having m in its intent with the name of the attribute. This labeling allows us to determine for each concept its extent and its intent: The extent [intent] of a concept contains all objects [attributes] whose object concepts [attribute concepts] can be reached from the concept on a descending [ascending] path of

no.	value	paid for	objective	health	date
3	42.00	Konni	ski-club	s	03.01.1995
20	641.26	Konni, Florian	office chairs	/	23.01.1995
27	68.57	Family	health insurance	hi	02.02.1995
34	688.85	Tobias	office table	/	06.02.1995
37	25.00	Father	gymn. club	s	08.02.1995
52	75.00	Konni	gymn. club	s	24.02.1995
73	578.60	Mother	Dr. Schmidt	d	10.03.1995
77	45.02	Tobias	Dr. Gram	d	17.03.1995
80	77.34	Parents	money due	/	21.03.1995

Fig. 2. Withdrawals from a private bank account

straight line segments. For instance, the concept labeled with CO has $\{$CO, SO_2, $NO_x\}$ as extent, and $\{$human toxicity potential, greenhouse effect, ecotoxicity$\}$ as intent. The concept lattice combines the view of different pollution scenarios with the influence of individual pollutants. Such an integrated view can be of interest for the planning of chimneys for plants generating specific pollutants.

In the following, we distinguish, for each formal concept $\mathfrak{c}$, between its *extent* (i.e., the set of all objects belonging to $\mathfrak{c}$) and its *contingent* (i.e., the set of all objects belonging to $\mathfrak{c}$ but not to any proper subconcept of $\mathfrak{c}$). In the standard line diagram, the contingent of a formal concept $\mathfrak{c}$ is the set of objects which is represented just below the point representing $\mathfrak{c}$. The extent of the largest concept is always the set of all objects. The extent of an arbitrary concept is exactly the union of all contingents of its subconcepts.

In many applications, the data table does not only allow Boolean attributes as in Fig. 1, but also many-valued attributes. In the next section, we show by means of an example how such *many-valued contexts* are handled by formal concept analysis.

3 The Conceptual Aspect of the Bank Account System

The basic example underlying this paper consists of a table of all withdrawals from a private bank account during several months. A small part of this table is shown in Fig. 2. As an example, the row numbered 20 contains information about a withdrawal of 641.26 DM for office chairs for the sons Konni and Florian paid on January 23, 1995. In formal concept analysis, data tables such as the one in Fig. 2 are formalized as *many-valued contexts.*

Definition. A *many-valued context* is a tuple $\mathbb{K} := (G, M, (W_m)_{m\in M}, I)$, where G, M, and W_m, $m \in M$, are sets, and $I \subseteq \{(g, m, w) \mid g{\in}G, m{\in}M, w{\in}W_m\}$ is a relation where $(g, m, w_1) \in I$ and $(g, m, w_2) \in I$ implies $w_1{=}w_2$. Thus, each $m \in M$ can be seen as a partial function. For $(g, m, w) \in I$ we say that "object g has value w for attribute m" and write $m(g) = w$.

	sport	doctor	health insurance	health
health=“s”	×			×
health=“d”		×		×
health=“hi”			×	×
health=“/”				

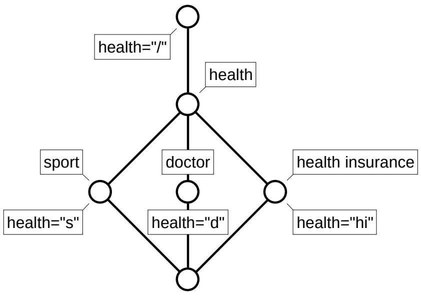

Fig. 3. The scale “health”

Clearly each finite many-valued context can be represented as a relational database table where the set G of objects occurs in the first field chosen as primary key.

In the following we construct a conceptual overview with the purpose to answer questions like “How much has been paid for health for each family member?”. Therefore we first introduce a conceptual language representing the meaning of the values occuring in the column health of Fig. 2. This language is represented by the formal context in Fig. 3. For example, the withdrawals labeled “s” in column health of Fig. 2 are assigned to “sport” and to “health”, while the withdrawals labeled “/” are not assigned to any attribute of this scale. The concept lattice of this formal context is represented by the line diagram in Fig. 3 which demonstrates graphically the intended distinction between the withdrawals not assigned to “health” and those assigned to “health” and the classification of these into three classes. This is an example of a conceptual scale in the sense of the following definition.

Definition. A *conceptual scale* for an attribute $m \in M$ of a many-valued context $(G, M, (W_m)_{m \in M}, I)$ is a formal context $\mathbb{S}_m := (W_m, M_m, I_m)$.

Conceptual scales serve for “embedding” the values of a many-valued attribute in a conceptual framework describing the aspects the user is interested in. But to embed also the original objects, in our example the withdrawals, into this framework we have to combine the partial mapping of the many-valued attribute m and the embedding of the values. This is done in the following definition of a realized scale.

Definition. Let $S_m = (W_m, M_m, I_m)$ be a conceptual scale of an attribute m of a many-valued context $(G, M, (W_m)_{m \in M}, I)$. The context (G, M_m, J) with $gJn :\iff \exists w \in W_m : (g, m, w) \in I \wedge (w, n) \in I_m$ is called the *realized scale* for the attribute m.

To construct the concept lattice of the realized scale we assign to each value w of m, hence to each object of the scale S_m, an SQL-query searching for all objects

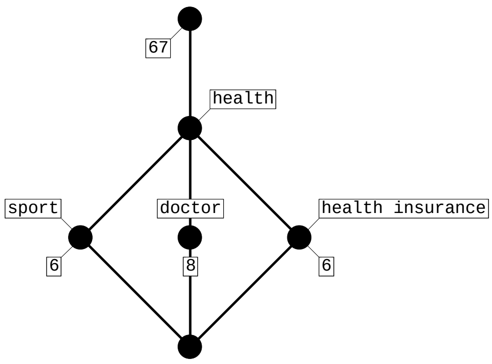

Fig. 4. Frequencies of withdrawals related to "health".

g in the given many-valued context such that $m(g) = w$. The concept lattice of the realized scale for "health" is shown in Fig. 4 where the contingents are replaced by their cardinalities, called *frequencies*.

Reading example: There are exactly six withdrawals assigned to "sport" and exactly 67 withdrawals not assigned to "health". Finally we remark that there are no withdrawals which are assigned to "health" but neither to "sport", "doctor" or "health insurance".

In TOSCANA, the user can choose conceptual scales from a menu. The database is queried by SQL-statements for determining the contingents of the concepts. Finally the results are displayed in a line diagram representing the embedding of the concept lattice of the realized scale in the concept lattice of the scale.

The line diagram in Fig. 4 is unsatisfactory insofar as we would like to see not only the frequencies of withdrawals but the amount of money paid. In the next section we shall discuss how this can be visualized.

4 The Numerical Aspect of the Bank Account System

For an efficient control of the household budget, the user needs an overview over the distribution of the money, and not of the number of withdrawals. Hence, for each contingent S, we display the sum over the corresponding entries in the column "value" instead of the frequency of S. The result of this computation is the left line diagram in Fig. 5. We can see for example that the withdrawals for "sport" sum up to 383 DM and the withdrawals not concerning "health" sum up to 41538 DM. The right line diagram shows, for each formal concept, the sum over the values of all withdrawals in the extent (instead of the contingent) of this concept, for instance the total amount of 45518 DM for all withdrawals in the given data table and the amount of 3980 DM for "health". To visualize also the amount of money paid for the family members (and for relevant groups

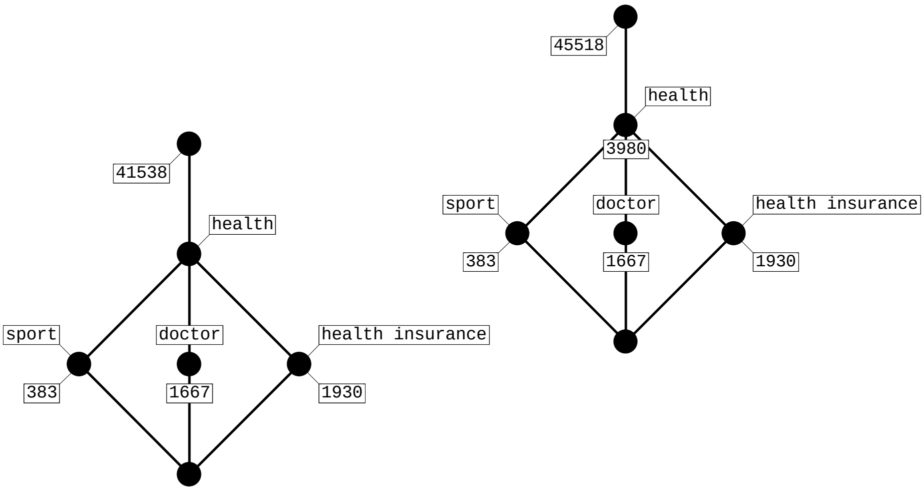

Fig. 5. Summing up book-values over contingents (left) and extents.

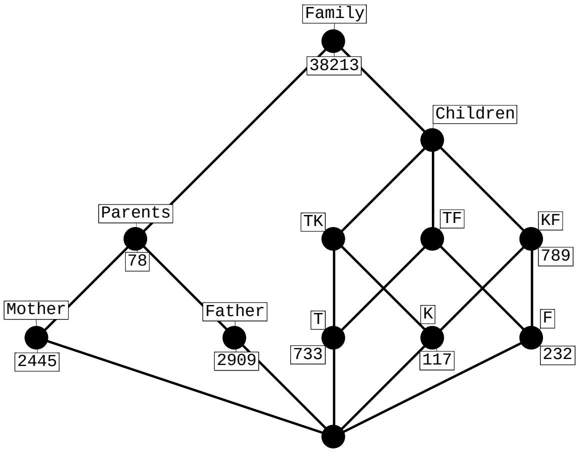

Fig. 6. Summing up book-values over contingents of the scale "family".

of them), we use the scale "family" in Fig. 6. This diagram shows for instance that there are withdrawals of 2445 DM for "Mother", that 78 DM are classified under "Parents", but not under "Mother" or "Father" (this is the "money due" in the last row of Fig. 2) and that 38213 DM appear for withdrawals classified under "Family" which are not specified further.

Next we combine the scales "family" and "health". The resulting *nested line diagram* is shown in Fig. 7. Now the withdrawals are classified with respect to the direct product of the scales for family and health. For instance, 733 DM expended for Tobias split into 45 DM for a doctor and 688 DM not concerning health, i.e., the amount spent on his office table (see Fig. 2). This nested line

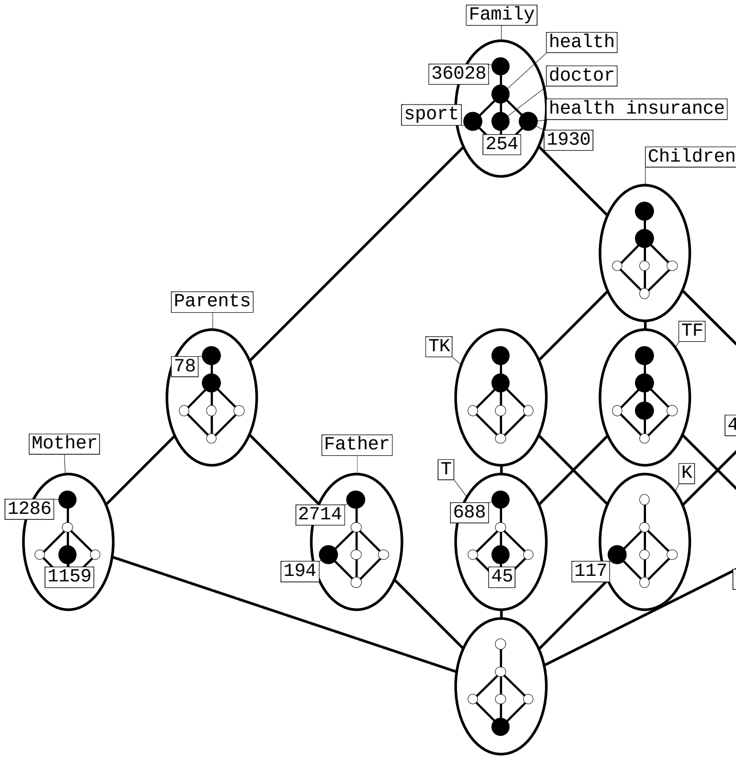

Fig. 7. Summing up over contingents in the nested line diagram of the scales "health" and "family".

diagram shows also that the withdrawals for health insurance (which amount to 1930 DM) are all summarized under the concept "Family" and are not specified further.

In the next section, we describe the formalization of numerical structures. This is the basis for generalizing the example in Section 6.

5 Relational Structures

In Section 2, we have seen how conceptual structures are formalized. Let us now consider the numerical aspect of the data. In fact, the formalization is a bit more general, such that it covers arbitrary relations and functions on arbitrary sets. It is based on the mathematical notion of *relational structures*.

In the bank account example, the bankbook values are real numbers, for which addition is defined. In general, for each $m \in M$, there may be functions and relations on the set W_m.

Definition. A *relational structure* $\mathbb{R} := (W, \mathcal{R}, \mathcal{F})$ consists of a set W, a set $\mathcal{R}$ of relations $R \subseteq W^{ar(R)}$ on W, and a set $\mathcal{F}$ of functions $f \colon W^{ar(f)} \to W$, where ar assigns to each relation and function its arity.

For instance, the data types implemented in the database management system (e. g., `Integer`, `Real`, `Boolean`, `Currency`, or `Datetime`) are relational structures. Hence, for each attribute $m \in M$, we can capture the algebraic structure of its possible attribute values by a relational structure $\mathbb{R}_m := (W_m, \mathcal{R}_m, \mathcal{F}_m)$, just as we captured their hierarchical relationships by a conceptual scale $\mathbb{S}_m$.

Definition. A *conceptual-relational scheme* of a family $(W_m)_{m \in M}$ of sets is a family $(\mathcal{R}_m, \mathbb{S}_m)_{m \in M}$ where, for each $m \in M$, $\mathbb{R}_m := (W_m, \mathcal{R}_m, \mathcal{F}_m)$ is a relational structure and $\mathbb{S}_m = (W_m, M_m, I_m)$ is a conceptual scale.

Here we should mention, that sometimes conceptual and relational aspects overlap. Depending on the purpose, they should be covered by a relational structure or by a conceptual scale, or by both. Time, for instance, can be captured by a linear order in a relational structure or by some scale (e. g., an inter-ordinal scale, if only certain time intervals are of interest).

Relational structures can be used for creating new scales. This *logical scaling* was developed by S. Prediger (cf. [5]). In this paper, however, we discuss only how relational structures may affect the data analysis process once the conceptual scales are created.

6 Conceptual Scaling Supported by Relational Structures

The bank account example and other applications show that it is useful not to analyze numerical and conceptual aspects of the data independently, but to combine them. In this section, we discuss how Conceptual Information Systems can be extended by a numerical component. Since the required functionalities differ from application to application, the idea is to delegate application-specific computations to an external system (e. g., book-keeping system, CAD system, control system, etc.). TOSCANA already provides an SQL-interface to the relational database management system in which the many-valued context is stored, so that we can use the numerical tools of the relational database system (as, for instance, in the bank account example).

In the process of going from the request of the user to the diagram shown on the screen, we can distinguish two consecutive, intermediary subprocesses. First, the chosen scale is imported from the conceptual scheme, and to each of its concepts, a subset of objects is assigned (by default, its extent or contingent). Second, for each of these sets, some algebraic operations may be performed. Most of the implemented Conceptual Information Systems only activate the first step. Our bank account system is an example where the second step is also activated. In the first step, we also can identify two actions where a numerical component can influence the analysis or retrieval process: the import of scales from the conceptual scheme, where parameters can be assigned to parametrized scales, and the import of objects from the database, which can be sorted out by filters. Finally, we can imagine a further action, following the display of the line diagram, which results in highlighting interesting concepts. These four activities which make an interaction between conceptual and numerical component possible now shall be discussed in detail.

6.1 Adapting Conceptual Scales to the Data

A conceptual scale represents knowledge about the structure of the set W_m of possible values of the attribute m. In general, it is independant from the values $m(G)$ that really appear in the database. In some situations however, it is desirable to construct the scale automatically depending on $m(G)$.

Inter-ordinal scales are typically used when a linear order (e. g., a price scale, a time scale) is divided into intervals with respect to their meaning. The boundaries of the intervals are usually fixed by a knowledge engineer. However, the range of possible attribute values is not always known a priori. Hence, for a first glance at the data, it has proved useful to query the database for the minimal and the maximal value and to split up this interval into intervals of equal length. Depending on the application, it might also be useful to fix the boundaries on certain statistical measures, as for instance average, median, quantiles. These “self-adapting scales” reduce the effort needed to create the conceptual scheme, since they are re-usable. It is planned to implement a user interface by means of which the user can edit parameters at runtime. For instance, he could first invoke an inter-ordinal scale with equidistant boundaries and then fine-tune it according to his needs.

This user interface leads to the second example, an application in control theory: Process data of the incineration plant of Darmstadt were analyzed in order to make the control system more efficient (cf. [2]). Process parameters like `ram velocity` and `steam` are stored in a database. The ram velocity does not influence the steam volume directly, but only with a certain time delay. When the time delay is kept variable, the user can change it via the interface during the runtime of TOSCANA. That can be used, for instance, for determining the time delay of two variables experimentally: The engineer examines the nested line diagram of the corresponding scales for ordinal dependencies. By varying the shift time, he tries to augment the dependencies, and to determine in this way the time delay.

The possibility of using parameters is also of interest for filters that control the data flow from the database to TOSCANA. They are discussed in the next subsection.

6.2 Filtering the Objects of the Many-valued Context

In many applications, users are interested in analyzing only a specific subset of objects of the many-valued context; for instance, if one is interested in the withdrawals from the bank account during the past quarter only. If such a subset is determined conceptually, being the extent (more rarely the contingent) of a concept of a suitable combination of conceptual scales, TOSCANA provides for the possibility of "zooming" into that specific concept by mouse click. In the sequel of the analysis only objects belonging to that concept are considered.

But if the interesting subset is not available as extent or contingent of some combinations of earlier constructed scales it is often easier to use a filter. Filters are designed to generate one single interesting subset of the set of objects while conceptual scales generate a whole set of interesting extents and all their intersections and contingents.

For such applications, the conceptual scheme should be extended by *filters*. In addition to conceptual scales, the user can choose filters from a menu. When a filter is activated, then objects are only considered for display if they pass the filter. A filter is realized as an SQL-fragment that is added by AND to the conditions provided by the chosen scales.

The remarks about parameters in the previous subsection apply to filters as well. An example for the use of parameters in filters is again the system of Sects. 2 and 3. As described above, we can construct a filter that only accepts withdrawals effected in a certain period, e. g., the last quarter. The interface for editing parameters introduced in Sect. 5.1 provides the possibility of examining the withdrawals of any period required. When the user activates the filter, he is asked for start and end date.

6.3 Focussing on Specific Aspects of the Objects

The bank account system is an example of focussing on different aspects of the data. There we focus not only on withdrawal numbers, but also on the sum of bankbook values. Now we discuss how this example fits into the formalization described in Sect. 4. Once the user has chosen one or more scales, TOSCANA determines for each concept of the corresponding concept lattice a set S of objects – in most cases its extent or its contingent. In Sect. 1 we mentioned that the user can choose for each concept whether all names of the objects in S shall be displayed or only the cardinality of S. A third standard aspect in TOSCANA is the display of relative frequencies. The last two aspects are examples of algebraic operations.

The focussing in the example of Sects. 2 and 3 can be understood as being composed of two actions: Firstly, instead of working on the set S, the sequence $(m(g) \in W_m)_{g \in S}$ is chosen. In the bank account example, this *projection* assigns to each withdrawal from S the corresponding book-value. Secondly, the sum $\sum(m(S)) := \sum_{g \in S} m(g)$ is computed (and displayed). The latter is done in the relational structure assigned to the corresponding attribute. In TOSCANA, this is realized by a modification of the way the SQL-queries are generated: the standard COUNT-command used for the computation of the frequency of S is replaced by a SUM-command operating on the column "value".

6.4 Highlighting Interesting Concepts

Focussing also can be understood in a different setting. It also means drawing the user's attention to those concepts where the frequency of objects (or the sum of book-values, etc.) is extraordinarily high (or low). The determination of these concepts is based on the frequency distribution of the nested line diagram. This distribution can be represented – without its conceptual order – by a contingency table with entry n_{ij} in cell (i, j) where i (j, resp.) is an object concept of the first (second) scale. As a refinement of Pearson's Chi- Square calculations for contingency tables we recommend calculating for each cell (i, j) the *expected frequency* $e_{ij} := (n_i n_j)/n$ ("expected" means "expected under independence assumption") where n_i (n_j, resp.) is the frequency of object concept i (j, resp.) and n is the total number of all objects. To compare the distribution of the observed frequencies n_{ij} and the expected frequencies e_{ij}, one should study the *dependency double matrix* (n_{ij}, e_{ij}). Pearson's Chi-Square calculations reduce the dependency double matrix to the famous $\chi^2 := \sum_{ij}((n_{ij} - e_{ij})^2/e_{ij})$. But the matrix also can be used as a whole in order to highlight interesting places in a nested line diagram:

- If the user wants to examine the dependency double matrix in detail, then he may choose to display their entries at the corresponding concepts. Additionally, one of the matrices of differences $n_{ij} - e_{ij}$, quotients $(n_{ij} - e_{ij})/e_{ij}$, or quotients n_{ij}/e_{ij} may be displayed in the same way. The conceptual structure represented by the line diagram helps us to understand the dependency double matrix.
- If a less detailed view is required, then the calculation component can generate graphical marks which indicate those concepts where the matrix entries are above or below a given threshold. A typical condition in applications is "$e_{ij} > k$ and $n_{ij}/e_{ij} > p$" where k and p are parameters which can be chosen on a suitable scale.

The Chi-Square formula is a very rough reduction of the information about dependencies, but, clearly, the degree of reduction depends on the purpose of the investigation. If one is interested not only in having an index showing whether there is a dependency, but in understanding the dependencies between two many-valued attributes with respect to chosen scales in detail, then one should carefully study the distribution of observed and expected frequencies. This can be done with the program DEPEND developed by C. Wehrle in his diploma thesis ([9], supervised by K. E. Wolff).)

7 Outlook

The connections between conceptual scales and relational structures should be studied extensively. Therefore, practical relevant examples containing both parts should be considered.

It is of particular interest to examine the compatibility of various conceptual scales and relational structures on the same set of attribute values. From a formal point of view both structures are of the same generality in the sense that each conceptual scale can be described as a relational structure and vice versa. But they are used differently: conceptual scales generate overviews for knowledge landscapes, while relational structures serve for computations.

This paper discussed how numerical components can support conceptual data processing. One should also investigate how, vice-versa, results of data analysis and retrieval activities in Conceptual Information Systems can be made accessible to other systems. This discussion may lead to hybrid knowledge systems composed of conceptual, numerical and also logical subsystems, each focussing on different aspects of the knowledge landscape inherent in the data.

References

1. B. Ganter, R. Wille: Formale Begriffsanalyse: Mathematische Grundlagen. Springer, Heidelberg 1996 (English translation to appear)
2. E. Kalix: Entwicklung von Regelungskonzepten für thermische Abfallbehandlungsanlagen. TH Darmstadt 1997
3. 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
4. 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
5. S. Prediger: Logical scaling in formal concept analysis. LNAI **1257**, Springer, Berlin
6. 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
7. 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
8. F. Vogt, R. Wille: TOSCANA — A graphical tool for analyzing and exploring data. LNCS **894**, Springer, Heidelberg 1995, 226–233
9. C. Wehrle: *Abhängigkeitsuntersuchungen in mehrwertigen Kontexten.* Diplomarbeit, Fachhochschule Darmstadt 1997
10. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets.* Reidel, Dordrecht–Boston 1982, 445–470
11. 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
12. R. Wille: Conceptual landscapes of knowledge: A pragmatic paradigm of knowledge processing. In: *Proc. KRUSE '97*, Vancouver, Kanada, 11.–13. 8. 1997, 2–14
13. K. E. Wolff: A first course in formal concept analysis – How to understand line diagrams. In: F. Faulbaum (ed.): *SoftStat '93, Advances in statistical software* **4**, Gustav Fischer Verlag, Stuttgart 1993, 429–438