

TOSCANA — A Graphical Tool for Analyzing and Exploring Data

Frank Vogt and Rudolf Wille

Forschungsgruppe Begriffsanalyse, Fachbereich Mathematik
Technische Hochschule Darmstadt

Abstract. TOSCANA is a computer program which allows an online interaction with larger data bases to analyse and explore data conceptually. It uses labelled line diagrams of concept lattices to communicate knowledge coded in given data. The basic problem to create online presentations of concept lattices is solved by composing prepared diagrams to nested line diagrams. A larger number of applications in different areas have already shown that TOSCANA is a useful tool for many purposes.

1 Formal Concept Analysis

Formal Concept Analysis has grown during the last fifteen years to a discipline of successful methods for analyzing and exploring data (cf. [3], [8], [11]). Those methods have the main advantage that they clearly unfold the inherent conceptual structures of data contexts always keeping the original data. The conceptual structures are represented by *labelled line diagrams* which have been proved in a large variety of applications as very useful communication tools (cf. [9]). To assist the computation and graphical representation of conceptual structures, computer programs have been developed which are already used by a large number of scientists in different areas of research and applications. TOSCANA is the most extensive program which allows an online interaction with larger data bases to explore and analyse the data conceptually (see [5]). A basic problem of such an interactive program is the online presentation of conceptual diagrams. In this paper we explain how this problem is solved in the program TOSCANA.

Formal concept analysis is based on the notion of a *formal context* which is defined as a triple (G, M, I) consisting of sets G and M together with a binary relation I between G and M ; the elements of G and M are called *objects* and *attributes*, respectively, and gIm (i. e., $(g, m) \in I$) is read: *the object g has the attribute m* . Following the traditional concept theory in philosophy, a *formal concept* of the context (G, M, I) is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A = \{g \in G \mid gIm \text{ for all } m \in B\}$, and $B = \{m \in M \mid gIm \text{ for all } g \in A\}$; A and B are called the *extent* and the *intent* of the formal concept (A, B) , respectively. The hierarchical relation *subconcept–superconcept* is mathematized by the definition:

$$(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2 (\Leftrightarrow B_1 \supseteq B_2)$$

The set of all formal concepts of (G, M, I) with this order relation is a complete lattice called the *concept lattice* of (G, M, I) and denoted by $\mathfrak{B}(G, M, I)$. Concept lattices are the main tools for analyzing and exploring data by methods of formal concept analysis. The first step is always to determine suitable data contexts which are appropriate for the desired aims and goals. Such data contexts may be given by cross tables as in the example of Fig. 1. The objects of this context are the digits $0, \dots, 9$ as they are displayed by a usual seven-segment display; the attributes are the segments of the display, and the relation indicates which digit uses which segment.

	a	b	c	d	e	f	g
0	x		x	x	x	x	x
1						x	x
2	x	x	x		x	x	
3	x	x	x			x	x
4		x		x		x	x
5	x	x	x	x			x
6		x	x	x	x		x
7	x					x	x
8	x	x	x	x	x	x	x
9	x	x		x		x	x

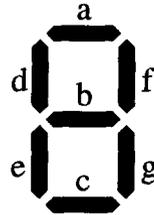


Fig. 1. The context of the digit display

The concept lattice of the context in Fig. 1 is represented in Fig. 2 by a labelled line diagram. The circles in the diagram represent the concepts of the context in Fig. 1. The subconcept–superconcept relation can be read from the diagram by following ascending paths of lines. Therefore, the relation of the context can be reconstructed from the line diagram (i. e., the object “3” has the attribute “a” because there is an ascending path from the black filled circle which is labelled by “3” up to the black filled circle which is labelled by “a”. The extent of each concept is obtained by collecting all objects which can be reached by descending paths, and the intent is obtained dually by collecting all attributes which can be reached by ascending paths. As an example, we consider the concept which is represented by the white circle at the right margin of the line diagram in Fig. 2. The extent of this concept is $\{4, 5, 6, 8, 9\}$, and its intent is $\{b, d, g\}$.

Let us remark that there is an efficient algorithm by B. Ganter (cf. [2]) to compute the concepts of a given formal context. However, in order to represent

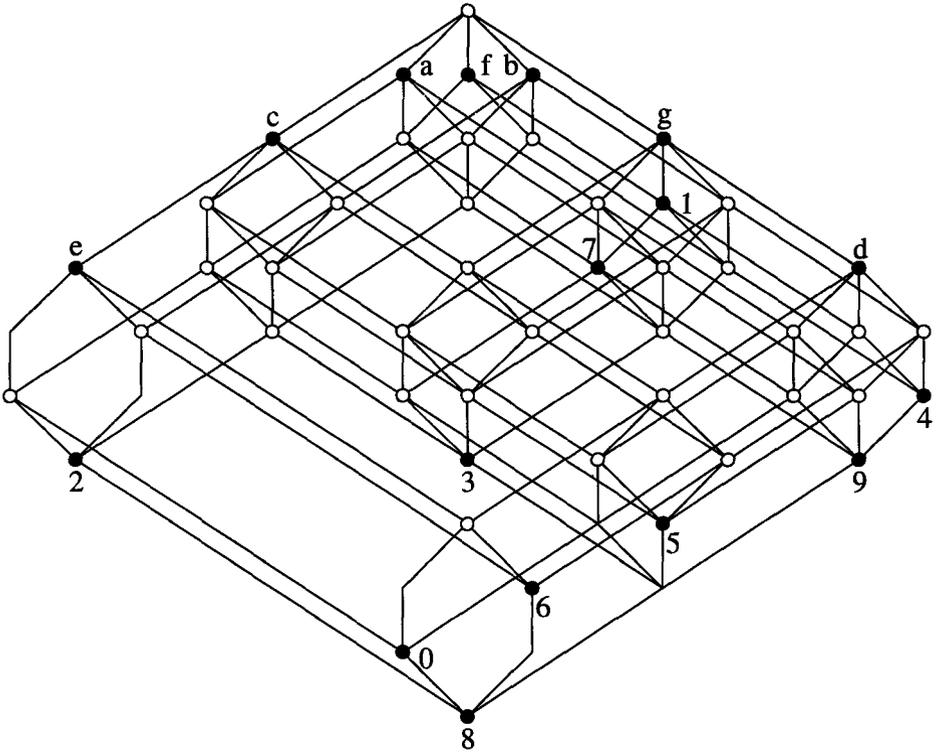


Fig. 2. A labelled line diagram of the concept lattice of the context in Fig. 1

the concept lattice by a labelled line diagram we need not only compute the concepts on the set-theoretical level but must also decide where to draw each circle and line on the drawing pad or the computer screen. There are several methods and algorithms which give partial solutions to this problem (cf. [10]). In TOSCANA, we follow the strategy to compose large diagrams from previously given smaller ones.

2 Nested Line Diagrams

The visual complexity of a labelled line diagram of a concept lattice can become very high although the underlying context is quite small. One way to reduce this complexity is to reduce the number of lines which must be drawn by introducing additional rules for reading the diagram. This idea leads to the notion of *nested line diagrams*. In Fig. 3, we see a nested line diagram which compares the bus types of 74 personal computers with their price. The personal computers are the objects of the underlying context. In the diagram, they are not referred by their name. Only the number of computers is given at the corresponding points. The data for this example are taken from an article in PC Magazine, 1993.

Let us remark that, originally, these data were given as a *many-valued context* which can be thought as a table with arbitrary entries instead of crosses. From a many-valued context we obtain a formal context by translating every many-valued attribute into a collection of one-valued attributes using a *conceptual scale* (see [4]).

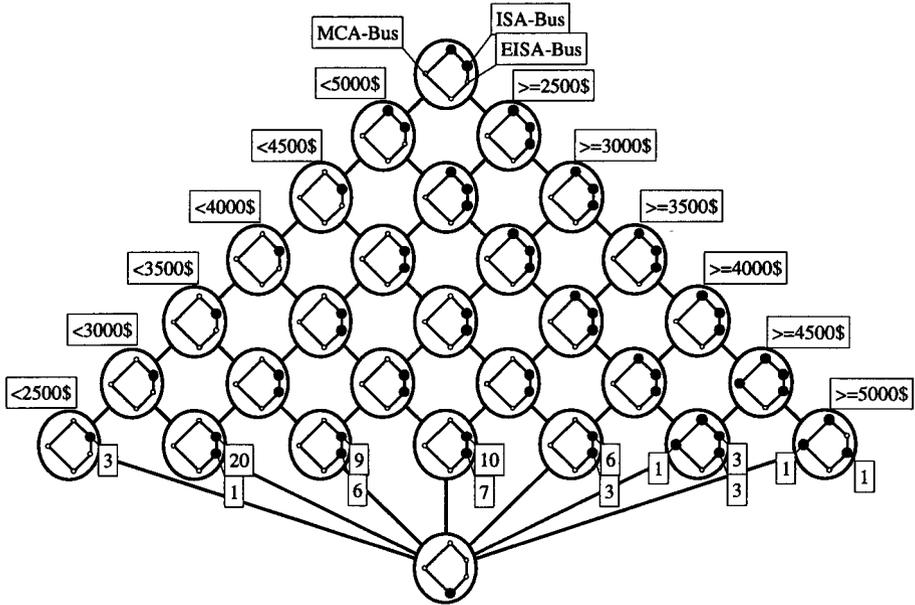


Fig. 3. A nested line diagram showing bus types and prices of personal computers

Let us explain how we read nested line diagrams and how we can obtain them from a formal context. The nested line diagram in Fig. 3 was obtained from the context by splitting its attributes in two groups: the attributes which are connected with the price and those which are connected with the bus type. From the first group we obtain the line diagram which constitutes the outer part (factor) of the nested line diagram whereas we get the line diagram of the inner part (factor) from the second group. The circles of the outer line diagram are enlarged to ellipses, and in each ellipse we draw a copy of the second line diagram. This way we get a representation of the direct product of the two concept lattices. For an ordinary line diagram of this direct product, we have to replace all lines between ellipses by parallel lines between the corresponding circles of the inner diagrams.

The concept lattice of the whole context can now be embedded into this direct product. This embedding is indicated in Fig. 3 by the black filled circles, i. e., only these circles represent concepts of the underlying formal context. We must extend the reading rules for labelled line diagrams as follows: A concept (A, B) is a subconcept of the concept (C, D) if there is an ascending path of

“outer” lines from the ellipse containing the circle which represents (A, B) to the ellipse containing the circle which represents (C, D) , and if there is an ascending path of “inner” lines between the corresponding circles. Observe that this rule simply reflects the comparability in direct products of ordered sets. Following this rule, we can read the extents and intents and reconstruct the relation of the context similarly to non-nested line diagrams. As an example, consider the single personal computer which is referred to by the “1” at the lower right corner of the nested line diagram in Fig. 3. This personal computer has, e. g., the attribute “ISA-Bus” because there are ascending outer lines to the ellipse at the top of the nested line diagram, and there is an ascending inner line from the circle which corresponds to the “1” up to the circle which is labelled with “ISA-Bus”.

3 Conceptual Files

Nested line diagrams provide an effective tool for automatic drawing of large line diagrams if there are enough small line diagrams prepared which can be used as factors in the nested line diagram. This method follows the paradigm that, in many cases, “good” diagrams cannot be drawn fully automatically and that we must provide small line diagrams which are drawn by hand for making them well readable.

Conceptual files formalize these ideas since they are understood as a collection of data which contains a many-valued context as well as scales with their diagrams for all attributes of the many-valued context. In order to obtain a visualization of the conceptual hierarchy of a part of the context, the corresponding attributes and scales are used to draw a nested line diagram (cf. [6], [7]). Fig. 4 shows a nested line diagram of the conceptual hierarchy of a part of a conceptual file containing data of an investigation about children suffering from diabetes. This investigation was made at the Children’s Hospital of the McGill University in Montréal, Canada (cf. [1]).

The nested line diagram in Fig. 4 represents the concept lattice which results from the combination of the three different conceptual scales for “coma”, “pH-level”, and “symptom duration”. Each of the three line diagrams of the scales was drawn by hand and stored into the conceptual file, together with the scales and line diagrams for all other many-valued attributes. For the question which was taken into consideration, these three scales were chosen. Then the nested line diagram was drawn automatically.

4 TOSCANA

TOSCANA is a management system for *conceptual data systems*. The notion of conceptual data systems was introduced in [6] and denotes the combination of a conceptual file together with the software tools which are necessary to explore the data in the conceptual file. With TOSCANA, a user can choose a sequence of

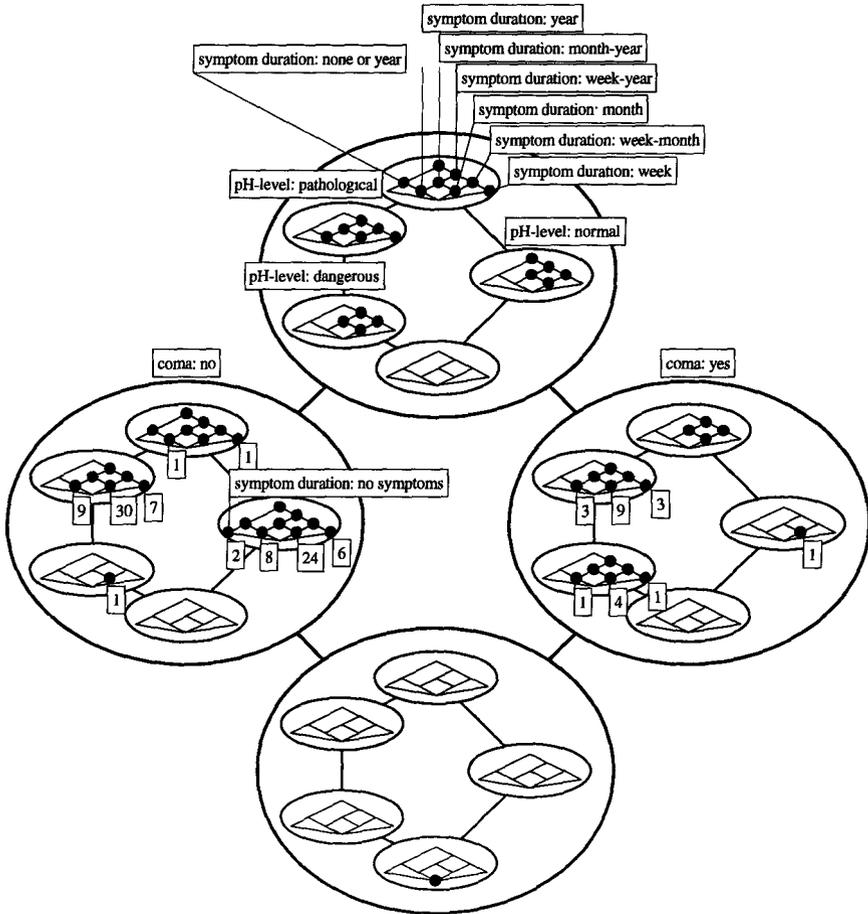


Fig. 4. An automatically drawn nested line diagram

scales of the current conceptual file. Then TOSCANA draws the outer factors of the resulting nested line diagram on the screen (see Fig. 5 for a typical screen). Now the user can zoom into one of the ellipses, i. e., he can choose an ellipse, and then TOSCANA draws the contents of that ellipse, extended by the next factor, on the whole screen. This way, the user can move up and down through the different levels of the nested line diagram and explore the data contained in the conceptual file.

It has turned out that this method of exploration matches the requirements of many users. Usually, only some attributes of a many-valued context are interesting for finding an answer to a specific question. With TOSCANA, the user can choose them and investigate the resulting nested line diagram with respect to his question. After that, he can change the selection of the scales for exploring the data with respect to another question. Although TOSCANA is still available

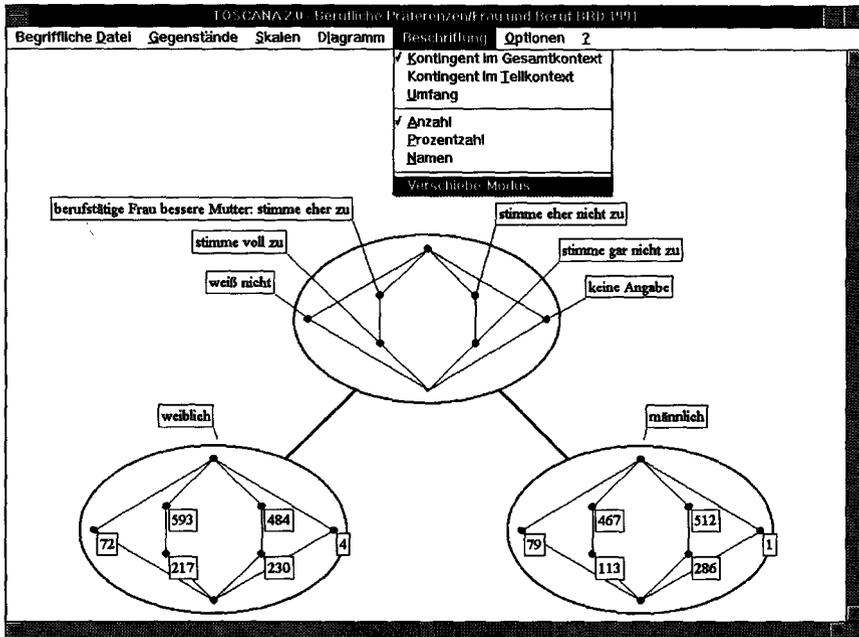


Fig. 5. A typical TOSCANA screen

“only” as a prototype, it already has become the tool of choice for dealing with large data collections and nested line diagrams. For example, also every nested line diagram in this paper is an original output of TOSCANA.

5 Further Developments

For certain applications (e. g. retrieval systems) an extension of TOSCANA turned out to be desirable which allows online creation of new scales with their graphical representation. One way to establish such an extension is to prepare a large library of abstract scales and corresponding line diagrams which may be adapted during the use of a concrete conceptual data system. Of course, not every new scale created by questions of a user will be in the library, but there should be suitable scales and line diagrams in which the new scale and its line diagram can be embedded. There exists already a large collection of conceptual line diagrams for such a library. Algorithms to search for embeddings into given scales and line diagrams have also been designed; methods to compose scales and diagrams may be used, too. Further research and development is necessary to establish the desired extension of TOSCANA. Another project to extend TOSCANA is concerned with components of knowledge inference and acquisition (cf. [9], [11]). Especially, knowledge acquisition needs new graphical tools to support the communication between the system and experts for the desired knowledge.

References

1. A. Ciampi, A. Schiffrin, J. Thiffault, H. Quintal, G. Weitzner, P. Poussier, and D. Lalla. *Cluster analysis of an insulin-dependent diabetic cohort towards the definition of clinical subtypes*. Children's Hospital, McGill University, Montréal, Preprint.
2. B. Ganter. Algorithmen zur Formalen Begriffsanalyse. In: B. Ganter, R. Wille, and K. E. Wolff (eds.), *Beiträge zur Begriffsanalyse*. B. I.-Wissenschaftsverlag, Mannheim, 1987, 241–254.
3. B. Ganter and R. Wille. *Formale Begriffsanalyse*. In preparation.
4. B. Ganter and R. Wille. Conceptual scaling. In: F. Roberts (ed.), *Applications of combinatorics and graph theory to the biological and social sciences*. Springer-Verlag, New York, 1989, 139–167.
5. W. Kollewe, M. Skorsky, F. Vogt, and R. Wille. TOSCANA — ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille and M. Zickwolff (eds.), *Begriffliche Wissensverarbeitung — Grundfragen und Aufgaben*. B. I.-Wissenschaftsverlag, Mannheim, 1994, 267–288.
6. P. Scheich, M. Skorsky, F. Vogt, C. Wachter, and R. Wille. Conceptual data systems. In: O. Opitz, B. Lausen, and R. Klar (eds.), *Information and classification*. Springer-Verlag, Heidelberg, 1993, 72–84.
7. F. Vogt, C. Wachter, and R. Wille. Data analysis based on a conceptual file. In: H.-H. Bock and P. Ihm (eds.), *Classification, data analysis, and knowledge organization*. Springer-Verlag, Berlin-Heidelberg, 1991, 131–140.
8. R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.), *Ordered sets*. Reidel, Dordrecht-Boston, 1982, 445–470.
9. R. Wille. Knowledge acquisition by methods of formal concept analysis. In: E. Diday (ed.), *Data analysis, learning symbolic and numeric knowledge*. Nova Science Publishers, New York-Budapest, 1989, 365–380.
10. 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.
11. R. Wille. Concept lattices and conceptual knowledge systems. *Computers & Mathematics with Applications* **23** (1992), 493–515.