A Geometrical Heuristic
for Drawing Concept Lattices

Gerd Stumme and Rudolf Wille

Technische Hochschule Darmstadt, Fachbereich Mathematik
Schloßgartenstr. 7, D–64289 Darmstadt

Abstract. Concept lattices are used in formal concept analysis to represent data conceptually so that the original data are still recognizable. Their line diagrams should reflect the semantical relationships within the data. Up to now, no satisfactory automatic drawing programs for this task exist. The geometrical heuristic is the most successful tool for drawing concept lattices manually. It uses a geometric representation as intermediate step between the list of upper covers and the line diagram of the lattice.

1 Introduction

Concept lattices have become a useful tool in data analysis and knowledge processing (cf. [7], [10], [6]). They allow to represent data conceptually so that the original data are still recognizable. This supports reliable data interpretations and opens possibilities of exploring data and retrieving information. Concept lattices can be graphically represented by *labelled line diagrams* which have been proved as useful communication tools in many applications (cf. [9]). Although concept lattices can be easily computed for given data contexts by using the *Next-Closure-Algorithm* of B. Ganter (cf. [3]), the automatic drawing of concept lattices is a great problem. The existing drawing programs are not satisfactory because they do not make the lattice structure sufficiently apparent. Also interactive procedures do not reach satisfactory results up to now. A serious problem is how to represent graphically concept lattices such that the semantical relationships within the data become mostly transparent. This problem cannot be solved by drawing algorithms which are only based on formal optimization strategies (e. g. minimizing line crossings).

Up to now, a geometrical heuristic yields the most successful method for drawing concept lattices. The idea is to sketch first a geometric representation of the given concept lattice which gives substantial insights into the structure of the concept lattice. Extensive experiences have shown that these insights enable to draw satisfactory line diagrams of concept lattices. It is the aim of this paper to explain this *"geometrical method"* (cf. [8]).

2 Concept Lattices

Formally, a concept lattice is derived from a *(formal) context* defined as a triple (G, M, I) where G and M are sets and I is a relation between G and M (i. e. $I \subseteq$

Triangles	equilateral	scalene	isosceles	oblique-angled	acute-angled	obtuse-angled	right-angled
T_1 $(0,0),(6,0),(3,1)$			×	×		×	
T_2 $(0,0),(1,0),(0,1)$			×				×
T_3 $(0,0),(4,0),(1,2)$		×		×	×		
T_4 $(0,0),(2,0)(1,\sqrt{3})$	×		×	×	×		
T_5 $(0,0),(2,0),(5,1)$		×		×		×	
T_6 $(0,0),(2,0),(1,3)$			×	×	×		
T_7 $(0,0),(2,0),(0,1)$		×					×

Fig. 1. Context of triangles

$G \times M$). The elements of G and M are called *objects* and *attributes*, respectively, and gIm $(:\iff (g,m) \in I)$ is read: *the object g has the attribute m*. The two following operators are needed to explain what we understand under a formal concept:

$$A \mapsto A' := \{m \in M \mid gIm \text{ for all } g \in A\} \text{ for } A \subseteq G$$

$$B \mapsto B' := \{g \in G \mid gIm \text{ for all } m \in B\} \text{ for } B \subseteq M$$

Now a *(formal) concept* of the context (G, M, I) is a pair (A, B) with $A \subseteq G, B \subseteq M$, $A' = B$, and $B' = A$. The set A is called the *extent* of the concept, the set B the *intent*. The hierarchical *subconcept-superconcept-relation* is mathematized by

$$(A_1, B_1) \le (A_2, B_2) :\iff A_1 \subseteq A_2 (\iff B_1 \supseteq B_2)$$

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

The following example shows how a line diagram of a concept lattice unfolds the conceptual relationships contained in the underlying data context. The formal context given by Fig. 1 has common properties of triangles as attributes and enough concrete triangles as objects to violate all non-valid implications between those properties (cf. [2]). The concept lattice of the context in Fig. 1 is shown in Fig. 2. The numbered circles indicate the concepts of the given context. A circle labelled by an object g represents the concept $\gamma g := (\{g\}'', \{g\}')$, i. e., the concept with the smallest extent containing g. A circle labelled with an attribute m represents the concept $\mu m := (\{m\}', \{m\}'')$, i. e., the concept with the smallest intent containing m. In general, the circle of a concept is linked by a descending path to the circles of all those objects which belong to the extent of the concept and by an ascending path to the circles of all those attributes which are contained in the intent of the concept. For example, concept 12 has
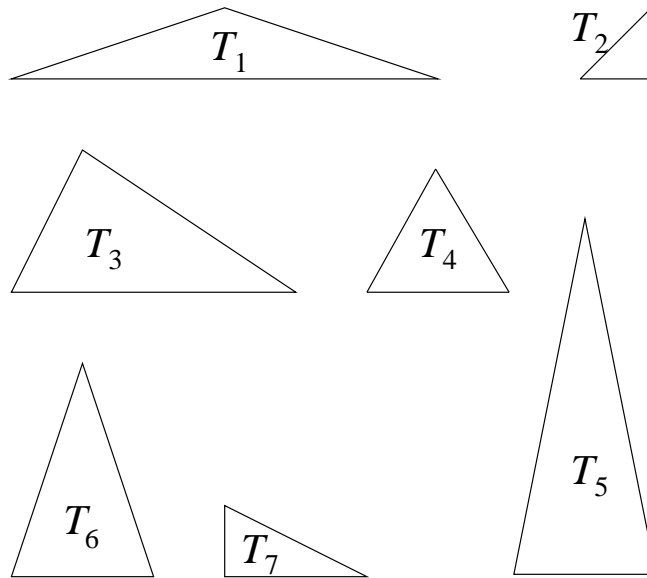
Fig. 2. Concept lattice of the context in Fig. 1

the objects T_1, T_2, and T_6 in its extent and the attributes *isosceles* and *scalene* in its intent. The whole information given by the context (G, M, I) can therefore be read from the line diagram of its concept lattice: An object g has an attribute m if and only if there is an ascending path from g to m in the line diagram.

3 A Geometric Representation

The considered geometric representation of lattices is best explained by imagining the following situation: One is sitting on the top element of a lattice and is looking downwards. The first elements which one sees are the lower covers of the top element. Those elements are viewed as "points". In general, every element having only one upper cover shall be recognized as a "point"(hidden partly by its upper cover if this is not the top element). Further downwards one may discover elements having more than one upper cover. An element with two upper covers is viewed as a line segment between those upper covers, an element with three upper covers as triangle and so on. An element having n upper covers shall therefore be understood as a $(n-1)$–simplex with its upper covers as vertices. Of course, these vertices themselves may be simplices! The geometric representation of the lattice is now a visualization of those geometric objects with their incidences. A geometric representation of the triangle concept lattice is shown in the upper left of Fig. 3. Concept 1 is not represented in the

diagram, because we have taken it as viewpoint. The concepts 2, 4, and 7 are represented by whole circles as they are the only "points" that are not "hidden" behind other concepts. The concepts 3, 8, 10, and 17 are also represented by circles as each of them has also only one upper cover, but they are lying below concepts different from concept 1 and are therefore partially hidden in the geometric representation. In the concept lattice we find seven concepts having two upper covers: these are the concepts 5, 6, 9, 11, 12, 13, and 15. In the geometric representation they are visualized by line segments linking their upper covers. The line segements are labelled by the corresponding concept numbers. Finally we find two triangles in the geometric representation. They are representing the concepts 14 and 16 which have three upper covers. The bottom element of the lattice, concept 18, is not visualized in the geometric representation. However we can easily deduce its upper covers: they are those concepts not beeing a vertex

of a simplex and not having a point lying behind them: these are the concepts 13, 15, 16, and 17.

The list of upper covers in Fig. 3 is an output of the basic program of Formal Concept Analysis (cf. [1]). The upper covers of a concept are listed in the line headed by the number of the concept. The list is sufficient for drawing the geometric representation. It is ordered in such a way that no concept appears as an upper cover of another concept before it is listed itself. So it is possible to obtain the geometric representation by working through the list of upper covers once from top to bottom. M. Kark has written an interactive program supporting the derivation of a geometric representation from a given list of upper covers (cf. [5]). An output of this program for the former example is also shown in Fig. 3.

4 From Geometric Representations to Line Diagrams

Geometric representations can effectively support the drawing of line diagrams. They have extensively been used for elaborating adequate line diagrams of concept lattices. For this purpose it is important that meaningful substructures of lattices can be recognized by corresponding substructures of geometric representations.
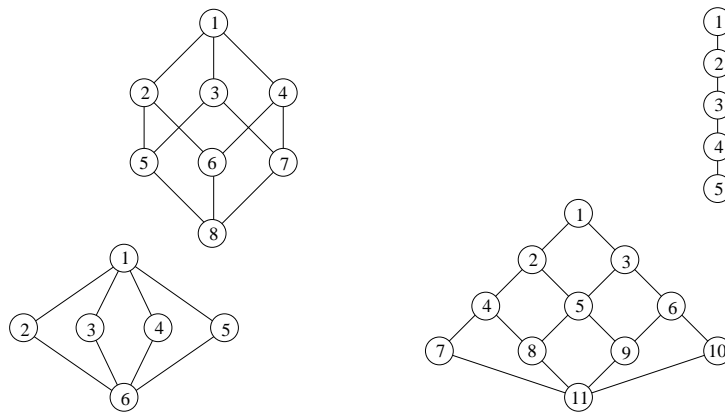

Fig. 4. Linediagrams and geometric representations of lattices

Figure 4 shows some important substructures of lattices and their corresponding geometric representations: The cube in the upper left indicates the independence of three attributes in a concept lattice. The chain at its right is typical for ordinally dependent attributes, while the diamond in the lower left signals that there are attributes excluding each other. The lattice in the lower right is caused by attributes describing an interval structure.

Fig. 5. Line diagram based on the geometric representation in Fig. 6

Let us now have a look how the geometric representation in Fig. 3 has helped us to draw the concept lattice in Fig. 2. As already described in the preceding section, the program of P. Burmeister calculated the list of upper covers in Fig. 3 from the context in Fig. 1. Then the geometric representation was sketched manually. It shows us that the lattice mainly consists of two cubes. If one already has some experience in reading geometric representations, one also knows how the two cubes are "glued" together. If not, one starts drawing the "upper" cube consisting of the concepts 1, 2, 4, 7, 5, 9, 12, and 14. Thereby one has the choice to place two of the concepts 2, 4, and 7 on the outside. We choose concept 2 and 7 as they have "points" as lower covers. This helps us keeping the line diagram in balance. Now we can attach concept 3 to concept 2 and complete the second cube by the concepts 6, 11, and 16. The concepts 7 and 12 are completed to a parallelogram by adding the concepts 8 and 13. Together with the concepts 9 and 14, the concepts 10 and 15 also form a parallelogram. The only which is left to do is to attach the concept 17 to concept 6 and to add the bottom element, concept 18.

Although drawing of parallelograms has high preference, the heuristic is not particularly based on specific lattice properties. So it may also be useful for elaborating appropriate drawings of arbitrary (partially) ordered sets. In general one does not have top and bottom elements in ordered sets; in this case the minimal and maximal elements of the ordered set has also to appear in the geometric representation.

5 A concrete example

The geometrical heuristic for drawing concept lattices has been applied in a great variety of applications. Here we can demonstrate this by only one example from a collaboration with the ministry of civil engineering of the German province *Nordrhein-Westfalen* to develop a retrieval system concerning all regulations for building constructions. Concept lattices are used in the retrieval system as conceptual nets leading the users to the regulations which they have to consider in a specific situation of the process of planing and designing. For our example we choose the concept lattice for the fundamental construction (*Rohbau*) of a family house. It was interesting that there have been attemps of drawing experts to create an adequate line diagram without the geometrical heuristic. The results were not bad, but they could be substantially improved by using the geometrical method. The lattice has 30 elements which are listed in Fig. 6 with their upper covers. The geometric representation in Fig. 6 is shown as it was produced directly from this list. The left–right–order in the geometric diagram gave the basic insight for the design of the line diagram (of course, there are many more instances which influenced the drawing of the line diagram). The labelled line diagram of the concept lattice which was also appreciated by our collaborating engineers is presented in Fig. 5.

Fig. 6. List of upper covers in the concept lattice *"Rohbau"* and geometric representation derived from it

References

1. P. Burmeister: Programm zur formalen Begriffsanalyse einwertiger Kontexte. TH Darmstadt 1987 (Latest version 1991 for Atari ST and MS DOS)
2. P. Burmeister: Merkmalsimplikationen bei unsicherem Wissen. In: W. Lex (ed.): *Arbeitstagung Begriffsanalyse und Künstliche Intelligenz, Informatik–Bericht 89/3.* TU Clausthal-Zellerfeld 1991, 15–46
3. B. Ganter: Algorithmen zur Begriffsanalyse. In: B. Ganter, R. Wille, K. E. Wolff (ed.): *Beiträge zur Begriffsanalyse.* B. I.-Wissenschaftsverlag, Mannheim–Wien–Zürich 1987, 241–254
4. B. Ganter, R. Wille: Formale Begriffsanalyse: Mathematische Grundlagen. B. I.-Wissenschaftsverlag, Mannheim (to appear)
5. M. Kark: Interaktives Zeichnen geometrischer Darstellungen von Begriffsverbänden. Diplomarbeit, FH Darmstadt 1989
6. W. Kollewe, M. Skorsky, F. Vogt, R. Wille: TOSCANA — ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille, M. Zickwolff: Begriffliche Wissensverarbeitung - Grundfragen und Aufgaben. B.I.-Wissenschaftsverlag, Mannheim 1994.
7. R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets.* Reidel, Dordrecht–Boston 1982, 445–470
8. R. Wille: Geometric representations of concept lattices. In: O. Opitz(ed.): *Conceptual and numerical analysis of data.* Springer, Berlin–Heidelberg 1989, 239–255

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

10. R. Wille: Concept lattices and conceptual knowledge systems. In: *Computers & Mathematics with Applications* **23**, 1992, 493–515