

# DimDraw – A Novel Tool for Drawing Concept Lattices

Dominik Dürschnabel, Tom Hanika, and Gerd Stumme

Knowledge & Data Engineering Group, University of Kassel, Germany

{duerrschnabel,hanika,stumme}@cs.uni-kassel.de

**Abstract** Concept lattice drawings are an important tool to visualize complex relations in data in a simple manner to human readers. Many attempts were made to transfer classical graph drawing approaches to order diagrams. Although those methods are satisfactory for some lattices they unfortunately perform poorly in general. In this work we present a novel tool to draw concept lattices that is purely motivated by the order structure.

**Keywords:** Formal Concept Analysis, Diagrams, Lattice-Drawing

## 1 Introduction

Line diagrams are a great tool for interpreting data ordered through formal concept analysis. In such diagrams every concept is visualized as a dot on the plane and the covering order relations are visualized as straight lines that are not allowed to touch other concept dots. Additional to these strong conditions, there are several soft conditions to improve the readability of diagrams for a human reader, for example, to minimize the number of crossing lines or to minimize the number of different slopes. Another desirable condition is to draw as many chains as possible on straight lines. Lastly, the distance of dots to (non-incident) lines should be maximized. Experience shows that in order to obtain a (human) readable drawing one has to balance those criteria. Based on this idea, there are algorithms and tools that work well on order diagrams. However, such automated drawings usually cannot compete to those created manually by an experienced human. Since such an expert is often not available, the task for creating a suitable (and easily readable) line diagram for a given concept lattice is a challenging task.

Here we step in with our novel tool **DimDraw**. We claim that it produces drawings that come close to the quality of hand-drawn line diagrams. In contrast to prior approaches it tries not only to optimize for some set of criteria but also employs the order structure of the concept lattice itself. More precisely, our tool computes a family of linear orders, called *realizer*, that entails the to-be-drawn

---

Authors are given in alphabetical order. No priority in authorship is implied.

Copyright © 2019 for this paper by its authors. Copying permitted for private and academic purposes.

lattice order as its intersection. From this family we derive coordinates for our lattice drawing. In fact, our tool does not only work on concept lattices but also on arbitrary orders.

**Related Work and Basics from Formal Concept Analysis.** We rely on notations from [4], i.e.,  $(G, M, I)$  is a *formal context* with sets  $G$  and  $M$ , and  $I \subseteq G \times M$ . Furthermore,  $\cdot': \mathcal{P}(G) \rightarrow \mathcal{P}(M), A \mapsto A' := \{m \in M \mid \forall g \in A : (g, m) \in I\}$  and  $\cdot': \mathcal{P}(M) \rightarrow \mathcal{P}(G), B \mapsto B' := \{g \in G \mid \forall m \in B : (g, m) \in I\}$  are *derivation operators* enabling  $\mathfrak{B}(G, M, I) = \{(A, B) \subseteq \mathcal{P}(G) \times \mathcal{P}(M) \mid A' = B \wedge B' = A\}$ , the set of all *formal concepts*. This set can be ordered through the inclusion order on  $G$ . Various approaches for drawing line diagrams can be found: The author in [5] employs an order approach which lays the foundation for our work. In [2] a rank function and a force-directed approach was used and the author in [3] focuses on additive diagrams.

## 2 Order Dimension Approach

Before introducing the algorithm behind `DimDraw` we need to recollect some basic notions on order dimension. Let  $(X, \leq_P)$  be an ordered set, i.e.,  $\leq_P \subseteq X \times X$  is an order relation. Two elements  $x, y \in X$  are called *incomparable* in  $P$ , if neither  $x \leq_P y$  nor  $y \leq_P x$ , otherwise they are called *comparable*. If  $(X, \leq_L)$  is an ordered set with  $\leq_P \subseteq \leq_L$ , such that no two elements are incomparable in  $\leq_L$ , then  $\leq_L$  is called a *linear extension* of  $\leq_P$ . The *intersection* of two order relations  $\leq_{L_1}$  and  $\leq_{L_2}$  on  $X$  is also an order relation on  $X$ . Let  $\mathcal{L}$  be a set of linear extensions of  $\leq_P$  with  $\bigcap_{\leq_L \in \mathcal{L}} \leq_L = \leq_P$ . Then  $\mathcal{L}$  is called a *realizer* of  $\leq_P$ . The *order dimension* of an order relation  $\leq_P$  is the cardinality of a minimal realizer of  $\leq_P$ .

A *Ferrers relation* is a relation  $F \subseteq G \times M$ , such that for all  $g, h \in G$  and  $m, n \in M$  the following condition holds:  $(g, m) \in F$  and  $(h, n) \in F \Rightarrow (g, n) \in F$  or  $(h, m) \in F$ . Following [4, Proposition 103] we know that  $F \subseteq G \times M$  is a Ferrers relation if and only if  $\mathfrak{B}(G, M, F)$  is a chain, i.e., all elements of  $\mathfrak{B}(G, M, F)$  are pairwise comparable. Those chains are essential for our drawing approach. The *Ferrers dimension* of a formal context is the smallest number  $k$ , such that there exists a set of  $k$  Ferrers relations with their intersection being  $I$ . The Ferrers dimension of a context is equal to the order dimension of its concept lattice, see [4, Theorem 46]. We further know that the order dimension of  $\mathfrak{B}(G, M, I)$  for some context  $(G, M, I)$  is at most  $d$ , if there are Ferrers relations  $F_1, \dots, F_d \subseteq G \times M$  with  $G \times M \setminus I = \bigcup_{i=1}^d F_i$ . This fact gives a handy way to compute the order dimension of a concept lattice. One has to cover all empty cells of the cross table of a concept with Ferrers relations. Note that those do not have to be disjoint. Unfortunately deciding the order dimension and the Ferrers dimension are  $\mathcal{NP}$ -complete if the dimensions are three or higher.

**Computing Coordinates using a Realizer.** First we have to compute a minimal realizer of the order relation of the concept lattice. This can be done with an algorithm described in [6]. Note that the algorithm described there is only exact for certain ordered sets. Alternatively one can use the Ferrers dimension: Try to cover all the empty spaces in a cross-table by as few Ferrers relations as possible. Then take the inverse Ferrers relations, compute the chains given by these relations and extend them to cover every element of  $\mathfrak{B}(G, M, I)$ .

Given a linear extension of a lattice, let the *position* of each concept in the linear extension be the cardinality of the set of sub-concepts. Embed the concept lattice into  $\mathbb{R}^d$  with  $d$  being its order dimension. The coordinates of concept  $C$  are given as  $(c_1, \dots, c_d)$ , such that  $c_i$  is the position of concept  $C$  in the linear extension  $L_i$ . Note that for each pair of concepts  $X, Y$  it holds that  $X$  is a sub-concept of  $Y$ , if and only if, for the coordinates  $X = (x_1, \dots, x_d)$  and  $Y = (y_1, \dots, y_d)$ , it holds that  $x_i \leq y_i$  for all  $1 \leq i \leq d$ . By rotating the resulting embedding we obtain a valid embedding of the concept lattice in  $\mathbb{R}^d$ . We have to carefully project this embedding to the plane to obtain our drawing. The theory we developed for the projection is based on work of [1]. It is very extensive and will be presented in a later work due to space constraints.

**Example.** Take the “Life in water” context from [4], and consider the three Ferrers-relations that cover all free spaces in the cross table:

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

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

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

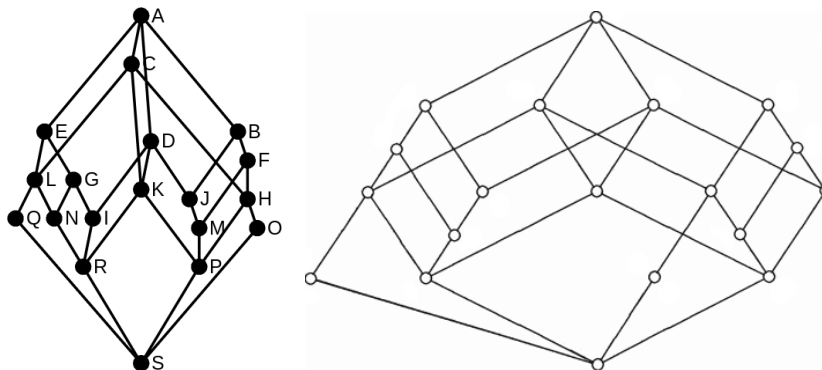
	a	b	c	d	e	f	g	h	i
1	x	x	3	3	3	x			
2	x	x	3	3	3	x	x		
3	x	x	3	3	3	x	x		
4	x	3	3	3	3	x	x	x	
5	x	x	x	3	x				
6	x	x	x	x					
7	x	x	x	x					
8	x	x	x	3	x				

These Ferrers relations give rise to the following (not unique) realizer:

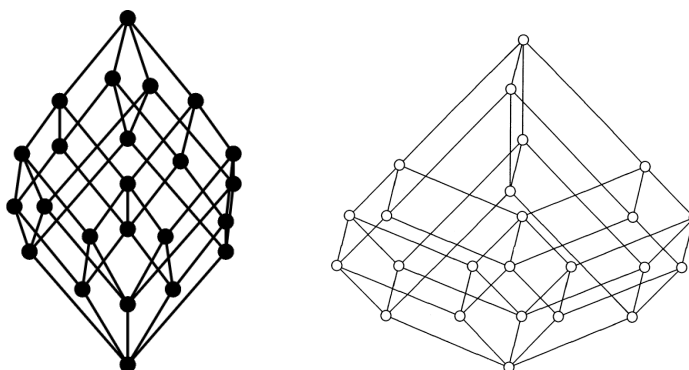
- $S \leq_1 Q \leq_1 O \leq_1 R \leq_1 P \leq_1 N \leq_1 H \leq_1 L \leq_1 K \leq_1 C \leq_1 M \leq_1 I \leq_1 G \leq_1 F \leq_1 J \leq_1 E \leq_1 B \leq_1 D \leq_1 A$
- $S \leq_2 P \leq_2 O \leq_2 M \leq_2 J \leq_2 H \leq_2 F \leq_2 B \leq_2 R \leq_2 K \leq_2 I \leq_2 D \leq_2 N \leq_2 G \leq_2 Q \leq_2 L \leq_2 E \leq_2 C \leq_2 A$
- $S \leq_3 R \leq_3 Q \leq_3 N \leq_3 I \leq_3 L \leq_3 G \leq_3 E \leq_3 P \leq_3 M \leq_3 K \leq_3 J \leq_3 D \leq_3 O \leq_3 H \leq_3 F \leq_3 C \leq_3 B \leq_3 A$

Note that this context is in fact 3-dimensional as the elements  $\{E, C, D, L, I, K\}$  form the ordered set  $S_3$ , which is well-known to be 3-dimensional. Now we embed the concepts based on their positions in the linear extensions as described above: A(18,18,18), B(16,7,17), C(9,17,16), D(15,16,7), E(15,16,7), F(13,6,15), G(12,13,6), H(6,5,14), I(11,10,4), J(14,4,11), K(8,9,10), L(7,15,5), M(10,3,9), N(5,12,3), O(2,2,13), P(4,1,8), Q(1,14,2), R(3,8,1), and S(0,0,0). Now the remaining task is to project this into the plane, which is a research task for itself.

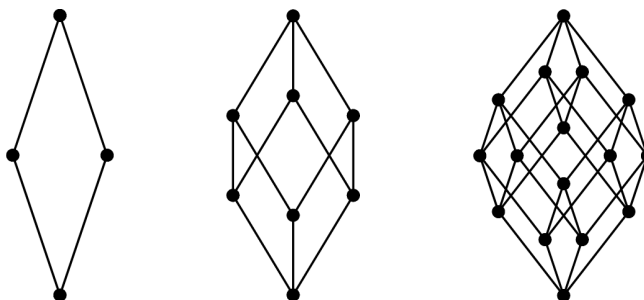
We obtain the following embedding (left) which is quite close to the hand drawn version (right):



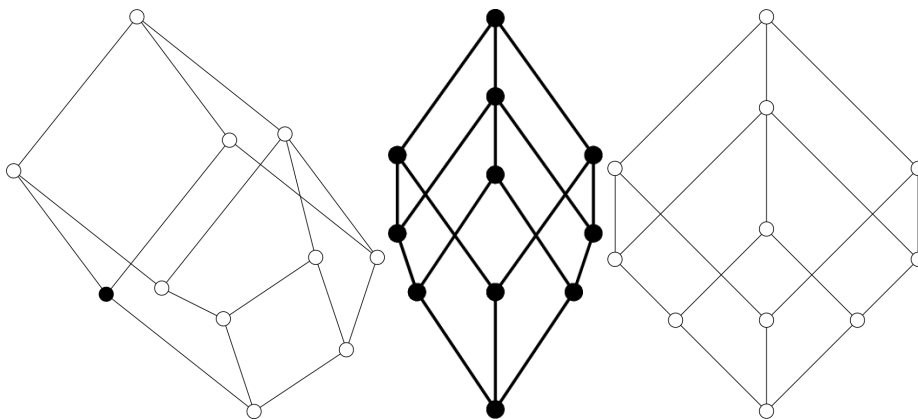
Another example, that is also taken from [4] is the “Drive concepts for motorcars” context. Here the embedding generated with our tool (left) is further away from the hand-drawn version (right), however it is again easy to read:



As another example we present the *contra nominal scales* on  $n$  objects and attributes, i.e.,  $([n], [n], \neq)$  where  $[n] := \{1, 2, \dots, n\}$ . It is easy to see that these exhibit order dimension  $n$ . We show line diagrams of the concept lattices computed by DimDraw for the two, three, and four dimensional contra nominal scales:



We conclude this section by comparing our tool to previous results in [3]. There the author employed an approach to generate additive line diagrams (left). We may note that our drawing (middle) is not additive, however, it resembles the hand drawn additive diagram (right) more accurately.



**Outlook** In this work we demonstrated a novel approach for drawing lattice diagrams automatically, which resulted in the work-in-progress software `DimDraw`. This tool will be part of the upcoming version of `conexp-clj` (<https://github.com/exot/conexp-clj>). The ongoing research for more sophisticated projection theories, as used by this tool, will be extended in future publications.

## References

- [1] S. Felsner and K. Reuter. “The Linear Extension Diameter of a Poset.” In: *SIAM Journal on Discrete Mathematics* 12.3 (Jan. 1999), pp. 360–373.
- [2] R. Freese. “Automated Lattice Drawing.” In: *Concept Lattices*. Ed. by P. Eklund. Vol. 2961. LNCS. Berlin/Heidelberg: Springer, 2004, pp. 112–127.
- [3] B. Ganter. “Conflict Avoidance in Additive Order Diagrams.” In: *Journal of Universal Computer Science*, 10.8 (2004), pp. 955–966.
- [4] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Berlin, 1999, pp. x+284.
- [5] R. Wille. “Lattices in Data Analysis: How to Draw Them with a Computer.” In: *Algorithms and Order*. Ed. by Ivan Rival. Dordrecht: Springer Netherlands, 1989, pp. 33–58.
- [6] J. Yáñez and J. Montero. “A poset dimension algorithm.” In: *Journal of Algorithms* 30.1 (1999), pp. 185–208.