Automated Lattice Drawing

Ralph Freese

University of Hawaii, Honolulu, HI 96822, USA,
ralph@math.hawaii.edu,
WWW home page: http://www.math.hawaii.edu/~ralph/

Abstract. Lattice diagrams, known as Hasse diagrams, have played an ever increasing role in lattice theory and fields that use lattices as a tool. Initially regarded with suspicion, they now play an important role in both pure lattice theory and in data representation. Now that lattices can be created by software, it is important to have software that can automatically draw them.

This paper covers:

- The role and history of the diagram.
- What constitutes a good diagram.
- Algorithms to produce good diagrams.

Recent work on software incorporating these algorithms into a drawing program will also be covered.

An *ordered set* $\mathbf{P} = (P, \leq)$ consists of a set P and a partial order relation $\leq$ on P. That is, the relation $\leq$ is reflexive ($x \leq x$), transitive ($x \leq y$ and $y \leq z$ imply $x \leq z$) and antisymmetric ($x \leq y$ and $y \leq x$ imply $x = y$). If P is finite there is a unique smallest relation $\prec$, known as the *cover* or *neighbor* relation, whose transitive, reflexive closure is $\leq$. (Graph theorists call this the transitive reduct of $\leq$.) A *Hasse diagram* of $\mathbf{P}$ is a diagram of the acyclic graph $(P, \prec)$ where the edges are straight line segments and, if $a < b$ in $\mathbf{P}$, then the vertical coordinate for a is less than the one for b. Because of this second condition arrows are omitted from the edges in the diagram.

A *lattice* is an ordered set in which every pair of elements a and b has a least upper bound, $a \vee b$, and a greatest lower bound, $a \wedge b$, and so also has a Hasse diagram.

These Hasse diagrams[1] are an important tool for researchers in lattice theory and ordered set theory and are now used to visualize data.

This paper deals the special issues involved in such diagrams. It gives several approaches that have been used to automatically draw such diagrams concentrating on a three dimension force algorithm especially adapted for ordered sets that does particularly well.

We begin with some examples.

[1] In the second edition of his famous book on lattice theory [3] Birkhoff says these diagrams are called Hasse diagrams because of Hasse's effective use of them but that they go back at least to H. Vogt, *Résolution algébrique des équation*, Paris, 1895.

1 Some Examples

Generally lattice theorists have a fairly precise idea of how their diagrams should look. Figure 1 gives an example.

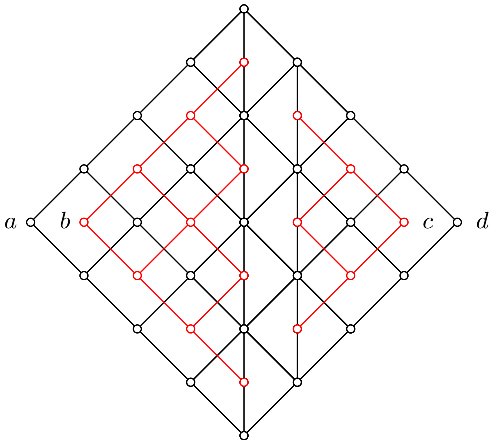

Fig. 1. The diagram of a small modular lattice

Of course the reflection of this diagram around the vertical axis would be considered equally acceptable but Figure 2 is another possible diagram of the same lattice.

While any permutation of the elements of a particular level would produce a correct representation of the order, almost any such permutation would make the diagram unrecognizable.

Most lattice theorists would draw the lattice of subsets of a three element set as the cube drawn on the left of Figure 3. Orthomodular lattice theorists often prefer to draw it as in the center but no lattice theorist would draw it as is done on the right.

Thus while a randomly chosen diagram will be mathematically correct, it is very likely to be useless. Prior to lattices being generated by computer programs this was not a problem: mathematicians knew (at least in most cases) how a particular lattice should be drawn and drew it that way.

2 A Brief History of the Diagram[2]

[2] Pseudo-history might be a better description of this section. The author's mathematical great grand parent is Eric Temple Bell who is known for his colorful, albeit not entirely accurate, accounts of the history of mathematics; see [18].

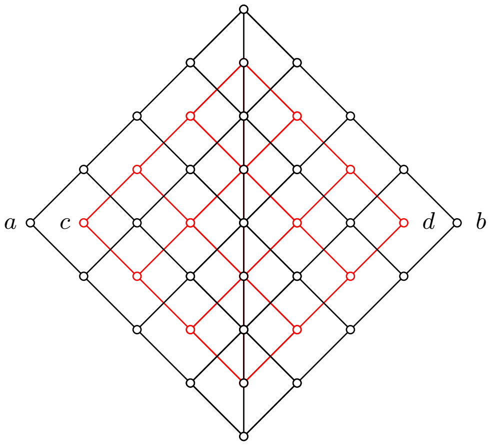

Fig. 2. A different diagram of the lattice of Figure 1

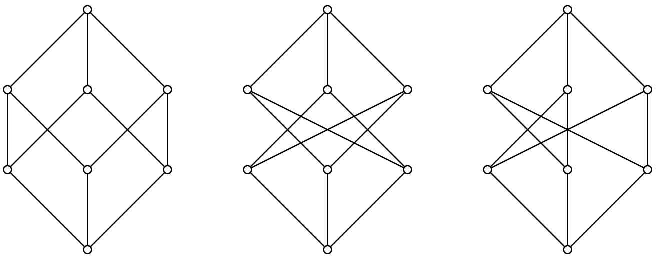

Fig. 3. The lattice of subsets of a three element set. Left: the usual diagram; Center: as sometimes drawn by orthomodular lattice theorists; Right: an ugly version

As mentioned earlier lattice diagrams are often called Hasse diagrams in honor of H. Hasse who used them effectively. Apparently the earliest lattice theorists such as R. Dedekind did not use diagrams to represent lattices. They began to be used in the 1930's but more as a tool for discovering new results; they rarely appeared in the literature.

The free modular lattice with three generators (which is drawn in Figure 8) was first described by Dedekind in [7]. O. Ore [17] gives something resembling a diagram by placing the elements of the free modular on the page in an arrangement reflecting the order but no lines are drawn. Ore, in connection with his work on direct decompositions of algebraic systems, asked R. P. Dilworth if a, b, c and d are elements of a modular lattice such that every pair meets to the least element and all pairs join to the top except possibly the pairs a, b and c, d, do all pairs join to the top? Dilworth drew a diagram to show this was false; see Figure 4. Ore, like many lattice theorists of his era, was skeptical of diagrams and asked Dilworth to find this lattice as a sublattice of the lattice of normal subgroups of a group.

Another interesting thing about this diagram was that Dilworth used both red and black to draw it. (If you are reading a print version of this paper, you can see the color version at the author's web site.) This use of color way back in the 1930's makes the diagram much easier to understand and better than the diagrams in journals even today.

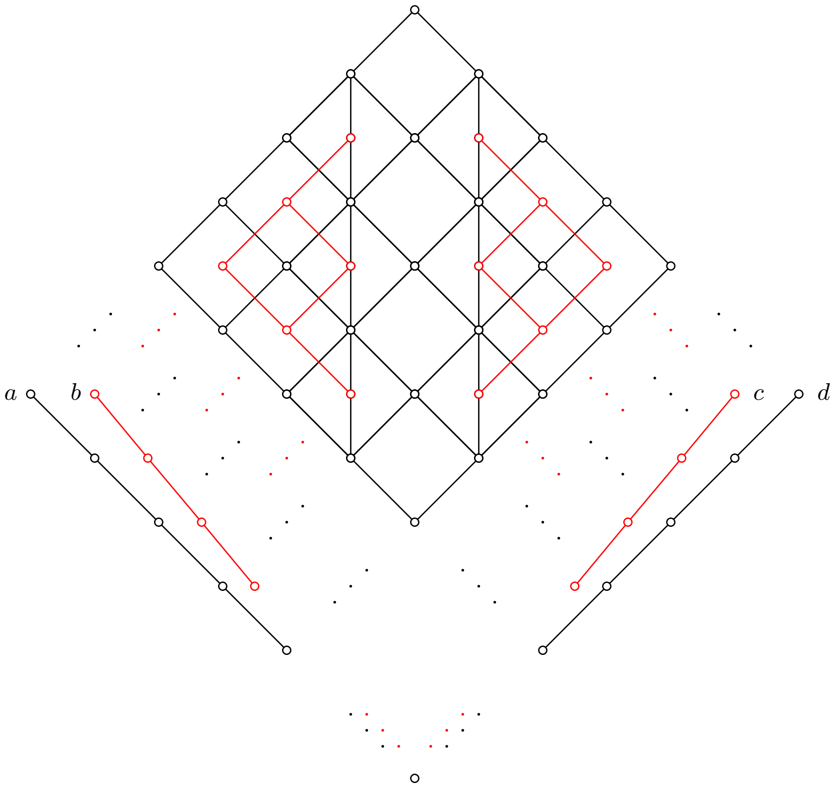

Fig. 4. A counter-example to Ore's question

A. Day, C. Herrmann, and R. Wille [4] constructed a lattice much like this one in their important study of four generated modular lattices. It shows, for example, that $\mathbf{M}_4$, the six element lattice with a least and greatest element and 4 incomparable elements, is not a projective modular lattice.

In [8] Dilworth cites as an example J. E. McLaughlin paper [16] in which he proves that an atomic lattice with unique comparable complements must be modular. (See [1] for a history of this important lattice theory problem.) As was often done at that time, McLaughlin used diagrams in finding his proof but omitted them from his paper. Dilworth's paper shows how much more understandable the proof can be with the use of diagrams.

By the 1970's diagrams had become an important tool in lattice theory. The appendix has several diagrams that have played an important role in research, biased toward my own work. Figure 8 shows the free modular lattice on three generators, $\mathbf{FM}(3)$, and the lattice freely generated by $\mathbf{M}_3$ and two extra ordered elements below one of the generators. Both are used repeatedly in the study of modular lattices.

Figure 10 shows the free modular lattice $\mathbf{FM}(\mathbf{2}+\mathbf{2})$ and free lattice $\mathbf{FL}(\mathbf{2}+\mathbf{2})$ generated by two 2-element chains. Of course the sublattice generated by two 2-element chains in any lattice must be a homomorphic image of $\mathbf{FL}(\mathbf{2}+\mathbf{2})$ and thus these lattices are a useful tool in lattice theory. It is interesting that $\mathbf{FL}(\mathbf{2}+\mathbf{2})$ was described by R. A. Dean in [6] but it was H. L. Rolf who diagrammed the lattice in [23] and since then it is known as Rolf's lattice.

The lattice on the left of Figure 11 is J. B. Nation's semidistributive lattice which is not a bounded image of a free lattice (answering a question of R. McKenzie). Nation and I constructed the lattice on the right of Figure 11 from this lattice using Alan Day's doubling construction. We knew this lattice had certain properties and hoped it had certain additional properties but we were having trouble proving that it did. To our surprise we found we were actually able to diagram this lattice. Once we had the diagram of the lattice, it was clear that it had the properties we wanted.

Figure 9 diagrams the congruence lattice of the free algebra on one generator in Polin's variety. This diagram played a critical role in the characterization of varieties with modular congruence lattice of [5].

3 Automatic Drawing

In the early 1980's J. B. Nation and I were working on free lattices. The algorithms for deciding equality of terms and for finding the canonical form of elements were tractable; that is, polynomial time, but rather unpleasant for humans. So we wrote programs to do this (mostly in Lisp). Associated with each element of a free lattice is a finite lattice that determines whether or not the element has lower covers (lower neighbors). For example the lattices associated with $(x \vee (y \wedge z)) \wedge (y \vee z)$ and with $(x \vee (y \wedge z)) \wedge (y \vee (x \wedge z))$ are diagrammed in Figure 5. (Since the second lattice is semidistributive but the first is not, $(x \vee (y \wedge z)) \wedge (y \vee z)$ has a lower cover but $(x \vee (y \wedge z)) \wedge (y \vee (x \wedge z))$ does not; see [10]).

I had programs to automatically calculate these lattices, but it did not give a diagrams. It just gave a list of the elements and covers. It is much more difficult to understand the lattice from this sort of data, just as in Formal Concept Analysis lattices help immensely in visualizing data. So I wrote a program to produce a lattice diagram. But was disappointed because even small, well known lattices were usually unrecognizable. For example the eight element Boolean algebra,

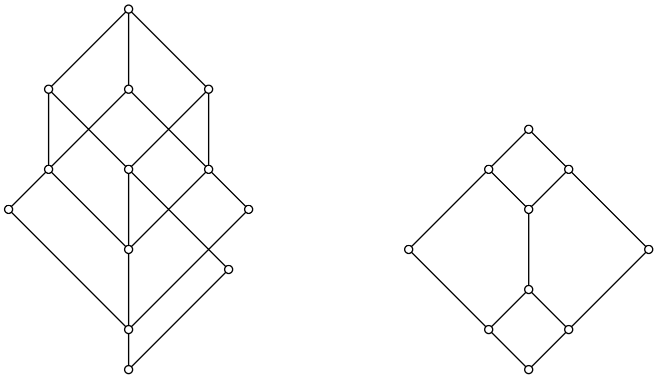

Fig. 5. The lattices $\mathrm{L}((x \vee (y \wedge z)) \wedge (y \vee z))$ and $\mathrm{L}((x \vee (y \wedge z)) \wedge (y \vee (x \wedge z)))$

drawn on the left of Figure 3 often looked like the center or right drawing. Clearly something better was needed.

4 Our Algorithm

Our algorithm is based on a combination of a mathematical rank function to determine the height and a modification of the "forces" method of graph theory.

The program first calculates the rank function on the ordered set and uses this to determine the height of the elements. It then places the points in three space using the rank for the height, i.e., for the z-coordinate. The points of the same rank are arranged around a circle on a plane parallel to the x–y plane. We now place imaginary forces on the elements. Comparable elements are attracted to each other while incomparable elements are repulsed.

These forces are applied several times in three phases. In the first phase the repulsive force is set to be very strong; in the second phase the attractive force is strong; and in the final phase the forces are balanced. Then an optimal projection to two space is chosen (either by the program or the user).

4.1 The Rank Function

As mentioned above the algorithm positions the elements of the ordered set $\mathbf{P}$ in 3-space and uses forces to adjust the x and y coordinates but the z coordinate (the height) is determined by a rank function on the ordered set given by

$$\mathrm{rank}(a) = \mathrm{height}(a) - \mathrm{depth}(a) + M$$

where $\mathrm{height}(a)$ is the length of the longest chain from a to a minimal element, $\mathrm{depth}(a)$ is the length of the longest chain from a to a maximal element. M can be any constant; it is usually the length of the longest chain in $\mathbf{P}$ so that the rank of the least element of a lattice is 0. Figure 6 gives an example of this rank function. Notice the rank function has a nice top to bottom symmetry.

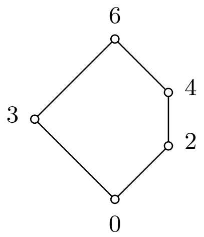

Fig. 6. The rank function for N_5

The usual algorithm to find a linear extension of $\mathbf{P}$ (graph theorists call this *topological sorting*) can be easily modified to calculate this rank function in linear time. (See Chapter 11 of [10] for a discussion of algorithms for ordered sets and lattices.)

4.2 Initialization and Force Scheme

We associate a point (x, y, z) in 3–space with each element of the ordered set. The z coordinate is determined by the rank function. Initially the points of the same rank are arranged with equal spacing (actually a slight perturbation is added to avoid unstable equilibria) around a circle on a plane parallel to the x–y plane with radius equal to the number of elements of that rank.

Then forces (with heavy damping—imagine the points lying in a thick syrup) are repeatedly applied to the points. Each point is attracted to the points it is comparable with. If the point has coordinates (x_0, y_0, z_0) and is comparable with a point with coordinates (x_1, y_1, z_1) the force on the first point is

$$c_{\mathrm{att}}\langle x_1 - x_0, y_1 - y_0, 0\rangle$$

If these two points are incomparable they are repulsed. The force on the first point is

$$\frac{c_{\mathrm{rep}}\langle x_0 - x_1, y_0 - y_1, 0\rangle}{|(x_1 - x_0)|^3 + |(y_1 - y_0)|^3 + |(z_1 - z_0)|^3}$$

Note the attraction of two comparable elements does not depend on the z–coordinate while the repulsion does. In both cases the force is in the x–y plane so that the z–coordinate remains fixed. The attraction is similar to the attraction of a spring (Hooke's Law) with natural length 0. But since the z–coordinate is unchanged the distance between the two points is always at least $|z_1 - z_0|$. Also note that the attraction is not just between elements and their upper and lower covers (neighbors) but between all pairs of comparable elements and does not depend on $z_1 - z_0$. The repulsion force uses an inverse square law and it does depend on the z–coordinate.

4.3 Iteration

The total force on each node is calculated and all of the nodes are moved. This process is iterated until the diagram becomes stable. Of course the attraction and repulsion can be modified by changing c_{att} and c_{rep} and the current algorithm goes through three stages: first with the repulsion force strong (that is c_{rep} large), then with the attraction force strong and finally with balanced forces.

Finally a projection of the form $\langle x, y, z\rangle \mapsto \langle x \cos\theta + y \sin\theta, z\rangle$ into the plane is chosen. θ can be chosen to maximize some niceness criterion or the user can rotate the picture to find a pleasing projection.

4.4 Comparisons

While the results of this algorithm do not match the quality of hand drawn diagrams they are at least recognizably the same and often close to being æthetically pleasing. For lattices with 10 or less elements the result is usually very close to the hand drawn version.

To illustrate the difference we have taken three figures (Figure 1.1, Figure 1.17, and Figure 1.25) from [11], and reproduced them along with the diagram drawn by the program in Figures 12, 13, and 14. The first two, while far from perfect, are easily recognized. The third one, Figure 14 is particularly interesting because the computer generated diagram is actually the way it is usually drawn; see, for example, Figure 1 of Knuth [15]. It better shows off the symmetry of the lattice. The diagram from [11] also has some obvious pluses.

5 What is a 'Nice' Diagram?

There is a large body of work on graph drawing. The Hasse diagram is a very restricted type of graph diagram: the edges must be straight lines and vertical coordinate must agree with the order (greater elements of the ordered set must be higher in the diagram). Since our goal is to really picture the lattice, we rarely diagram lattices with more than 50 elements so asymptotic results are not always relevant. Nevertheless there are results of interest to lattice drawing. We discuss one: minimizing the number of edge crossings.

5.1 Edge Crossings

Finding a diagram with the minimum number of edge crossings is NP–hard as was shown by Garey and Johnson [13]. In [9] P. Eades and N. Wormald showed that the problem of minimizing the edge crossings in a bipartite graph where one of the layers has a prescribed order is also NP–hard. It is easy to see that these results imply that the problem of minimizing crossings in an ordered set is NP–hard. However this does not immediately imply that the crossing problem for lattices is hard.[3] It was Ivan Rival who brought the crossing problem for lattice to my attention. While the result of Garey and Johnson can modified in a straight-forward way to apply to lattices, extending Eades and Wormald's result takes more care. We outline how to do it here. Stating the problem:

Decision Crossing Problem for Lattices (DCPL)

Instance: A lattice **L**, an ordering on the atoms of **L** and an integer M.

Question: Is there a diagram of **L**, that is, a map from the elements of **L** into the plane, such that the vertical coordinates are given by the rank function and the left to right order of the atoms determined by the horizontal coordinate is the given ordering of the atoms and such that the number of crossings is at most M?

Theorem 1. DCPL *is* NP*–complete.*

Proof. The problem is clearly in the class NP. The proof that it is NP–complete follows the proof of Theorem 1 of [9] so we will only point out how to modify that theorem to prove this theorem. To prove this problem is NP–complete we give a polynomial-time reduction to the following known NP–complete problem: given a directed graph **D** and a positive integer K is there a set of at most K arcs whose removal would make **D** acyclic? This is known as the Feedback Arc Set problem and is NP–complete; see p. 192 of [12].

So let $\mathbf{D} = (U, B)$ be a directed graph with vertex set $U = \{u_1, \ldots, u_n\}$ and arc set B and let K be a positive integer. For each arc $a = (u_r, u_s) \in B$ let $C(a)$, let the "clump" associated with a be the set

$$C(a) = \{c_1^a, c_2^a, c_5^a, c_6^a\} \cup \{c_{3,i}^a, c_{4,i}^a : 1 \leq i \leq n,\ i \neq r, s\}.$$

The atoms of **L** consist of the union of the $C(a)$'s. The left to right order of these atoms keeps the elements of each clump together and within a clump the order is the order on the first index and within the $c_{3,i}^a$'s on the second index but for the $c_{4,i}^a$'s it is the *reverse* order of the second index.

The coatoms of **L** are the elements of U. If $a = (u_r, u_s) \in B$ then in **L**

$$\begin{aligned} c_1^a, c_5^a &< u_r \\ c_2^a, c_6^a &< u_s \\ c_{3,t}^a, c_{4,t}^a &< u_t \qquad \text{for } t \neq r, s \end{aligned}$$

Since each atom has exactly one upper cover under this order, **L** is a lattice once we add a least and greatest element.

Let $x(u_i)$ be the x-coordinate of u_i in a Hasse diagram of **L**. This determines a (left to right) order on the coatoms since the rank of all the coatoms is the

[3] Satisfying the lattice axioms places strong restrictions on a ordered set. While the diagram of an ordered set with n elements can have as many as $n^2/4$ edges, that of a lattice can have at most $n^{3/2}$; see [10].

same. Let B' denote $\{(u_i, u_j) \in B : x(u_i) > x(u_j)\}$. Letting $\beta = |B|$, the number of crossings in the diagram is

$$4\binom{\beta}{2}\binom{n}{2} + 2\beta\binom{n-2}{2} + 4\beta(n-1) + \beta + 2|B'|$$

The proof of this is similar to the proof of Lemma 1 of [9]. For example, if $a = (u_r, u_s)$ and $\{i, j\} \cap \{r, s\} = \emptyset$ then, because of the reverse order of the $c^a_{4,k}$'s, there there are 2 crossings of the 4 edges from u_i and u_j into $C(a)$ regardless of whether $i < j$ or not. These crossings give rise to the $2\beta\binom{n-2}{2}$ term.

Of course the graph $(U, B - B')$ has a linear extension and so is acyclic. Thus there is a set of K arcs of $\mathbf{D}$ whose removal makes it acyclic if and only there is a left to right ordering of the coatoms of $\mathbf{L}$ so that the diagram of $\mathbf{L}$ with this layout of the coatoms has at most

$$4\binom{\beta}{2}\binom{n}{2} + 2\beta\binom{n-2}{2} + 4\beta(n-1) + \beta + 2K$$

crossings. Thus DCPL is NP–complete.

5.2 Heuristic Niceness

Of course I am more interested in a useful tool than in theoretical results about diagrams (which is also the goal of FCA) and so tend to concentrate on heuristics. What constitutes a "nice" diagram is, of course, a question of æthetics, not mathematics. R. Wille and his colleagues at Darmstadt suggested various criteria. One was minimizing the number of different slopes of the edges. Looking at the diagrams in this paper we see that this is very important. But it is interesting to note that some minimum slope diagrams can be bizarre. Figure 7 from [25] shows an Escher-like diagram of the 8-element Boolean algebra.

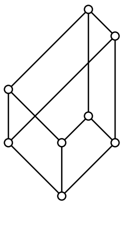

Fig. 7. An Escher-like diagram of the eight element Boolean algebra.

In [11] this idea is refined into two rules: the *rule of parallelograms* and the *rule of lines*. While the algorithm we presented does not follow these rules, it often produces diagrams that approximate them.

In [2] the authors present an algorithm for minimizing total length of the edges when the elements are restricted to lie on a grid which produces nice results.

6 Software

Our algorithm was originally implemented in Lisp, but in 1996 we converted it to a Java applet; see `http://www.math.hawaii.edu/∼ralph/LatDraw/`. It has been used as a component in several other programs:

- JavaMath, a web based application using the applet to draw lattice (particularly subgroup lattices) directly from Maple and GAP.
- Universal Algebra Calculator, an application for universal algebra incorporating our program to draw congruence lattices.
- JaLaBA, Java Lattice Building Application. This allows you to enter your own formal context and it will generate the lattice.

There are more details on the site above.

We are presently reconstituting our program as a component which can be easily plugged into other programs. The goals of the new program include:

- Bring the Java up-to-date: the original applet was written in the first Java (version 1.02) and as such uses the old event model and does not use Graphics2D.
- Add new, often requested, features such as dragging the elements, interval viewing, saving in various formats, printing.
- Two way communication between the component and the client program using it: the program would be aware of events in the component so, for example, it could display information about a node when the user clicks it. In the other direction, the program can request that the component change the color of certain elements in the lattice based on a user's gesture in the program. At the programming level, there will be higher level events so, for example, the program will be able to listen for higher events such as "vertex clicked" or "edge clicked" rather than just mouse clicked.

I plan is to make this open source and to put it on sourceforge.net. Sample programs using the drawing component will be included.

References

1. M. E. Adams, *Uniquely complemented lattices*, The Dilworth Theorems, Selected Papers of Robert P. Dilworth (K. Bogart, R. Freese, and J. Kung, eds.), Birkhäuser, Basel, 1990, pp. 79–84.
2. A. Aeschlimann and J. Schmid, *Drawing orders using less ink*, Order **9** (1992), 5–13.
3. G. Birkhoff, *Lattice theory*, Amer. Math. Soc., Providence, R. I., 1948, rev. ed., Colloquium Publications.
4. A. Day, Ch. Herrmann, and R. Wille, *On modular lattices with four generators*, Algebra Universalis **2** (1972), 317–323.
5. Alan Day and Ralph Freese, *A characterization of identities implying congruence modularity, I*, Canad. J. Math. **32** (1980), 1140–1167.

6. R. A. Dean, *Completely free lattices generated by partially ordered sets*, Trans. Amer. Math. Soc. **83** (1956), 238–249.
7. R. Dedekind, *Über die von drei Moduln erzeugte Dualgruppe*, Math. Annalen **53** (1900), 371–403.
8. R. P. Dilworth, *The role of order in lattice theory*, Ordered Sets (I. Rival, ed.), D. Reidel, Dordrecht, Holland, 1982, Proc. of the Banff Symposium on Ordered Sets.
9. P. Eades and N. Wormald, *Edge crossings in drawings of bipartite graphs*, Algorithmica **11** (1994), no. 4, 379–403.
10. Ralph Freese, Jaroslav Ježek, and J. B. Nation, *Free lattices*, Amer. Math. Soc., Providence, 1995, Mathematical Surveys and Monographs, vol. 42.
11. B. Ganter and R. Wille, *Formal Concept Analysis*, Springer, Berlin, Heidelberg, New York, 1999, ISBN 3-540-62771-5.
12. M. Garey and D. Johnson, *Computers and intractability, a guide to the theory of NP-completeness*, W. H. Freeman and Company, San Francisco, 1979.
13. M. R. Garey and D. S. Johnson, *Crossing number is NP-complete*, SIAM J. Algebraic Discrete Methods **4** (1983), no. 3, 312–316.
14. A. Garg and R. Tamassia, *Advances in graph drawing*, Algorithms and Complexity, Springer Verlag, Berlin-New York, 1994, Second Italian Conference, CASC '94, Rome Italy, pp. 12–21.
15. Donald E. Knuth, *The art of computer programming. Vol. 3*, second ed., Addison-Wesley Publishing Co., Reading, Mass., 1998, Searching and Sorting, Addison-Wesley Series in Computer Science and Information Processing.
16. J. E. McLaughlin, *Atomic lattices with unique comparable complements*, Proc. Amer. Math. Soc. **7** (1956), 864–866.
17. O. Ore, *On the foundations of abstract algebra, I*, Ann. Math. **36** (1935), 406–437.
18. Constance Reid, *The search for E. T. Bell: also known as John Taine*, Math. Assoc. Amer., Washington, D. C., 1993.
19. I. Rival, *The diagram*, Graphs and order, NATO Adv. Sci. Inst. Ser. C: Math. Phys. Sci., vol. 147, Reidel, Dordrecht, Boston, 1985, pp. 103–133.
20. I. Rival, *Reading, drawing, and order*, Algebras and orders, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., vol. 389, Kluwer Acad. Publ., Dordrecht, 1991, pp. 359–404.
21. I. Rival and B. Sands, *Pictures in lattice theory*, Algebraic and geometric combinatorics **65** (1982), 341–355.
22. I. Rival and R. Wille, *Lattices freely generated by partially ordered sets: which can be "drawn"?*, J. Reine Angew. Math. **310** (1979), 56–80.
23. H. L. Rolf, *The free lattice generated by a set of chains*, Pacific J. Math. **8** (1958), 585–595.
24. M. Skorsky, *Endliche Verbände—Diagramme und Eigenschaften*, Ph.D. thesis, Technische Hochschule Darmstadt, 1992.
25. J. Stephan, *Liniendiagramme von verbänden*, Technische Hochschule Darmstadt, Darmstadt, 1987, Diplomarbeit.
26. R. Wille, *Lattices in data analysis: how to draw them with a computer*, Algorithms and order (I. Rival, ed.), Univ. Ottawa, 1989, pp. 33–58.

Appendix A: More Diagrams

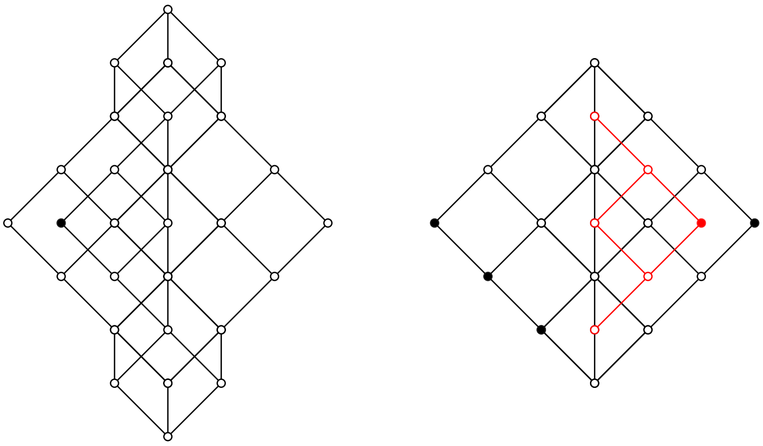

Fig. 8. $\mathbf{FM}(3)$ and the lattice freely generated by $\mathbf{M}_3$ and two extra ordered elements below a generator.

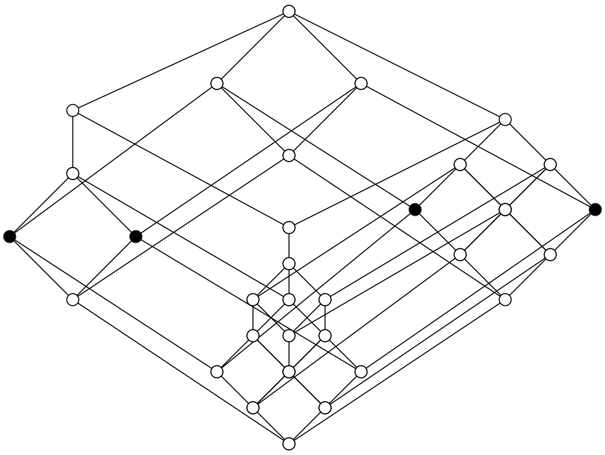

Fig. 9. The congruence lattice of the free algebra on one generator in Polin's variety.

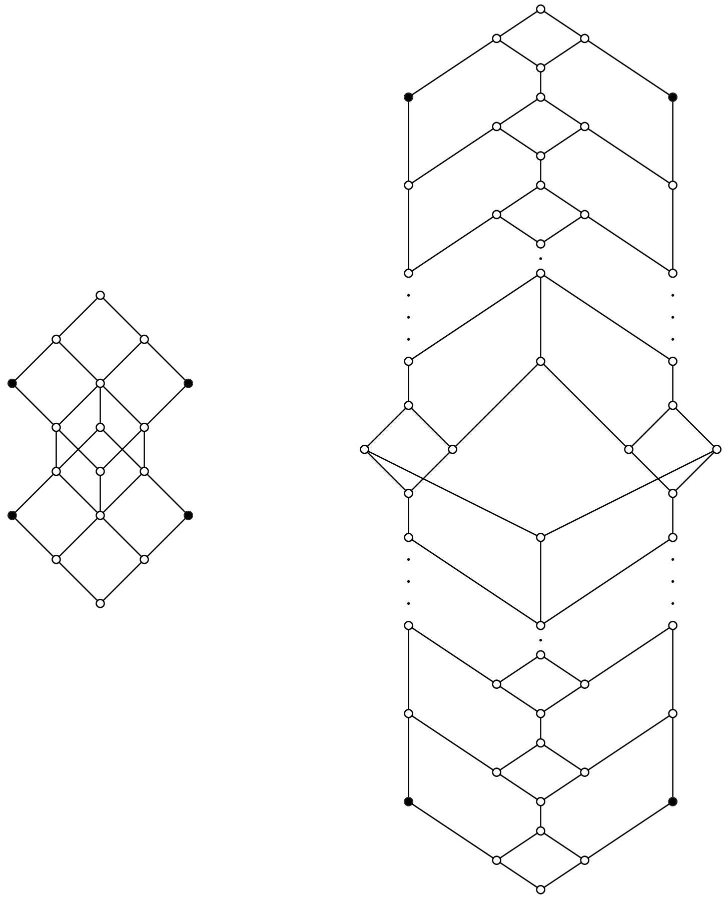

Fig. 10. **FM**(**2** + **2**) and **FL**(**2** + **2**).

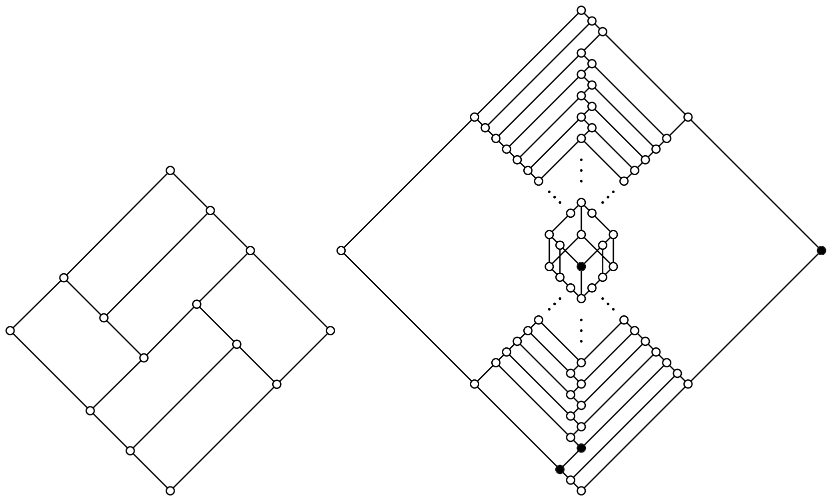

Fig. 11. Nation's example of a finite semidistributive lattice which is not a bounded homomorphic image of a free lattice and its W–cover

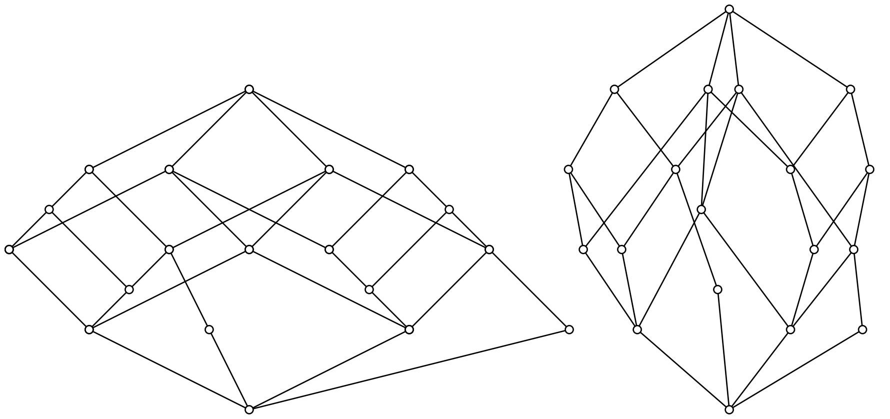

Fig. 12. Left: Figure 1.1 of [11]. Right: as drawn by the program

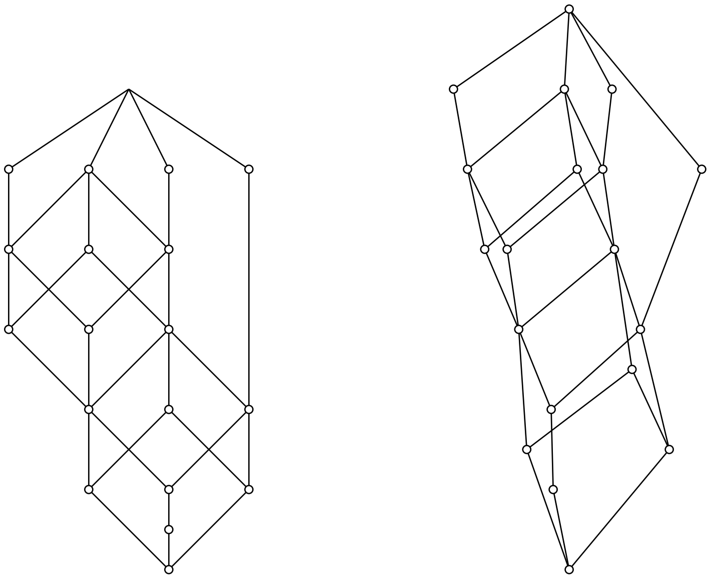

Fig. 13. Left: Figure 1.17 of [11]. Right: as drawn by the program

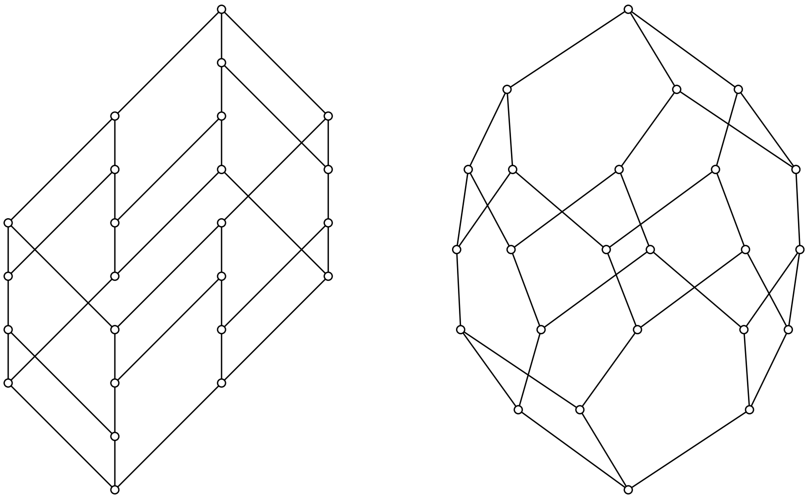

Fig. 14. Left: Figure 1.25 of [11]. Right: as drawn by the program