ALGORITHMS FOR FINITE, FINITELY PRESENTED AND FREE LATTICES

RALPH FREESE

Abstract. In this talk we will present and analyze the efficiency of various algorithms in lattice theory. For finite lattices this will include recognition of various properties such as subdirect irreducibility, semidistributivity, and boundedness (in the sense of McKenzie) as well as efficient algorithms for computing the congruence lattice. For free and finitely presented lattices we will discuss algorithms for such things as finding all the lower and upper covers of an element and recognizing if a finitely presented lattice is finite. Several interesting open problems will be given.

We will also discuss how (computer implementations of) these algorithms have been used to prove results in various branches of lattice theory.

I first became interested in computer programs to help with lattice theory when Nation and I were studying free lattices. Putting a lattice term into (Whitman) canonical form can be tedious, so we developed a program to do this and calculations in free lattices in general. This program was expanded to include finite and finitely presented lattices, congruence lattices, and automatic lattice drawing and proved extremely useful. As part of our book [12] on free lattices we decided to write a chapter on lattice algorithms for free lattices and those aspects of finite lattices related to free lattices. We expanded the chapter to include a general introduction to algorithms for lattices. We found the literature on this subject spread over both computer science and mathematical journals. Algorithms on directed graphs have been extensively developed and many are basic to ordered sets and algorithms from the theory of data bases play a role.

1. Finite Lattices

For a finite lattice $\mathbf{L}$, $\mathrm{J}(\mathbf{L})$ denotes the set of nonzero join irreducible elements and $\mathrm{M}(\mathbf{L})$ the set of nonunit meet irreducible elements. These sets are ordered by the induced order from $\mathbf{L}$. If $a \in \mathrm{J}(\mathbf{L})$ then it has a unique lower cover in $\mathbf{L}$ which we denote by a_* and similarly if $q \in \mathrm{M}(\mathbf{L})$ then q^* is the unique upper cover of q. The cover relation is denote by $a \prec b$; $\mathrm{Cg}\,(x, y)$ is the smallest congruence identifying x and y.

For an ordered set $\mathbf{P}$ we define

$$E_{\leq} = \{\langle a, b\rangle : a \leq b\} \qquad e_{\leq} = |E_{\leq}|$$
$$E_{\prec} = \{\langle a, b\rangle : a \prec b\} \qquad e_{\prec} = |E_{\prec}|.$$

For clarity we sometimes write $E_{\leq}(\mathbf{P})$, etc. Of course $\mathbf{P}$ is determined from its underlying set P and any 'edge' set E with

$$E_{\prec} \subseteq E \subseteq E_{\leq} \tag{1}$$

The transitive, reflexive closure of any such E is $E_{\leq}$ and $E_{\prec}$ is the smallest set whose transitive, reflexive closure is $E_{\leq}$. $E_{\prec}$ is alternately known as the *covering relation*, the *transitive reduct*, and the *Hasse diagram* of $\mathbf{P}$. For a set E satisfying (1) we let $e = |E|$.

A linear extension of an ordered set $\mathbf{P} = \langle P, E\rangle$ can be be found in time $O(n + e)$, that is, linear time; see [20] and [23]. Most algorithms require that P be linearly ordered in order to run efficiently. Note the input size of $\mathbf{P} = \langle P, E\rangle$ is $n + e$; if the Hasse diagram is given as the input, this is $n + e_{\prec}$. While for an ordered set $e_{\prec}$ can be as big as $n^2/4$, a much stronger theorem, proved independently by W. Klotz and L. Lutz [22] and P. Goralčík, A. Goralčíková, V. Koubek, and V. Rödl, [17], holds for lattices:

Theorem 1. *If $e_{\prec}$ is the number of covers in a semilattice with n elements, then*

$$n - 1 \leq e_{\prec} \leq n^{3/2}.$$

Klotz and Lutz used this result as the starting point for obtaining an asymptotic formula for the number of labeled lattices on n elements. Stronger results results were obtained by D. Keitman and K. Winston [21].

Clearly lattices can be represented in space proportional to n^2 and the results on the asymptotic number of lattices show that we could not hope for better than $n^{3/2}$. M. Talamo and P. Vocca [30] and [29] have a representation of lattice of size $O(n^{3/2})$ in which $x < y$ (but not $x \vee y$) can be evaluated in constant time.

The Mac Neille completion of the ordered set of elements which are either join or meet irreducible gives back the lattice. This is the starting point of *concept analysis* and can (but does not always) give a very economical representation for lattices; see [15].

Using Theorem 1 it is possible to decide if an ordered set is a lattice and to construct all the basic data structures such as the $\leq$–relation and the join and meet tables, all in time $O(n^{5/2})$. Is there a faster algorithm?

Problem 1. *Can one decide if an ordered set of size n is a lattice and, if it is, find the basic data structures in time faster than $O(n^{5/2})$?*

For projective planes (as lattices) $e_{\prec}$ is asymptotically $\sqrt{2}/4\ n^{3/2}$, so Theorem 1 cannot be substantially improved. Nevertheless, lattices with $e_{\prec}$ near this bound seem to be rare.

Theorem 2. *Let $\mathbf{L}$ be a lattice with n elements.*

1. *If* **L** *satisfies either semidistributive law then*
$$e_{\prec} \leq \frac{1}{2} n \log_2 n$$
with equality if and only if **L** *is a Boolean lattice.*
2. *If* **L** *has width at most k then*
$$e_{\prec} \leq n(\frac{1}{2} + \sqrt{k}).$$

For projective three spaces $e_{\prec}$ is asymptotically $n^{5/4}$ and for partition lattice it is $O(n(\log_2 n)^2)$.

There are much faster algorithms for for certain restricted classes of lattices. For distributive lattices, at least for the recognition problem. J.-P. Bordat [1] has an $O(e_{\prec} \log \log n)$ algorithm. O. Karpushev improves this to a linear time algorithm. In [18] M. Habib and L. Nourine give fast algorithms for computing in distributive lattices.

2. Upper Covers

Consider Algorithm 1. This is a straightforward algorithm which finds the covering relation from the $\leq$ relation.

```
%   P is topologically sorted.
Q ← P
for a ∈ P do
    pop(Q)
    S ← ∅   %   S will be the upper covers of a.
    for x ∈ Q do
        if a ≤ x then
          T ← S
          while T ≠ ∅ and first(T) ≰ x do
                pop(T)
          endwhile
          if T = ∅ then push(x, S) endif
        endif
    endfor
    UC[a] ← S
endfor
```

Algorithm 1: Finding the Covering Relation

A straightforward analysis shows this runs in $O(ne_{\prec} + n^2)$ time. By Theorem 1 this is $O(n^{5/2})$ for lattices. But with a more ingenious argument we can show the algorithm runs in time $O(n^2)$ for lattices; see [12].

Theorem 3. Algorithm 1 *calculates the upper covers in* **P** *form the* $\leq$*–relation in time* $O(ne_{\prec} + n^2)$ *time. If* **P** *is a lattice, this algorithm runs in time* $O(n^2)$*.*

3. Congruences on Lattices

In this section let $\mathbf{L}$ denote a finite lattice with n elements. $\mathrm{J}(\mathbf{L})$ denotes the set of nonzero, join irreducible elements and $\mathrm{M}(\mathbf{L})$ denotes the nonunit meet irreducible elements of $\mathbf{L}$. If $a \in \mathrm{J}(\mathbf{L})$ and $q \in \mathrm{M}(\mathbf{L})$, we let a_* denote the unique lower cover of a and q^* denote the unique upper cover of q. Recall the following elementary facts about congruence on lattices:

- If $a \prec b$ then $\mathrm{Cg}\,(a, b)$ is join irreducible in $\mathbf{Con\ L}$.
- If $\theta \in \mathrm{J}(\mathbf{Con\ L})$ then $\theta = \mathrm{Cg}\,(a, a_*) = \mathrm{Cg}\,(q, q^*)$ for some $a \in \mathrm{J}(\mathbf{L})$ and some $q \in \mathrm{M}(\mathbf{L})$.

Thus the map $a \mapsto \mathrm{Cg}\,(a, a_*)$ maps $\mathrm{J}(\mathbf{L})$ onto $\mathrm{J}(\mathbf{Con\ L})$.

Let $\theta \in \mathrm{Con}\ \mathbf{L}$, let $\mathbf{K} = \mathbf{L}/\theta$, and let $f : \mathbf{L} \twoheadrightarrow \mathbf{K}$ be the natural epimorphism. For $b \in K$ let $\beta(b)$ denote the least preimage of b. Then it is easy to check that β is one-to-one and preserves joins. Thus $\mathbf{K}$ is isomorphic to a join subsemilattice of $\mathbf{L}$. Moreover, $f(\beta(y)) = y$ and $\beta(f(x)) \leq x$. Let

$$S_\theta = \beta(\mathrm{J}(\mathbf{K})).$$

Lemma 4.

$$S_\theta = \{a \in \mathrm{J}(\mathbf{L}) : \langle a, a_* \rangle \notin \theta\}$$

and

$$\beta f(x) = \bigvee \{a \in S_\theta : a \leq x\}.$$

This shows that each congruence θ on $\mathbf{L}$ can be represented by a subset S_θ of $\mathrm{J}(\mathbf{L})$. Of course S_θ can be represented as a bitvector V_θ indexed by $\mathrm{J}(\mathbf{L})$. We take the value of this vector at $a \in \mathrm{J}(\mathbf{L})$ to be 0 if $a \in S_\theta$ and 1 otherwise so that the order agrees with that of $\mathbf{Con\ L}$. Since $\mathbf{Con\ L}$ is a distributive lattice the join and meet of two such bitvectors can be calculated as the bitwise *or* and *and*.

If we identify $\mathbf{K}$ with the join subsemilattice of $\mathbf{L}$ as above, β is then the identity map and the natural map f associated with θ is

$$f(x) = \bigvee \{a \in S_\theta : a \leq x\}. \tag{2}$$

This also shows that it is easy to determine if $x\ \theta\ y$ from our representation.

In order for all this to work we need to find V_θ (or S_θ) for each congruence of the form $\mathrm{Cg}\,(a, a_*)$, $a \in \mathrm{J}(\mathbf{L})$. This is not just the bitvector which is 1 at a and 0 elsewhere; it is 1 at b if (and only if) $\mathrm{Cg}\,(b, b_*) \leq \mathrm{Cg}\,(a, a_*)$. To do this we use a modification of a 'dependency' relation introduced by Jónsson in his studies of sublattices of free lattice and by Nation and myself in our studies of covers in free lattices as well as in other application. The relations we are about to define were also used in connection with concept analysis; see [16].

For a, $b \in \mathrm{J}(\mathbf{L})$ and $q \in \mathrm{M}(\mathbf{L})$ we define two relations by

$$a \nearrow q \quad \text{if and only if} \quad a \leq q^* \text{ and } a \nleq q \tag{3}$$

$$q \searrow b \quad \text{if and only if} \quad q \geq b_* \text{ and } q \ngeq b. \tag{4}$$

Note that $a \nearrow q$ if and only if $a \vee q = q^*$ and $q \searrow b$ if and only if $q \wedge b = b_*$. Define $\mathbf{G}(\mathbf{L}) = \langle V, \rightarrow\rangle$ to be the directed graph whose vertex set V is the disjoint union of (copies of) $\mathrm{J}(\mathbf{L})$ and $\mathrm{M}(\mathbf{L})$ and whose relation $\rightarrow$ is the union of $\nearrow$ and $\searrow$. Of course this graph also determines a quasiorder. Let $\equiv$ be the equivalence relation for this quasiorder: $x \equiv y$ if and only if $x \rightarrow^* y$ and $y \rightarrow^* x$. We let $\mathbf{G}(\mathbf{L})/{\equiv}$ denote the induced ordered set on the equivalence classes.

Theorem 5. $\mathbf{G}(\mathbf{L})/{\equiv}$ *and* $\mathrm{J}(\mathbf{Con\ L})$ *are isomorphic ordered sets.*

Using R. Tarjan's breadth first search algorithm we can find the $\equiv$-classes in time $O(|\mathrm{J}(\mathbf{L})| \cdot |\mathrm{M}(\mathbf{L})|)$. Thus we can find $\mathrm{J}(\mathbf{Con\ L})$ in time $O(n^2)$ provided we know, for example, the $\leq$ relation on $\mathbf{L}$. More precisely, we can calculate a set E such that

$$E_{\prec}(\mathrm{J}(\mathbf{Con\ L})) \subseteq E \subseteq E_{\leq}(\mathrm{J}(\mathbf{Con\ L})). \tag{5}$$

Now in order to test if $\mathrm{Cg}\,(b, b_*) \leq \mathrm{Cg}\,(a, a_*)$ we need the the transitive closure $E_{\leq}(\mathrm{J}(\mathbf{Con\ L}))$ of E. Before returning to this problem we note that many properties of $\mathbf{L}$ can be determined from E alone and thus can be tested in time (n^2).

Theorem 6. *Let* $\mathbf{L}$ *be a lattice with* n *elements. Then any of the following properties can be tested in time* $O(n^2)$*:*

1. $\mathbf{L}$ *is simple.*
2. $\mathbf{L}$ *is subdirectly irreducible.*
3. $\mathbf{L}$ *is directly indecomposable.*
4. $\mathbf{L}$ *is semidistributive (upper or lower or both).*
5. $\mathbf{L}$ *is a bounded (upper or lower or both) homomorphic image of a free lattice.*
6. $\mathbf{L}$ *is a splitting lattice.*

Proof: For example, $\mathbf{L}$ is subdirectly irreducible if and only if $\mathrm{J}(\mathbf{Con\ L})$ has a unique minimal element and is simple if and only if $|\mathrm{J}(\mathbf{Con\ L})| = 1$ which can be tested quickly without calculating $E_{\leq}(\mathrm{J}(\mathbf{Con\ L}))$. $\mathbf{L}$ is a lower image of a free lattice if and only if $|\mathrm{J}(\mathbf{Con\ L})| = |\mathrm{J}(\mathbf{L})|$ and upper bounded if and only if $|\mathrm{J}(\mathbf{Con\ L})| = |\mathrm{M}(\mathbf{L})|$. $\mathbf{L}$ is a splitting lattice if it is bounded and subdirectly irreducible.

Item 3 uses a different ordered set due to Markowsky [24]. ∎

Now we return to the problem of finding the order relation on $\mathrm{J}(\mathbf{Con\ L})$, not just a set E satisfying (5), that is, finding the transitive closure of E. Let

$$m = |\mathrm{J}(\mathbf{Con\ L})| \qquad e_{\prec} = e_{\prec}(\mathrm{J}(\mathbf{Con\ L})) \qquad e_{\leq} = e_{\leq}(\mathrm{J}(\mathbf{Con\ L}))$$

It is straightforward to find $E_{\leq}(\mathrm{J}(\mathbf{Con\ L}))$ in time $O(m^2 + me_{\prec} + e_{\leq})$. Now $m \leq n$ (in fact $m \leq |\mathrm{J}(\mathbf{L})|$) so $e_{\prec} \leq n^2$ and thus $E_{\leq}(\mathrm{J}(\mathbf{Con\ L}))$ can be found in time (n^3) but a much better theorem is true:

Theorem 7. *For a finite lattice* $\mathbf{L}$ *with* n *elements,* $\mathrm{J}(\mathbf{Con\ L})$ *and its order relation can be found in time* $O(n^2 \log n)$*.*

This is proved in [11]. The main idea is proving a stronger relation than the obvious relation: $e_\prec \leq n^2$. In fact we show

$$e_\prec(\mathrm{J}(\mathbf{Con\ L})) \leq 2n \log_2 n$$

from which the theorem follows.

While the size N of $\mathbf{Con\ L}$ may be exponential in the size of $\mathbf{L}$, $\mathbf{Con\ L}$ is distributive and so is isomorphic to the lattice of order ideals of $\mathrm{J}(\mathbf{Con\ L})$. It is straightforward to compute this in time $O(mN)$ but there is a faster algorithm in R. Medina and L. Nourine [26]; see also [18].

4. Free and Finitely Presented Lattices

Equational complexity. Consider the problem of deciding if two lattice terms $s(x_1, \dots, x_n)$ and $t(x_1, \dots, x_n)$ are always equal when evaluated in every lattice in some class of lattices $\mathcal{K}$. For the three classical classes of lattices, all lattices, modular lattices, and distributive lattices, the complexity of the equational theory forms an interesting trichotomy:

Table 1. Equational complexity of three classes of lattices

Class	Complexity
distributive lattices	co-NP complete
modular lattices	undecidable
all lattices	polynomial

The word problem for free lattices was first shown to be solvable by Thoralf Skolem in 1920, see [27], although this escaped the attention of lattice theorists until recently. Skolem's work is especially interesting because it includes a (uniform) solution for the word problem for finitely presented lattices which is polynomial time. Philip Whitman initiated a deeper study of free lattices in 1941, [31] and [32]. He gave the solution for the word problem for free lattices which we use today. We let $\mathbf{FL}(X)$ denote the free lattice generated by X. He showed that each element of $\mathbf{FL}(X)$ has a unique shortest term representing it, known as the canonical form.

While Whitman's algorithms for testing $v \leq u$ and for putting terms into canonical form are polynomial time, they are not particularly fun to do by hand and it was this that motivated me to write a program to do these tasks. While there are some interesting computational aspects of these concepts, we will not discuss them here; instead we will concentrate on the covering relation.

Finitely presented lattices. A partial lattice $\mathbf{P}$ is an ordered set with a partial join and meet operation. This entails that if the partial join operation is defined on a set of elements, its value must be the least upper bound (in the ordering of $\mathbf{P}$) of the set and similarly for the partial meet operation. We let $\mathbf{FL}(\mathbf{P})$ denote the lattice freely generated by $\mathbf{P}$. Clearly $\mathbf{FL}(\mathbf{P})$ is finitely presented when $\mathbf{P}$ is finite, and conversely, there is a polynomial time algorithm to convert any finitely presented lattice to one on this form.

In many respects finitely presented lattices are similar to free lattices: there is a solution to the word problem similar to Whitman's, [8], and each element has a canonical form, [10].

5. Covers in Free Lattices, Bounded Lattices

Whitman found a few covers in free lattices; for example, he showed that $x \wedge (y \vee z) \prec x$ in $\mathbf{FL}(x, y, z)$. In his work on the lattice of lattice varieties [25] McKenzie needed to find covers in free lattices. He defined the concept of a *bounded epimorphism*. A homomorphism

$$f : \mathbf{F} \twoheadrightarrow \mathbf{L}$$

is *bounded* if $f^{-1}(a)$ has a least and greatest element in $\mathbf{F}$, for each $a \in L$. The least preimage of a is denoted $\beta(a)$ and the greatest preimage is denoted $\alpha(a)$. If $a \prec b$ in $\mathbf{L}$ then $\alpha(a) \prec \alpha(a) \vee \beta(b)$ and so give rise to covers in $\mathbf{F}$. A lattice is called *bounded* if it is there is a bounded homomorphism from a finitely generated free lattice onto $\mathbf{L}$.

Now if $v \prec u$ in $\mathbf{F}$, then there is a unique largest congruence $\psi(u, v)$ separating u from v. The quotient lattice $\mathbf{F}/\psi(u, v)$ is always subdirectly irreducible. When $\mathbf{F}$ is a free lattice these quotient lattices are very important. They are finite and the natural homomorphism from the free lattice onto them is bounded. Conversely every finite, subdirectly irreducible bounded lattice arises in this way. This important connection between these simple ideas allowed McKenzie to characterize finite project lattices and show that one could decide if $v \prec u$ in a free lattice. His algorithm for the latter was highly exponential. In [14] Nation and I made a thorough study of covers in free lattices giving more general algorithms which run in polynomial time. In [10] these ideas were extended to finitely presented lattices.

How does one find covers in $\mathbf{FL}(\mathbf{P})$ (and thus in $\mathbf{FL}(X)$)? The problem can be reduced to deciding when a join irreducible element of $\mathbf{FL}(\mathbf{P})$ has a lower cover. That is, when it is completely join irreducible. If $w \in \mathbf{FL}(\mathbf{P})$ we associate a set $\mathrm{J}(w)$ of join irreducibles (essentially the join irreducible subterms of w). If we close $\mathrm{J}(w)$ under joins we obtain a lattice $\mathbf{L}(w)$ which is a join subsemilattice of $\mathbf{FL}(\mathbf{P})$. The map $f : \mathbf{FL}(\mathbf{P}) \twoheadrightarrow \mathbf{L}(w)$ given by

$$f(u) = \bigvee \{v \in \mathrm{J}(w) : v \leq u\} \tag{6}$$

is a homomorphism. (Note the similarity with our representation of congruences on finite lattices, especially (2).)

Theorem 8. *A join irreducible element w in the finitely presented lattice* $\mathbf{FL}(\mathbf{P})$ *is completely join irreducible if the homomorphism* (6) *is bounded.*

We have already noted that one can test if $\mathbf{L}(w)$ is a bounded image of a free lattice in polynomial time and a similar result holds for testing if (6) is bounded. However $\mathbf{L}(w)$ is the join closure of $\mathrm{J}(w)$ and so may be exponential in the size of w.

Despite this there is a polynomial time algorithm, at least for free lattices. For a join irreducible element w of a free lattice let

$$w_\dagger = \bigvee \{u \in \mathrm{J}(w) : u < w\}$$
$$\mathrm{K}(w) = \{v \in \mathrm{J}(w) : w_\dagger \vee v \not\geq w\}$$

Notice that $w_\dagger$ is the lower cover of w in $\mathbf{L}(w)$.

Theorem 9. *Let w be a join irreducible element of* $\mathbf{FL}(X)$*. Then w is completely join irreducible in* $\mathbf{FL}(X)$ *(that is, it has a lower cover) if and only if the following two conditions are satisfied:*

1. *every* $u \in \mathrm{J}(w) - \{w\}$ *is completely join irreducible,*
2. $w \not\leq \bigvee \mathrm{K}(w)$.

Moreover if w is completely join irreducible then

$$\kappa(w) = \bigvee \{x \in X : w_\dagger \vee x \not\geq w\} \vee \bigvee \{k^\dagger \wedge \kappa(v) : v \in \mathrm{J}(w) - \{w\},\ w \not\leq \kappa(v)\}$$

where

$$k^\dagger = \bigwedge \left\{\kappa(v) : v \in \mathrm{J}(w) - \{w\},\ \kappa(v) \geq \bigvee \mathrm{K}(w)\right\}$$

Here $\kappa(w)$ is the largest element of $\mathbf{FL}(X)$ above the lower cover w_* of w but not above w which exists by the semidistributivity of $\mathbf{FL}(X)$.

For the finitely presented lattices $\mathbf{FL}(\mathbf{P})$ we can find the lower covers of an element in polynomial time *provided we know the lower covers of the elements of* $\mathbf{P}$. The following is open:

Problem 2. *Is there a polynomial time algorithm to determine if an element w of a finitely presented lattice* $\mathbf{FL}(\mathbf{P})$ *is completely join irreducible, uniform in w and* $\mathbf{P}$*?*

Alan Day's doubling construction. If $\mathbf{L}$ is a lattice and I is an interval in $\mathbf{L}$ then $\mathbf{L}[I]$ denotes the lattice obtained by replacing I by $I \times 2$ and using the (more or less) obvious order; see [6] or [12]. Day used his construction to give a simple solution to the word problem for free lattices, [4]. But more importantly he noted that a bounded homomorphism $\mathbf{F} \to \mathbf{L}$ could be factored through $\mathbf{L}[I]$, $\mathbf{F} \to \mathbf{L}[I] \to \mathbf{L}$ and the map $\mathbf{F} \to \mathbf{L}[I]$ was still bounded. This allowed him to prove that *free lattices are weakly atomic* [5]. (A lattice is *weakly atomic* if every nontrivial interval contains a covering.)

Finitely presented lattices need not be weakly atomic. In [10] there is an example of a finitely presented lattice with no cover.

Problem 3. *Is there an algorithm to determine if a finitely presented lattice is weakly atomic?*

We have an algorithm to determine if a finitely presented lattice is bounded but would the following is open:

Problem 4. *Is there a polynomial time algorithm to determine if a finitely presented lattice is bounded?*

If such an algorithm exists we could decide in polynomial time if a finitely presented lattice is projective.

Semidistributive lattices. A lattice is semidistributive if it satisfies the following condition and its dual:

$$a \vee b = a \vee c \quad \text{implies} \quad a \vee b = a \vee (b \wedge c)$$

Free lattices are semidistributive and from this it follows that bounded lattices are semidistributive. McKenzie asked if the converse were true for finite lattices. It took some time to answer this. J. B. Nation found a counter-example.

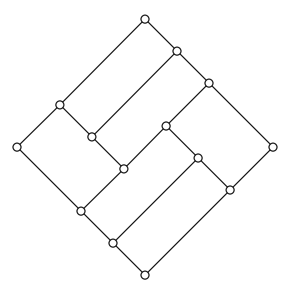

FIGURE 1. A semidistributive lattice which is not bounded.

The fact that it is hard to find semidistributive lattices which are not bounded is reflected in several theorem which state that, under certain additional hypotheses, semidistributivity does imply boundedness. The deepest result of this nature is Nation's result that if a finite semidistributive lattice satisfies Whitman's condition (W) (see [12]) then it is bounded. This result has several important consequences.

It is also true that congruence lattices of finite semilattice are upper bounded; see [13]. N. Caspard [3] recently showed that the lattice of permutations is bounded.

6. APPLICATIONS

Most of the algorithms alluded to in this paper have been implemented (primarily in Lisp) and the resulting programs have been very useful in solving problems in lattice theory. We will illustrate this with big lattices.

A finite lattice is called *big* if it is a maximal sublattice of an infinite lattice. At first we did not think such a lattice could exist. For example,

I. Rival has shown that if $\mathbf{L} \leq \mathbf{K}$ and both lattices are distributive, then $|\mathbf{K}| \leq (3/2)|\mathbf{L}|$. Despite this there big lattices. In [9] we characterize these lattices.

Theorem 10. *There are* 145 *minimal big lattices and every big lattice contains one of these as a sublattice.*

Not surprisingly our computer programs played a big role in finding these lattices. As an example suppose we want to know if the 8 element Boolean algebra is big. The following lemma is obvious.

Lemma 11. *A sublattice* $\mathbf{L}$ *of* $\mathbf{K}$ *is maximal if and only if* $\mathbf{K}$ *is generated by* $\{x\} \cup L$ *for each* $x \in K - L$.

So we might look at the finitely presented lattice $\mathbf{FL}(\mathbf{P})$ where $\mathbf{P}$ is the partial lattice consisting of the 8 element Boolean algebra with all its joins and meets defined together with one addition element below one of the atoms. Figure 2 diagrams $\mathbf{FL}(\mathbf{P})$.

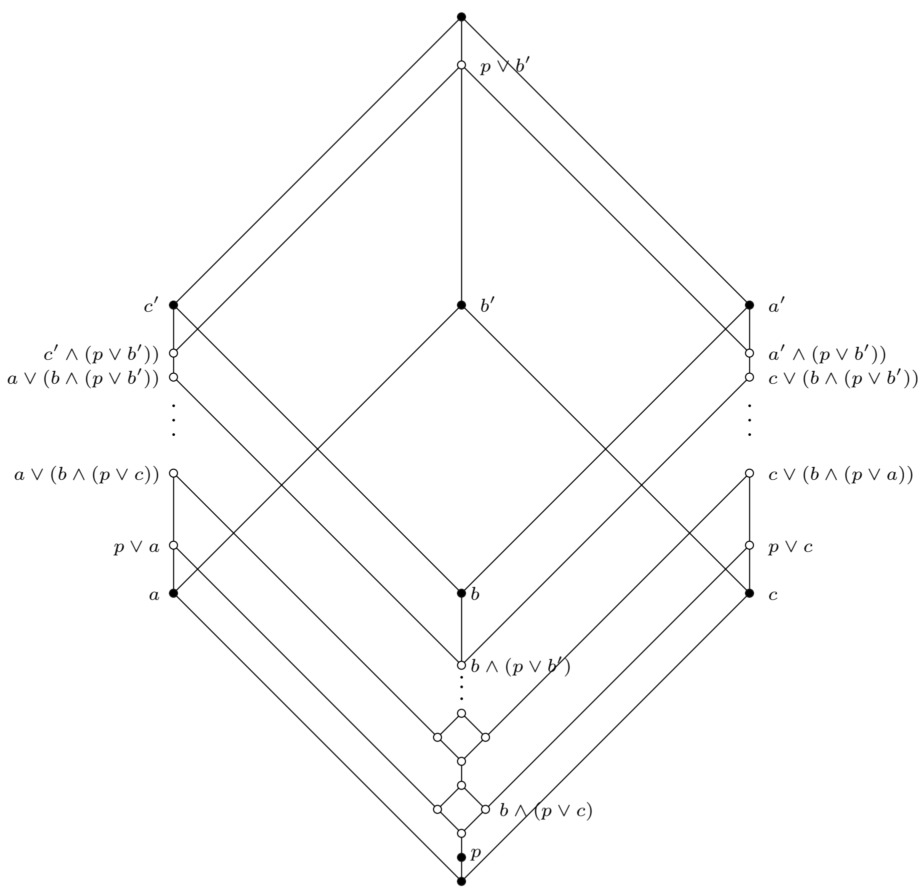

FIGURE 2. The free lattice generated by $\mathbf{B}_3$ and p, $0 \leq p \leq b$.

While this is certainly an infinite extension of $\mathbf{B}_3$, it is not a minimal extension: the condition of the lemma fails. In fact $\mathbf{B}_3$ is not big.

In order to produce minimal extensions we use a different idea. Suppose x and y are incomparable elements of $\mathbf{L}$ and that $L = [0, y] \cup [x, 1]$. Let $\mathbf{P}$ be the partial lattice on $P = L \cup \{p, q\}$ with all the joins and meets of $\mathbf{L}$ defined and in addition:

$$x \wedge y \leq p \leq x \qquad y \leq q \leq x \vee y \tag{7}$$

$$p = x \wedge q \qquad q = y \vee p \tag{8}$$

Then $\mathbf{L}$ is a maximal sublattice of $\mathbf{FL}(\mathbf{P})$. Thus if $\mathbf{FL}(\mathbf{P})$ is infinite then $\mathbf{L}$ is big. For $\mathbf{L} = \mathbf{B}_3$, $\mathbf{FL}(\mathbf{P})$ is drawn in Figure 3. In fact this is the largest minimal extension of $\mathbf{B}_3$.

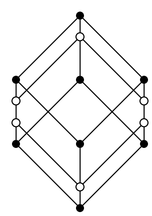

FIGURE 3

V. Slavík [28] has an algorithm to determine if $\mathbf{FL}(\mathbf{P})$ is finite but it is not very efficient. This leads to the following problem:

Problem 5. *Is there a polynomial time algorithm to determine if a finitely presented lattice is finite?*

When we want to know if a finitely presented lattice $\mathbf{FL}(\mathbf{P})$ is finite we use our program to alternately close it under joins and meets. If it stops then $\mathbf{FL}(\mathbf{P})$ is finite. If not the program looks for a join irreducible element which is not completely join irreducible (or dually). Of course if it finds such an element, $\mathbf{FL}(\mathbf{P})$ is infinite. However, we do not know if this procedure always gives an answer.

Problem 6. *If every join irreducible element of a finitely presented lattice is completely join irreducible and dually is it finite?*

Interestingly when $\mathbf{FL}(\mathbf{P})$ is infinite the growth of the closures seems to divide into two cases: either it is linear (only a fixed number of elements is added with each closure) or it is at least exponential. The former leads to lattices which can be drawn such as the one in Figure 2.

7. Automatic Lattice Drawing

Finitely presented lattices have finite associated with them; for example, $\mathbf{L}(w)$ which determines if w has any lower covers. Our programs can automatically generate such lattices but it can be difficult to 'see' what the lattice is just from a list of elements and a procedure for testing $\leq$. So it is nice to have diagrams of them. While it is easy to produce a correct diagram, the lattice may be unrecognizable even for small familiar lattice. So we wrote a program to automatically produce 'nice' pictures of lattices. A description of the program and a Java applet illustrating it is on my web page:

`http://www.math.hawaii.edu/∼ralph/LatDraw/`

References

[1] J.-P. Bordat, *Efficient polynomial algorithms for distributive lattices*, Discrete Appl. Math. **32** (1991), 31–50.

[2] N. Caspard, *A characterization theorem for all interval doubling schemes of the lattice of permutations*, to appear.

[3] N. Caspard, *The lattice of permutations is bounded*, to appear.

[4] A. Day, *A simple solution of the word problem for lattices*, Canad. Math. Bull. **13** (1970), 253–254.

[5] A. Day, *Splitting lattices generate all lattices*, Algebra Universalis **7** (1977), 163–170.

[6] A. Day, *Doubling constructions in lattice theory*, Canad. J. Math. **44** (1992), 252–269.

[7] Alan Day and Ralph Freese, *A characterization of identities implying congruence modularity, I*, Canad. J. Math. **32** (1980), 1140–1167.

[8] R. A. Dean, *Free lattices generated by partially ordered sets and preserving bounds*, Canad. J. Math. **16** (1964), 136–148.

[9] R. Freese, J. Ježek, and J. B. Nation, *Lattices with large minimal extensions*, Algebra Universalis, to appear.

[10] Ralph Freese, *Finitely presented lattices: canonical forms and the covering relation*, Trans. Amer. Math. Soc. **312** (1989), 841–860.

[11] Ralph Freese, *Computing congruence lattices of finite lattices*, Proc. Amer. Math. Soc. **125** (1997), 3457–3463.

[12] Ralph Freese, Jaroslav Ježek, and J. B. Nation, *Free lattices*, Amer. Math. Soc., Providence, 1995, Mathematical Surveys and Monographs, vol. 42.

[13] Ralph Freese, Keith Kearnes, and J. B. Nation, *Congruence lattices of congruence semidistributive algebras*, Lattice Theory and its Applications (K. A. Baker and R. Wille, eds.), Heldermann Verlag, Lemgo, Germany, 1995, Darmstadt, Germany, June 13–17, 1991, pp. 63–78.

[14] Ralph Freese and J. B. Nation, *Covers in free lattices*, Trans. Amer. Math. Soc. **288** (1985), 1–42.

[15] B. Ganter and R. Wille, *Formale Begriffsanalyse*, Mathematische Grundlagen, Springer Verlag, Berlin, 1996.

[16] W. Geyer, *The generalized doubling construction and formal concept analysis*, Algebra Universalis **32** (1994), 341–367.

[17] P. Goralčík, A. Goralčíková, V. Koubek, and V. Rödl, *Fast recognition of rings and lattices*, Fundamentals of Computation Theory (F. Gécseg, ed.), Springer Verlag, New York, 1981, Lecture Notes in Computer Science, **117**, pp. 137–145.

[18] Michel Habib and Lhouari Nourine, *Tree structure for distributive lattices and its applications*, Theoret. Comput. Sci. **165** (1996), no. 2, 391–405.

[19] Christian Herrmann and Marcel Wild, *A polynomial algorithm for testing congruence modularity*, Internat. J. Algebra Comput. **6** (1996), no. 4, 379–387.

[20] A. B. Kahn, *Topological sorting of large networks*, Comm. ACM **5** (1962), 558–562.

[21] D. J. Kleitman and K. J. Winston, *The asymptotic number of lattices*, Combinatorical Mathematics, Optimal Designs and their Applications (J. Srivastava, ed.), Ann. Discrete Math., 6, North-Holland, Amsterdam, 1980, pp. 243–249.

[22] Walter Klotz and Lutz Lucht, *Endliche Verbände*, J. Reine Angew. Math. **247** (1971), 58–68.
[23] D. E. Knuth, *The art of computer programming, Vol. I: Fundamental algorithms*, Addison Wesley, Reading, Mass., 1975, second edition.
[24] G. Markowsky, *The factorization and representation of lattices*, Trans. Amer. Math. Soc. **203** (1975), 185–200.
[25] R. McKenzie, *Equational bases and non-modular lattice varieties*, Trans. Amer. Math. Soc. **174** (1972), 1–43.
[26] R. Medina and L. Nourine, *Algorithme efficace de génération des idéaux d'un ensemble ordonné*, C. R. Acad. Sci. Paris Sér. I Math. **319** (1994), 1115–1120.
[27] T. Skolem, *Select works in logic*, Scandinavian University Books, Oslo, 1970.
[28] V. Slavík, *Lattices with finite w–covers*, Algebra Universalis **36** (1996), 286–315.
[29] M. Talamo and P. Vocca, *A data structure for lattice representation*, Theoret. Comput. Sci. **175** (1997), no. 2, 373–392, Orders, algorithms and applications (Lyon, 1994).
[30] Maurizio Talamo and Paola Vocca, *Fast lattice browsing on sparse representations*, Orders, algorithms, and applications (Lyon, 1994), Springer, Berlin, 1994, pp. 186–204.
[31] Ph. M. Whitman, *Free lattices*, Ann. of Math. (2) **42** (1941), 325–330.
[32] Ph. M. Whitman, *Free lattices II*, Ann. of Math. (2) **43** (1942), 104–115.

Ralph Freese
University of Hawaii
Honolulu, HI 96822, U. S. A.
ralph@math.hawaii.edu
http://www.math.hawaii.edu/∼ralph/