11-2015

Generating All Finite Modular Lattices of a Given Size

Peter Jipsen
Chapman University, jipsen@chapman.edu

Nathan Lawless
Chapman University

Recommended Citation
P. Jipsen and N. Lawless, "Generating all finite modular lattices of a given size," Algebra Univers., vol. 74, no. 3–4, pp. 253–264, 2015. doi: 10.1007/s00012-015-0348-x

Generating All Finite Modular Lattices of a Given Size

Generating all finite modular lattices of a given size

Peter Jipsen and Nathan Lawless

Dedicated to Brian Davey on the occasion of his 65th birthday

Abstract. Modular lattices, introduced by R. Dedekind, are an important subvariety of lattices that includes all distributive lattices. Heitzig and Reinhold [6] developed an algorithm to enumerate, up to isomorphism, all finite lattices up to size 18. Here we adapt and improve this algorithm to construct and count modular lattices up to size 23, semimodular lattices up to size 22, and lattices of size 19. We also show that 2^{n-3} is a lower bound for the number of nonisomorphic modular lattices of size n.

1. Introduction

Enumeration of finite mathematical structures is an important tool since it allows testing new hypotheses and searching for counterexamples. Additionally, it provides insight into the properties of these structures. Here we concentrate on constructing, up to isomorphism, all modular lattices with a given number of elements. The algorithm we develop is a modification of the approach of Heitzig and Reinhold [6] who enumerated (up to isomorphism) all lattices with up to 18 elements. The number of distributive lattices of size up to 49 were calculated by Erné, Heitzig and Reinhold [3]. In the Online Encyclopedia of Integer Sequences (`oeis.org`) the relevant sequences are A006981, A006966 and A006982, but the sequence for the number of modular lattices was given only up to $n = 11$. For $n = 12$ there are 766 nonisomorphic modular lattices, as was reported in [1]. We extend this result to $n = 23$ and also count the number of semimodular lattices up to size $n = 22$ (see Table 1).

Our algorithm uses an improved method for removing isomorphic copies which allowed us to recalculate the numbers in [6] for all lattices up to $n = 18$ and go one step further to find the number of nonisomorphic lattices with 19 elements. The calculations were done on a cluster of 64 processors and took 26 hours for $n = 18$ and 19 days for $n = 19$.

In the remainder of this section, we define some properties and recall some basic results of (semi)modular lattices. In Section 2, we give an outline of the algorithm used by [6] to generate finite lattices up to isomorphism. Then, in

Presented by ...
Received ...; accepted in final form ...
2010 *Mathematics Subject Classification*: Primary: 06C05; Secondary: 06C10, 05A15.
Key words and phrases: Modular lattices, semimodular lattices, counting up to isomorphism, orderly algorithm.

Section 3, we adapt this algorithm to generate modular lattices up to isomorphism by adding a series of constraints to the algorithm. Section 4 contains an improvement for the algorithm used by [6] by employing the canonical construction path introduced in [8]. In Section 5, the algorithm is adjusted to generate only vertically indecomposable modular lattices.

A *modular lattice* L is a lattice which satisfies the modular law

$$a \geq c \text{ implies } a \wedge (b \vee c) = (a \wedge b) \vee c \text{ for all } a, b, c \in L.$$

Weaker conditions of modularity are semimodularity and lower semimodularity. As usual, we write $a \prec b$ if a is *covered by* b.

A lattice L is *semimodular* if for all $a, b \in L$

$$a \wedge b \prec a, b \text{ implies that } a, b \prec a \vee b.$$

Dually, L is *lower semimodular* if for all $a, b \in L$

$$a, b \prec a \vee b \text{ implies that } a \wedge b \prec a, b.$$

Recall that a *chain* in a lattice L is a subset of L such that all elements in the subset are comparable. The *length* of a lattice is the supremum of the cardinalities of all chains in the lattice. The *depth*, respectively *height*, of an element x in a lattice is the supremum of all chains in the filter, respectively ideal, generated by x. The next two well-known results below can be found for example in [5].

Proposition 1.1. *A lattice of finite length is modular if and only if it is semimodular and lower semimodular.*

A chain C in a poset P is *maximal* if whenever $C \subseteq D \subseteq P$ and D is a chain in P, then $C = D$. In a finite lattice, a maximal chain is a chain from bottom to top such that each element in the chain, other than the top, is covered by some element in the chain.

Theorem 1.2 (Jordan-Hölder Chain Condition)**.** *Let L be a finite semimodular lattice. Then, for any maximal chains C and D in L, $|C| = |D|$.*

2. Generating finite lattices

There are many ways to represent finite lattices and to construct bigger lattices from smaller lattices. An algorithm that constructs up to isomorphism all combinatorial objects of a certain kind and of a given size is called an *orderly algorithm* if it produces exactly one member of each isomorphism class without testing that this member is nonisomorphic to previously constructed objects. Such algorithms were first introduced by Faradzhev [4] and Read [10] for enumerating finite graphs. Heitzig and Reinhold [6] developed an orderly algorithm to enumerate all finite lattices up to isomorphism and used it to count the number of lattices up to size 18. Since our first algorithm for modular lattices is based on their approach, we recall some of the details here.

Let L be a lattice. A nonempty antichain $A \subseteq L \setminus \{0\}$ is a *lattice-antichain* if $a \wedge b \in \{0\} \cup \uparrow A$ for all $a, b \in \uparrow A$, where $\uparrow A = \{b \mid b \geq a, a \in A\}$. A finite lattice is called an n-lattice if its set of elements is $\{0, 1, 2, \ldots, n-1\}$, where 0 and 1 are the bottom and top elements.

Given a lattice antichain A and an n-lattice L, a poset L^A with $n+1$ elements is constructed by adding an element n to L as an atom with A as the set of its covers. Furthermore, the following lemma states that L^A is a lattice.

Lemma 2.1 ([6]). *A subset $A \subseteq L \setminus \{0\}$ of an n-lattice L is a lattice-antichain if and only if L is a subposet of an $(n+1)$-lattice L^A in which the element n is an atom and A is the set of its covers.*

In order to generate only one copy of each lattice up to isomorphism, the weight $w(L) = (w_2(L), \ldots, w_{n-1}(L))$ of an n-lattice L is defined by setting $w_i(L) = \sum_{i \prec j} 2^j$.

With this weight, for two n-lattices L and M, $w(L)$ is said to be (lexicographically) smaller than $w(M)$ if there is an $i \leq n-1$ such that $w_i(L) < w_i(M)$ and $w_k(L) = w_k(M)$ for all $k = 2, \ldots, i-1$. An n-lattice C is called a canonical lattice if there is no n-lattice isomorphic to C that has a smaller weight. In order to check whether an n-lattice L is canonical, one has to check whether there is a permutation of the elements of L that yields an isomorphic copy of L with a smaller weight.

With these definitions, a recursive algorithm is formulated in [6] which generates exactly all canonical lattices of order equal to n for a given natural number $n \geq 2$.

```
next_lattice(integer m, canonical m-lattice L)
begin
    if m < n then
        for each lattice-antichain A of L do
            if L^A is a canonical lattice then
                next_lattice(m + 1, L^A)
    if m = n then output L
end
```

Algorithm 1

The set of all maximal elements in a finite poset P is called the *first level* of P and is denoted by $lev_1(P)$. The $(m+1)$-*th level* of P is recursively defined by

$$lev_{m+1}(P) = lev_1(P \setminus \bigcup_{i=1}^{m} lev_i(P)).$$

It can also be seen that the previous definition of depth of $p \in P$ is equivalent to a number k such that $p \in lev_k(P)$. We denote this by $dep_P(p)$.

We say an n-lattice L is *levelized* if

$$dep_L(i) \leq dep_L(j) \text{ for all } i, j \in L \setminus \{0\} \text{ with } i \leq j.$$

In other words, the levels form a partition on $L \setminus \{0\} = \{1, 2, \ldots, n-1\}$ of the form $\{1 \mid 2, \ldots, m_2 \mid m_2+1, \ldots, m_3 \mid \ldots \mid m_{k-1}+1, \ldots, m_k\}$, where k is the number of levels in L.

Throughout the rest of the paper, we consider the bottom level to be $lev_k(L)$, unless indicated otherwise. The following lemma gives us an important property when generating levelized lattices, since it tells us we only need to consider lattice-antichains that have at least one element in the two bottom levels.

Lemma 2.2 ([6]). *For an n-lattice L and a lattice-antichain A, L^A is levelized if and only if $A \cap (lev_{k-1}(L) \cup lev_k(L)) \neq \emptyset$.*

3. Generating finite modular lattices

In order to construct only modular lattices of size n using this algorithm, we start by selecting only the lattices which are modular when we get to size n. However, modular lattices constitute a very small fraction of the total number of lattices. Therefore, it is important to add constraints in order to minimize the generation of non-modular lattices. In order to do this, we present a series of results to decide when a subtree in a search tree can be cut off and which lattice antichains must be considered in each step. During this section, we refer to *descendants* of a lattice L as those lattices generated through the element extension described in [6], together with any additional constraints introduced in this section.

Lemma 3.1. *For an n-lattice L, and a lattice-antichain $A \subseteq L$, if there exist $a, b \in A$ such that $dep_L(a) \neq dep_L(b)$, then all descendants of L^A are non-semimodular. Specifically, they are non-modular.*

Proof. Assume without loss of generality that $dep_L(a) < dep_L(b)$. Let C_a and C_b be the chains of maximal cardinality from 1 to a and b respectively.

For any $x \in L$, $dep_L(x)$ is equal to the cardinality of the longest chain from x to 1, hence $|C_a| < |C_b|$.

Next, in L^A, we have $n \prec a$ and $n \prec b$. Let M be a descendant of L^A, and choose any chain D from 0 to n.

Then $D_a := D \cup C_a$ and $D_b := D \cup C_b$ are maximal chains of different length since $|D_a| = |D| + |C_a| < |D| + |C_b| = |D_b|$. By Theorem 1.2, it follows that M is not semimodular, and therefore is non-modular. □

From the preceding lemma we may conclude the following result.

Corollary 3.2. *For the construction of (semi-)modular lattices using the orderly algorithm of* [6]*, it suffices consider lattice-antichains A such that $A \subseteq lev_{k-1}(L)$ or $A \subseteq lev_k(L)$.*

Lemma 3.3. *Let L be an n-lattice where $k = dep_L(n-1)$ is the bottom nonzero level, and let $A \subseteq lev_k(L)$ be a lattice-antichain of L. If there is an atom of L in $lev_{k-1}(L)$ then all descendants of L^A are non-semimodular, and hence non-modular.*

Proof. Let $b \in lev_{k-1}(L)$ be an atom of L, and choose any $a \in A \subseteq lev_k(L)$. Then there exist chains C_a from a to 1 and C_b from b to 1 of length k and $k-1$ respectively.

Since the new element n is in a new level $lev_{k+1}(L^A)$, and $dep_{L^A}(b) = k-1 = (k+1)-2$, b is contained in the third lowest level of L^A. Therefore, by Corollary 3.2, it is not used in the generation of any descendants, and b remains as an atom in all descendants. Hence, the maximal chain $D_b := \{0\} \cup C_b$ is of constant length $1 + (k-1) = k$ for any descendant of L^A.

Let M be a descendant of L^A. Choose any chain C_n from 0 to n, then $|C_n| \geq 2$. Therefore, for the maximal chain $D_a := C_n \cup C_a$,

$$|D_a| = |C_n| + |C_a| \geq 2 + k > k = |D_b|.$$

By Theorem 1.2, since both D_a and D_b are maximal chains, it follows that M is non-semimodular and therefore non-modular. □

An observation which significantly decreases the search space is based on the following property of the algorithm. Since elements are always added below a lattice antichain, if two elements in the antichain fail semimodularity, then those two elements also fail semimodularity in any of the descendants. Therefore, when adding a new element below a lattice antichain, we should check that we are not generating a non-semimodular lattice.

Lemma 3.4. *For an n-lattice L and a lattice antichain $A \subseteq L$, if there exist $a, b \in A$ which do not have a common cover, then all descendants of L^A are non-semimodular.*

Proof. In L^A, for the new element n, $n \prec a$ and $n \prec b$. However, $a \not\prec a \vee b$ or $b \not\prec a \vee b$. Therefore, L^A is not semimodular. Furthermore, for any descendant M of L^A, it is not possible to add a common cover to a, b. Hence, M is not semimodular (and consequently, not modular). □

Similarly, we can consider when it is not possible to make a non-lower semimodular lattice into a lower semimodular lattice.

Lemma 3.5. *Let L be an n-lattice, and let k be its bottom non-zero level. If there exist $a, b \in lev_{k-2}(L)$ which do not satisfy lower semimodularity, then all descendants of L are non-lower semimodular (and hence non-modular).*

Proof. Given an a, b such that $a, b \prec a \vee b$ but $a \wedge b \not\prec a$ or $a \wedge b \not\prec b$, the algorithm can make a, b satisfy lower semimodularity by adding an element below a, b. However, by Corollary 3.2, we only consider lattice antichains in $lev_k(L)$ and $lev_{k-1}(L)$. Therefore, if $a, b \in lev_{k-2}(L)$, we cannot add a common co-cover, and all descendants M of L are non-lower semimodular. □

This lemma can be incorporated into the algorithm by checking that all elements of $lev_{k-1}(L)$ satisfy lower semimodularity each time a new level is added.

The preceding results are summarized in the following theorems.

Theorem 3.6. *When generating semimodular lattices, for a lattice L, we only consider lattice-antichains A which satisfy all of the following conditions:*

(A1) $A \subseteq lev_{k-1}(L)$ *or* $A \subseteq lev_k(L)$.
(A2) *If* $A \subseteq lev_k(L)$, *there are no atoms in* $lev_{k-1}(L)$.
(A3) *For all* $x, y \in A$, x *and* y *have a common cover.*

Theorem 3.7. *When generating modular lattices, for a lattice L, we only consider lattice-antichains A which satisfy* (A1), (A2), (A3) *and*

(A4) *If* $A \subseteq lev_k(L)$, *then* $lev_{k-1}(L)$ *satisfies lower semimodularity (i. e., for all* $x, y \in lev_{k-1}(L)$, $x, y \prec x \vee y$ *implies* $x \wedge y \prec x, y$*).*

Another improvement can be implemented in the last step when generating lattices of size n from those of size $n-1$, by only considering lattice-antichains $A \subseteq lev_{k-1}(L)$ and $A = lev_k(L)$. This is due to the following result.

Lemma 3.8. *For an n-lattice L and a lattice antichain $A \subsetneq lev_k(L)$, the $n+1$-lattice L^A is non-modular.*

Proof. Since $A \subsetneq lev_k(L)$, there exists $b \in lev_k(L)$ such that $b \notin A$. Let $a \in A$. Since $a, b \in lev_k(L)$, there exist chains C_a and C_b from a to 1 and b to 1 respectively of length k both.

In L^A, $n \prec a$, but $n \not\prec b$. Thus, for the maximal chains $D_a := \{0, n\} \cup C_a$ and $D_b := \{0\} \cup C_b$,

$$|D_a| = 2 + k > 1 + k = |D_b|,$$

thus L^A is non-modular. □

4. Dealing with isomorphisms

When generating finite (modular) lattices using Algorithm 1, the majority of the time is spent in testing if the lattice L^A is canonical, an operation of order $O(n!)$. An approach that speeds-up the algorithm significantly, while still generating exactly one isomorphic copy of each (modular) lattice is via generation by *canonical construction path* which was introduced by McKay [8].

This canonical construction has two components. The first one is to use only one representative of each orbit in the lattice antichains of L. In other words, if there is an automorphism g on L such that $\{g(a) \mid a \in A\} = B$ for lattice-antichains A, B, only one of these antichains is chosen arbitrarily.

Secondly, after the extension of any lattice L using A, L^A is checked to see if L is the inverse through a "canonical deletion". This uses the canonical

labeling of the program `nauty` [9]. In general, a *canonical labeling* associates with each n-lattice L a permutation c_L on $\{0,\ldots,n-1\}$ such that for any n-lattice M we have $L \cong M$ if and only if

$$\{(c_L(x), c_L(y)) \mid x \leq y \text{ in } L\} = \{(c_M(x), c_M(y)) \mid x \leq y \text{ in } M\},$$

i. e., the permutation maps each lattice to a fixed representative of its isomorphism class. When a new $(n+1)$-lattice L^A is generated from L and A, a canonical labeling c_{L^A} of L^A is generated using a partition by levels in `nauty`. Let $n' := c_{L^A}^{-1}(n)$ denote the element which maps to n under the canonical labeling. We consider the set $m(L^A) = \{\langle L^A, a\rangle \mid f(a) = n', f \in Aut(L^A)\}$, where $\langle L^A, a\rangle$ denotes the lattice obtained by removing a from L^A. Note that $L = \langle L^A, n\rangle$. If $L \in m(L^A)$, we say L^A is *canonical* and keep it, otherwise it is discarded. Note that such a canonical lattice can have a different labeling of the nodes than a canonical lattice from Algorithm 1.

Using this construction, Theorem 1 in [8] states that starting from any lattice, exactly one isomorphic copy of each descendant will be output. Thus, starting with the two-element lattice, we can generate exactly one isomorphic copy of each lattice of a given size n. This has an advantage over the construction used in [6] since it does not require checking all permutations of a lattice, and it uses canonical labeling by `nauty`, which is generally considered the most efficient canonical labeling program for small combinatorial structures. Furthermore, this construction is orderly since it only considers the lattices L and L^A. This is beneficial during computations because it does not require storage of previously generated lattices or communication between nodes during parallel computations. Given this, Algorithm 1 can be modified:

```
next_lattice2(integer m, canonical m-lattice L)
begin
    if m < n then
        LAC := {A | A is a lattice-antichain of L}
        for each orbit O of the action of Aut(L) on LAC
            select any A ∈ O
            c := canonical labeling of L^A
            n' := c^{-1}(n)
            if f(n) = n' for some f ∈ Aut(L^A) then
                next_lattice2(m + 1, L^A)
    if m = n then output L
end
```

Algorithm 2

5. Vertically indecomposable modular lattices

We say a lattice L is *vertically decomposable* if it contains an element which is neither the greatest nor the least element of L but is comparable with every

element of L. A lattice which is not vertically decomposable is said to be *vertically indecomposable*.

Let $m^v(n)$ be the number of unlabeled vertically indecomposable modular lattices. Then the recursive formula [6] can be used to compute the number of unlabeled modular lattices from the number of unlabeled vertically indecomposable modular lattices.

$$m(n) = \sum_{k=2}^{n} m^v(k) \cdot m(n-k+1), \quad n \geq 2$$

In order to avoid generating vertically decomposable modular lattices, we only need to avoid using $lev_k(L)$ as a lattice antichain of L, since then $n \in L^A$ would be comparable to all elements in L^A. However, not using these lattice antichains would cut off branches of the canonical path that could potentially generate vertically indecomposable canonical lattices. Lemma 5.1 tells us we can safely avoid using them when $|lev_k(L)| = 1$.

Lemma 5.1. *Given an n-lattice L with only one atom $n-1$, then all descendants of $L^{\{n-1\}}$ are vertically decomposable.*

Proof. It is clear that $lev_k(L^A) = \{n\}$ and $lev_{k-1}(L^A) = \{n-1\}$, where $n \leq n-1$. Under our construction, only lattice-antichains $\{n\}$ and $\{n-1\}$ are considered. Therefore, for any descendant M of L^A and any new element $m \in M$ such that $m \notin L^A$ or $m = 0$, $m \leq n-1$. Additionally, for all $a \in L \setminus \{0\}$, $n-1 \leq a$. Thus, M is vertically decomposable. □

What this means is that we only construct vertically decomposable lattices where the only comparable element is a single atom. However, these are ignored during the count of vertically indecomposable lattices.

Note that in the last step, by Lemma 3.8, we only have to consider lattice-antichains in $lev_{k-1}(L)$.

6. A lower bound on the number of modular lattices

Let m_n denote the number of modular lattices of size n (up to isomorphism). In this section we give a simple argument for a lower bound of this sequence.

Theorem 6.1. *For all n, $2^{n-3} \leq m_n$.*

Proof. Let L_3 be the three element lattice with 0 and 1 as bottom and top respectively. Consider the following two extensions of an n-lattice L:

$$L_\alpha := L^A \text{ where } A = \{x \in L \mid x \succ 0\}$$

$$L_\beta := L^{\{b\}} \text{ for an arbitrary } b \text{ such that } b \succ n-1$$

We declare an n-lattice L to be an α-lattice or a β-lattice if it is obtained through the α or β construction respectively.

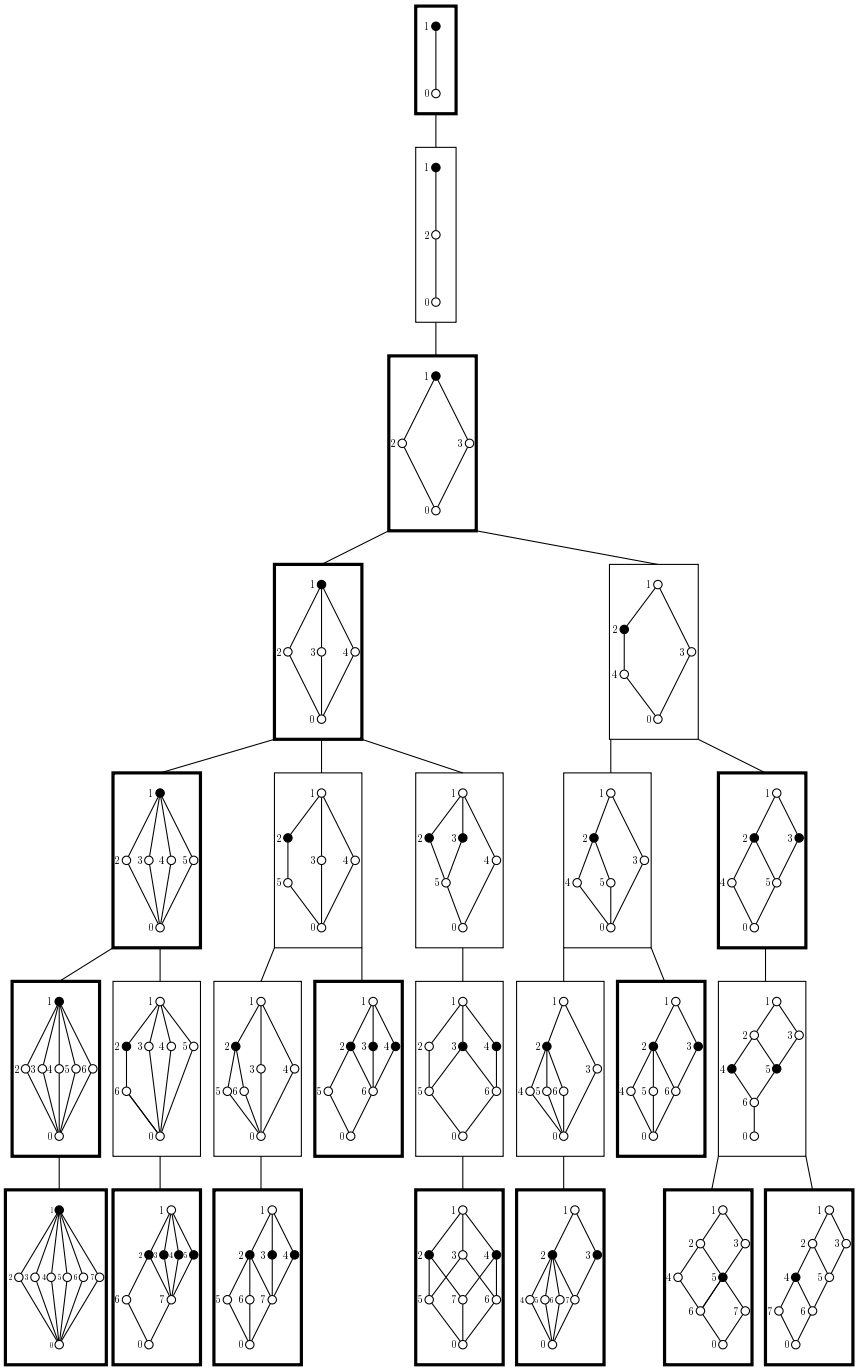

FIGURE 1. Example of the generation of all vertically indecomposable modular lattices up to size $n = 8$. Black circles indicate lattice antichain used in the previous step. Thick rectangles indicate vertically indecomposable modular lattices.

n	All Lattices	Semimodular	V.I. Semimod.	Modular	V.I. Modular
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	0	1	0
4	2	2	1	2	1
5	5	4	1	4	1
6	15	8	2	8	2
7	53	**17**	**4**	16	**3**
8	222	**38**	**9**	34	**7**
9	1 078	**88**	**21**	72	**12**
10	5 994	**212**	**53**	157	**28**
11	37 622	**530**	**139**	343	**54**
12	262 776	**1 376**	**384**	766	**127**
13	2 018 305	**3 693**	**1 088**	**1 718**	**266**
14	16 873 364	**10 232**	**3 186**	**3 899**	**614**
15	152 233 518	**29 231**	**9 596**	**8 898**	**1 356**
16	1 471 613 387	**85 906**	**29 601**	**20 475**	**3 134**
17	15 150 569 446	**259 291**	**93 462**	**47 321**	**7 091**
18	165 269 824 761	**802 308**	**301 265**	**110 024**	**16 482**
19	**1901910625578**	**2 540 635**	**990 083**	**256 791**	**37 929**
20		**8 220 218**	**3 312 563**	**601 991**	**88 622**
21		**27 134 483**	**11 270 507**	**1 415 768**	**206 295**
22		**91 258 141**	**38 955 164**	**3 340 847**	**484 445**
23				**7 904 700**	**1 136 897**

TABLE 1. Number of lattices and (vertically indecomposable = V.I.) (semi)modular lattices up to isomorphism. New numbers are in bold.

We want to show that, starting with L_3, in each step this construction generates two more modular lattices that are nonisomorphic to all other lattices in the collection.

It is clear that if L is a modular lattice, L_α is also modular since it is the same lattice with an element added at the bottom.

For L_β, we consider the cases where L is a modular α-lattice and a modular β-lattice obtained through this construction. If L is an α-lattice, then there are two atoms n and $n-1$ in L_β, both of which are covered by b, independently of the choice of b. Therefore, n and $n-1$ satisfy (lower) semimodularity. Since the new element n is not the common cover or co-cover of any two elements in L_β and L is modular, it follows that L_β is modular.

If L is a β-lattice, then there is only one choice of b (the element used in the previous step) since there is only one cover for $n-1$. Notice that the first β step starting from an α-lattice (or L_3) will generate an M_2 sublattice formed by the bottom element 0, the two atoms n and $n-1$ and the cover b used. After k successive β steps, there will be an M_{k+1} sublattice formed by 0, b, and the atoms $n-k,\ldots,n$. Therefore, after any β step, the new element will share a common cover (b) with all the atoms it shares a common co-cover (0) with, and vice versa. Since all other elements of L_β satisfy (lower) semimodularity by modularity of L, it follows that L_β is modular.

Next, we want to show that L_α and L_β are not isomorphic to any other lattice obtained from this construction. Consider two modular lattices L and

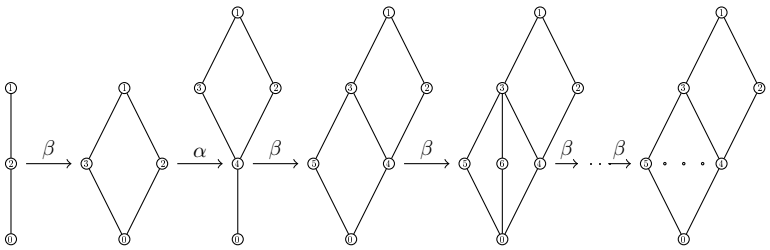

FIGURE 2. Example of a particular path of the $\alpha - \beta$ construction. Note that the α construction adds a single join-irreducible atom, and the β construction adds an extra element to the lower M_k sublattice.

M generated by the α-β construction. Since L_α has a unique atom and M_β has more than one atom, it follows that $L_\alpha \ncong M_\beta$.

Suppose $L_\alpha \cong M_\alpha$. Then, L and M can be reconstructed by removing an atom in L_α and M_α respectively. Clearly, $L \cong M$, since there is only one atom to be removed.

If $L_\beta \cong M_\beta$, there is more than one choice of atom. Let $k+1$ be the number of atoms in both, then they both have been obtained through k β steps, and the k atoms added through β steps are automorphic. Therefore removal of any of these atoms in L_β and M_β will generate two isomorphic modular lattices, hence $L \cong M$. Consequently no two non-isomorphic lattices can generate isomorphic lattices through the α-β construction.

We conclude the proof by induction. For $n = 3$, there are $2^{3-3} = 1$ modular lattices in the $\alpha-\beta$ construction (the initial L_3 lattice). Assume there are 2^{n-3} non-isomorphic modular lattices of size n constructed via the α-β construction. Then each of these lattices will produce 2 new modular lattices which are not isomorphic to any of the lattices produced by any other non-isomorphic lattice. As a result, there are 2^{n+1-3} non-isomorphic modular lattice of size $n+1$, thus completing the induction. □

Let l_n and d_n be the number of (nonisomorphic) lattices and distributive lattices of size n. Lower and upper bounds for these sequences are given in [7] and [3] respectively:

$$(2^{\sqrt{2}/4})^{(n-2)^{3/2}+o((n-2)^{3/2})} < l_n < 6.11343^{(n-2)^{3/2}}$$

$$1.81^{n-4} < d_n < 2.46^{n-1}.$$

The lower bound for modular lattices obtained in the preceding theorem can be improved slightly for $n > 7$ by counting a larger class of planar modular lattices. However, it seems that currently the best known upper bound for (semi)modular lattices is the same as the one for all lattices.

We would like to thank Michael Fahy and Nikos Hatzopoulus for help with operating the computing clusters. We also thank the Summer Undergraduate Research Fellowship at Chapman University for financial support.

References

[1] R. Belohlavek and V. Vychodil: Residuated lattices of size ≤ 12, Order 27 (2010), 147–161.
[2] R. Dedekind: Über die von drei Moduln erzeugte Dualgruppe, Math. Ann. 53 (1900), 371–403.
[3] M. Erné, J. Heitzig and J. Reinhold: On the number of distributive lattices, Electron. J. Combin. 9 (2002), 23 pp.
[4] I. A. Faradzhev, Constructive enumeration of combinatorial objects, Internat. Colloq. CNRS No.260, Combinatoire et Theorie des Graphes, Paris (1976), 131–135.
[5] G. Grätzer, "General Lattice Theory", 2nd edition, Birkhäuser, 1998.
[6] J. Heitzig and J. Reinhold: Counting finite lattices, Algebra Univers. 48 (2002) 43–53.
[7] D. J. Kleitman and K. J. Winston: The asymptotic number of lattices, Annals of Discrete Math. 6 (1980) 243–249.
[8] B. D. McKay: Isomorph-free exhaustive generation, J. Algorithms 26 (1998) 306–324.
[9] B. D. McKay and A. Piperno: Practical graph isomorphism, II, http://arxiv.org/abs/1301.1493, (2013), 22 pp.
[10] R. C. Read: Every one a winner, or how to avoid isomorphism search when cataloguing combinatorial configurations, Annals of Discrete Math. 2 (1978) 107–120.

Peter Jipsen
Chapman University, Orange, CA, United States
e-mail: jipsen@chapman.edu
URL: http://www1.chapman.edu/~jipsen

Nathan Lawless
Chapman University, Orange, CA, United States
e-mail: lawle108@mail.chapman.edu
URL: http://www.nlawless.com