An orderly algorithm to enumerate finite (semi)modular lattices

BLAST 2013

Peter Jipsen and Nathan Lawless

Chapman University

October 6, 2013

Outline

- The original algorithm: Generating all finite lattices
- Generating modular and semimodular lattices
- Lower bound on modular lattices
- Results for other classes of lattices
- Planar modular lattices of size n

Modular Lattices

- A **modular lattice** M is a lattice that satisfies the modular law for all $x, y, z \in M$:

$$x \geq z \text{ implies } x \wedge (y \vee z) = (x \wedge y) \vee z$$

or equivalently:

$$x \wedge [y \vee (x \wedge z))] = (x \wedge y) \vee (x \wedge z).$$

Modular Lattices

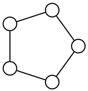
- A **modular lattice** M is a lattice that satisfies the modular law for all $x, y, z \in M$:

$$x \geq z \text{ implies } x \wedge (y \vee z) = (x \wedge y) \vee z$$

or equivalently:

$$x \wedge [y \vee (x \wedge z))] = (x \wedge y) \vee (x \wedge z).$$

- An alternative way to view modular lattices is by **Dedekind's Theorem**: *L is a nonmodular lattice iff N_5 can be embedded into L.*



N_5

Modular Lattices

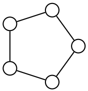
- A **modular lattice** M is a lattice that satisfies the modular law for all $x, y, z \in M$:

$$x \geq z \text{ implies } x \wedge (y \vee z) = (x \wedge y) \vee z$$

or equivalently:

$$x \wedge [y \vee (x \wedge z))] = (x \wedge y) \vee (x \wedge z).$$

- An alternative way to view modular lattices is by **Dedekind's Theorem**: *L is a nonmodular lattice iff N_5 can be embedded into L.*



N_5

- Standard examples of modular lattices are:

Modular Lattices

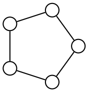
- A **modular lattice** M is a lattice that satisfies the modular law for all $x, y, z \in M$:

$$x \geq z \text{ implies } x \wedge (y \vee z) = (x \wedge y) \vee z$$

or equivalently:

$$x \wedge [y \vee (x \wedge z))] = (x \wedge y) \vee (x \wedge z).$$

- An alternative way to view modular lattices is by **Dedekind's Theorem**: *L is a nonmodular lattice iff N_5 can be embedded into L.*



N_5

- Standard examples of modular lattices are:
    - Lattices of subspaces of vector spaces.

Modular Lattices

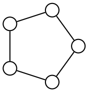
- A **modular lattice** M is a lattice that satisfies the modular law for all $x, y, z \in M$:

$$x \geq z \text{ implies } x \wedge (y \vee z) = (x \wedge y) \vee z$$

  or equivalently:

$$x \wedge [y \vee (x \wedge z))] = (x \wedge y) \vee (x \wedge z).$$

- An alternative way to view modular lattices is by **Dedekind's Theorem**: *L is a nonmodular lattice iff N_5 can be embedded into L.*



N_5

- Standard examples of modular lattices are:
  - Lattices of subspaces of vector spaces.
  - Lattices of normal subgroups of a group.

Modular Lattices

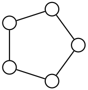
- A **modular lattice** M is a lattice that satisfies the modular law for all $x, y, z \in M$:

$$x \geq z \text{ implies } x \wedge (y \vee z) = (x \wedge y) \vee z$$

  or equivalently:

$$x \wedge [y \vee (x \wedge z))] = (x \wedge y) \vee (x \wedge z).$$

- An alternative way to view modular lattices is by **Dedekind's Theorem**: *L is a nonmodular lattice iff N_5 can be embedded into L.*



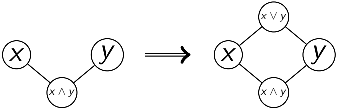
N_5

- Standard examples of modular lattices are:
  - Lattices of subspaces of vector spaces.
  - Lattices of normal subgroups of a group.
  - Lattices of ideals of a ring.

Semimodular Lattices

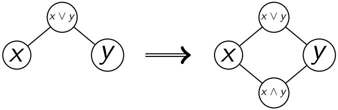
- A lattice L is **semimodular** if for all $x, y \in L$

$$x \wedge y \prec x, y \text{ implies that } x, y \prec x \vee y.$$



- A lattice L is **lower semimodular** if for all $x, y \in L$

$$x, y \prec x \vee y \text{ implies that } x \wedge y \prec x, y.$$



- **Theorem:** A finite lattice L is modular if and only if it is semimodular and lower semimodular.

Generating Finite Lattices

Heitzig and Reinhold [2000] developed an **orderly algorithm** to enumerate all finite lattices and used it to count the number of lattices up to size 18. To explain their algorithm, we recall some basic definitions:

- b is a **cover** of a if $a < b$ and there is no element c such that $a < c < b$, and denote this by $a \prec b$.

Generating Finite Lattices

Heitzig and Reinhold [2000] developed an **orderly algorithm** to enumerate all finite lattices and used it to count the number of lattices up to size 18. To explain their algorithm, we recall some basic definitions:

- b is a **cover** of a if $a < b$ and there is no element c such that $a < c < b$, and denote this by $a \prec b$.
- an element is an **atom** if it covers the bottom element.

Generating Finite Lattices

Heitzig and Reinhold [2000] developed an **orderly algorithm** to enumerate all finite lattices and used it to count the number of lattices up to size 18. To explain their algorithm, we recall some basic definitions:

- b is a **cover** of a if $a < b$ and there is no element c such that $a < c < b$, and denote this by $a \prec b$.
- an element is an **atom** if it covers the bottom element.
- $\uparrow A = \{x \in L \mid a \leq x \text{ for some } a \in A\}$ = the **upper set** of A.

Generating Finite Lattices

Heitzig and Reinhold [2000] developed an **orderly algorithm** to enumerate all finite lattices and used it to count the number of lattices up to size 18. To explain their algorithm, we recall some basic definitions:

- b is a **cover** of a if $a < b$ and there is no element c such that $a < c < b$, and denote this by $a \prec b$.
- an element is an **atom** if it covers the bottom element.
- $\uparrow A = \{x \in L \mid a \leq x \text{ for some } a \in A\}$ = the **upper set** of A.
- An **antichain** is a subset of L in which any two elements in the subset are incomparable.

Generating Finite Lattices

Heitzig and Reinhold [2000] developed an **orderly algorithm** to enumerate all finite lattices and used it to count the number of lattices up to size 18. To explain their algorithm, we recall some basic definitions:

- b is a **cover** of a if $a < b$ and there is no element c such that $a < c < b$, and denote this by $a \prec b$.
- an element is an **atom** if it covers the bottom element.
- $\uparrow A = \{x \in L \mid a \leq x \text{ for some } a \in A\}$ = the **upper set** of A.
- An **antichain** is a subset of L in which any two elements in the subset are incomparable.
- The set of all maximal elements in L is called the first level of L ($Lev_1(L)$). The **(m+1)-th level** of L can be recursively defined by $Lev_{m+1}(L) = Lev_1(L - \bigcup_{i=1}^{m} Lev_i(L))$.

Let A be an antichain of a lattice L. If A satisfies **(A1)**, we call it a **lattice-antichain**.

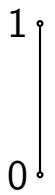
$$\textbf{(A1)} \quad \text{For any } a, b \in \uparrow A,\ a \wedge b \in \uparrow A \cup \{0\}.$$

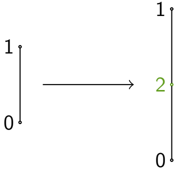
L^A is the poset constructed from L by adding an atom which is covered by all elements in A.

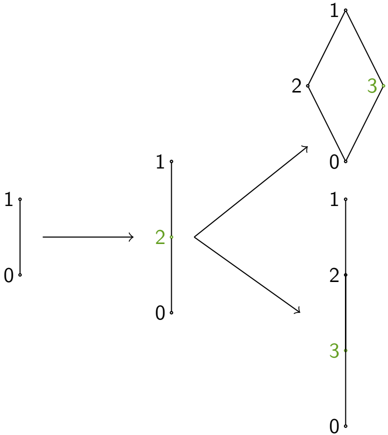
Lemma: [Heitzig, Reinhold 2000] L^A is a lattice iff A satisfies **(A1)**

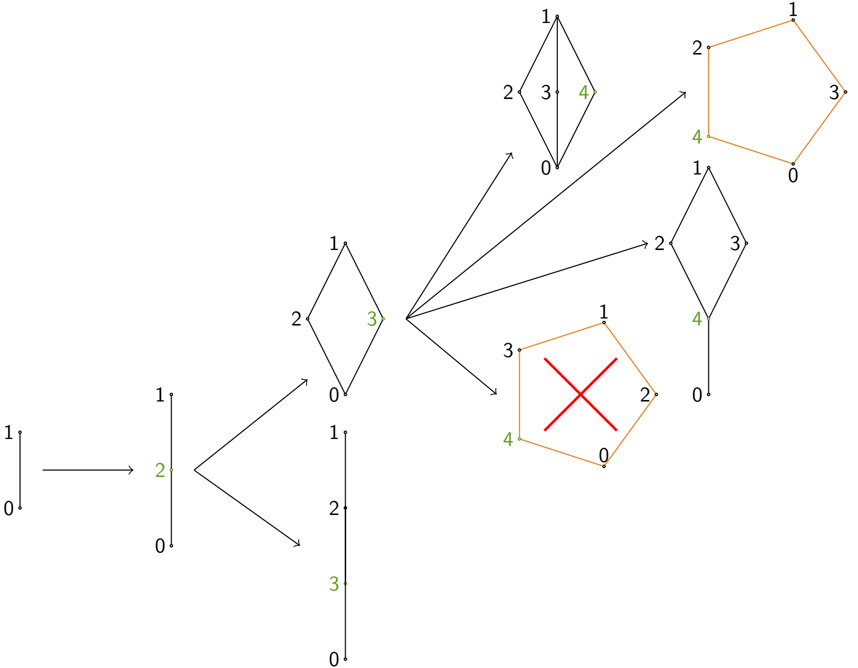
A recursive algorithm that generates for a given natural number $n \geq 2$ exactly all canonical lattices up to n elements starting with the two element lattice:

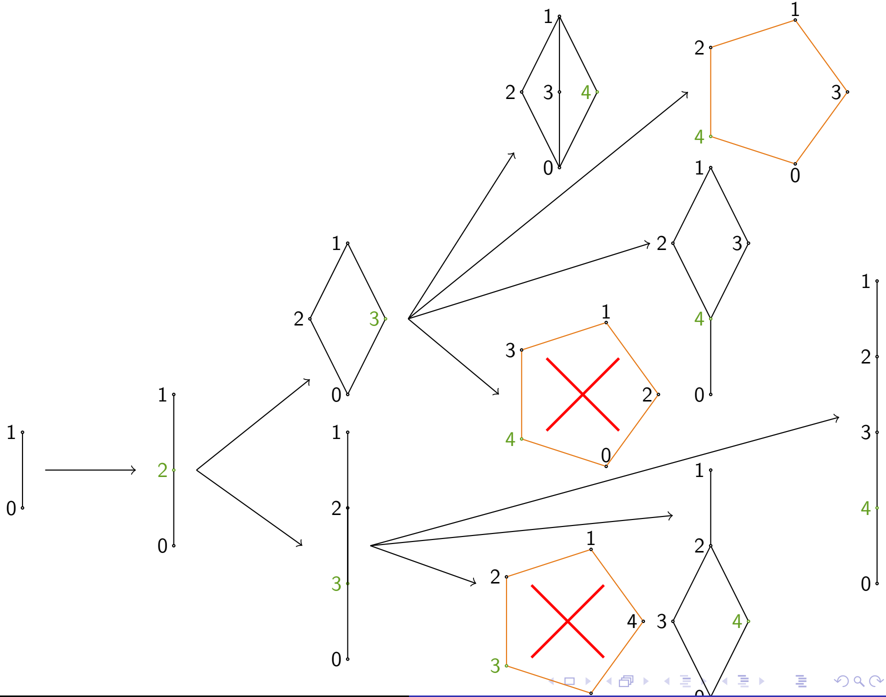
```
next_lattice(integer m, canonical m-lattice L)
begin
    if m < n then
        for each lattice-antichain A of L do
            if L^A is a canonical lattice then
                next_lattice (m+1, L^A)
    if m = n then output L
end
```


1
0


1
0
1
2
0


1
0
1
2
0
1
2
3
0
1
2
3
0


1
0
1
2
0
1
2
3
0
1
2
3
0
1
2
3
4
0
1
2
3
4
0
1
2
3
4
0
1
3
2
4
0



Dealing with Isomorphisms

- In order to select one isomorphic copy, a weight is defined for each lattice. If a lattice has the lowest weight among all it's permutations, it is considered canonical.
- However, this is an expensive check since it requires checking all permutations for each lattice (with some restrictions).
- The algorithm runtime can be improved by using a *canonical path extension*, introduced by McKay (1998):
  - Use only one (arbitrary) representative of each orbit in the lattice antichains of L.
  - When L^A is generated, perform a "canonical deletion". If L is automorphic to the generated lattice, then L^A is considered canonical.

Counting Finite Lattices: Semimodular Lattices

This algorithm can be modified such that when a lattice of size n is generated, the algorithm checks if it is (semi)modular.

Since semimodular and modular lattices are a very small fraction of all lattices, we present some results to reduce the search space of the algorithm. Here, $Lev_k(L)$ and $Lev_{k-1}(L)$ denote the bottom and second bottom levels of L respectively.

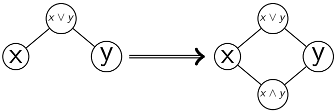
- **Semimodular Lattices Theorem:** When generating semimodular lattices, for a lattice L, we only consider antichains A which satisfy **(A1)** and all of the following conditions:
  (A2) $A \subseteq Lev_{k-1}(L)$ or $A \subseteq Lev_k(L)$.
  (A3) If $A \subseteq Lev_k(L)$, there are no atoms in $Lev_{k-1}(L)$.
  (A4) For all $x, y \in A$, x and y have a common cover.

Counting Finite Lattices: Modular Lattices

- **Modular Lattices Theorem:** When generating modular lattices, for a lattice L, we only consider antichains A which satisfy **(A1-4)** and

  (A5) If $A \subseteq Lev_k(L)$, $Lev_{k-1}(L)$ satisfies lower semimodularity

  (ie: for all $x, y \in Lev_{k-1}(L)$, $x, y \prec x \vee y$ implies $x \wedge y \prec x, y$)



Runtime Analysis

- Calculation of modular lattices of size n takes approximately 5.5 times the time used to generate all modular lattices of size $n-1$.
- In order to reach higher numbers, the algorithm was parallelized using the Message Passing Interface (MPI).
- Approximately **50 hours** were required to calculate all modular lattices of size 22 running the algorithm in parallel on 64 CPUs. It is estimated it would have taken **1 month** with the serial version.

n	# Lattices	# Semimod. Latt.	# Mod. Latt.
1	1	**1**	1
2	1	**1**	1
3	1	**1**	1
4	2	**2**	2
5	5	**4**	4
6	15	**8**	8
7	53	**17**	16
8	222	**38**	34
9	1,078	**88**	72
10	5,994	**212**	157
11	37,622	**530**	343
12	262,776	**1,376**	766

n	# Lattices	# Semimod. Latt.	# Mod. Latt.
1	1	**1**	1
2	1	**1**	1
3	1	**1**	1
4	2	**2**	2
5	5	**4**	4
6	15	**8**	8
7	53	**17**	16
8	222	**38**	34
9	1,078	**88**	72
10	5,994	**212**	157
11	37,622	**530**	343
12	262,776	**1,376**	766
13	2,018,305	**3,693**	**1,718**
14	16,873,364	**10,232**	**3,899**
15	152,233,518	**29,231**	**8,898**
16	1,471,613,387	**85,906**	**20,475**
17	15,150,569,446	**259,291**	**47,321**
18	165,269,824,761	**802,308**	**110,024**

n	# Lattices	# Semimod. Latt.	# Mod. Latt.
1	1	**1**	1
2	1	**1**	1
3	1	**1**	1
4	2	**2**	2
5	5	**4**	4
6	15	**8**	8
7	53	**17**	16
8	222	**38**	34
9	1,078	**88**	72
10	5,994	**212**	157
11	37,622	**530**	343
12	262,776	**1,376**	766
13	2,018,305	**3,693**	**1,718**
14	16,873,364	**10,232**	**3,899**
15	152,233,518	**29,231**	**8,898**
16	1,471,613,387	**85,906**	**20,475**
17	15,150,569,446	**259,291**	**47,321**
18	165,269,824,761	**802,308**	**110,024**
19	**1,901,910,625,578**	**2,540,635**	**256,791**

n	# Lattices	# Semimod. Latt.	# Mod. Latt.
1	1	**1**	1
2	1	**1**	1
3	1	**1**	1
4	2	**2**	2
5	5	**4**	4
6	15	**8**	8
7	53	**17**	16
8	222	**38**	34
9	1,078	**88**	72
10	5,994	**212**	157
11	37,622	**530**	343
12	262,776	**1,376**	766
13	2,018,305	**3,693**	**1,718**
14	16,873,364	**10,232**	**3,899**
15	152,233,518	**29,231**	**8,898**
16	1,471,613,387	**85,906**	**20,475**
17	15,150,569,446	**259,291**	**47,321**
18	165,269,824,761	**802,308**	**110,024**
19	**1,901,910,625,578**	**2,540,635**	**256,791**
20	–	**8,220,218**	**601,991**
21	–	**27,134,483**	**1,415,768**
22	–	**91,258,141**	**3,340,847**

n	# Lattices	# Semimod. Latt.	# Mod. Latt.
1	1	**1**	1
2	1	**1**	1
3	1	**1**	1
4	2	**2**	2
5	5	**4**	4
6	15	**8**	8
7	53	**17**	16
8	222	**38**	34
9	1,078	**88**	72
10	5,994	**212**	157
11	37,622	**530**	343
12	262,776	**1,376**	766
13	2,018,305	**3,693**	**1,718**
14	16,873,364	**10,232**	**3,899**
15	152,233,518	**29,231**	**8,898**
16	1,471,613,387	**85,906**	**20,475**
17	15,150,569,446	**259,291**	**47,321**
18	165,269,824,761	**802,308**	**110,024**
19	**1,901,910,625,578**	**2,540,635**	**256,791**
20	–	**8,220,218**	**601,991**
21	–	**27,134,483**	**1,415,768**
22	–	**91,258,141**	**3,340,847**
23	–	–	**7,904,700**
24	–	–	**18,752,942**

Lower Bound on Modular Lattices

- **Theorem:** The number of unique modular lattices of size n up to isomorphism is greater or equal to 2^{n-3}.
- **Outline of proof:** Let L_3 be the three element lattice with 0 and 1 as bottom and top respectively, and let $n-1$ the last element added. Consider the following two extensions of an n-lattice L:

  $L_\alpha = L^A$ where $A = \{x \in L \mid a \prec 0\}$

  $L_\beta =$

  $$\begin{cases} L^1 & \text{if } L = L_3 \\ L^{\{a\}} \text{ for an arbitrary } a \text{ such that } a \succ n-1 & \text{otherwise} \end{cases}$$

  Idea: Each modular lattice L will generate two unique modular lattices L_α and L_β.

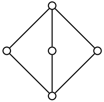
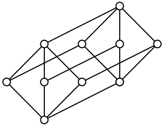
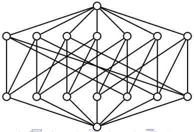
Open Question: Upper Bound on Modular Lattices?

Current upper bound is the upper bound for the number of all lattices up to isomorphism, which is approximately

$$6.112^{[(n-2)^{3/2}+o((n-2)^{3/2})]} \quad \text{(Kleitman, 1980)}$$

Alternative approach

- Finite distributive lattices have been counted up to size 49 (Erné, Heitzig, Reinhold 2002) using the duality with finite posets.
- It is possible to generate modular lattices in a similar way.
- Faigle and Herrmann [1981] define partially ordered geometries that are dual to finite length modular lattices.
- These are posets with a collection of subsets called *lines*, but it is not clear how efficiently nonisomorphic collections can be enumerated.
- Another approach is to use Herrmann's [1973] S-glued sums to build all modular lattices from products of projective subspace lattices.

M_3  $M_3 \times 2$  $Lat(\mathbb{Z}_2^3)$ 

Generating Other Lattices

- The algorithm for generating all lattices along with the implementation of the canonical path construction provides a tool to generate any type of lattice up to size 19, such as:
  1. Semidistributive lattices
  2. Weakly distributive lattices
  3. Almost distributive lattices
  4. 2-distributive lattices
  5. Self-dual lattices

Number of lattices compared

Modular, semidistributive, weakly distributive, 2-distributive and self-dual lattices compared to all lattices

s.i. = subdirectly irreducibles

n	Lattices	s.i.	Mod	s.i.	SD$\wedge$	s.i.	WD$\wedge$	s.i.	2-distr	sdual
5	5	2	4	1	4	1	4	1	5	3
6	15	4	8	1	9	0	10	1	15	7
7	53	16	16	1	23	1	29	5	53	13
8	222	69	34	2	65	3	96	12	222	36
9	1078	360	72	3	197	7	347	42	1075	76
10	5994	2103	157	4	636	11	1359	149	5952	232
11	37622	13867	343	7	2171	31	5679	551	37086	562
12	262776	100853	766	15	7756	89	25003	2160	256203	

Enumerating (finite) planar modular lattices

Quackenbush [1973] gave a characterization of planar lattices.
In the modular case this is just a sublattice of $C_m \times C_n$ with doubly irreducible elements added in any of the "squares"
For a vertically indecomposable planar modular lattice of size n:

1. Choose the number $k < n$ of squares; let $[k] = \{1, 2, \ldots, k\}$
2. Choose $u \in [k]^d$ for $0 < d \leq \lfloor (k+1)/2 \rfloor$ such that $\sum u_i = k$
3. Choose a vector $v \in \prod_{i=2}^{|u|} [\min(u_{i-1}, u_i) - 1]$
4. Let $m = n - 2k - 2 + \sum v_i$.
5. If $m \geq 0$ then choose $w \in \{0, \ldots, m\}^k$ s.t. $\sum w_i = m$

Theorem: For $m \geq 0$ the above data determines a unique planar modular lattice of size n with k squares arranged in $|u|$ (diagonal) columns of height $|u_i|$, shifted v_i and with w_j doubly irreducibles added to the j-th square.

This construction of planar modular lattices is very efficient

Probably can obtain a formula for the number of planar modular lattices of size n

All modular: 1,1,1,2,4,8,16,34,72,157,343,766,1718,3899,8898
Planar modular: 1,1,1,2,4,8,16,33,70,151,329,725,1613,3619,8176
s.i. planar mod: 1,1,0,0,1,1, 1, 2, 3, 4, 7, 15, 27, 49, 96

So about 92% of all modular lattices of size 15 are planar

What is the limit of (planar modular)/(all modular) as $n \to \infty$?

Grätzer and Quackenbush [2010] characterize the subdirectly irreducibles in the variety generated by all planar modular lattices

What is the limit of (s.i. planar modular)/(all planar modular) as $n \to \infty$?

References

1. R. Belohlavek and V. Vychodil, *Residuated Lattices of Size $\leq$ 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. J. Heitzig and J. Reinhold, *Counting Finite Lattices*, Algebra Univers. 48 (2002) 43–53.
5. D. J. Kleitman and K. J. Winston, *Combinatorial mathematics, optimal designs and their applications*, Ann. Discrete Math. 6 (1980), 243249.
6. B. D. McKay, *Isomorph-free exhaustive generation*, J. Algorithms 26 (1998) 306–324.

Preprint available on arXiv (search "jipsen lawless")