**Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)**

Scales behind computational intelligence: exploring properties of finite lattices

Radim Belohlavek, Vilem Vychodil
Dept. Computer Science, Palacky University, Tomkova 40, CZ-779 00, Olomouc, Czech Republic
Email: {radim.belohlavek, vilem.vychodil}@upol.cz

Dedicated to Professor Ivan Chajda on the occasion of his 60th birthday

***Abstract*— Finite lattices are fundamental structures which can be found in many fields of information science and computational intelligence: data analysis, data mining, multiple-valued logics, fuzzy logics, hierarchical modelling, graph theory, combinatorics, etc. Surprisingly, not much attention has been paid to structural properties of finite lattices. In this paper we study properties of finite lattices up to eleven elements. We present algorithms for generating non-isomorphic finite lattices up to a given size. We propose heuristic tests of non-isomorphism of finite lattices and examine their performance. We present a summary of selected properties of finite lattices.**

I. Introduction

Motivations and problem setting Ordered sets and lattices play crucial role in several areas of computer science, e.g. in visualization of data, data analysis, uncertainty modeling, many-valued and fuzzy logics [1], [6], graph theory, etc. In particular, lattices play the role of scales in various types of uncertainty theories. A favorite choice is the unit interval $[0, 1]$ which is used, for instance, in probability theory, Dempster-Shafer theory of evidence, fuzzy logics, etc. The elements $a \in [0, 1]$ are called degrees of probability, degrees of plausibility, degrees of truth, etc., respectively.

Linearly ordered scales, e.g. $[0, 1]$, are not always appropriate. Namely, degrees in the respective theories may not be comparable, see e.g. [8]. Natural arguments from the point of view of the respective theories lead to the requirement that a scale be a lattice, i.e. infima and suprema exist. Moreover, an important role is payed by "small" lattices. Namely, according to Miller's 7 ± 2 phenomenon well-known from psychology [10], humans are able to assign degrees in a consistent manner provided the scale of degrees contains up to 7 ± 2 elements. With more than 7 ± 2 elements, the assignments become inconsistent. Another argument supporting the importance of finite lattices comes from fuzzy logic applications. While using $[0, 1]$ is satisfactory in many cases, quite a lot of problems leads to infinite structures if $[0, 1]$ is used (consider just the simple fact that the set of all fuzzy sets in a finite universe is uncountable when $[0, 1]$ is used as a set of truth degrees). Quite often, a natural solution, which is computationally tractable, is to take a finite scale truth degrees instead of $[0, 1]$.

These facts bring us to finite lattices with a reasonably small number of elements (7 ± 2, perhaps a bit more). Surprisingly, little has been done in a systematic study of finite lattices and their properties. A systematic study of small lattices is the main topic of our paper. We are interested in fundamental questions related to finite lattices and their properties. For instance, how many non-isomorphic lattices with n elements are there? Can we generate lattices in an efficient way? Which properties of lattices are frequent for small lattices? Our paper answers several questions of this type. We study finite lattices with up to 11 elements. We present an efficient algorithm for generating all non-isomorphic finite lattices with a given number of elements. Furthermore, we analyze selected properties of all the lattices generated by our algorithm.

The paper is organized as follows. Section II presents preliminaries from partially ordered sets and lattices, Section III describes generation of finite lattices. In Section IV we present a summary of selected properties of generated lattices.

Related work Preliminary version of the method presented in this paper is described in [9]. In [7] the authors investigated numbers of finite lattices but they did not present any further analysis of properties of the generated lattices.

II. Preliminaries

A binary relation $R \subseteq U \times U$ in a set U is called a *partial order* (*in* U) if it is reflexive, antisymmetric, and transitive. If $\leq$ is a partial order in U, we write $a \leq b$ instead of $\langle a, b\rangle \in \leq$, and write $a < b$ if $a \leq b$ and $a \neq b$. If $\leq$ is a partial order in U, the pair $\mathbf{U} = \langle U, \leq\rangle$ is called a *partially ordered set.* Elements $a, b \in U$ are called *incomparable* (*in* $\mathbf{U} = \langle U, \leq\rangle$), written $a \parallel b$, if $a \not\leq b$ and $b \not\leq a$.

Let $\langle U, \leq\rangle$ be a partially ordered set and $A \subseteq U$. An element $a \in A$ is called a *least* element of A (with respect to $\leq$) if for each $b \in A$ we have $a \leq b$; a *greatest* element of A (with respect to $\leq$) if for each $b \in A$ we have $b \leq a$. If there exists a least or a greatest element of U, it is denoted by 0 or by 1, respectively. For each $A \subseteq U$ we define sets $\mathcal{L}(A), \mathcal{U}(A) \subseteq U$ as follows:

$$\mathcal{L}(A) = \{b \in U \,|\, b \leq a \text{ for each } a \in A\}, \tag{1}$$

$$\mathcal{U}(A) = \{b \in U \,|\, a \leq b \text{ for each } a \in A\}. \tag{2}$$

Set $\mathcal{L}(A)$ is called a *lower cone* of A (*in* $\langle U, \leq\rangle$), $\mathcal{U}(A)$ is called an *upper cone* of A (*in* $\langle U, \leq\rangle$). An element $b \in \mathcal{L}(A)$ is called a *lower bound of* A (*in* $\langle U, \leq\rangle$), an element $b \in \mathcal{U}(A)$ is called an *upper bound of* A (*in* $\langle U, \leq\rangle$).

If $\mathcal{L}(A)$ has a greatest element a, then a is called an *infimum* of A in $\langle U, \leq\rangle$, denoted by $\bigwedge A$. Dually, if $\mathcal{U}(A)$ has a least

element a, then a is called a *supremum* of A in $\langle U, \leq \rangle$, denoted by $\bigvee A$. Note that if the least element 0 of U exists then $\bigwedge U = 0 = \bigvee \emptyset$ by definition. Dually, if 1 (the greatest element of U) exists then $\bigwedge \emptyset = 1 = \bigvee U$.

A partially ordered set $\langle U, \leq \rangle$ is called *lattice ordered* if infimum and supremum exist for any two elements in U. Lattice ordered set $\langle U, \leq \rangle$ will be called a *lattice.* If $\langle U, \leq \rangle$ is a lattice with finite U, then $\langle U, \leq \rangle$ is called a *finite lattice.* If $A = \{a, b\}$ then $\bigwedge A$ and $\bigvee A$ will be denoted by $a \wedge b$ and $a \vee b$, respectively.

Remark 1: A partially ordered set $\langle U, \leq \rangle$ is called *completely lattice ordered* (a *complete lattice*) if infimum and supremum exist for any subset of U. It is a well-known fact that each finite lattice is complete, because for each subset $A = \{a_1, \ldots, a_n\} \subseteq U$, we have $\bigwedge A = a_1 \wedge a_2 \wedge \cdots \wedge a_n$, i.e. in case of finite lattices, the existence of infima for any two elements from U implies the existence of infima for any subset of U (and dually for suprema), see [2], [5].

From now on, we denote support sets of (finite) lattices by L (possibly with indices). Each finite lattice $\langle L, \leq \rangle$ has the least and greatest element of L. Indeed, for $L = \{a_1, a_2, \ldots, a_n\}$, the least element (denoted 0) equals $a_1 \wedge a_2 \wedge \cdots \wedge a_n$, the greatest element (denoted 1) equals $a_1 \vee a_2 \vee \cdots \vee a_n$, see [5]. If $\mathbf{L} = \langle L, \leq \rangle$ is a lattice, then $\mathbf{L}^{-1} = \langle L, \leq^{-1} \rangle$, where $\leq^{-1}$ is the inverse relation to $\leq$, is a lattice which is called a *dual lattice to* $\mathbf{L}$.

A partially ordered set $\langle U, \leq \rangle$ is said to be *linearly ordered* (or a *chain*) if for every $a, b \in U$ we have $a \leq b$ or $b \leq a$, i.e. every two elements are comparable. In this case, $\leq$ is called a *linear order.* Each chain is a lattice with $\wedge$ and $\vee$ given by

$$a \wedge b = \begin{cases} a & \text{if } a \leq b, \\ b & \text{otherwise,} \end{cases} \qquad a \vee b = \begin{cases} b & \text{if } a \leq b, \\ a & \text{otherwise.} \end{cases}$$

In this paper we are interested in general properties of lattices disregarding the "names of elements" the lattices consist of. Lattices which differ only in names of their elements will be treated as indistinguishable. In order to define the notion of an indistinguishability of lattices precisely, we use lattice isomorphisms. Let $\mathbf{L}_1 = \langle L_1, \leq_1 \rangle$ and $\mathbf{L}_2 = \langle L_2, \leq_2 \rangle$ be lattices. A mapping $h : L_1 \to L_2$ is called a *lattice isomorphism* (between $\mathbf{L}_1$ and $\mathbf{L}_2$) if (i) h is a bijection, and (ii) for each $a, b \in L_1$, we have

$$a \leq_1 b \quad \text{iff} \quad h(a) \leq_2 h(b). \tag{3}$$

Lattices $\mathbf{L}_1$ and $\mathbf{L}_2$ are said to be *isomorphic,* written $\mathbf{L}_1 \cong \mathbf{L}_2$, if there is a lattice isomorphism between $\mathbf{L}_1$ and $\mathbf{L}_2$. Throughout the paper, we will consider lattices "up to isomorphism" which means that we tacitly identify all isomorphic lattices.

Remark 2: A partially ordered set $\mathbf{U} = \langle U, \leq \rangle$ with finite U (that includes finite lattices) may be visualized using so-called *Hasse diagram*: each element $a \in U$ is depicted as a node (possibly labeled by "a") in such a way that if a is covered by b, i.e. $a < b$ and there is no $c \in U$ such that $a < c$ and $c < b$, then we connect the nodes of a and b, and put the node of a below the node of b. From properties of isomorphic lattices, we get that Hasse diagrams of two isomorphic lattices can be drawn the same way. Fig. 1 shows Hasse diagrams of all six-element lattices (up to isomorphism). The lattice in the lower-right corner of Fig. 1 is a six-element chain.

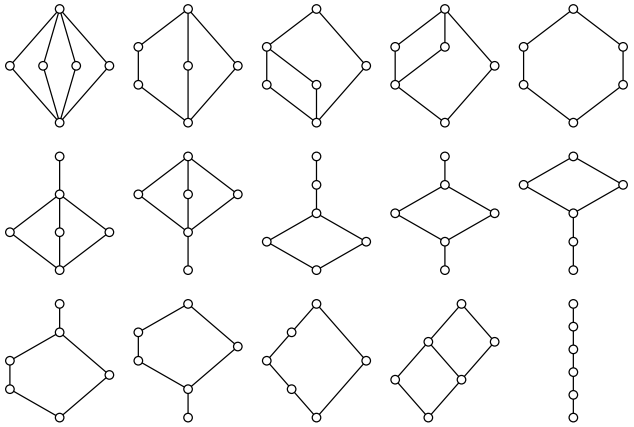


Fig. 1. Hasse diagrams of all six-element lattices (up to isomorphism).

We introduced lattices as partially ordered sets in which infima and suprema exist for any two elements. Alternatively, lattices can be described as algebras. An algebra $\mathbf{M} = \langle M, \sqcap, \sqcup \rangle$ with binary operations $\sqcap$ and $\sqcup$ in M is called a *lattice* if, for each $a, b, c \in M$, we have

$$\begin{aligned} a \sqcup b &= b \sqcup a, & a \sqcap b &= b \sqcap a, \\ a \sqcup (b \sqcup c) &= (a \sqcup b) \sqcup c, & a \sqcap (b \sqcap c) &= (a \sqcap b) \sqcap c, \\ a \sqcup (a \sqcap b) &= a, & a \sqcap (a \sqcup b) &= a. \end{aligned}$$

That is, $\mathbf{M} = \langle M, \sqcap, \sqcup \rangle$ is a lattice, if (i) both $\sqcap$ and $\sqcup$ are commutative; (ii) both $\sqcap$ and $\sqcup$ are associative; (iii) $\sqcap$ and $\sqcup$ mutually satisfy the law of absorption.

If $\mathbf{L} = \langle L, \leq \rangle$ is a lattice ordered set, put $a \sqcap b = a \wedge b$ and $a \sqcup b = a \vee b$. Then, $\langle L, \sqcap, \sqcup \rangle$ is a lattice. Conversely, if $\mathbf{M} = \langle M, \sqcap, \sqcup \rangle$ is a lattice, put $a \leq b$ iff $a \sqcap b = a$ (or, which is equivalent, iff $a \sqcup b = b$). Then $\langle M, \leq \rangle$ is a lattice ordered set such that $a \wedge b = a \sqcap b$ and $a \vee b = a \sqcup b$. Moreover, these constructions are mutually inverse, see [2], [5].

III. Generation of non-isomorphic finite lattices

In this section we propose a method for generation of non-isomorphic finite lattice of a given size. The process of generation of finite lattices includes several problems. First, we need an efficient representation of finite lattices that can be implemented in computers. Second, we need a procedure that generates lattices one by one so that for all isomorphic lattices, the procedure generates just one representative of them. To solve the latter problem, we need an efficient method for checking lattice isomorphism. All these issues will be discussed in this section. We start by introducing representation of finite lattices.

$\leq$	0	a	b	c	d	1
0	×	×	×	×	×	×
a		×			×	×
b			×	×	×	×
c				×		×
d					×	×
1						×

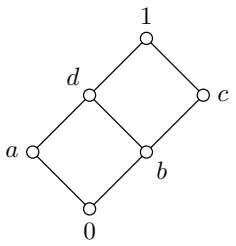


Fig. 2. Finite lattice (right) and its adjacency table (left).

A. Representation of finite lattices

A finite lattice $\mathbf{L} = \langle L, \leq \rangle$ is given by its support set L and a particular binary relation $\leq$ in L. If $\mathbf{L}$ is an n-element lattice, i.e. if $|L| = n$, then $\leq$ can be represented by an *adjacency table* (*adjacency matrix*) consisting of $n \times n$ entries. In more detail, we can consider a table with n rows and n columns corresponding to elements from L. Table entries are crosses "×" and blanks. Table entry given by row corresponding to element $a \in L$ and by column corresponding to element $b \in L$ contains "×" iff $a \leq b$. Hence, entry given by row corresponding to $a \in L$ and by column corresponding to $b \in L$ is blank iff $a \not\leq b$.

An adjacency table for $\mathbf{L} = \langle L, \leq \rangle$ is not uniquely given because we can have various adjacency tables that differ in the ordering of their rows and columns. In what follows we focus on adjacency tables which are in a special triangular form. Our representation will take advantage of the following assertion.

Theorem 3: Let $\mathbf{L} = \langle L, \leq \rangle$ be a finite lattice. Then there is a linear order $\preccurlyeq$ in L such that, for each $a, b \in L$,

$$a \leq b \quad \text{implies} \quad a \preccurlyeq b. \tag{4}$$

Proof: See [2], [5]. ■

Theorem 3 says that the lattice order $\leq$ can be extended to a linear order $\preccurlyeq$. Condition (4) ensures that $\leq$ is a subrelation of $\preccurlyeq$. Now, for each finite lattice $\mathbf{L} = \langle L, \leq \rangle$ we can consider an adjacency table of $\leq$ such that the rows and columns of the table are listed in the order given by $\preccurlyeq$ from Theorem 3. That is, a column denoted a precedes a column denoted b iff $a \preccurlyeq b$, and analogously for rows. For illustration, see Fig. 2. In Fig. 2, we have a Hasse diagram of a finite lattice with $L = \{0, a, b, c, d, 1\}$. An adjacency table whose rows and columns are ordered by a linear order $\preccurlyeq$ extending $\leq$ such that $0 \preccurlyeq a \preccurlyeq b \preccurlyeq c \preccurlyeq d \preccurlyeq 1$ is depicted in Fig. 2 (left). Let us note, that $\preccurlyeq$ defined e.g. by $0 \preccurlyeq b \preccurlyeq a \preccurlyeq d \preccurlyeq c \preccurlyeq 1$ would also satisfy the conditions of Theorem 3. Observe that since $\preccurlyeq$ extends $\leq$, i.e. condition (4) is true, the adjacency table is in the *upper triangular form.*

So far, we have observed that in order to represent $\leq$, it is enough to consider an adjacency table of $\leq$ in the upper triangular form. For $|L| = n$, the upper triangle contains exactly $\frac{n(n+1)}{2}$ elements. In addition to that, not all crosses in the upper triangular adjacency table carry interesting information. From properties of finite lattice orders, for each $a \in L$, we have $0 \preccurlyeq a$, $a \preccurlyeq a$, and $a \preccurlyeq 1$. Thus, the upper triangular adjacency table can be further reduced to non-trivial records only which are in case of our illustrative example contained in the gray area. In general, for an n-element lattice ($n \geq 4$), it is enough to keep track of

$$3 + \frac{n(n-5)}{2} \tag{5}$$

entries of the upper triangular adjacency table of $\leq$ (for each $|L| \leq 3$, there is exactly one lattice up to isomorphism). Such table entries can be encoded by a binary vector whose length is given by (5). For instance, the lattice from Fig. 2 can be represented by a binary vector 001110 (i.e., by a concatenation of binary vectors 001, 11, and 0 representing relevant bits from rows a, b, and c of the adjacency table).

Table in Fig. 3 shows lengths of binary vectors given by (5) that are used to represent n-element lattices as described above. Obviously, there are 2^k mutually different binary vectors of length k which encode 2^k mutually different binary relations some of which (not all of them) are the desired lattice orders.

size of L	1	2	3	4	5	6	7	8	9	10	11
vector length	0	0	0	1	3	6	10	15	21	28	36
possible relations	2^0	2^0	2^0	2^1	2^3	2^6	2^{10}	2^{15}	2^{21}	2^{28}	2^{36}

Fig. 3. Lengths of vectors encoding finite lattices up to 11 elements.

Our representation of lattices by binary vectors representing portions of upper triangular adjacency tables has several advantages. It is concise and it allows us to have efficient algorithms for determining $\leq$, $\wedge$, and $\vee$. For instance, the fact that $a \leq b$ is true (or not) can be checked in a constant time. Moreover, the upper triangular form ensures that if $a \prec b$ (i.e., if $a \preccurlyeq b$ and $a \neq b$), then $b \not\leq a$. Hence, the fact that $b \not\leq a$ can sometimes be decided even without looking in the adjacency table (binary vector).

Suprema and infima can be computed with asymptotic time complexity $O(n)$. Consider $L = \{a_1, \ldots, a_n\}$ and $\preccurlyeq$ such that $a_1 \preccurlyeq a_2 \preccurlyeq \cdots \preccurlyeq a_n$. Fig. 4 depicts algorithms for computing infima (meet) and suprema (join) of elements a_i and a_j provided that $a_i \preccurlyeq a_j$ (i.e., $i \leq j$). As we can see, due to our representation of $\leq$, it is not necessary to compute the cones given by (1) and (2) and then to determine their greatest and least elements—both the tasks are done simultaneously in less than n elementary steps (proof of soundness of the algorithms will be presented in a full version of this paper).

We conclude this subsection by showing that our representation of lattices is in fact a universal one. The following assertion shows that all n-element lattices (up to isomorphism) can be generated using a fixed L with $|L| = n$ and considering a fixed linear order $\preccurlyeq$ on L which serves as ordering of columns and rows of adjacency tables.

Theorem 4: Fix L with $|L| = n$ and let $\preccurlyeq$ be a linear order in L. Then for each n-element lattice $\mathbf{L}' = \langle L', \leq' \rangle$ there is a lattice order $\leq$ in L such that

```
procedure meet (a_i, a_j):
  if a_i ≤ a_j:
    return a_i
  else:
    for k from i − 1 downto 1:
      if a_k ≤ a_i and a_k ≤ a_j:
        return a_k

procedure join (a_i, a_j):
  if a_i ≤ a_j:
    return a_j
  else:
    for k from j + 1 upto n:
      if a_i ≤ a_k and a_j ≤ a_k:
        return a_k
```

Fig. 4. Implementation of "meet" and "join" operations.

(i) $\preccurlyeq$ extends $\leq$ and

(ii) $\mathbf{L}' = \langle L', \leq' \rangle$ is isomorphic to $\mathbf{L} = \langle L, \leq \rangle$.

Proof: Because of the limited scope of this paper, we present only a sketch of the proof. Denote $L = \{a_1, \ldots, a_n\}$ and $L' = \{a'_1, \ldots, a'_n\}$ and assume $a_1 \preccurlyeq a_2 \preccurlyeq \cdots \preccurlyeq a_n$. Take a linear order $\preccurlyeq'$ that extends $\leq'$. We can write $a'_{i_1} \preccurlyeq' a'_{i_2} \preccurlyeq' \cdots \preccurlyeq' a'_{i_n}$, where $\{i_1, \ldots, i_n\} = \{1, \ldots, n\}$. Define mapping $h \colon L \to L'$ by $h(a_j) = a'_{i_j}$ $(j = 1, \ldots, n)$. Finally, for $a_j, a_k \in L$, put $a_j \leq a_k$ iff $a'_{i_j} \leq' a'_{i_k}$. Now it is routine to check that h is a lattice isomorphism between $\mathbf{L}' = \langle L', \leq' \rangle$ and $\mathbf{L} = \langle L, \leq \rangle$. ∎

Due to Theorem 4, each n-element lattice is isomorphic to a lattice ordered set in a fixed L that can be encoded by a binary vector as described above. Thus, if we generate all non-isomorphic lattices on L represented by binary vectors (for a fixed $\preccurlyeq$), we get all n-element lattices (up to isomorphism).

B. Characteristic vectors of finite lattices

In the previous section we discussed a representation of lattices. In this section we turn our attention to important properties of lattices which will be used to quickly distinguish non-isomorphic lattices. Isomorphism tests generally belong to hard problems. For instance, the problem of graph isomorphism is NP-complete. In this paper we propose a method which can help speed up tests of non-isomorphism. With each finite lattice we associate its characteristic vector which somehow encodes properties of the lattice which are shared by all its isomorphic copies but which are highly unlikely to be shared between non-isomorphic lattices. In this section we describe the characteristic vectors and in the next section we show a heuristic test of non-isomorphism based on these characteristic vectors.

Consider a finite lattice $\mathbf{L} = \langle L, \leq \rangle$ and a linear order $\preccurlyeq$ extending $\leq$. We define a subset $\mathcal{P}(L)$ of $\preccurlyeq$ by

$$\mathcal{P}(L) = \{\langle a, b \rangle \in \preccurlyeq \mid a \neq 0 \text{ and } b \neq 1 \text{ and } a \neq b\}. \quad (6)$$

Observe that $\mathcal{P}(L)$ is related to the non-trivial part of the upper triangular adjacency table of $\leq$. Namely, $\mathcal{P}(L)$ is exactly the set of pairs $\langle a, b \rangle$ which are encoded in the binary vector representing $\leq$ (see previous subsection).

Using $\mathcal{P}(L)$, for each $a \in L$, we define four non-negative integers $v_1(a), \ldots, v_4(a)$ characterizing properties of $a \in L$:

$$v_1(a) = |\mathcal{L}(\{a\})| = |\{b \in L \mid b \leq a\}|, \quad (7)$$

$$v_2(a) = |\mathcal{U}(\{a\})| = |\{b \in L \mid a \leq b\}|, \quad (8)$$

$$v_3(a) = |\{\langle b, c \rangle \in \mathcal{P}(L) \mid a = b \wedge c\}|, \quad (9)$$

$$v_4(a) = |\{\langle b, c \rangle \in \mathcal{P}(L) \mid a = b \vee c\}|. \quad (10)$$

By definition, $v_1(a)$ and $v_2(a)$ represent the numbers of elements which are lower/greater than or equal to a. Values of $v_3(a)$ and $v_4(a)$ represent the numbers of non-trivial pairs of elements from L whose infimum/supremum gives a. Thus, non-negative integers $v_1(a), \ldots, v_4(a)$ represent some properties of $a \in L$ which are, roughly speaking, given by the "position" of $a \in L$ in the lattice. Moreover, for each $a \in L$, we consider a tuple of integers as follows:

$$v(a) = \langle v_1(a), v_2(a), v_3(a), v_4(a) \rangle. \quad (11)$$

Note that values of $v_i(a)$ and $v(a)$ depend on the lattice order $\leq$ on L, i.e. two different $\leq_1$ and $\leq_2$ on L can yield different values of $v_i(a)$ and $v(a)$. In order to make $\mathbf{L} = \langle L, \leq \rangle$ explicit, we denote $v_i(a)$ and $v(a)$ by $v_i^{\mathbf{L}}(a)$ and $v^{\mathbf{L}}(a)$.

Example 5: Recall the finite lattice from Fig. 2. The values of v_i's can be expressed by a table of the following form:

$\mathbf{L}$	0	a	b	c	d	1
v_1	1	2	2	3	4	6
v_2	6	3	4	2	2	1
v_3	2	1	3	0	0	0
v_4	0	0	0	1	3	2

Columns of the table correspond to elements from L. Rows of the table are labeled by $v_1, \ldots, v_4$. A table entry corresponding to v_i and a contains the value of $v_i(a)$. For instance, we can read from the table that $v_1(0) = 1$, $v_2(0) = 6$, $v_3(0) = 2$, $v_4(0) = 0$, i.e. $v(0) = \langle 1, 6, 2, 0 \rangle$; $v_1(c) = 3$, $v_2(c) = 2$, $v_3(c) = 0$, $v_4(c) = 1$, i.e. $v(c) = \langle 3, 2, 0, 1 \rangle$, etc.

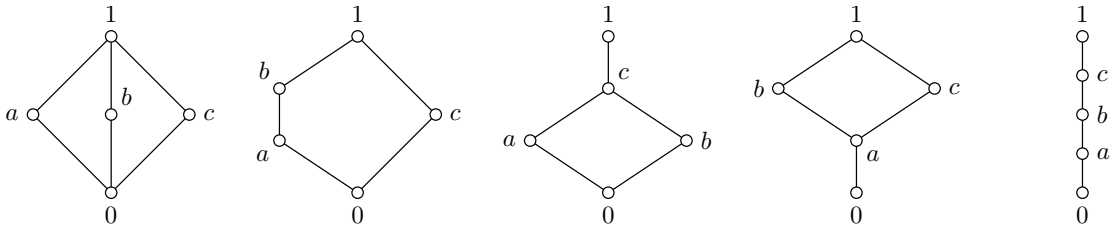
Vectors $v(a)$ will be used by the heuristic non-isomorphism test that will be introduced in next section. In order to determine equality of such vectors (for all $a \in L$), we sort these vectors according to their lexical ordering. In more detail, we can define a *lexical* (*linear*) *order* $\leqslant_{\text{lex}}$ on four-tuples of integers as follows. For $x = \langle x_1, x_2, x_3, x_4 \rangle \in \mathbb{Z}^4$ and $y = \langle y_1, y_2, y_3, y_4 \rangle \in \mathbb{Z}^4$ we put $x \leqslant_{\text{lex}} y$ iff either $x = y$ (tuples are identical) or there is $i \in \{1, \ldots, 4\}$ such that, for each $j < i$, $x_j = y_j$ and $x_i < y_i$. One can easily verify that $\leqslant_{\text{lex}}$ is indeed a linear order on $\mathbb{Z}^4$. Since $v(a)$ and $v(b)$ given by (11) are also four-tuples of integers, we either have $v(a) \leqslant_{\text{lex}} v(b)$ or $v(b) \leqslant_{\text{lex}} v(a)$.

We now introduce the characteristic vectors. A *characteristic vector of a finite lattice* $\mathbf{L} = \langle L, \leq \rangle$ is a vector of integers given by concatenation of vectors $v(a)$ $(a \in L)$ listed in the lexical order $\leqslant_{\text{lex}}$.

Example 6: (i) In case of lattice from Fig. 2, we can see that $v(0) \leqslant_{\text{lex}} v(a) \leqslant_{\text{lex}} v(b) \leqslant_{\text{lex}} v(c) \leqslant_{\text{lex}} v(d) \leqslant_{\text{lex}} v(1)$. That is, the characteristic vector is a concatenation of vectors $v(0)$, $v(a)$, $v(b)$, $v(c)$, $v(d)$, and $v(1)$ in this order:

$\langle 1, 6, 2, 0, 2, 3, 1, 0, 2, 4, 3, 0, 3, 2, 0, 1, 4, 2, 0, 3, 6, 1, 0, 2 \rangle$.

(ii) Fig. 5 depicts all five-element lattices together with tables describing values of v_i's. The columns of the tables are already



$\mathbf{L}_{5,1}$	0	a	b	c	1
v_1	1	2	2	2	5
v_2	5	2	2	2	1
v_3	3	0	0	0	0
v_4	0	0	0	0	3

$\mathbf{L}_{5,2}$	0	c	a	b	1
v_1	1	2	2	3	5
v_2	5	2	3	2	1
v_3	2	0	1	0	0
v_4	0	0	0	1	2

$\mathbf{L}_{5,3}$	0	a	b	c	1
v_1	1	2	2	4	5
v_2	5	3	3	2	1
v_3	1	1	1	0	0
v_4	0	0	0	3	0

$\mathbf{L}_{5,4}$	0	a	b	c	1
v_1	1	2	3	3	5
v_2	5	4	2	2	1
v_3	0	3	0	0	0
v_4	0	0	1	1	1

$\mathbf{L}_{5,5}$	0	a	b	c	1
v_1	1	2	3	4	5
v_2	5	4	3	2	1
v_3	0	2	1	0	0
v_4	0	0	1	2	0

Fig. 5. All five-element lattices (up to isomorphism) and their characteristic vectors written in tables.

listed in the lexical order $\leqslant_{\text{lex}}$. For instance, characteristic vector of $\mathbf{L}_{5,2}$ is given by concatenation of $v(0)$, $v(c)$, $v(a)$, $v(b)$, and $v(1)$, i.e. $\langle 1,5,2,0,2,2,0,0,2,3,1,0,3,2,0,1,5,1,0,2\rangle$.

The problem of determining the characteristic vector of a given n-element lattice is tractable. In fact, there is an algorithm with asymptotic complexity $O(n^3)$ that can do the job. Hint: traversing trough the binary vector is done in $O(n^2)$ steps, each step requires computation of infima and suprema, which can be done in $O(n)$ steps; finally, an efficient sorting algorithm like heap-sort can be used to sort vectors with respect to $\leqslant_{\text{lex}}$ in $O(n \log n)$ steps. We postpone full description of the algorithm to a full version of this paper.

C. Heuristic tests of non-isomorphism

A direct procedure to test whether two n-element lattices are isomorphic, which is given by the definition of an isomorphism, leads to an algorithm with asymptotic time complexity $O(2^n)$, because in worst case, it is necessary to generate $n!$ bijective mappings between two n-element lattices and to check whether some of them is an isomorphism. In this section we propose a procedure which can quickly disqualify most of non-isomorphic lattices, avoiding thus going through all $n!$ bijections. The proposed procedure heuristically tests *non-isomorphism* of lattices. In general, it cannot be used to decide whether two lattices *are* isomorphic. In the latter case, the brute force checking of bijections between lattices must occur. However, the advantage of our procedure is that even if we are compelled to the checking of bijections, we can restrict ourselves only to "isomorphism candidates" which can be read of the characteristic vectors. The number of isomorphism candidates is in most cases much smaller than $n!$. We begin with the following

Theorem 7: Let $\mathbf{L}_1$ and $\mathbf{L}_2$ be lattices, $h\colon L_1 \to L_2$ be a lattice isomorphism. Then, for each $a \in L_1$, we have

$$v^{\mathbf{L}_1}(a) = v^{\mathbf{L}_2}(h(a)). \tag{12}$$

As a consequence, two isomorphic finite lattices have the same characteristic vectors.

Proof: The claim can be proved by checking (7)–(10). The proof is omitted due to a limited scope of this paper. ■

An immediate consequence of Theorem 7 is that two finite lattices having different characteristic vectors cannot be isomorphic. Thus, a *quick non-isomorphism test* can be done by checking the inequality of characteristic vectors. If the characteristic vectors of $\mathbf{L}_1$ and $\mathbf{L}_2$ are equal, in general we cannot tell whether $\mathbf{L}_1 \cong \mathbf{L}_2$ or not. Nevertheless, in the next section we will see that for lattices up to 11 elements it is highly unlikely to have the same characteristic vectors for non-isomorphic lattices.

Suppose the characteristic vectors of $\mathbf{L}_1$ and $\mathbf{L}_2$ are equal and both $\mathbf{L}_1$ and $\mathbf{L}_2$ are defined on the same universe set L linearly ordered with a fixed $\preccurlyeq$. In order to confirm/deny the isomorphism of $\mathbf{L}_1$ and $\mathbf{L}_2$, it is not necessary to go through all permutations of L (bijections $h\colon L \to L$) because Theorem 7 says that the elements corresponding under any isomorphism of $\mathbf{L}_1$ and $\mathbf{L}_2$ must have the same values of $v(\cdots)$, see (12). Thus, it suffices to generate and check only permutations satisfying (12). We call such permutations isomorphism candidates: a permutation $h : L \to L$ is called an *isomorphism candidate* if (12) is satisfied for each $a \in L_1$.

Example 8: (i) Consider the lattice from Fig. 2 and the table containing values of its characteristic vector from Example 5. All columns of the table in Example 5 are pairwise distinct. Therefore, in case we would arrive at a lattice with the same characteristic vector as the lattice from Fig. 2, then, in order to decide their isomorphism, it suffices to check just one bijection (just one isomorphism candidate).

(ii) In case of lattice $\mathbf{L}_{5,3}$ from Fig. 5 we would have to check two isomorphism candidates, because two columns in the corresponding table are equal. In case of $\mathbf{L}_{5,1}$, we would have to check $3! = 6$ candidates, because three columns of the table are equal, etc. Of course, in case of five-element lattices, we can see that the non-isomorphic lattices have pairwise different characteristic vectors. Hence, strictly speaking, no checking of candidates is necessary. However, for lattices with $|L| \geq 8$, there are situations when the heuristic test can fail as we will see later (i.e., checking all isomorphism candidates is necessary if $|L| \geq 8$).

The heuristic test of non-isomorphism takes two lattices $\mathbf{L}_1 = \langle L, \leq_1\rangle$ and $\mathbf{L}_2 = \langle L, \leq_2\rangle$ as its input, and produces an answer "true" (may or may not be isomorphic) or "false" (not

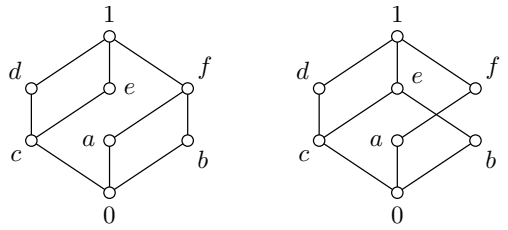


Fig. 6. Eight element lattices that fail the heuristic non-isomorphism test.

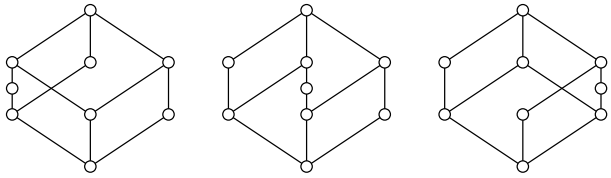


Fig. 7. Nine element lattices that fail the heuristic non-isomorphism test.

isomorphic). The test can be summarized as follows:

(i) Count 1's in the binary vector encoding $\mathbf{L}_1$ and in the binary vector encoding $\mathbf{L}_2$. If the numbers of 1's in both the vectors are different, then return "false" immediately. Otherwise go to step (ii).
(ii) Determine characteristic vectors of $\mathbf{L}_1$ and $\mathbf{L}_2$. Return "true" if the vectors coincide; return "false" otherwise.

Denote the output of the heuristic test for $\mathbf{L}_1$ and $\mathbf{L}_2$ by $\mathrm{Iso}_H(\mathbf{L}_1, \mathbf{L}_2)$. Clearly, if $\mathrm{Iso}_H(\mathbf{L}_1, \mathbf{L}_2)$ equals "false" then, due to our previous observations, we know that $\mathbf{L}_1$ and $\mathbf{L}_2$ are not isomorphic. If $\mathrm{Iso}_H(\mathbf{L}_1, \mathbf{L}_2)$ equals "true", we employ the exact test of isomorphism. The exact test accepts as its input the lattices together with their characteristic vector (which was computed by the previous use of the heuristic test). The output of the exact test is "false" (not isomorphic) "true" (isomorphic). The exact test goes as follows:

(i) Initialize the generator of isomorphism candidates given by the characteristic vector. Proceed with step (ii).
(ii) If there are no more isomorphism candidates left to check, return "false". Otherwise, take first isomorphism candidate h and go to step (iii).
(iii) For each $a, b \in L_1$, check condition (3). If the condition is true for each $a, b \in L_1$, return "true". Otherwise go to step (ii).

Now, we will be interested in situations when the heuristic test fails. By a *failure of the heuristic test* we mean a situation, when $\mathrm{Iso}_H(\mathbf{L}_1, \mathbf{L}_2)$ is "true" for lattices $\mathbf{L}_1$ and $\mathbf{L}_2$ which are *not isomorphic*. In a situation like this, the heuristic test alone is not sufficient to decide isomorphism of lattices. We will be interested in the frequency of these "pathological situations". First, let us show that such situations really do occur.

Example 9: (i) Consider eight-element lattices from Fig. 6 (denote the left-hand side lattice by $\mathbf{L}_1$ and the right-hand side one by $\mathbf{L}_2$). Both the lattices have the same characteristic vector. Namely, the following table contains entries of the characteristic vector:

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	2	5	15	53	220	1049	5682	34502
2	0	0	0	0	0	0	0	1	13	125	1159
3	0	0	0	0	0	0	0	0	1	18	212
4	0	0	0	0	0	0	0	0	0	2	28
5	0	0	0	0	0	0	0	0	0	0	6
6	0	0	0	0	0	0	0	0	0	0	4

Fig. 8. Numbers of characteristic vectors of given orders.

$\mathbf{L}_1$	0	a	b	c	d	e	f	1
$\mathbf{L}_2$	0	a	b	c	d	f	e	1
v_1	1	2	2	2	3	3	4	8
v_2	8	3	3	4	2	2	2	1
v_3	10	1	1	3	0	0	0	0
v_4	0	0	0	0	1	1	4	10

As we can see, two pairs of elements have the same columns, i.e. the exact test of isomorphism would go through $2! \cdot 2! = 4$ isomorphism candidates. Each of the four candidates will be a mapping $h\colon L_1 \to L_2$ where $h(0) = 0$, $\{h(a), h(b)\} = \{a, b\}$, $h(c) = c$, $\{h(d), h(e)\} = \{d, f\}$, $h(f) = e$, and $h(1) = 1$. Thus, we either have $h(a) = a$ and $h(b) = b$, or $h(a) = b$ and $h(b) = a$. In either case, h cannot be an isomorphism: (i) if $h(a) = a$, then we get $a \leq_1 f$ and $h(a) = a \not\leq_2 e = h(f)$, which violates (3); (ii) if $h(a) = b$, then $h(b) = a$, i.e. we get $b \leq_1 f$ and $h(b) = a \not\leq_2 e = h(f)$, which again violates (3). Hence, $\mathbf{L}_1$ and $\mathbf{L}_2$ are not isomorphic while having the same characteristic vector. Let us note that $\mathbf{L}_1$ and $\mathbf{L}_2$ are the only eight-element lattices (up to isomorphism) that fail the heuristic test and, at the same time, are the least lattices (with respect to the numbers of their elements) that fail the test.

(ii) Fig. 7 shows three non-isomorphic nine-element lattices which share the same characteristic vector. Hence, any two distinct lattices from the three depicted in Fig. 7 would fail the heuristic test.

A question is, whether or not it is rare that two lattices fail the heuristic test. We have investigated this problem for the generated lattices with up to 11 elements. Suppose we have a characteristic vector c of some finite lattice. By an *order of* c, denoted $||c||$, we mean the number of pairwise non-isomorphic lattices whose characteristic vector is exactly c. For instance, we have $||c|| = 2$ for the characteristic vector c from Example 9 (i), and $||c|| = 3$ for c from Example 9 (ii). If $||c|| = 1$, there is just one finite lattice (up to isomorphism) with c in which case the heuristic test never fails.

For small sizes of lattices, we have that $||c|| = 1$ for each characteristic vector, i.e. the isomorphism can be decided by the heuristic test. The table in Fig. 8 shows the numbers of characteristic vectors of given orders. The columns of the table correspond to sizes of lattices, rows correspond to orders of characteristic vectors, and table entries show how many characteristic vectors (of orders given by rows and sizes given by columns) there are. We can see that for $n \leq 7$, there are only vectors of order 1.

size of L	8	9	10	11
probability	$4.06 \cdot 10^{-5}$	$2.75 \cdot 10^{-5}$	$1.06 \cdot 10^{-5}$	$2.94 \cdot 10^{-6}$

Fig. 9. Probability of failure of the heuristic test.

To sum up, for $|L| \leq 7$ the heuristic test never fails. For $8 \leq |L| \leq 11$ the heuristic test can fail, but the probability of failure (see the probability values in table in Fig. 9) is almost negligible and seems to be going down with the increasing size of L (at least this is true for $8 \leq |L| \leq 11$).

D. Generation of finite lattices: main algorithm

Non-isomorphic finite lattices of a given size are generated in the following way. First, we create an *initial lattice* which is encoded by a binary vector (see Section III-A) containing all 0s and add it to a list of generated lattices. A partially ordered set given by this binary vector is indeed a lattice which, for $|L| = n$, contains an antichain of $n - 2$ elements (i.e., $n - 2$ elements of the lattice are incomparable). For $|L| = 6$, the initial lattice is the lattice in the upper-left corner of Fig. 2. Then we go through all possible binary vectors of a given length. For each of the vectors we check if it represents a lattice which has not yet been generated. If so, we add the lattice to the set of generated lattices and continue with next binary vector. The procedure goes on until we generate a *final lattice,* which is the lattice encoded by the binary vector full of 1s. Such a vector encodes an n-element chain. For instance, in case of $|L| = 6$, it is the lattice in the lower-right corner of Fig. 2. During the lattice generation, we use the isomorphism tests described in Section III-C.

Our method of generation of lattices can be described by the following recursive procedure:

1) Set binary vector for the initial n-element lattice.
2) Check if the current binary vector represents a lattice. If it represents a lattice order $\leq$, go to step 3). Otherwise, go to step 4).
3) Check if $\leq$ (lattice order represented by the current binary vector) is isomorphic to a lattice which has already been generated—use the heuristic and exact tests described in Section III-C.
   - If $\leq$ is not isomorphic to any of the generated lattices, add $\leq$ to the set of generated lattices and go to step 4).
   - If $\leq$ is isomorphic to some previously generated $\leq'$ and if in addition $\leq$ equals $\leq'$ (i.e., $\leq$ has already been found), then end this branch of recursion.
   - Otherwise (i.e., $\leq$ is isomorphic to some previously generated $\leq'$ but $\leq$ differs from $\leq'$), go to step 4).
4) Loop over all bits of the binary vector $\leq$ which equal 0:
   a) Make a copy $\leq'$ of $\leq$ (copy of binary vectors).
   b) Set to 1 the current bit in $\leq'$ which equals 0.
   c) Make transitive closure of $\leq'$.
   d) Recursively call 2) for $\leq'$.

The numbers of the generated non-isomorphic lattices are in the table in Fig. 10 (row denoted "all lattices"). These results agree with observations concerning the numbers of lattices from [7].

	1	2	3	4	5	6	7	8	9	10	11
all lattices	1	1	1	2	5	15	53	222	1078	5994	37622
modular	1	1	1	2	4	8	16	34	72	157	343
distributive	1	1	1	2	3	5	8	15	26	47	82
complemented	1	1	0	1	2	6	18	71	307	1594	9446
boolean	1	1	0	1	0	0	0	1	0	0	0
relat. compl.	1	1	0	1	1	1	1	2	2	4	6
pseudo-compl.	1	1	1	2	4	10	29	99	391	1775	9214
rel. ps. compl.	1	1	1	2	3	5	8	15	26	47	82

Fig. 10. Numbers of non-isomorphic lattices with selected properties.

Remark 10: (i) Our test of isomorphism which is based on the heuristic and exact tests seems to be very efficient (at lest for lattices up to 11 elements). For instance, generation of 9-element lattices using our isomorphism tests takes under 2 minutes while the same algorithm which uses only the exact test needs over 8 hours to do the same job.

(ii) An interesting thing to note is the average number of isomorphism candidates that are used during each isomorphism test. Recall that when the heuristic test is positive, we must decide the isomorphism by finding an appropriate bijection between lattices (this is a part of the exact test). Using characteristic vectors, we can restrict ourselves only to certain bijections, so-called isomorphism candidates described in Section III-C. The average number of tested isomorphism candidates is surprisingly low: during generation of 9-element lattices, we check approximately 9 isomorphism candidates per each of the 10 isomorphism tests. That means less than one isomorphism candidate per one isomorphism test.

(iii) Not each binary vector encoding $\leq$ represents a lattice order. Nevertheless, in case of $|L| \leq 5$ it happens that each partial order encoded by a binary vector of the appropriate length is a lattice order. The shortest binary vector encoding a partial order on $|L| = 6$ which is not a lattice order is 011110.

IV. Selected properties of the generated lattices

In this section we briefly present summary of properties of the generated lattices we focused on. The first type of properties which may be of interest and which can be seen directly from Hasse diagrams of lattices are their heights and widths. By a *height* (or *width*) of a lattice we mean the length of the longest maximal chain (or antichain) contained in that lattice. From the generated lattices we have observed that the numbers of lattices with a given width and height follows a normal distribution. The situation for $|L| = 11$ is depicted in Fig. 11. From Fig. 11 we can read that there is exactly one lattice with height 11 (an 11-element chain), and exactly one lattice with width 9 (initial 11-element lattice, see Section III-D). The table in Fig. 12 shows average characteristics of the generated lattices. Rows of the table correspond to properties (see the legend in Fig. 12, for the notions involved we refer to [2], [5]), columns of the table correspond to sizes of lattices. Table entries are the average values. An interesting observation is that the average number of irreducible elements increases

	1	2	3	4	5	6	7	8	9
3	0	0	0	0	0	0	0	0	1
4	0	0	0	0	123	159	72	15	0
5	0	0	83	2212	3294	1138	126	0	0
6	0	0	2295	8464	4387	518	0	0	0
7	0	164	4413	5339	973	0	0	0	0
8	0	374	2133	805	0	0	0	0	0
9	0	217	280	0	0	0	0	0	0
10	0	36	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0

Fig. 11. Numbers of 11-element lattices of given height(row) and width(col.).

	1	2	3	4	5	6	7	8	9	10	11
ht	1.00	2.00	3.00	3.50	4.00	4.47	4.91	5.30	5.67	6.00	6.31
wd	1.00	1.00	1.00	1.50	2.00	2.40	2.75	3.09	3.41	3.74	4.06
at	0.00	1.00	1.00	1.50	1.80	2.00	2.15	2.28	2.40	2.51	2.61
ir	1.00	2.00	3.00	3.50	4.20	4.93	5.60	6.25	6.88	7.51	8.12
pr	1.00	2.00	3.00	3.50	3.40	3.27	3.15	2.97	2.78	2.60	2.43
mc	1.00	1.00	1.00	1.50	2.00	2.47	3.00	3.58	4.22	4.91	5.66
ma	1.00	2.00	3.00	3.50	4.00	4.67	5.43	6.36	7.46	8.77	10.32

Fig. 12. Average characteristics of lattices (legend: ht = avg. height of lattices, wd = avg. width of lattices, at = avg. number of (co)atoms, ir = avg. number of meet/join irreducible elements, pr = avg. number of meet/join prime elements, mc/ma = avg. number of maximal chains/antichains.

with the increasing size of L while the average number of prime elements drops down (this may be surprising because both the notions are defined in a similar way, see [5]).

The table in Fig. 10 gives a summary of the numbers of non-isomorphic lattices satisfying additional conditions: modularity, distributivity, existence of complements, Boolean property (distributivity and existence of complements), existence of relative complements, pseudo-complements, and relative pseudo-complements. These properties are of interest, e.g., when lattices are considered as structures of truth values in multiple-valued logics and fuzzy logics.

The table in Fig. 10 shows the numbers of lattices having each property but does not show, e.g., how many modular lattices are pseudo-complemented. Such information can be found in the table in Fig. 13. Here, columns denote properties considered in Fig. 10, the left-most column contains numbers of lattices with a given combination of properties. Each row of the table represents one combination of properties (properties which are present are marked by $\times$). From table in Fig. 13 we can see that some combinations of properties are relatively

cnt.	MOD	DIS	COM	BOO	REL	PCO	RPC
4	×	×	×	×	×	×	×
8	×		×		×		
8			×		×		
187	×	×				×	×
204	×						
236	×					×	
447			×			×	
10653						×	
10980			×				
22267							

Fig. 13. Groups of properties shared by lattices.

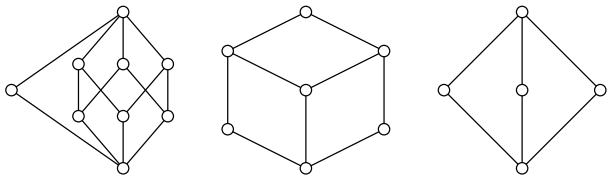


Fig. 14. Least lattices that have a specific group of properties.

rare. In addition to that, some combinations of properties are not shared by "small" lattices (up to certain number of elements). For instance, the least lattice which is only relatively complemented and (in consequence) complemented has 9 elements and it is depicted in Fig. 14 (left). The least lattice which does not satisfy any of the properties MOD–RPC (see Fig. 13) has 7 elements and is depicted in Fig. 14 (middle). The lattice in Fig. 14 (right) is the least lattice which is only modular, complemented, and relatively complemented.

V. Conclusion and future research

We have presented a method for generation of finite lattices up to a given size. The generated lattices were used for a preliminary exploration of their quantitative properties (their average heights, widths, numbers of atoms, etc.). Our database can be used to find finite lattices satisfying certain conditions which can be further used in applied and theoretical research. In our future work we will focus of the following topics:

- incremental algorithms that use $(n-1)$-element lattices to generate n-element lattices;
- exploration of further properties of the generated lattices;
- improvements of heuristic tests of non-isomorphism;
- combination of our approach with other methods, see [7];
- generation of lattices extended by additional operations (e.g., residuated lattices, see [1], [6]).

A database of generated lattices is available at:
`http://vychodil.inf.upol.cz/res/devel/finlat/`

Acknowledgment

Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

References

[1] Belohlavek R.: *Fuzzy Relational Systems: Foundations and Principles*. Kluwer, Academic/Plenum Publishers, New York, 2002.
[2] Birkhoff G.: *Lattice Theory.* Publ. AMS, Providence, RI (1967).
[3] Carpineto C., Romano G.: *Concept Data Analysis. Theory and Applications.* J. Wiley, 2004.
[4] Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
[5] Gratzer G. A.: *General Lattice Theory.* Birkhauser (1998, 2nd ed.).
[6] Hájek P.: *Metamathematics of Fuzzy Logic.* Kluwer, Dordrecht, 1998.
[7] Heitzig J., Reinhold J.: Counting Finite Lattices. *Algebra Universalis* **48**(1)2002, 43–53.
[8] Kramosil I.: *Probabilistic Analysis of Belief Functions.* Springer, New York, 2001.
[9] Kure M.: *Computer-aided study of finite posets.* UP Olomouc (MSc. thesis, in Czech), 2004.
[10] Miller G. T.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *The Psychological Review* **63**(1956), 81–97.