Swiss Federal Institute of Technology Lausanne

Department of Physics

A program for computing lattices applied to the orthogonal geometry

Stage in Computer Science

Patrik Ferrari

in collaboration with Dr. Remo Moresi

Research Center for Mathematics and Physics, Locarno

Locarno

July - October 1999

Contents

1 Mathematical basis

A *partially ordered set*[1], briefly *poset*, is a set L with a relation $\leq$ satisfying the following properties:

1. Reflexivity: $a \leq a$
2. Antisymmetry: $a \leq b$ and $b \leq a$ imply that $a = b$
3. Transitivity: $a \leq b$ and $b \leq c$ imply that $a \leq c$

where a, b, c are elements of L.

We introduce two binary operations:

1. Infimum[2] $\wedge : L^2 \to L : a, b \mapsto a \wedge b \doteqdot \inf\{a, b\}$
2. Supremum[3] $\vee : L^2 \to L : a, b \mapsto a \vee b \doteqdot \sup\{a, b\}$

They satisfy the following properties:

1. Idempotency: $a \wedge a = a$ and $a \vee a = a$
2. Commutativity: $a \wedge b = b \wedge a$ and $a \vee b = b \vee a$
3. Associativity: $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ and $(a \vee b) \vee c = a \vee (b \vee c)$
4. Absorption identity: $a \wedge (a \vee b) = a$ and $a \vee (a \wedge b) = a$

where a, b, c are elements of L.
The absorption identity can be derived using: $a \leq b \iff a \wedge b = a$ or $a \vee b = b$.

Definition 1.1 (Lattice) *A poset $\langle L; \leq \rangle$ is called a lattice if* $\sup\{a, b\}$ *and* $\inf\{a, b\}$ *exist for all $a, b \in L$.*

Remark 1.2 *Equivalently, a lattice can be defined as an algebra:*
An algebra $\langle L; \wedge, \vee \rangle$ is called a lattice iff: L is a non void set, $\wedge$ and $\vee$ are binary operations on L satisfying idempotency, commutativity, associativity and absorption.

In what follows we will assume that all (semi)lattices have universal bounds 0, 1.

A finite lattice can always be described by the infimum and the supremum table. An alternative way is to describe the partial ordering, that is, all pairs $\langle x, y \rangle$ with $x \leq y$. We can simplify the list of these pairs and list only the covering relations. We say that a covers b or b is covered by a iff $a > b$ and for no x, $a > x > b$.

If we use only one of the binary operations, then we only have semilattices. They are called meet (join) semilattices if we use only the $\wedge$ ($\vee$) operation.

[1] We used [1] for general reference concerning lattice theory
[2] It is also called *meet*.
[3] It is also called *join*.

Definition 1.3 (Modular lattice) *A lattice L is called modular if it satisfies the following condition:*

$$a \leq c \Rightarrow (a \vee b) \wedge c = a \vee (b \wedge c) \quad \forall a, b, c \in L \tag{1}$$

Definition 1.4 (Distributive lattice) *A lattice L is called distributive if it satisfies the following condition:*

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \quad \forall a, b, c \in L \tag{2}$$

Remark 1.5 *Each distributive lattice is a modular lattice.*

Definition 1.6 (Galois (meet semi)lattice) *A Galois (meet semi)lattice is a (meet semi)lattice L with a unary operation $\perp$ that satisfies the following property:*

$$x \leq (x^\perp \wedge y)^\perp \quad \forall x, y \in L \tag{3}$$

Remark 1.7 *We will assume that all Galois (semi)lattices are non-degenerate in the sense that $1^\perp = 0$.*

Remark 1.8 *The condition (3) on the orthogonal operation is also equivalent to one of the following properties:*

1. $x \leq y \Rightarrow y^\perp \leq x^\perp$ *and* $y \leq x^\perp \Rightarrow x \leq y^\perp$
2. $x \leq y \Rightarrow y^\perp \leq x^\perp$ *and* $x \leq x^{\perp\perp}$

An element x is *closed* if $x = x^{\perp\perp}$. All the elements of the form $x^\perp$ are closed. If L is a Galois lattice, we can derive another identity:

$$(x \vee y)^\perp = x^\perp \wedge y^\perp \tag{4}$$

Definition 1.9 *Let L be a lattice and $x, y \in L$ such that they are in relation, i.e. that $x \leq y$ (or $y \leq x$). The interval y/x (resp. x/y) is defined by $y/x \doteqdot \{z \in L \mid x \leq z \leq y\}$*

Definition 1.10 (Index function) *Let L be a lattice, let I be the set of intervals of L and let Γ be the set of cardinal numbers $\leq \aleph_0$. $\delta : I \to \Gamma$ is called an index function of type $\aleph_0$ if the following properties are satisfied $\forall x, y, z \in L$:*

1. $\delta(x/y) \geq \delta(x \wedge z/y \wedge z)$.
2. $\delta(x/y) \geq \delta(x \vee z/y \vee z)$.
3. $\delta(x/y) \geq \delta(y^\perp/x^\perp)$.
4. $\delta(x/y) + \delta(y/z) = \delta(x/z)$.
5. $\delta(x/y) = 0 \iff x = y$.

Definition 1.11 (Hermitian lattice of type $\aleph_0$) *A Hermitian lattice of type* $\aleph_0$ *is a structure* $\langle L, 0, 1, \wedge, \vee, \perp, b, \delta \rangle$ *such that:*

1. $\langle L, 0, 1, \wedge, \vee, \perp \rangle$ *is a modular Galois lattice with universal bounds* 0, 1.
2. $b \in L$ *is a nullary operation with*

$$x \wedge x^{\perp} \leq b \quad \forall x \in L \tag{5}$$

3. δ *is an index function of type* $\aleph_0$.

The induced structure on the meet semilattice (without considering the operation $\vee$) gives the concept of a Hermitian (meet)semilattice of type $\aleph_0$.

2 The program

2.1 The program's goal

The program's goal is the computation of a given kind of (semi)lattices starting from a set of generators and relations.

2.2 Coding of the lattice elements

The lattice generators are coded by a small alphabetical letter (e.g. “a”,“b”,“x”,“y”), the orthogonal operation by the big letter “T”, the infimum by “$<$” and the supremum by “$>$”.
An element of a lattice is given by a sequence of letters of the generators and the operations. I used the Polish coding, a code that does not need parenthesis. I choose this code because it is easier to treat it with a computer program. More precisely we have:

- the orthogonal of “x”, $x^\perp$, is coded by “Tx”,
- the infimum between “x” and “y”, $x \wedge y$, is coded by “$<xy$” and
- the supremum between “x” and “y”, $x \vee y$, is coded by “$>xy$”.

To obtain the elements in the standard form starting from the coded one, it is sufficient to read it from right to left. If there is a “T” then we create the orthogonal of the element and if there are two elements (preceded by a binary operation), we write them by leaving a little space and then we add the operator between them (e.g. $<bT>aTb$ is equal to $(b^\perp \wedge a)^\perp \wedge b$).
An element can be written in different ways, because of the associativity and the commutativity properties, the absorption identity and other relations depending on the particularities of the lattice. For example $a \wedge a^\perp =< aTa$ is equal to $a^\perp \wedge a =< Taa$. Then an element corresponds to a *class of words* and not to *one* word. The representative is the one that is lexicographically the biggest, where the order relation in the alphabet set is “$<$”, “$>$”, “T”, “z”, “y”, $\dots$, “b”, “a” (from the least to the biggest). The choice of the operation symbols is related to the ASCII code. In fact “$<$” corresponds to 60, “$>$” to 62, “T” to 84 and the small letters are between 97 and 122.
When two elements are in relation, $x \leq y$, the program creates a pair of elements in which the first is x and the second is y (see 2.4).

2.3 The coding of the index function

The index function assigns to each lattice interval a cardinal number, that can be finite or $\aleph_0$ (see 1.10).
It is used for determining the effects of fixing the finiteness of an interval on a lattice. But the added condition does not determine the nature of all the lattice intervals.

This is the reason why the coding of indices allows three possibilities:

1. if an interval is finite, it is coded using a string formed by D plus a number (e.g. $D1$, $D2$, ...),
2. if the interval cardinality is $\aleph_0$, then it is coded by $A0$,
3. if the interval finiteness is not determined, then it is coded by ND.

The coding is necessary because only the finiteness of an interval is known and not his numeric value.

2.4 The structure of the objects

For the program, I used several lists of elements and pairs (see 2.5). There are two kinds of basic objects:

1. `PElemento`: an object with two fields:
   (a) `numero`: a pointer, used to identify the element.
   (b) `nome`: a string, which is the name of the element.
2. `PCoppia`: an object with three fields:
   (a) `numero`: a pointer used to identify the pair.
   (b) `coppia`: an array of strings. It contains the pairs of elements that are in relation.
   (c) `indice`: a string, used to save the coded index.

`PElemento` is used for the collections of elements and indices, and `PCoppia` is used for the lists of elements and indices pairs.

2.5 The used lists

In order to achieve the program's goal I need the following lists[4]:

- a list of elements: `Insieme`,
- a list of relations between the elements (i.e. a list of pairs): `ListaCoppie`,
- a list of substitutions of elements (e.g. if $\delta(a/0) < \aleph_0$ then $a^{\perp\perp} = a$): `ListaSost`,
- a list of substitutions of words determined by the commutativity and the associativity laws (e.g. $a \wedge a^{\perp}$ is replaced by $a^{\perp} \wedge a$): `ListaRin`,
- a list of pairs of indices: `IndexCoppie`,
- a list of substitutions of indices: `IndexSost`,
- a list of the eliminated elements: `ElEliminati`.

[4]The names in `typewriter font` correspond to the names of the lists

I also need some basic procedures in order to:

- create and/or eliminate pairs and elements from the lists,
- join two lists of elements or pairs,
- print the elements and the pairs on files.

2.6 How to obtain the equivalence of elements

The determination of the equivalence of two elements is achieved in two different ways:

1. using the pairs: if we find the pairs (x, y) and (y, x) (i.e. $x \leq y$ and $y \leq x$) then we know that $x = y$. Or if we find a pair (TTx, x), using the relation $x \leq x^{\perp\perp}$ then we know that $x = x^{\perp\perp}$,
2. using the indices: if we find an equation for finite indices, like $\delta_i = \delta_i + \delta_j$, then $\delta_j = 0$ and the corresponding elements are equivalents (e.g. $\delta_j = \delta(v/w) = 0 \Rightarrow v = w$).

2.7 The program "skeleton"

In this sub-chapter the program's main structure is shortly described.

Repeat until the number of elements is unchanged:

- Operation (orthogonal, infimum or supremum): create new elements and pairs.
- Union of the lists of elements and pairs.
- Create the new pairs, depending on the kind of lattice:
  - if the lattice is modular: use the relation (1),
  - if it is a Galois lattice: use the relation (4),
  - if the (semi)lattice is Hermitian: use the relation (5),
  - if it is a Galois (semi)lattice: use the relation $x \leq x^{\perp\perp}$,
  - use the transitivity property of the binary operations.
- `Transitivita`[5]: repeated until the number of pairs is unchanged. It creates new pairs using the transitivity property.
- `Equivalenze`[6]: repeated until the number of elements is unchanged:
  - `FunzioneIndice`: creates new indices (if $\delta(x/y) < \aleph_0$ then all the subintervals of x/y have finite indices), creates new indices relations using the properties 1. to 3. of the index function, deletes the indices that are no longer used.

[5]The names of the procedures and functions are written in `typewriter` font.

[6]There are used the properties listed in 2.6

  - `ControllaEquivalenze`: controls the equivalence of elements using the pairs of elements *ListaCoppie*[7].
  - `EquivalenzeIndici`: controls the equivalence of indices using the pairs of indices *IndexCoppie*.
  - `Equazioni`: searches the equivalences of elements using *IndexCoppie*.
  - `Riduci`: does the substitutions in the elements of *Insieme* and in the pairs of *ListaCoppie*, substitutes the indices and deletes double elements and pairs.

- `OrdinaElementi`: Orders the elements of *Insieme*.
- `SostSost`: substitution only in the first field of *coppia* (see 2.4) of *ListaSost* (the list of substitutions).
- Creates a reserve file from which it is possible to resume the computation.

[7]The names of lists are written in *italic* font.

3 Some results

The computer program was used to compute several (semi)lattices. On the one hand, in order to test the program, various known (semi)lattices were recomputed; on the other hand, some new semilattices were computed. In this chapter we present a test example (see example 1) and three examples of new semilattices (see examples 2 to 4). The second and third examples were obtained without using the index function, whereas the fourth needs this function in order to be finite. The third example is discussed by Moresi in [5]. All examples are homomorphic images of the free Hermitian semilattice generated by a single element. We chose in each case some conditions, because the free semilattice is infinite (see [3]).

In the second example the elements are written below in the classical form. In the other ones the elements are written in the coded form, except for $a^\perp, b^\perp, a^{\perp\perp}$. They are also written like an orthogonal or a biorthogonal of an element or an infimum of two elements.

3.1 Example one

This is the most general Hermitian (meet)semilattice generated by a satisfying the following conditions:

1. $a \leq b$
2. $b = b^{\perp\perp}$
3. $\delta(1/b) < \aleph_0$

There are 34 elements in the semilattice, as shown in figure 1, page 14:

1: $\mathbf{0}$	18: $T<TaTTa = 5^\perp$
2: $\mathbf{1}$	19: $T<TbTT<aTa = 6^\perp$
3: $<TTaT<bT<aTa = a^{\perp\perp} \wedge 24$	20: $T<TbTTa = 7^\perp$
4: $<TaT<bTa = a^\perp \wedge 27$	21: $T<aTa = 8^\perp$
5: $<TaTTa = a^\perp \wedge a^{\perp\perp}$	22: $T<aTb = 9^\perp$
6: $<TbTT<aTa = b^\perp \wedge 29$	23: $T<b<TaT<bTa = 1^\perp$
7: $<TbTTa = b^\perp \wedge a^{\perp\perp}$	24: $T<bT<aTa = 11^\perp$
8: $<aTa = a \wedge a^\perp$	25: $T<bT<bT<aTa = 12^\perp$
9: $<aTb = a \wedge b^\perp$	26: $T<bT<bTa = 13^\perp$
10: $<b<TaT<bTa = b \wedge 4$	27: $T<bTa = 14^\perp$
11: $<bT<aTa = b \wedge 27$	28: $T<bTb = 15^\perp$
12: $<bT<bT<aTa = b \wedge 26$	29: $TT<aTa = 8^{\perp\perp}$
13: $<bT<bTa = b \wedge 27$	30: $TTa = a^{\perp\perp}$
14: $<bTa = b \wedge a^\perp$	31: $Ta = a^\perp$
15: $<bTb = b \wedge b^\perp$	32: $Tb = b^\perp$
16: $T<TTaT<bT<aTa = 3^\perp$	33: a
17: $T<TaT<bTa = 4^\perp$	34: b

The relation between the indices of the intervals are the following:

$$
\begin{array}{lllllllll}
\delta_1 & \doteqdot & \delta(\mathbf{1}/22) & = & \delta(9/\mathbf{0}) & & & & \\
\delta_2 & \doteqdot & \delta(22/19) & = & \delta(6/9) & & & & \\
\delta_3 & \doteqdot & \delta(19/20) & = & \delta(21/16) & = & \delta(3/29) & = & \delta(7/6) \\
\delta_4 & \doteqdot & \delta(20/28) & = & \delta(16/25) & = & \delta(18/23) & = & \delta(a^\perp/26) \\
 & = & \delta(13/a^{\perp\perp}) & = & \delta(10/5) & = & \delta(12/3) & = & \delta(15/7) \\
\delta_5 & \doteqdot & \delta(28/b) & = & \delta(25/11) & = & \delta(23/17) & = & \delta(26/14) \\
 & = & \delta(27/13) & = & \delta(4/10) & = & \delta(24/12) & = & \delta(b^\perp/15)
\end{array}
$$

3.2 Example two

This is the most general Hermitian (meet)semilattice generated by a that satisfy the following conditions:

1. $b \leq a$,
2. $a = a^{\perp\perp}$ and $b = b^{\perp\perp}$, i.e. a and b are closed.

There are 22 elements in the semilattice, as shown in figure 2, page 15:

1: $\mathbf{0}$	12: $\left(\left(a \wedge b^\perp\right)^\perp \wedge \left(a \wedge \left(a \wedge b^\perp\right)^\perp\right)^\perp\right)^\perp = 4^\perp$
2: $\mathbf{1}$	13: $\left(b^\perp \wedge \left(a \wedge b^\perp\right)^\perp\right)^\perp = 5^\perp$
3: $\left(a \wedge b^\perp\right)^\perp \wedge \left(b^\perp \wedge \left(a \wedge b^\perp\right)^\perp\right)^\perp = 16 \wedge 13$	14: $\left(a \wedge \left(a \wedge b^\perp\right)^\perp\right)^\perp = 6^\perp$
4: $\left(a \wedge b^\perp\right)^\perp \wedge \left(a \wedge \left(a \wedge b^\perp\right)^\perp\right)^\perp = 16 \wedge 14$	15: $\left(a \wedge \left(b \wedge b^\perp\right)^\perp\right)^\perp = 7^\perp$
5: $b^\perp \wedge \left(a \wedge b^\perp\right)^\perp = b^\perp \wedge 16$	16: $\left(a \wedge b^\perp\right)^\perp = 8^\perp$
6: $a \wedge \left(a \wedge b^\perp\right)^\perp = a \wedge 16$	17: $\left(b \wedge a^\perp\right) = 9^\perp$
7: $a \wedge \left(b \wedge b^\perp\right)^\perp = a \wedge 18$	18: $\left(b \wedge b^\perp\right)^\perp = 10^\perp$
8: $a \wedge b^\perp$	19: $a^\perp$
9: $b \wedge a^\perp$	20: $b^\perp$
10: $b \wedge b^\perp$	21: a
11: $\left(\left(a \wedge b^\perp\right)^\perp \wedge \left(b^\perp \wedge \left(a \wedge b^\perp\right)^\perp\right)^\perp\right)^\perp = 3^\perp$	22: b

3.3 Example three

This is the most general Hermitian (meet)semilattice generated by a satisfying the following conditions:

1. $a \leq b$
2. $b = b^{\perp\perp}$
3. $b \wedge \left(b \wedge a^{\perp}\right)^{\perp} = \left(a^{\perp} \wedge \left(b \wedge b^{\perp}\right)^{\perp}\right)^{\perp}$

We computed this semilattice because Moresi found a chain that could be infinite: $c_0 \doteqdot b \wedge b^{\perp}$ and $c_n \doteqdot b^{\perp} \wedge \left(a^{\perp} \wedge c_{n-1}^{\perp}\right)^{\perp}$ (see [5]). To end this chain we imposed the third condition.
There are 55 elements in the semilattice, as shown in figure 3, page 16:

1: $\mathbf{0}$

2: $\mathbf{1}$

3: $<T<aTaT<TbTTa = 38 \wedge 37$

4: $<T<bT<aTaT<T<bTbT<TaTTa = 42 \wedge 30$

5: $<T<bT<bT<aTaT<TaTTa = 45 \wedge 35$

6: $<T<bTbT<TTaT<bT<aTa = 48 \wedge 33$

7: $<T<bTbT<TaTTa = 48 \wedge 35$

8: $<T<bTbT<aTa = 48 \wedge 38$

9: $<TTaT<T<bTbT<aTa = a^{\perp\perp} \wedge 31$

10: $<TTaT<bT<aTa = a^{\perp\perp} \wedge 42$

11: $<TaT<bTa = a^{\perp} \wedge 47$

12: $<TaTTa = a^{\perp} \wedge a^{\perp\perp}$

13: $<TbTT<aTa = b^{\perp} \wedge 49$

14: $<TbTTa = b^{\perp} \wedge a^{\perp\perp}$

15: $<aTa = a \wedge a^{\perp}$

16: $<aTb = a \wedge b^{\perp}$

17: $<b<TaT<bTa = b \wedge 11$

18: $<bT<TaTTa = b \wedge 35$

19: $<bT<aTa = b \wedge 38$

20: $<bT<b<TaT<bTa = b \wedge 40$

21: $<bT<bT<TaTTa = b \wedge 41$

22: $<bT<bT<aTa = b \wedge 42$

23: $<bT<bTa = b \wedge 47$

24: $<bTa = b \wedge a^{\perp}$

25: $<bTb = b \wedge b^{\perp}$

26: $T<T<aTaT<TbTTa = 3^{\perp}$

27: $T<T<bT<aTaT<T<bTbT<TaTTa = 4^{\perp}$

28: $T<T<bT<bT<aTaT<TaTTa = 5^{\perp}$

29: $T<T<bTbT<TTaT<bT<aTa = 6^{\perp}$

30: $T<T<bTbT<TaTTa = 7^{\perp}$

31: $T<T<bTbT<aTa = 8^{\perp}$

32: $T<TTaT<T<bTbT<aTa = 9^{\perp}$

33: $T<TTaT<bT<aTa = 10^{\perp}$

34: $T<TaT<bTa = 11^{\perp}$

35: $T<TaTTa = 12^{\perp}$

36: $T<TbTT<aTa = 13^{\perp}$

37: $T<TbTTa = 14^{\perp}$

38: $T<aTa = 15^{\perp}$

39: $T<aTb = 16^{\perp}$

40: $T<b<TaT<bTa = 17^{\perp}$

41: $T<bT<TaTTa = 18^{\perp}$

42: $T<bT<aTa = 19^{\perp}$

43: $T<bT<b<TaT<bTa = 20^{\perp}$

44: $T<bT<bT<TaTTa = 21^{\perp}$

45: $T<bT<bT<aTa = 22^{\perp}$

46: $T<bT<bTa = 23^{\perp}$

47: $T<bTa = 24^{\perp}$

48: $T<bTb = 25^{\perp}$

49: $TT<aTa = 15^{\perp\perp}$

50: $TT<aTb = 16^{\perp\perp}$

51: $TTa = a^{\perp\perp}$

52: $Ta = a^{\perp}$

53: $Tb = b^{\perp}$

54: a

55: b

3.4 Example four

This is the most general Hermitian (meet)semilattice generated by a satisfying the following conditions:

1. $a \leq b$
2. $b = b^{\perp\perp}$
3. $\delta(b \wedge (b \wedge a^{\perp})^{\perp}/a^{\perp\perp}) < \aleph_0$

We computed this semilattice because its finite nature is determined only by fixing the index $\delta(b \wedge (b \wedge a^{\perp})^{\perp}/a^{\perp\perp})$ to be finite.
There are 63 elements in the semilattice, as shown in figure 4, page 17:

1: $\mathbf{0}$
2: $\mathbf{1}$
3: $<T<aTaT<TbTTa = 46 \wedge 45$
4: $<T<bT<aTaT<TaT<bTb = 50 \wedge 42$
5: $<T<bTaT<TaT<bTa = 55 \wedge 41$
6: $<T<bTaT<bT<bTa = 55 \wedge 54$
7: $<T<bTbT<aTa = 56 \wedge 46$
8: $<TTaT<bT<aTa = a^{\perp\perp} \wedge 50$
9: $<TaT<TaT<bT<bT<aTa = a^{\perp} \wedge 40$
10: $<TaT<TaT<bTb = a^{\perp} \wedge 42$
11: $<TaT<b<TaT<bTa = a^{\perp} \wedge 48$
12: $<TaT<bT<bT<TaTTa = a^{\perp} \wedge 52$
13: $<TaT<bT<bT<aTa = a^{\perp} \wedge 53$
14: $<TaT<bTa = a^{\perp} \wedge 55$
15: $<TaT<bTb = a^{\perp} \wedge 56$
16: $<TaTTa = a^{\perp} \wedge a^{\perp\perp}$
17: $<TbTT<aTa = b^{\perp} \wedge 57$
18: $<TbTTa = b^{\perp} \wedge a^{\perp\perp}$
19: $<aTa = a \wedge a^{\perp}$
20: $<aTb = a \wedge b^{\perp}$
21: $<b<TaT<bTa = b \wedge 14$
22: $<bT<TaTTa = b \wedge 43$
23: $<bT<aTa = b \wedge 46$
24: $<bT<b<TaT<bTa = b \wedge 48$
25: $<bT<bT<TaTTa = b \wedge 49$
26: $<bT<bT<aTa = b \wedge 50$
27: $<bT<bTa = b \wedge 55$
28: $<bTa = b \wedge a^{\perp}$
29: $<bTb = b \wedge b^{\perp}$
30: $T<T<aTaT<TbTTa = 3^{\perp}$
31: $T<T<bT<aTaT<TaT<bTb = 4^{\perp}$
32: $T<T<bTaT<TaT<bTa = 5^{\perp}$
33: $T<T<bTaT<bT<bTa = 6^{\perp}$
34: $T<T<bTbT<aTa = 7^{\perp}$
35: $T<TTaT<bT<aTa = 8^{\perp}$
36: $T<TaT<TaT<bT<bT<aTa = 9^{\perp}$
37: $T<TaT<TaT<bTb = 10^{\perp}$
38: $T<TaT<b<TaT<bTa = 11^{\perp}$
39: $T<TaT<bT<bT<TaTTa = 12^{\perp}$
40: $T<TaT<bT<bT<aTa = 13^{\perp}$
41: $T<TaT<bTa = 14^{\perp}$
42: $T<TaT<bTb = 15^{\perp}$
43: $T<TaTTa = 16^{\perp}$
44: $T<TbTT<aTa = 17^{\perp}$
45: $T<TbTTa = 18^{\perp}$
46: $T<aTa = 19^{\perp}$
47: $T<aTb = 20^{\perp}$
48: $T<b<TaT<bTa = 21^{\perp}$
49: $T<bT<TaTTa = 22^{\perp}$
50: $T<bT<aTa = 23^{\perp}$
51: $T<bT<b<TaT<bTa = 24^{\perp}$
52: $T<bT<bT<TaTTa = 25^{\perp}$
53: $T<bT<bT<aTa = 26^{\perp}$
54: $T<bT<bTa = 27^{\perp}$
55: $T<bTa = 28^{\perp}$
56: $T<bTb = 29^{\perp}$
57: $TT<aTa = 19^{\perp\perp}$
58: $TT<aTb = 20^{\perp\perp}$
59: $TTa = a^{\perp\perp}$
60: $Ta = a^{\perp}$
61: $Tb = b^{\perp}$
62: a
63: b

The relation between the indices of the intervals are the following:

$$\begin{array}{ccccccccc}
\delta_1 & \doteqdot & \delta(42/a^{\perp\perp}) & = & \delta(10/16) & = & \delta(4/7) & = & \delta(34/30) \\
 & = & \delta(29/18) & = & \delta(45/56) & = & \delta(2/6) & = & \delta(35/31) \\
 & = & \delta(43/37) & = & \delta(a^{\perp}/15) & & & & \\
\delta_2 & \doteqdot & \delta(40/42) & = & \delta(9/10) & = & \delta(26/4) & = & \delta(31/53) \\
 & = & \delta(37/36) & = & \delta(15/13) & & & & \\
\delta_3 & \doteqdot & \delta(39/40) & = & \delta(25/9) & = & \delta(36/52) & = & \delta(13/12) \\
\delta_4 & \doteqdot & \delta(38/39) & = & \delta(21/25) & = & \delta(51/49) & = & \delta(22/24) \\
 & = & \delta(52/48) & = & \delta(12/11) & & & & \\
\delta_5 & \doteqdot & \delta(5/38) & = & \delta(11/32) & & & & \\
\delta_6 & \doteqdot & \delta(27/5) & = & \delta(14/6) & = & \delta(33/41) & = & \delta(31/54)
\end{array}$$

3.5 Diagrams

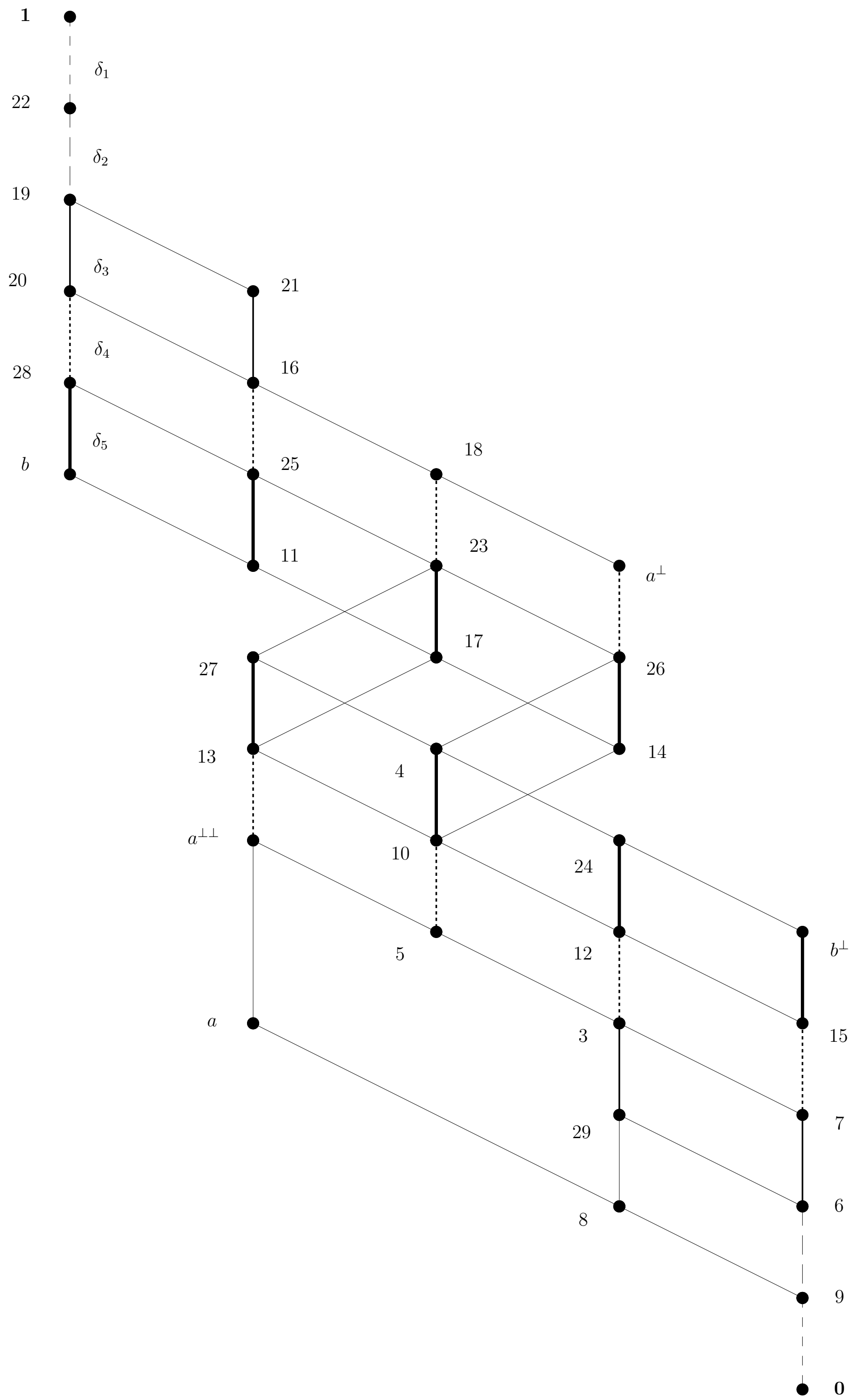

Figure 1: Semilattice with $a \leq b$, $b = b^{\perp\perp}$ and $\delta(1/b) < \aleph_0$

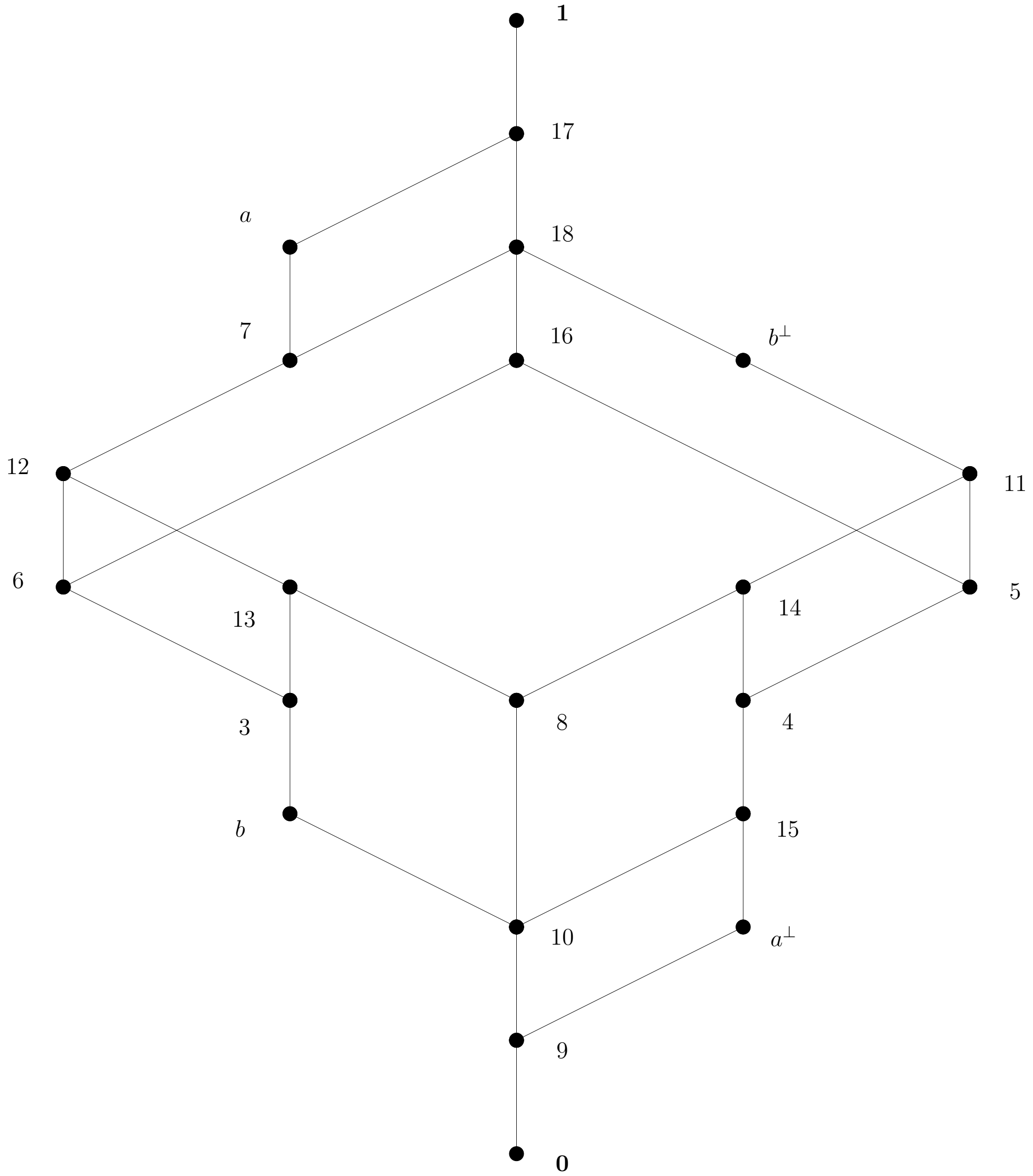

Figure 2: Semilattice with $b \leq a$, $a = a^{\perp\perp}$ and $b = b^{\perp\perp}$

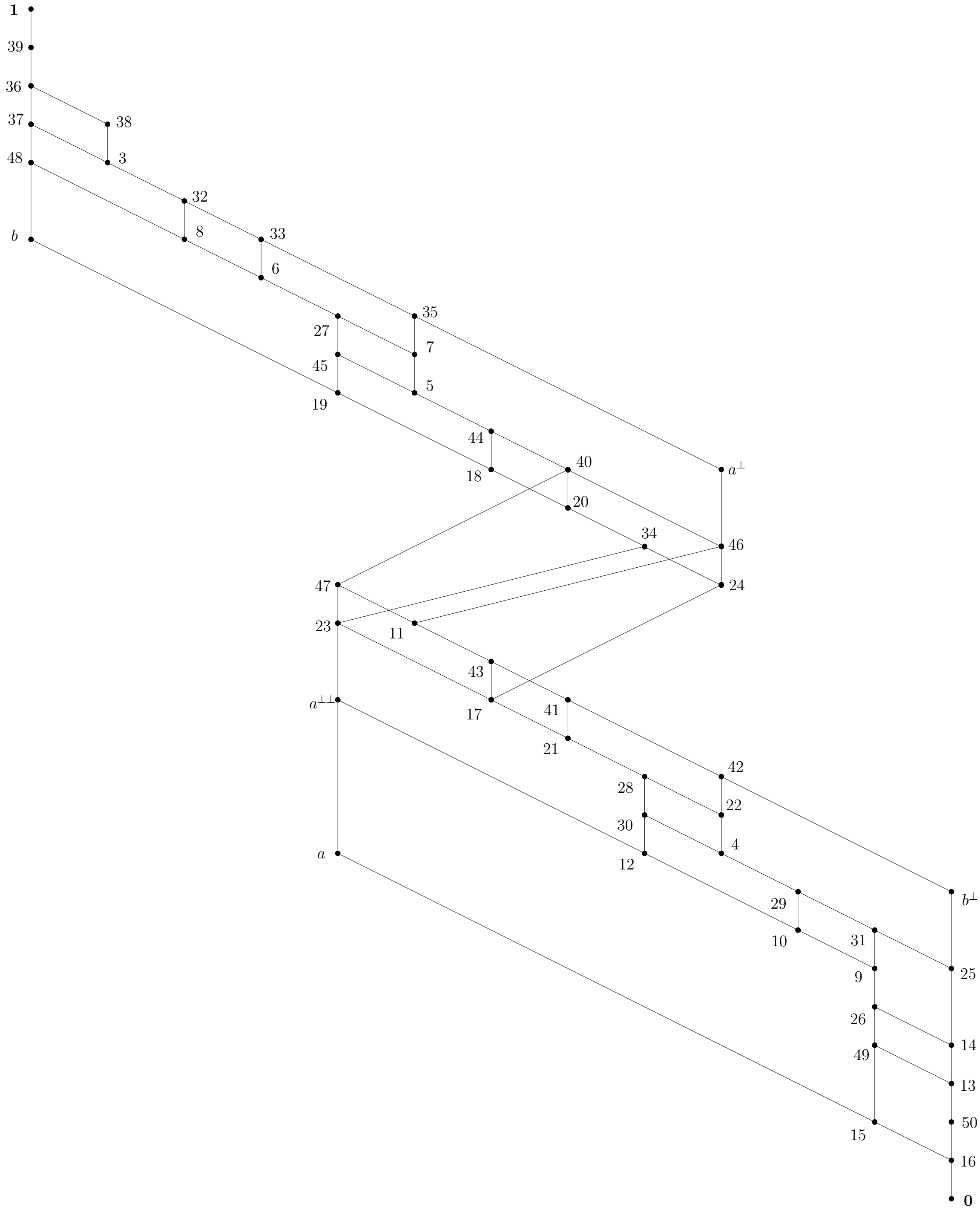

Figure 3: Semilattice with $a \leq b$, $b = b^{\perp\perp}$ and $b \wedge \left(b \wedge a^{\perp}\right)^{\perp} = \left(a^{\perp} \wedge \left(b \wedge b^{\perp}\right)^{\perp}\right)^{\perp}$

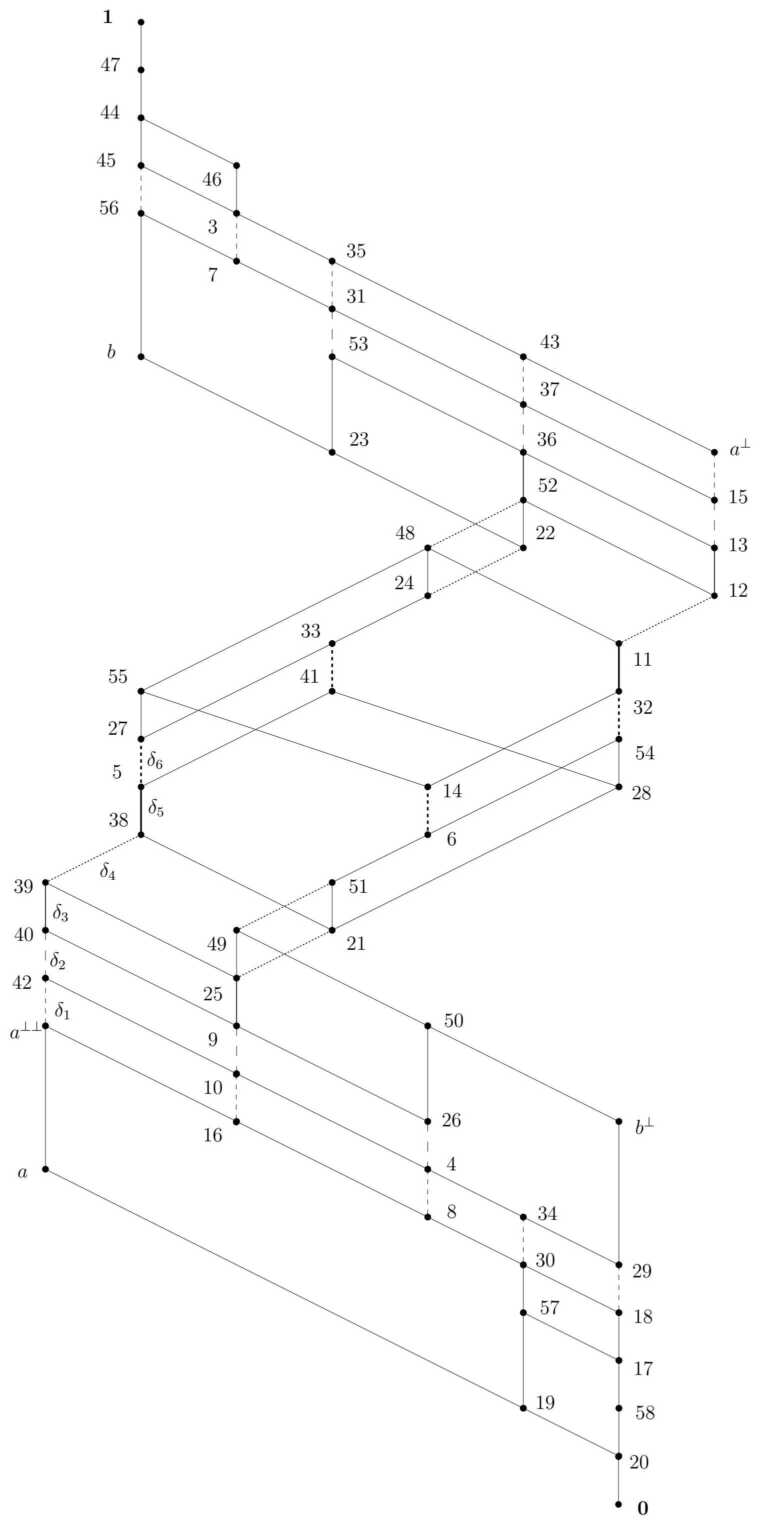

Figure 4: Semilattice with $a \leq b$, $b = b^{\perp\perp}$ and $\delta(b \wedge (b \wedge a^{\perp})^{\perp}/a^{\perp\perp}) < \aleph_0$

Acknowledgments

It gives me a great pleasure to thank Dr. Remo Moresi who suggested such an interesting subject and for his collaboration during this work. Moreover, I wish to thank Prof. Shristi D. Chatterji for his disponibility. Finally, I'm very grateful to my brother Christian for a critical reading of the manuscript and useful advice.

References

[1] G. Grätzer, *General lattice theory*, Basel Stuttgart Birkhäuser 1978.

[2] P. Grogono, *La programmation en Pascal*, Addison-Wesley Europe, 1988.

[3] Ch. Herrmann, Galois Lattices, *Note di Matematica e Fisica 7*, 229-234, 1994.

[4] H.A. Keller, U.-M. Künzi, M. Wild: *Orthogonal geometry in infinite dimensional vector spaces*, Bayreuther mathematische Schriften, Heft 53, 1998.

[5] R. Moresi, Beispiele Hermitescher Halbverbände, to appear in *Note di Matematica e Fisica 10* (proceedings of the meeting "Mathematik-Tagung in Memoriam von Herbert Gross (1936-1989)", Locarno 28.-29. 8. 1999).

[6] R. Moresi, Hermitean (semi)lattices with index function, *Note di Matematica e Fisica 7*, 235-242, 1994.

[7] The help guide of Borland Pascal.