An Explicit Formula for Sorting and its Application to Sorting in Lattices

Jens Gerlach

Fraunhofer FOKUS
jens.gerlach@fokus.fraunhofer.de

Abstract In a totally ordered set the notion of sorting a finite sequence is defined through the existence of a suitable permutation of the sequence's indices. A drawback of this definition is that it only implicitly expresses how the elements of a sequence are related to those of its sorted counterpart. To alleviate this situation we prove a simple formula that explicitly describes how the kth element of a sorted sequence can be computed from the elements of the original sequence. As this formula relies only on the minimum and maximum operations we use it to define the notion of sorting for lattices. A major difference of sorting in lattices is that it does *not* guarantee that sequence elements are only rearranged. To the contrary, sorting in general lattices may introduce new values into a sequence or completely remove values from it. We can show, however, that other fundamental properties that are associated with sorting are preserved. Furthermore, we address the problem that the direct application of our explicit formula for sorting leads to an algorithm with exponential complexity. We present therefore for distributive lattices a recursive formulation to compute the sort of a sequence. This alternative formulation, which is inspired by the identity $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ that underlies Pascal's triangle, allows for sorting in lattices with quadratic complexity and is in fact a generalization of *insertion sort* for lattices.

1 Introduction

In this paper we present the results of two preprints [1,2] where we outline basic principles of a theory of sorting in lattices.

Sorting a sequence in a total order $(X, \leq)$ is typically defined through the existence of a suitable permutation (cf. [3, p. 4]). There exists for each sequence x of length n in a totally ordered set a permutation φ of $[1, n] = \{1, \ldots, n\}$ such that $x \circ \varphi$ is a increasing sequence. If x is injective, then φ is uniquely determined, and vice versa. However, regardless whether there is exactly one permutation, the rearrangement $x^{\uparrow} = x \circ \varphi$ is uniquely determined and we thus refer to it as the *increasing sort of x*.

Sorting defines a map $x \mapsto x^{\uparrow}$ from X^n to the subset of increasing sequences. This map has several interesting properties. First of all, it is idempotent

$$\left(x^{\uparrow}\right)^{\uparrow} = x^{\uparrow} \tag{1}$$

and thus a projection. Secondly, for each permutation ψ of $[1, n]$ we have

$$(x \circ \psi)^{\uparrow} = x^{\uparrow} \tag{2}$$

The definition of sorting through the existence of a suitable permutation only provides an implicit relationship between the elements of x and $x^{\uparrow}$. However, sometimes we prefer explicit relationships.

If, for example, someone asked whether there is for the numbers a and b and the exponent n a general relationship between the value $(a+b)^n$ and the powers a^n and b^n, then the (obvious) answer is that this relationship is captured by the Binomial Theorem

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k \tag{3}$$

which also shows that other powers of a and b are involved.

When looking for an explicit relationship between the elements of x and the elements of its increasingly sorted counterpart $x^{\uparrow} = \left(x_1^{\uparrow}, \ldots, x_n^{\uparrow}\right)$, one can provide an easy answer for the first and last elements of $x^{\uparrow}$. In fact, we know that $x_1^{\uparrow}$ is the least element of $\{x_1, \ldots, x_n\}$

$$x_1^{\uparrow} = x_1 \wedge \ldots \wedge x_n = \bigwedge_{k=1}^{n} x_k, \tag{4}$$

whereas $x_n^{\uparrow}$ is the greatest element of x

$$x_n^{\uparrow} = x_1 \vee \ldots \vee x_n = \bigvee_{k=1}^{n} x_k. \tag{5}$$

In Section 2 we prove Identity (7) that explicitly states how the elements $x_1^{\uparrow}, \ldots, x_n^{\uparrow}$ are related to $x_1, \ldots, x_n$. This formula only uses the minimum and maximum operations on finite sets. Based on this observation, we define in Section 3 the notion of *sorting of sequences in a lattice* through simply replacing the minimum/maximum operations by the infimum/supremum operations, respectively. We also show that sorting in lattices in general not just reorders the elements of a sequence but really changes them. However, we are able to prove that our definition satisfies various properties that are associated with sorting.

The direct application of Identity (7) leads to an algorithm with exponential complexity (cf. Section 4). In order to address this problem, we prove the *recursive* Identity (19) for the case of *bounded distributive lattices*. This identity is closely related to the well-known fact that the binomial coefficient

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

can be efficiently computed through the recursion

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

which underlies Pascal's triangle.

Furthermore, we prove that a lattice, in which the recursive Identity (19) holds, is necessarily distributive. The main advantage of our recursive identity is that it allows for an algorithm for sorting in lattices with quadratic complexity. In fact, this algorithm is a generalization of *insertion sort* for lattices (cf. Section 5).

2 A formula for sorting

Let $(X, \leq)$ be a totally ordered set, then each nonempty finite subset A of X contains a *least* and a *greatest* element [4, R. 6.5]. We also speak of the minimum and maximum of A and refer to these special elements as $\bigwedge A$ and $\bigvee A$, respectively. The following inequalities hold for all $a \in A$

$$\bigwedge A \leq a \leq \bigvee A \tag{6}$$

For $A = \{x, y\}$ we use the notation $x \wedge y$ and $x \vee y$ to denote the minimum and maximum of x and y, respectively.
The main results of this paper depend on a particular family of finite sets.

Definition 1. *For $k \in [1, n]$ we denote with $\mathbb{N}\binom{n}{k} := \left\{A \subset [1, n] \;\middle|\; |A| = k\right\}$ the set of subsets of $[1, n]$ that contain exactly k elements. The set $\mathbb{N}\binom{n}{k}$ consists of $\binom{n}{k}$ elements.*

Proposition 1. *Let $(x_1, \ldots, x_n)$ be a sequence in a totally ordered set, then the following identity holds for the elements of the sequence $\left(x_1^{\uparrow}, \ldots, x_n^{\uparrow}\right)$*

$$x_k^{\uparrow} = \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i. \tag{7}$$

Before we prove Proposition 1 we introduce an abbreviation for the right hand side of Identity (7). For a sequence x of length n we define for $1 \leq k \leq n$

$$x_k^{\vartriangle} := \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i. \tag{8}$$

With this notation Proposition 1 reads

$$x^{\uparrow} = x^{\vartriangle}. \tag{9}$$

We remark that because $(X, \leq)$ is a total order, we know that each element of $x^{\vartriangle}$ is also an element of x. When applying Identity (8) it is sometimes convenient to use a slightly more explicit way to write the elements of $x^{\vartriangle}$.

$$x_k^{\vartriangle} = \bigwedge_{1 \leq i_1 < \ldots < i_k \leq n} x_{i_1} \vee \ldots \vee x_{i_k} \tag{10}$$

We see then that $x_1^{\vartriangle}$ is the least element of x and thus equals $x_1^{\uparrow}$ (cf. Identity (4)), whereas $x_n^{\vartriangle}$ is the greatest element of x and thus equals $x_n^{\uparrow}$ (cf. Identity (5)). This means that Identity (7) is satisfied for $k = 1$ and $k = n$.

Lemma 1. *If x is a sequence of length n in a totally ordered set $(X, \leq)$, then $x^{\vartriangle}$ is a increasing sequence.*

Proof. Let $1 \le k < n$ and I be an arbitrary subset of $[1,n]$ with $k+1$ elements. If J is a subset of I with k elements, then we have by Inequality (6) and $J \subset I$

$$x_k^{\vartriangle} = \bigwedge_{L \in \mathbb{N}\binom{n}{k}} \bigvee_{l \in L} x_l \le \bigvee_{j \in J} x_j \le \bigvee_{i \in I} x_i.$$

Since I is an arbitrary set of $k+1$ elements we obtain from here

$$x_k^{\vartriangle} \le \bigwedge_{I \in \mathbb{N}\binom{n}{k+1}} \bigvee_{i \in I} x_i = x_{k+1}^{\vartriangle},$$

which shows that $x^{\vartriangle}$ is increasing. $\square$

Note that in the proof of Lemma 1 we have only used the fact that the minimum of a set is a *lower bound* for all elements of that set (cf. Inequality (6)).

Proof (Proposition 1). We will show that for each k with $1 \le k \le n$ both $x_k^{\vartriangle} \le x_k^{\uparrow}$ and $x_k^{\uparrow} \le x_k^{\vartriangle}$ hold. Let φ be a permutation of $[1,n]$ with

$$x^{\uparrow} = x \circ \varphi \tag{11}$$

and let $J \subset [1,n]$ be the subset for which

$$J = \varphi([1,k]) \tag{12}$$

holds. From the fact that J contains exactly k elements we conclude

$$\begin{aligned}
x_k^{\vartriangle} = \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i \le \bigvee_{j \in J} x_j & \qquad \text{by Inequality (6)} \\
= \bigvee_{j \in J} x^{\uparrow}\left(\varphi^{-1}(j)\right) & \qquad \text{by Identity (11)} \\
= \bigvee_{i \in [1,k]} x_i^{\uparrow} & \qquad \text{by Identity (12)} \\
= x_k^{\uparrow} & \qquad \text{by monotonicity of } x^{\uparrow}.
\end{aligned}$$

This finishes the first part of the proof.
Conversely, we conclude from the fact that $(X, \le)$ is a total order and Identity (11) that there exists a subset B of $[1,n]$ with exactly k elements such that

$$x_k^{\vartriangle} = \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i = \bigvee_{i \in B} x_i = \bigvee_{i \in B} x^{\uparrow}\left(\varphi^{-1}(i)\right) = \bigvee_{j \in \varphi^{-1}(B)} x_j^{\uparrow}$$

holds. Since $x^{\uparrow}$ is increasing we have $\bigvee_{j \in \varphi^{-1}(B)} x_j^{\uparrow} = x_m^{\uparrow}$ where $m = \bigvee(\varphi^{-1}(B))$ is the greatest element of $\varphi^{-1}(B)$. We have, thus,

$$x_k^{\vartriangle} = x_m^{\uparrow}. \tag{13}$$

However, since $\bigvee(\varphi^{-1}(B))$ is a subset of $[1,n]$ that contains exactly k elements we obtain $k \le m$. Since $x^{\uparrow}$ is increasing we conclude $x_k^{\uparrow} \le x_m^{\uparrow}$. This inequality and Identity (13) imply $x_k^{\uparrow} \le x_k^{\vartriangle}$, which completes the proof. $\square$

3 Sorting in lattices

Let $(X, \leq)$ be a partially ordered set that is also a *lattice* $(X, \wedge, \vee)$, then for each $x, y \in X$ there exists the infimum $x \wedge y$ and the supremum $x \vee y$ (cf. [5, Chapter 3]). These operations are commutative and associative and they satisfy for all $x, y \in X$ the so-called absorption properties $x \vee (x \wedge y) = x$ and $x \wedge (x \vee y) = x$. If $(X, \leq)$ is a *total order*, then $\wedge$ and $\vee$ are the minimum and maximum operations of Section 2.

In a lattice, the infimum and supremum exist for every finite subset A and are denoted by $\bigwedge A$ and $\bigvee A$, respectively (cf. [5, p. 49]). We therefore know that for a sequence x of length n the value

$$x_k^{\vartriangle} = \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i$$

from Identity (8) is well-defined in a lattice. This motivates the following definition.

Definition 2. *If x is a sequence of length n in a lattice $(X, \wedge, \vee)$, then we refer to $x^{\vartriangle}$ as defined by Identity* (8) *as the* increasing sort of *x with respect to the lattice* $(X, \wedge, \vee)$.

Before we start to investigate which properties that are traditionally associated with sorting are maintained by our definition we want to point out a major difference: In a lattice the value $x_k^{\vartriangle}$ might be different from the original values $x_1, \ldots, x_n$. The reason for this is the following: While in a lattice the inequalities $x \wedge y \leq x, y \leq x \vee y$ generally hold, there might be also the case that the set $\{x \wedge y, x \vee y\}$ is different from the set $\{x, y\}$. In a total order these two sets are always equal.

Examples of sorting in lattices As a first example we consider the finite set $X = \{x, y, z\}$. Figure 1 shows the lattice of all subsets of X. Let x be the sequence $a = (\{x\}, \{y\}, \{z\})$, then $a^{\vartriangle} = (\emptyset, \emptyset, X)$. Thus, $a^{\vartriangle}$ is a increasing sequence that consists of elements that are completely different from those of a.

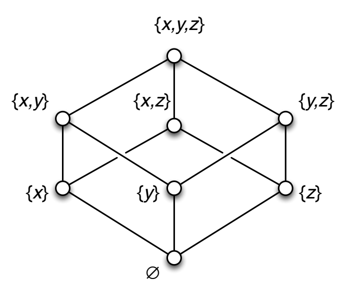

Figure 1. The lattice of $\{x, y, z\}$

x	$x^{\vartriangle}$
(1)	(1)
(1, 2)	(1, 2)
(1, 2, 3)	(1, 1, 6)
(1, 2, 3, 4)	(1, 1, 2, 12)
(1, 2, 3, 4, 5)	(1, 1, 1, 2, 60)
(1, 2, 3, 4, 5, 6)	(1, 1, 1, 2, 6, 60)
(1, 2, 3, 4, 5, 6, 7)	(1, 1, 1, 1, 2, 6, 420)
(1, 2, 3, 4, 5, 6, 7, 8)	(1, 1, 1, 1, 2, 2, 12, 840)

Table 1. Sorting in the lattice $(\mathbb{N}, \gcd, \operatorname{lcm})$

As a second example we consider the lattice $(\mathbb{N}, \gcd, \operatorname{lcm})$ where $\gcd(x, y)$ and $\operatorname{lcm}(x, y)$ denote the *greatest common divisor* and *least common multiple* of x and y, respectively. The associated partial order of this lattices is defined by divisibility of natural numbers. Table 1 shows some examples of our definition of sorting for different sequences in $(\mathbb{N}, \gcd, \mathrm{lcm})$. Again we see that sorting in a lattice may change the elements in a sequence.

Elementary properties of sorting in lattices The following lemma states that $x^{\vartriangle}$ is indeed a increasing sequence with respect to the partial order $(X, \leq)$ of the lattice $(X, \wedge, \vee)$.

Lemma 2. *If x is a finite sequence in a lattice $(X, \wedge, \vee)$ with associated partial order $(X, \leq)$, then Identity* (8) *defines a increasing sequence $x^{\vartriangle}$.*

Proof. In order to prove this lemma we can proceed exactly as in the proof of Lemma 1 where $(X, \leq)$ is a total order. As remarked on Page 2, we have used only the fact that $\bigwedge A$ is a lower bound of A which by definition also holds for lattices. □

A simple consequence of Lemma 2 is the following Lemma 3 which states that sorting in lattices respects lower and upper bounds of the original sequence.

Lemma 3. *Let x be a sequence of length n in a lattice $(X, \wedge, \vee)$ with associated partial order $(X, \leq)$. If for $1 \leq i \leq n$ holds $a \leq x_i \leq b$, then $a \leq x_i^{\vartriangle} \leq b$ holds as well.*

Proof. From Identity (10) follows that $x_n^{\vartriangle}$ is the supremum of the elements $x_1, \ldots, x_n$. Thus, we have $x_n^{\vartriangle} \leq b$. Lemma 2 ensures that $x_n^{\vartriangle}$ is the largest element of $x^{\vartriangle}$. Thus we have $x_i^{\vartriangle} \leq b$ for $1 \leq i \leq n$. The case for the lower bound a is treated analogously. □

The following lemma restates the idempotence of sorting for the case of lattices (cf. Identity (1)).

Lemma 4. *If x is a finite sequence in a lattice $(X, \wedge, \vee)$, then $(x^{\vartriangle})^{\vartriangle} = x^{\vartriangle}$.*

Proof. We know from Lemma 2 that $x^{\vartriangle}$ is a increasing sequence in the partial order $(X, \leq)$. Thus, the relation $\leq$ is a *total order* on the set $\left\{x_1^{\vartriangle}, \ldots, x_n^{\vartriangle}\right\} \subset X$. In other words we can sort $x^{\vartriangle}$ in the classical sense. From this follows by Identity (7)

$$x^{\vartriangle} = (x^{\vartriangle})^{\uparrow} = (x^{\vartriangle})^{\vartriangle}.$$

□

We can also show the invariance of sorting in lattices under permutations (cf. Identity (2)).

Lemma 5. *If x is a sequence of length n in a lattice and ψ a permutation of $[1, n]$, then $(x \circ \psi)^{\vartriangle} = x^{\vartriangle}$ holds.*

Proof. We have for $1 \leq k \leq n$

$$(x \circ \psi)_k^{\vartriangle} = \bigwedge_{A \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in A} (x \circ \psi)_i = \bigwedge_{A \in \mathbb{N}\binom{n}{k}} \bigvee_{j \in \psi(A)} x_j = \bigwedge_{B \in \psi\left(\mathbb{N}\binom{n}{k}\right)} \bigvee_{j \in B} x_j$$

Because ψ is a permutation of $[1, n]$ we find that $\psi\left(\mathbb{N}\binom{n}{k}\right) = \mathbb{N}\binom{n}{k}$ and conclude

$$(x \circ \psi)_k^{\vartriangle} = \bigwedge_{B \in \mathbb{N}\binom{n}{k}} \bigvee_{j \in B} x(j) = x_k^{\vartriangle}.$$

□

4 Recursive sorting in lattices

The definition of $x^{\vartriangle}$ through Identity (8) is nice and succinct, but it is also quite impractical to use in computations. Table 2 shows simple performance measurements (conducted on a notebook computer) for computing $(1, \ldots, n)^{\vartriangle}$ in $(\mathbb{N}, \gcd, \operatorname{lcm})$. The reason for this dramatic slowdown is of course the exponential complexity inherent in Identity (8): In order to compute $x^{\vartriangle}$ from x it is necessary to consider all $2^n - 1$ nonempty subsets of $[1, n]$.

sequence length	20	21	22	23	24	25	26
time in s	0.6	1.3	2.7	5.8	11.8	25.5	51.6

Table 2. Wall-clock time for computing $(1, \ldots, n)^{\vartriangle}$ according to Identity (8)

For the remainder of this paper we assume that $(X, \wedge, \vee, \bot, \top)$ is a *bounded* lattice. Here $\bot$ is the least element of X and the neutral element of join, that is,

$$x = \bot \vee x = x \vee \bot \qquad \forall x \in X \tag{14}$$

whereas $\top$ is the greatest element of X and the neutral element of meet, that is,

$$x = \top \wedge x = x \wedge \top \qquad \forall x \in X. \tag{15}$$

We now introduce a notation that allows us to concisely refer to individual elements of both $(x_1, \ldots, x_n)^{\vartriangle}$ and $(x_1, \ldots, x_{n-1})^{\vartriangle}$. Here again, it is convenient to employ the notation for the binomial coefficient $\binom{n}{k}$ in the context of sorting in lattices. For a sequence x of length n we define for $0 \le m \le n$

$$x^{\vartriangle}\binom{m}{k} := \begin{cases} \bot & k = 0 \\ (x_1, \ldots, x_m)^{\vartriangle}(k) & k \in [1, m] \\ \top & k = m + 1 \end{cases} \tag{16}$$

We know from Identity (8) that $(x_1, \ldots, x_m)^{\vartriangle}(k) = \bigwedge_{I \in \mathbb{N}\binom{m}{k}} \bigvee_{i \in I} x_i$ holds for $1 \le k \le m$.

We therefore have

$$x^{\vartriangle}\binom{m}{k} = \bigwedge_{I \in \mathbb{N}\binom{m}{k}} \bigvee_{i \in I} x_i. \tag{17}$$

In particular, the following identity holds for $1 \le k \le n$

$$x^{\vartriangle}\binom{n}{k} = x^{\vartriangle}_k. \tag{18}$$

The main result of this section is Proposition 2, which states in Identity (19), how the kth element of $(x_1, \ldots, x_n)^{\vartriangle}$ can be computed from $(x_1, \ldots, x_{n-1})^{\vartriangle}$ and x_n by simply applying one *join* and one *meet*. The proof of Proposition 2 relies on the fact that the lattice under consideration is both *bounded* and *distributive*.

Proposition 2. *If* $(X, \wedge, \vee, \perp, \top)$ *is a bounded distributive lattice and if* x *is a sequence of length* n*, then for* $1 \leq k \leq n$ *holds*

$$x^{\vartriangle}\binom{n}{k} = x^{\vartriangle}\binom{n-1}{k} \wedge \left(x^{\vartriangle}\binom{n-1}{k-1} \vee x_n\right) \tag{19}$$

Proof. For $k = 1$, we have

$$\begin{aligned}
x^{\vartriangle}\binom{n}{1} &= \bigwedge_{i=1}^{n} x_i && \text{by Identity (17)}\\
&= \left(\bigwedge_{i=1}^{n-1} x_i\right) \wedge x_n && \text{by associativity}\\
&= x^{\vartriangle}\binom{n-1}{1} \wedge x_n && \text{by Identity (17)}\\
&= x^{\vartriangle}\binom{n-1}{1} \wedge \left(\perp \vee x_n\right) && \text{by Identity (14)}\\
&= x^{\vartriangle}\binom{n-1}{1} \wedge \left(x^{\vartriangle}\binom{n-1}{0} \vee x_n\right) && \text{by Identity (16).}
\end{aligned}$$

We deal similarly with the case $k = n$ (cf. [2, p. 5]). In the general case of $1 < k < n$, we first remark that if A is a subset of $[1, n]$, which consists of k elements, then there are two cases possible:

1. If n does not belong to A, then A is a subset of $\mathbb{N}\binom{n-1}{k}$.
2. If n is an element of A, then the set $B := A \setminus \{n\}$ belongs to $\mathbb{N}\binom{n-1}{k-1}$.

In other words, $\mathbb{N}\binom{n}{k}$ can be represented as the following (disjoint) union

$$\mathbb{N}\binom{n}{k} = \mathbb{N}\binom{n-1}{k} \cup \left\{B \cup \{n\} \;\middle|\; B \in \mathbb{N}\binom{n-1}{k-1}\right\}. \tag{20}$$

We obtain therefore

$$\begin{aligned}
x^{\vartriangle}\binom{n}{k} &= \bigwedge_{I \in \mathbb{N}\binom{n}{k}} \bigvee_{i \in I} x_i && && \text{by Identity (17)}\\
&= \bigwedge_{I \in \mathbb{N}\binom{n-1}{k}} \bigvee_{i \in I} x_i &\wedge& \bigwedge_{I \in \mathbb{N}\binom{n-1}{k-1}} \bigvee_{i \in I \cup \{n\}} x_i && \text{by Identity (20)}\\
&= x^{\vartriangle}\binom{n-1}{k} &\wedge& \bigwedge_{I \in \mathbb{N}\binom{n-1}{k-1}} \bigvee_{i \in I \cup \{n\}} x_i && \text{by Identity (17)}\\
&= x^{\vartriangle}\binom{n-1}{k} &\wedge& \bigwedge_{I \in \mathbb{N}\binom{n-1}{k-1}} \left(\bigvee_{i \in I} x_i \vee x_n\right) && \text{by associativity}\\
&= x^{\vartriangle}\binom{n-1}{k} &\wedge& \left(\left(\bigwedge_{I \in \mathbb{N}\binom{n-1}{k-1}} \bigvee_{i \in I} x_i\right) \vee x_n\right) && \text{by distributivity}\\
&= x^{\vartriangle}\binom{n-1}{k} &\wedge& \left(x^{\vartriangle}\binom{n-1}{k-1} \vee x_n\right) && \text{by Identity (17)}
\end{aligned}$$

which completes the proof. □

The following Proposition 3 states that the converse of Proposition 2 also holds.

Proposition 3. *Let* $(X, \wedge, \vee, \perp, \top)$ *be a bounded lattice which is* not *distributive. Then there exists a sequence* $x = (x_1, x_2, x_3)$ *in* X *such that Identity* (19) *is not satisfied.*

Proof. According to a standard result on distributive lattices [5, Theorem 4.7], a lattice is *not* distributive, if and only if it contains a sublattice which is isomorphic to either N_5 or M_3 (cf. Figure 2).

Figure 2. The non-distributive lattices N_5 and M_3

From Identity (10) follows for the elements of $x^\vartriangle = \left(x_1^\vartriangle, x_2^\vartriangle, x_3^\vartriangle\right)$

$$x_1^\vartriangle = x_1 \wedge x_2 \wedge x_3 \tag{21a}$$

$$x_2^\vartriangle = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3) \tag{21b}$$

$$x_3^\vartriangle = x_1 \vee x_2 \vee x_3. \tag{21c}$$

If X contains the sublattice N_5, then we consider the sequence $x = (c, d, b)$ and its subsequence (c, d). From Identity (21) then follows

$$(c, d, b)^\vartriangle = (a, d, e) \qquad \text{and} \qquad (c, d)^\vartriangle = (a, e).$$

Thus, we have

$$x^\vartriangle \binom{3}{2} = d \qquad x^\vartriangle \binom{2}{2} = e \qquad x^\vartriangle \binom{2}{1} = a.$$

However, applying Identity (19) we obtain

$$\begin{aligned} x^\vartriangle \binom{3}{2} &= x^\vartriangle \binom{2}{2} \wedge \left(x^\vartriangle \binom{2}{1} \vee x_3\right) \\ &= e \wedge (a \vee b) = e \wedge b = b \end{aligned}$$

instead of d.

If X contains the sublattice M_3, then we consider the sequence $x = (b, c, d)$ and its subsequence (b, c). From Identity (21) then follows

$$(b, c, d)^\vartriangle = (a, e, e) \qquad \text{and} \qquad (b, c)^\vartriangle = (a, e).$$

We therefore have

$$x^{\triangle}\binom{3}{2} = e \qquad\qquad x^{\triangle}\binom{2}{2} = e \qquad\qquad x^{\triangle}\binom{2}{1} = a.$$

Again, applying Identity (19) we obtain

$$\begin{aligned} x^{\triangle}\binom{3}{2} &= x^{\triangle}\binom{2}{2} \wedge \left(x^{\triangle}\binom{2}{1} \vee x_3\right) \\ &= e \wedge (a \vee d) = e \wedge d = d \end{aligned}$$

instead of e. □

Using Identity (19), we can prove the following Lemma 6, which generalizes a known fact known from sorting in a total order: If one knows that x_n is greater or equal that the preceding elements $x_1, \ldots, x_{n-1}$ then sorting the sequence $(x_1, \ldots, x_n)$ can be accomplished by sorting $(x_1, \ldots, x_{n-1})$ and simply appending x_n.

Lemma 6. *Let $(X, \wedge, \vee, \bot, \top)$ be a bounded distributive lattice and x be a sequence of length n. If the condition $x_i \leq x_n$ holds for $1 \leq i \leq n-1$, then the identities*

$$\begin{aligned} x^{\triangle}\binom{n}{i} &= x^{\triangle}\binom{n-1}{i} \\ x^{\triangle}\binom{n}{n} &= x_n \end{aligned}$$

hold.

Proof. The first equation follows directly from the fact that $x_n^{\triangle}$ is the supremum of the values $x_1, \ldots, x_n$. Regarding the second equation, we know from Lemma 2 that if for $1 \leq i \leq n-1$ the inequality $x_i \leq x_n$ holds, then

$$x^{\triangle}\binom{n-1}{i} \leq x_n.$$

This inequality is also valid for $i = 0$ because $x^{\triangle}\binom{n-1}{0} = \bot$ holds by Identity (16). From general properties of meet and join then follows that

$$\begin{aligned} x^{\triangle}\binom{n-1}{i} \vee x_n &= x_n \\ x^{\triangle}\binom{n-1}{i} \wedge x_n &= x^{\triangle}\binom{n-1}{i} \end{aligned}$$

holds for $0 \leq i \leq n-1$. We can therefore simplify Identity (19) as follows

$$\begin{aligned} x^{\triangle}\binom{n}{i} &= x^{\triangle}\binom{n-1}{i} \wedge \left(x^{\triangle}\binom{n-1}{i-1} \vee x_n\right) \\ &= x^{\triangle}\binom{n-1}{i} \wedge x_n \\ &= x^{\triangle}\binom{n-1}{i}. \end{aligned}$$

□

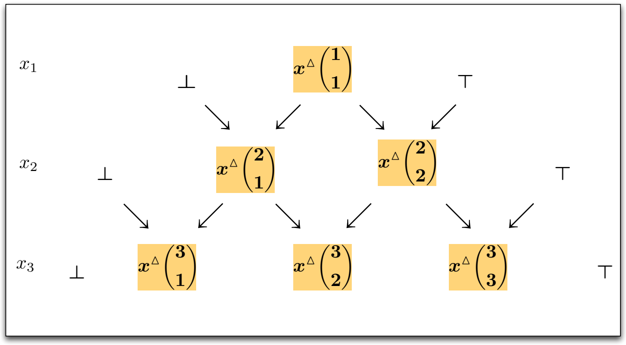

Figure 3. Graphical representation of Identity (19)

5 Insertion sort in lattices

Figure 3 graphically represents Identity (19) in a form that emphasizes its close relationship to Pascal's triangle. Whenever an arrow $\searrow$ and and arrow $\swarrow$ meet, the values are combined by a *meet*. In the case of an arrow $\searrow$, however, first the value at the origin of the arrow is combined with the sequence value x_n through a *join*.

Formula (22) outlines an algorithm that is based on Identity (19). The algorithm starts from $x_1 = (x_1)^{\triangle}$ and successively computes

$$(x_1, \dots, x_{i-1})^{\triangle}, x_i \quad \mapsto \quad (x_1, \dots, x_{i-1}, x_i)^{\triangle}. \tag{22}$$

From Identity (19) follows that in step i exactly i joins and i meets must be performed. Thus, altogether there are

$$\sum_{i=2}^{n} 2 * i = n(n+1) - 2$$

applications of join and meet. In other words, such an implementation has quadratic complexity. This algorithm can be considered as *insertion sort* [3, § 5.2.1] for lattices because one element at a time is added to an already "sorted" sequence. Table 3 shows some performance measurements for this algorithm in the bounded and distributive lattice $(\mathbb{N}, \gcd, \mathrm{lcm}, 1, 0)$.

sequence length	100	1000	10000	100000
time in s	0	0	3.4	420

Table 3. Wall-clock time for computing $(1, \dots, n)^{\triangle}$ according to Identity (19)

These results show that sorting in lattices can now be applied to much larger sequences than those shown in Table 2 before the limitations of an algorithm with quadratic complexity become noticeable.

6 Conclusions

Proposition 1 states through Identity (7) a simple explicit relationship between the elements of a finite sequence in a totally ordered sets to its sorted counterpart.

A sorting algorithm that directly uses Identity (7) would have exponential complexity. Thus, Identity (7) appears not relevant for implementing computationally efficient algorithms. The reader should bear in mind, however, that this is also true for the Binomial Theorem. In fact, directly computing $(x+y)^n$ is normally more efficient than computing the expansion

$$x^n + nx^{n-1}y + \frac{n(n-1)}{2}x^{n-2}y^2 + \ldots + y^n$$

A more interesting aspect of Identity (7) is therefore that it allows to *generalize the notion of sorting finite sequences to lattices*. Compared to sorting in a totally ordered set, sorting in lattices is a more *invasive* procedure because it may change sequence elements. While this may be considered as a major drawback one should bear in mind that generalizations often lead to surprising properties. The real criterion for accepting a generalization is whether it provides new insights or has useful applications. With respect to sorting in lattices, the latter question has not been addressed in this paper and remains a topic of future research.

We are able to show that our definition of sorting in lattices maintains many properties that are associated with sorting. Another important results of this paper are Proposition 2, which proves Identity (19) for bounded distributive lattices, and Proposition 3, which shows that the distributivity is necessary for Identity (19) to hold. The remarkable points of Identity (19) are that it

- exhibits a strong analogy between sorting and Pascal's triangle,
- allows to sort in lattices with quadratic complexity, and that it
- is in fact a generalization of *insertion sort* for lattices.

I would like to thank the reviewers for their comments. I am also very grateful for the many corrections and valuable suggestions of my colleagues Jochen Burghardt and Hans Werner Pohl: Jochen Burghardt's suggestion to investigate whether the distributivity in Proposition 2 is really necessary led to Proposition 3. Hans Werner Pohl pointed out the analogy of the algorithm in Equation 22 to insertion sort.

References

1. J. Gerlach. Sorting in Lattices. *ArXiv e-prints*, March 2013. http://arxiv.org/abs/1303.5560.
2. J. Gerlach. Recursive Sorting in Lattices. *ArXiv e-prints*, May 2013 http://arxiv.org/abs/1306.0019.
3. Donald E. Knuth. *The Art of Computer Programming, Volume III: Sorting and Searching*. Addison-Wesley, 1973.
4. Bourbaki, N., *Elements of Mathematics, Theory of Sets*, Addison-Wesley, Reading, MA, 1968.
5. S. Roman. *Lattices and Ordered Sets*. Springer-Verlag New York, 2008.