Fast Möbius inversion in semimodular lattices and ER-labelable posets

Petteri Kaski and Jukka Kohonen

Department of Computer Science
Aalto University
Espoo, Finland

{petteri.kaski,jukka.kohonen}@aalto.fi

Thomas Westerbäck

Department of Mathematics and Systems Analysis
Aalto University
Espoo, Finland

thomas.westerback@aalto.fi

Submitted: Mar 8, 2016; Accepted: Jul 29, 2016; Published: Aug 19, 2016
Mathematics Subject Classifications: 06C10, 06A07, 68W40, 68R05

Abstract

We consider the problem of fast zeta and Möbius transforms in finite posets, particularly in lattices. It has previously been shown that for a certain family of lattices, zeta and Möbius transforms can be computed in $O(e)$ elementary arithmetic operations, where e denotes the size of the covering relation. We show that this family is exactly that of geometric lattices. We also extend the algorithms so that they work in e operations for all semimodular lattices, including chains and divisor lattices. Finally, for both transforms, we provide a more general algorithm that works in e operations for all ER-labelable posets.

1 Introduction

Fast methods for computing zeta and Möbius transforms in finite posets are important for many algorithms of combinatorial nature, such as graph coloring [1] and fast Fourier transforms on inverse semigroups [9]. From an algebraic perspective, these transforms are basis-changing isomorphisms analogous to the fast Fourier transform [2]. From a computational perspective, the zeta transform is related to *ensemble computation*, where one is required to compute several different products of inputs, possibly sharing common subexpressions, and the challenge is to minimize the number of elementary multiplications perfomed (see Garey and Johnson [5]).

Let $(P, \leqslant)$ be a finite partially ordered set, or *poset*, with $|P| = v$ elements, and let $f : P \to A$ be a function to an abelian group A. The *zeta transform* of f is the function $g : P \to A$ such that, for all $y \in P$,

$$g(y) = \sum_{x \leqslant y} f(x).$$

For computational purposes, the function f can be represented as a v-element row vector $\vec{f} = (f(x))_{x \in P}$, and g similarly. The transformation $\vec{f} \mapsto \vec{g}$ is linear, and defined by the $v \times v$ matrix ζ, where $\zeta_{xy} = 1$ if $x \leqslant y$, and 0 otherwise. The obvious way to compute the zeta transform is to perform the matrix-vector multiplication $\vec{g} = \vec{f}\zeta$, incurring $O(v^2)$ elementary additions. However, the special structure of the transformation gives hope of performing it faster, in particular for posets of some specific form. It should perhaps be emphasized that a method of *computing* the zeta transform need not involve any explicit construction of the matrix ζ.

For a given poset, we represent the computation from f to its zeta transform g as a *straight-line program* (see Bürgisser *et al.* [4]), a sequence of elementary pairwise arithmetic operations (additions and subtractions) to be executed in turn, without loops or conditional statements. We seek to minimize the length of the program, that is, the number of arithmetic operations performed.

The zeta transform is always invertible, since ζ is an upper triangular matrix with ones on the main diagonal, assuming the matrix lists the elements of P in a linear order that is an extension of P. We also want to construct a short straight-line program for the inverse transform $\mu = \zeta^{-1}$, called the *Möbius transform.* The matrices ζ and μ are in fact representations of the zeta and Möbius functions of the poset. We refer to Stanley [12] for further discussion.

1.1 Measures of the complexity

We relate the program length to certain parameters characterizing the complexity of the poset: the number of elements or vertices (v), the number of join-irreducible elements (n), and the number of edges in the Hasse diagram of the poset (e).

In a lattice we always have $v - 1 \leqslant e \leqslant vn$.[1] As an example where e is large, consider the lattice of subsets of an n-element set. This lattice has $v = 2^n$ elements, of which n are join-irreducible (the singletons), and $e = 2^{n-1}n = \Theta(vn)$ edges. Both zeta and Möbius transforms can be computed in $\Theta(e) = \Theta(vn)$ operations by Yates's construction [14]. As an example where e is small, consider the v-element chain, which has $n = e = v - 1$, thus $vn = \Theta(e^2)$. In a chain both transforms are easily computed in e operations.

The parameter e also appears in a lower bound by Kennes [8]: for any lattice, any sequence of additions that computes the zeta transform has length at least e. Note that

[1] The first inequality holds because every non-maximal element has at least one upward edge. The latter inequality follows by considering the upward edges of an arbitrary element x and the spectrum map φ, as defined in Section 2. If both $y \gtrdot x$ and $z \gtrdot x$, then $\varphi(y) \setminus \varphi(x)$ and $\varphi(z) \setminus \varphi(x)$ must be disjoint, otherwise y and z would have two incomparable lower bounds. Hence x has at most n upward edges, and the whole lattice has $e \leqslant vn$.

this lower bound does not apply to posets in general; as a counterexample, Björklund *et al.* [2, §6] exhibit a non-lattice bipartite poset whose zeta transform can be computed in $O(\sqrt{e})$ additions, using a construction by Valiant [13].

For any lattice, Björklund *et al.* [2] construct both programs with length $O(vn)$. This upper bound is always valid but possibly quite crude. A tighter length bound $O(e)$ applies for lattices fulfilling the condition that for any element x and any join-irreducible element i that is not below x, the join $x \vee i$ covers x. Examples of such lattices include the subset lattice, the partition lattice, and the lattice of subspaces of a finite vector space. For some lattices, the resulting program length is highly dependent upon the choice of certain details in the construction. As a simple example, with the v-element chain one choice leads to the optimal $O(e)$ additions, while another choice leads to $\Theta(e^2)$ additions. We seek here a better understanding of posets that admit fast computation of the zeta and Möbius transforms.

1.2 Our results

First we show that the aforementioned condition by Björklund *et al.* [2] is equivalent to the the lattice being *geometric* (semimodular and atomic). Thus in any geometric lattice both transforms can be done in $O(e)$ arithmetic operations. This result also provides an alternative "single-axiom" characterization of geometric lattices.

Second, we show that atomicity is not needed: in any *semimodular* lattice, the construction can be done in a manner that yields a program of length e, matching Kennes's lower bound. The optimal straight-line programs for chains now follow as a special case, as well as for other semimodular lattices such as the lattice of positive divisors of an integer.

Third, for further generality, we show that if a poset admits a certain kind of *edge labeling*, then the transforms can be done in exactly e arithmetic operations by following the edges in an order implied by their labels. The idea of such labelings goes back to Stanley [11], who studied edge labelings in the context of counting chains in a lattice. Edge labelings were further studied and generalized in a series of papers by Björner and Wachs; for bibliography we refer to Stanley [12, Chapter 3] and Björner and Wachs [3, Section 5]. The class of posets admitting zeta and Möbius transforms in e operations now encompasses the family of *ER-labelable* posets. Semimodular lattices are included as a special case, as well as lower semimodular lattices (by duality) and supersolvable lattices.

The new results are summarized in Table 1 in context with previous results. Note that the family of ER-labelable posets subsumes all other families in the table (except for "all lattices").

2 Preliminaries

For a general treatment on posets and lattices we refer to Grätzer [6] and Stanley [12]. In the present work, a poset P or a lattice L is always finite. The covering relation is denoted by $\lessdot$, and the set of covering pairs, or edges of the Hasse diagram, is $E = \{(x, y) : x \lessdot y\}$.

Poset family	Program length	Reference
all lattices	$O(v^2)$	matrix-vector multiplication
subset lattices	$O(vn) = O(e)$	Yates [14]
distributive lattices (zeta only)	$O(vn)$	Parviainen and Koivisto [10]
all lattices	$O(vn)$	Björklund *et al.* [2]
lattices with a certain condition	$O(e)$	Björklund *et al.* [2]
geometric lattices	$O(e)$	new (§2)
semimodular lattices	$O(e)$	new (§3)
ER-labelable posets	$O(e)$	new (§4)

Table 1: Program lengths for the zeta and Möbius transforms.

An element $x \in P$ is *join-irreducible* if it covers exactly one element, and I is the set of join-irreducible elements. Poset size is characterized by three quantities: the number of elements $v = |P|$; the number of join-irreducible elements $n = |I|$; and the number of edges $e = |E|$.

Without loss of generality, we assume that the join-irreducible elements of a lattice are denoted by integers $I = [n]$. The *spectrum map* of an element $x \in L$ is $\varphi(x) = \{h \in I : h \leqslant x\}$, and for $i \leqslant n$, the *prefix spectrum map* is $\varphi_i(x) = \{h \in [i] : h \leqslant x\}$, with $\varphi_0(x) = \varnothing$.

In a lattice the minimum element is $\hat{0}$, and an element that covers $\hat{0}$ is an *atom*. A lattice is *atomic* if every element is a join of atoms. A lattice is *(upper) semimodular*, if for any two elements x and y such that x covers $x \wedge y$, the join $x \vee y$ in turn covers y. The dual of a semimodular lattice is *lower semimodular*. An atomic semimodular lattice is *geometric*. Examples of geometric lattices include subset lattices and partition lattices. Examples of non-atomic semimodular lattices include the chain of three or more elements, and the divisor lattice of an integer that is not a product of distinct primes (see Stanley [12, §3.3]).

Theorem 1.3 of Björklund *et al.* [2] shows that both zeta and Möbius transforms can be done in $O(e)$ arithmetic operations in any finite lattice L fulfilling the following condition:

$$x \vee i \gtrdot x \qquad \text{for all } x \in L,\ i \in I,\ x \not\geqslant i. \tag{1}$$

The following theorem provides an alternative characterization.

Theorem 1. *A finite lattice fulfills condition* (1) *if and only if it is geometric.*

Proof. For the "if" direction, let L be a finite geometric lattice, $x \in L$, and $i \in I$ such that $x \not\geqslant i$. Since L is atomic, i is an atom, and $i \gtrdot \hat{0} = x \wedge i$. Now by semimodularity $x \vee i \gtrdot x$.

For the "only if" direction, let L be a finite lattice where (1) holds. For any join-irreducible i, choosing $x = \hat{0}$ shows that $i \gtrdot \hat{0}$, thus L is atomic.

Let x and y be elements such that $y \gtrdot x \wedge y = m$. Choose a join-irreducible element i such that $i \leqslant y$ but $i \not\leqslant x$. Now $y = m \vee i$, so $x \vee y = x \vee m \vee i = x \vee i$, which covers x by condition (1). Thus L is semimodular. □

3 Fast Möbius inversion in semimodular lattices

We shall next show that atomicity is unnecessary for fast zeta and Möbius transforms: semimodularity by itself suffices to ensure programs of length e.

3.1 Fast zeta transform

We recall the fast zeta transform algorithm of Björklund *et al.* [2, §3.2]. While the original algorithm is described by embedding the lattice L into a set family $\mathcal{L}$, for the sake of transparency we describe the algorithm directly in terms of lattice operations.

Algorithm 1 Fast zeta transform

```
Input: Lattice $L$ with join-irreducibles $I = [n]$
Output: Straight-line program that, given $f$, computes its zeta transform $g$
 1: for all $x \in L$ do
 2:    print "$g(x) \leftarrow f(x)$"                         {Initialization}
 3: end for
 4: for $i = 1, 2, \ldots, n$ do
 5:    for all $x \in L$ such that $x \ngeq i$ do
 6:       $y \leftarrow x \vee i$
 7:       if $\varphi_{i-1}(x) = \varphi_{i-1}(y)$ then
 8:          print "$g(y) \leftarrow g(y) + g(x)$"                {Addition}
 9:       end if
10:    end for
11: end for
```

Algorithm 1 takes as input a representation of the lattice structure, and outputs a sequence of additions to be performed. Theorem 1.1 of Björklund *et al.* [2] shows that the resulting straight-line program indeed computes the zeta transform for any lattice, regardless of how I is ordered. The outer loop beginning on line 4 is executed n times, and the inner loop at most $|L| = v$ times, so at most $O(vn)$ additions are used; but the precise number may be much smaller, depending on lattice structure and the ordering of I. Consider a chain of v elements, with $n = e = v - 1$. Bottom-up traversal incurs e additions, which is the optimal number; each addition proceeds along an edge of the chain. In contrast, top-down traversal is suboptimal and incurs $\Theta(e^2)$ additions [2], many of which involve two distant elements that do not share an edge.

The observation that bottom-up traversal leads to fewer additions can be generalized to all semimodular lattices. The following theorem provides the optimal arrangement, whereby each addition corresponds to an edge of the lattice.

Theorem 2. *In a semimodular lattice with e edges, the join-irreducible elements can be ordered so that Algorithm 1 generates exactly e additions.*

Proof. Order the join-irreducible elements by increasing rank, breaking ties arbitrarily. Consider the situation when the condition on line 7 succeeds.

Since i is join-irreducible, there is a unique element k such that $k \lessdot i$. Since k has a rank strictly smaller than i, it follows that $\varphi_{i-1}(k) = \varphi(k)$. Now, because $k \leqslant i \leqslant x \vee i = y$, we have

$$\varphi(k) = \varphi_{i-1}(k) \subseteq \varphi_{i-1}(y) = \varphi_{i-1}(x) \subseteq \varphi(x),$$

where the equality $\varphi_{i-1}(y) = \varphi_{i-1}(x)$ is true due to the condition on line 7.

Since $\varphi(k) \subseteq \varphi(x)$, it follows that $k \leqslant x$. Thus since $i \nleqslant x$ we must have $x \wedge i = k$, implying $i \gtrdot k = x \wedge i$, and then $x \vee i \gtrdot x$ by semimodularity.

We have seen that whenever an addition is generated on line 8, it involves elements x and y such that $x \lessdot y$, so we can associate the addition with an edge of the Hasse diagram. No two additions are associated with the same edge, so the number of additions is at most e. Conversely, each edge is visited by the algorithm, since if $x \lessdot y$, then $\varphi(x) \neq \varphi(y)$, and the edge is visited with the smallest i such that $\varphi_i(x) \neq \varphi_i(y)$. Hence the algorithm perfoms exactly e additions. □

Figure 1 illustrates how the zeta transform proceeds in a semimodular lattice that has 9 edges. Nine additions are performed in four phases, corresponding to the four join-irreducible elements of the lattice.

3.2 Fast Möbius transform

Next we shall show the corresponding result for the Möbius transform, which computes f from g. Again we start with the algorithm by Björklund *et al.* [2, §3.6] expressed in terms of lattice operations.

Algorithm 2 Fast Möbius transform

```
Input: Lattice $L$ with join-irreducibles $I = [n]$
Output: Straight-line program that, given $g$, computes its Möbius transform $f$
 1: for all $x \in L$ do
 2:   print "$f(x) \leftarrow g(x)$"                        {Initialization}
 3: end for
 4: for $i = n, n-1, \dots, 1$ do
 5:   for all $x \in L$ such that $x \ngeqslant i$ do
 6:     $y \leftarrow x \vee i$
 7:     if $\varphi_{i-1}(x) = \varphi_{i-1}(y)$ then
 8:       print "$f(y) \leftarrow f(y) - f(x)$"              {Subtraction}
 9:     end if
10:   end for
11: end for
```

Algorithm 2 is similar to the zeta transform algorithm, with a few crucial changes. The join-irreducible elements are visited in reverse order, the roles of f and g are inversed, and subtraction replaces the addition on line 8. Since the conditions on lines 5–7 are the same as before, the next theorem follows.

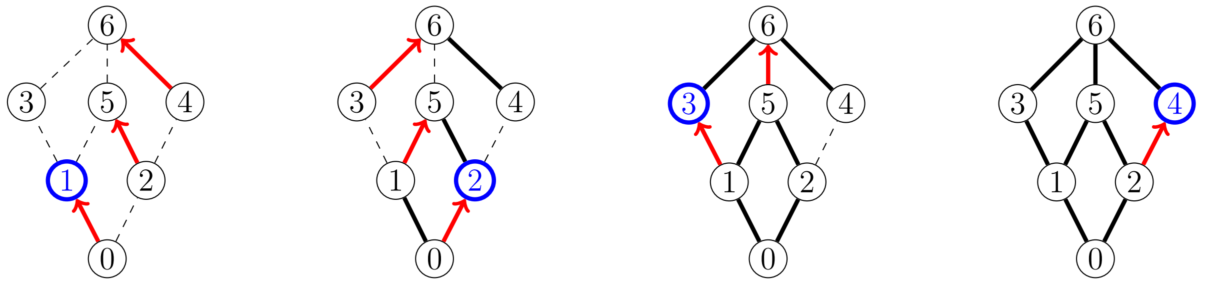

Figure 1: Fast zeta transform in a semimodular lattice. In each phase the join-irreducible element i being considered is highlighted in blue. Red arrows indicate addition along edges. Thick black edges have been visited already, and dashed edges are yet unvisited.

Straight-line program	
Phase 1	$g(1) \leftarrow g(1) + g(0)$
	$g(5) \leftarrow g(5) + g(2)$
	$g(6) \leftarrow g(6) + g(4)$
Phase 2	$g(2) \leftarrow g(2) + g(0)$
	$g(5) \leftarrow g(5) + g(1)$
	$g(6) \leftarrow g(6) + g(3)$
Phase 3	$g(3) \leftarrow g(3) + g(1)$
	$g(6) \leftarrow g(6) + g(5)$
Phase 4	$g(4) \leftarrow g(4) + g(2)$

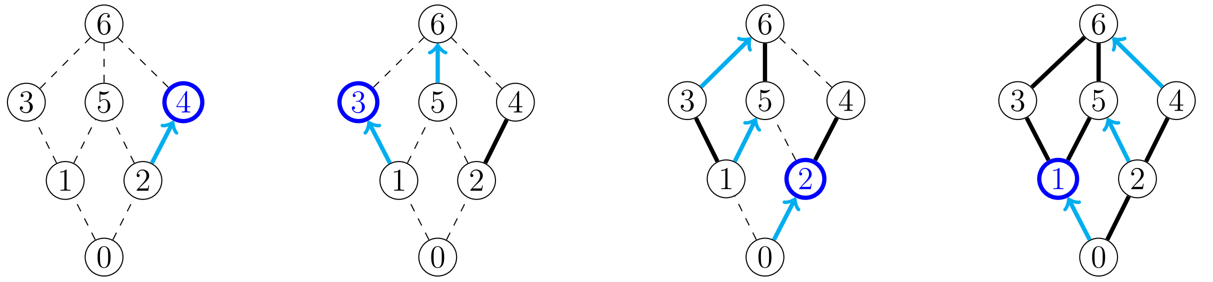

Figure 2: Fast Möbius transform in a semimodular lattice. In each phase the join-irreducible element i being considered is highlighted in blue. Cyan arrows indicate subtraction along edges. Thick black edges have been visited already, and dashed edges are yet unvisited.

Straight-line program	
Phase 4	$f(4) \leftarrow f(4) - f(2)$
Phase 3	$f(6) \leftarrow f(6) - f(5)$
	$f(3) \leftarrow f(3) - f(1)$
Phase 2	$f(6) \leftarrow f(6) - f(3)$
	$f(5) \leftarrow f(5) - f(1)$
	$f(2) \leftarrow f(2) - f(0)$
Phase 1	$f(6) \leftarrow f(6) - f(4)$
	$f(5) \leftarrow f(5) - f(2)$
	$f(1) \leftarrow f(1) - f(0)$

Theorem 3. *In a semimodular lattice with e edges, the join-irreducible elements can be ordered so that Algorithm 2 generates exactly e subtractions.*

Proof. Order and name the join-irreducible elements as $1, 2, \ldots, n$ by increasing rank, breaking ties arbitrarily. When the condition on line 7 succeeds, it holds that $x \lessdot y$, by the same reasoning as in Theorem 2. Thus the subtraction on line 8 can be associated with that edge, and total the number of subtractions performed equals the number of edges. □

Figure 2 illustrates how the Möbius transform proceeds on a semimodular lattice. Note that join-irreducible elements are considered in the reverse order $4, 3, 2, 1$.

4 Fast Möbius inversion in posets labelable with unique rising chains

Our next addition algorithm (Algorithm 3) is constructed to proceed along the edges of a poset. By this we mean that the straight-line program consists of e lines of the form

$$g(y) \leftarrow g(y) + g(x),$$

one for each edge (x, y), where $x \lessdot y$. The edges are visited in an order specified by an *edge labeling* $\lambda : E \to \mathbb{Z}$. The algorithm requires the labels to be distinct so that the order is unambiguous. The inverse transform is obtained by reversing the order of operations and replacing additions with subtractions (Algorithm 4).

Algorithm 3 Add along edges

```
Input: Poset $P$ with injective edge labeling $\lambda : E \to \mathbb{Z}$
Output: Straight-line program
for all $x \in P$ do
    print "$g(x) \leftarrow f(x)$"                          {Initialization}
end for
for all $(x, y) \in E$ in increasing order of $\lambda$ do
    print "$g(y) \leftarrow g(y) + g(x)$"                   {Addition}
end for
```

Algorithm 4 Subtract along edges

```
Input: Poset $P$ with injective edge labeling $\lambda : E \to \mathbb{Z}$
Output: Straight-line program
for all $x \in P$ do
    print "$f(x) \leftarrow g(x)$"                          {Initialization}
end for
for all $(x, y) \in E$ in decreasing order of $\lambda$ do
    print "$f(y) \leftarrow f(y) - f(x)$"                   {Subtraction}
end for
```

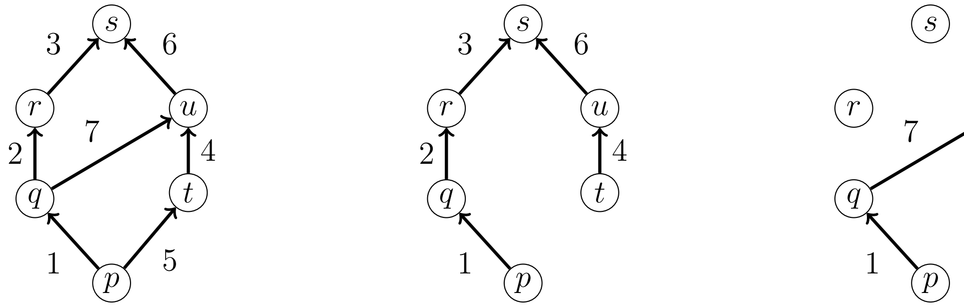

Figure 3: **Left:** A poset with an injective ER-labeling. **Center:** The rising chains that propagate input terms to $g(s)$. Note that the input term from each element below s appears exactly once in the result. For example, $f(q)$ propagates to $g(s)$ along the rising chain $q \lessdot r \lessdot s$, but not along the non-rising chain $q \lessdot u \lessdot s$. **Right:** The rising chains that propagate input terms to $g(u)$.

Both Algorithms 3 and 4 generate exactly e operations by design, in contrast to Algorithms 1 and 2, which do so for semimodular lattices but not in general. We will next formulate a sufficient condition to ensure that Algorithm 3 computes the zeta transform (and that consequently Algorithm 4 computes the Möbius transform).

Consider two comparable elements $x \leqslant y$ and an unrefinable chain C from x to y,

$$x = x_0 \lessdot x_1 \lessdot \cdots \lessdot x_m = y.$$

With a labeling $\lambda : E \to \mathbb{Z}$, we say that C is *rising* if its labels are strictly increasing,

$$\lambda(x_0, x_1) < \lambda(x_1, x_2) < \cdots < \lambda(x_{m-1}, x_m).$$

An edge labeling $\lambda : E \to \mathbb{Z}$ is an *ER-labeling* (edge rising labeling) if, for each pair of elements $x \leqslant y$, there is a unique rising chain from x to y.[2]

If λ is an injective ER-labeling and C is a rising chain from x to y, then Algorithm 3 performs additions along C in the chain order. Thus the input term $f(x)$ propagates along C to the output $g(y)$. Conversely, if C is not rising, then some of the additions in the chain are performed out of the chain order, and $f(x)$ does *not* propagate to $g(y)$ along C. Figure 3 illustrates this in a poset equipped with an ER-labeling. Hence we have following sufficient condition.

Proposition 4. *If λ is an ER-labeling, then the straight-line programs from Algorithms 3 and 4 compute the zeta and Möbius transforms, respectively.*

Proof. For each pair $x \leqslant y$, the straight-line program from Algorithm 3 propagates $f(x)$ up to $g(y)$ exactly once, along the unique rising chain from x to y. Thus for each element

[2] The name "ER-labeling" is due to Björner and Wachs [3, Definition 5.2]. Strictly speaking their definition applies to a *chain-edge labeling* induced by an edge labeling. We use here a simpler definition as edge labelings are sufficient to our present purposes.

$y \in P$, the output $g(y)$ equals $\sum_{x \leqslant y} f(x)$ as required. Hence the result is the zeta transform.

The program from Algorithm 4 consists of subtractions that undo the additions in reverse order, hence it performs the Möbius transform. □

A poset that admits an ER-labeling is *ER-labelable.* An ER-labeling is not necessarily injective as required by Algorithms 3 and 4, but an injective ER-labeling can be produced as follows.

Proposition 5. *If a poset has an ER-labeling, then it also has an injective ER-labeling.*

Proof. Let P be a poset with an ER-labeling λ. Order the edges by λ; among edges that have the same label, order by their first elements in descending order (with respect to the poset order $\leqslant$). Break any remaining ties arbitrarily. With the edges thus ordered, define an injective labeling λ' by assigning labels $1, 2, \ldots, e$ in order.

Consider now an unrefinable chain C. If C is not rising under λ, then it contains three elements $s \lessdot t \lessdot u$ such that either $\lambda(s,t) > \lambda(t,u)$, or $\lambda(s,t) = \lambda(t,u)$. In both cases, by construction, we have $\lambda'(s,t) > \lambda'(t,u)$, so C is not rising under λ'.

Conversely, if C is rising under λ, then with any three consecutive elements $s \lessdot t \lessdot u$ in C, we have $\lambda(s,t) < \lambda(t,u)$. Then by construction $\lambda'(s,t) < \lambda'(t,u)$. Hence C is rising under λ'.

Since λ and λ' have the same rising chains, λ' is also an ER-labeling. □

By combining Propositions 4 and 5 we obtain our main result for ER-labelable posets.

Theorem 6. *In an ER-labelable poset that has e edges, the zeta transform can be computed in e additions, and the Möbius transform in e subtractions.*

ER-labelable posets are a generalization of *R-labelable* posets: an R-labelable poset is a graded ER-labelable poset. Thus Theorem 6 provides fast zeta and Möbius transforms for all R-labelable posets, which include supersolvable lattices and semimodular lattices (see Stanley [12, §3.14]). For a semimodular lattice an ER-labeling is obtained by naming the join-irreducible elements as $1, 2, \ldots, n$ in an order compatible with the lattice (for example, ordering by rank), and defining

$$\lambda(s,t) = \min\{i : s \vee i = t\}.$$

Lower semimodular lattices are ER-labelable as well. More generally, duality preserves ER-labelability: if λ is an ER-labeling for poset P, then $\lambda^*(s,t) = -\lambda(t,s)$ is an ER-labeling for P^*. This is illustrated in Figure 4. If we take this induced ER-labeling, convert it to an injective one, and run Algorithm 3, we obtain the zeta transform for the lower semimodular lattice shown on the right in 9 additions. In comparison, Algorithm 1 can use up to 11 additions (depending on how I is ordered).

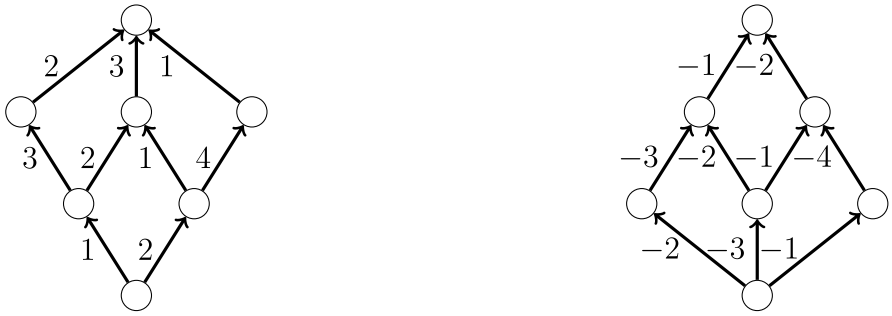

Figure 4: An upper semimodular lattice with an ER-labeling (left), and its dual with the induced ER-labeling (right).

An ER-labelable poset need not be graded. As an example, the pentagon lattice (Figure 5, left) is not graded but has an ER-labeling, facilitating both transforms in 5 operations. The hexagon lattice (Figure 5, right) cannot be ER-labeled. This would require rising chains from p to r and from q to s, implying that the chain $p \lessdot q \lessdot r \lessdot s$ is rising; but similarly $p \lessdot t \lessdot u \lessdot s$ would be rising, so there would be two rising chains from p to s.

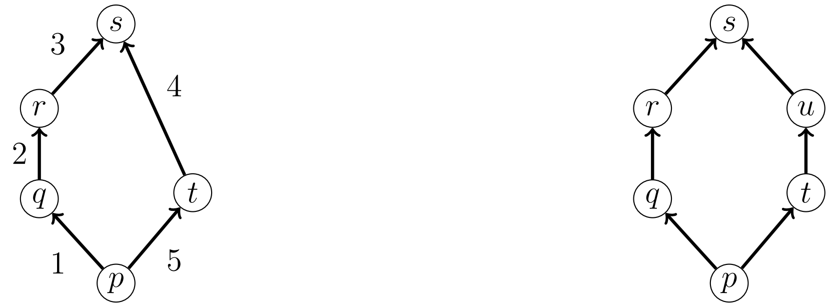

Figure 5: The pentagon is ER-labelable (left), but the hexagon is not (right).

In the hexagon it is also impossible to compute the zeta transform in e pairwise operations (additions or subtractions); at the minimum, $7 = e + 1$ operations are required. This can be seen by observing that at least four operations are required to compute the four outputs at q, r, t, and u, and then enumerating the possibilities of computing the remaining output $g(s)$.

Proposition 4 can in fact be reversed, giving a method of recognizing whether a given edge labeling is an ER-labeling. If an edge labeling is not injective, it can first be converted into an injective one by Proposition 5. Care must be taken to specify how the additions are performed. Consider, for example, addition in a Boolean algebra: there might well be two or more rising chains from x to y, but since $f(x) + f(x) = f(x)$, the result might still be the zeta transform of the input. A simple solution is to perform the additions in a free abelian group with all input values distinct.

Proposition 7. *Let λ be an injective edge labeling in poset P, and let $f(x) = x$ in the free abelian group whose basis is P. If Algorithm 3 computes the zeta transform, then λ is an ER-labeling.*

Proof. Let g be the result from running Algorithm 3. Consider any two comparable elements $a \leqslant b$ in P. Since g is the zeta transform of f, in particular $g(b)$ contains the term $f(a) = a$ exactly once. This implies that there is exactly one rising chain from a to b, otherwise $g(b)$ would contain a either zero times or more than once. □

The converse of Theorem 6 does not hold for posets in general. There are posets that are not ER-labelable, but admit the zeta transform in e operations or less (for example the bipartite poset mentioned in the introduction, augmented with a hexagon on top). However, for lattices this seems to be an open question: if a lattice admits the zeta transform in e operations, is it necessarily ER-labelable?

Another open question concerns whether there are any posets where the zeta transform requires more than $2e$ operations. The hardest known instance, in terms of e, seems to be a lattice constructed from k parallel chains of k elements, adjoined with a common bottom and a common top, with a total of $e = k(k+1)$ edges. If only addition is available, then Theorem 1 of Järvisalo *et al.* [7] implies that computing the zeta transform for this lattice requires $2e - O(\sqrt{e})$ additions.

Acknowledgements

The research that led to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement 338077 "Theory and Practice of Advanced Search and Enumeration."

We thank Marcus Greferath for the conjecture that became Theorem 1, and Matti Karppa and the anonymous referee for invaluable comments.

References

[1] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: Fast subset convolution. In *Proc. 39th Annual ACM Symposium on Theory of Computing*, STOC'07, pages 67–74, 2007.

[2] Andreas Björklund, Thore Husfeldt, Petteri Kaski, Mikko Koivisto, Jesper Nederlof, and Pekka Parviainen. Fast zeta transforms for lattices with few irreducibles. *ACM Transactions on Algorithms*, 12:4:1–19, 2015.

[3] Anders Björner and Michelle L. Wachs. Shellable nonpure complexes and posets. I. *Transactions of the American Mathematical Society*, 348:1299–1327, 1996.

[4] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic Complexity Theory*. Springer, 1997.

[5] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979.

[6] George Grätzer. *Lattice Theory: Foundation*. Birkhäuser, 2011.

[7] Matti Järvisalo, Petteri Kaski, Mikko Koivisto, and Janne H. Korhonen. Finding efficient circuits for ensemble computation. In *Proc. 15th International Conference on Theory and Applications of Satisfiability Testing*, SAT'12, pages 369–382, 2012.

[8] Robert Kennes. Computational aspects of the Möbius transformation of graphs. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:201–223, 1992.

[9] Martin E. Malandro. Fast Fourier transforms for finite inverse semigroups. *Journal of Algebra*, 324:282–312, 2010.

[10] Pekka Parviainen and Mikko Koivisto. Bayesian structure discovery in Bayesian networks with less space. In *Proc. 13th International Conference on Artificial Intelligence and Statistics*, AISTATS'10, pages 589–596, 2010.

[11] Richard P. Stanley. Supersolvable lattices. *Algebra Universalis*, 2:197–217, 1972.

[12] Richard P. Stanley. *Enumerative Combinatorics: Volume 1*. Cambridge University Press, second edition, 2011.

[13] Leslie G. Valiant. Negation is powerless for Boolean slice functions. *SIAM Journal on Computing*, 15:531–535, 1986.

[14] Frank Yates. *The Design and Analysis of Factorial Experiments*. Imperial Bureau of Soil Science, 1937.