Embedding Partially Ordered Sets into Chain-Products

Michel Habib, Marianne Huchard and Lhouari Nourine

Département d'Informatique Fondamentale
LIRMM, Université Montpellier II — CNRS UMR C09928
161 rue Ada, 34392 Montpellier Cedex 5, FRANCE.
(email: (habib,huchard,nourine)@lirmm.fr).

Abstract. Embedding partially ordered sets into chain-products is already known to be NP-complete (see Yannakakis [30] for dimension or Stahl and Wille [26] for 2-dimension). In this paper, we introduce a new dimension parameter and show that encoding using terms (or k-dimension) is not better than bit-vector (or 2-dimension) and vice versa. A decomposition theory is introduced using coatomic lattices. An algorithm is provided to compute the associated decomposition tree. Such a tree is unique for a lattice and we show how it allows bit-vectors encoding computations. In the meantime a conjecture of Caseau for 2-dimension is discussed.

Keywords: Atomic, Embedding, Hierarchy Encoding, Partially ordered sets, Dedekind-MacNeille Completion, k-Dimension.

In recent applications in computer science (cf. Aït-Kaci *et al* [3] for logic programming, Caseau [5] for object programming, Agrawal *et al* [1] for databases, Ellis [9] for conceptual graphs management and Mattern [23] for distributed systems), the problem of partial order encoding has come into light. In those applications big hierarchies have to be efficiently stored in a computer. Efficient here means that the total storage is optimal with respect to fast answers for reachability queries (i.e. $x \leq y$?). In some particular cases, the hierarchy is the directed covering graph of a lattice and some extra operations are required such as the computation of $x \vee y$ and $x \wedge y$ for any two elements x and y (see Caseau [5]). For a survey of these applications and encoding techniques, see Fall [10]. Another well studied particular case is obtained for trees in which efficient nearest common ancestor computations are needed (see Harel and Tarjan [17]).

1 Notations

A partially ordered set $P = (X, \leq_P)$ is a reflexive, antisymmetric and transitive binary relation on a set X. We denote by $<_P$ the strict ordering associated with P. When necessary, we may consider P as a directed graph (X, E) where $E \subset X^2$ and $(x, y) \in E$ iff $x \leq_P y$. Two distinct elements x and y are said to be comparable if $x <_P y$ or $y <_P x$. Otherwise they are said to be incomparable (denoted by $x||y$). We say that y covers x (denoted by $\prec$) iff $x <_P y$ and there is no z such that $x <_P z <_P y$; if y covers x then x is an immediate predecessor of y. The directed graph $(X, \prec)$ is called the directed covering graph of P.

An element $z \in X$ is an upper bound of $x, y \in X$ if $x \leq_P z$ and $y \leq_P z$. The element z is called the least upper bound or *join* of x and y if $z \leq_P t$ for all upper bounds t of x and y. The greatest lower bound or *meet* is defined dually. We denote $x \vee y$ (resp. $x \wedge y$) the least upper bound (resp. greatest lower bound) of x and y. A non-empty ordered set P is called a lattice if $x \vee y$ and $x \wedge y$ exist for all $x, y \in X$. It is clear that a finite lattice has one minimal element and one maximal element denoted respectively by $\perp$ and $\top$. Since we are dealing with algorithms the lattices we consider are supposed to be finite. For a recent book on lattice theory the reader is referred to Davey and Priestley [6].

Let L be a lattice. An element $x \in L$ is said to be join-irreducible if $x \neq \perp$ and $x = y \vee z$ implies $x = y$ or $x = z$ for all $y, z \in L$. In other words x covers only one element. Meet-irreducible elements are defined dually. We denote the set of join-irreducible elements of L by $J(L)$ and the set of meet-irreducible elements by $M(L)$. An atom is a join-irreducible element which covers $\perp$, a coatom is a meet-irreducible element which is covered by $\top$. Definitions on orders and lattices not given here, can be found in Davey and Priestley [6].

In this paper we first discuss the general notion of coding a partially ordered set and then introduce a new general parameter (namely: *encoding dimension*). The remainder of the paper is devoted to bit-vectors encoding (dim_2), and we propose a general framework for an heuristic based on a decomposition theory for lattices. In the meantime some Caseau's conjecture is detailed and commented.

2 The general coding problem

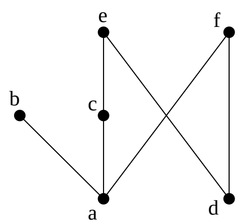
Let us settle this general problem:

Encoding Problem: *Given an order $P = (X, \leq)$. Find an embedding (injective map) $f : X \to Y$ such that $x \leq_P y$ iff $\rho(f(x), f(y)) = 1$ where $\rho : Y \times Y \to \{0, 1\}$.*

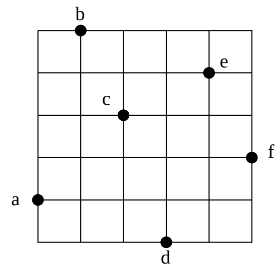
Such a pair (f, ρ) is called an encoding of P. In fact P is order-isomorphic to a substructure —i.e. $(f(X), \rho)$— of Y, more precisely $\rho(f(x), f(y)) = 1$ can be written as $f(x) \leq_\rho f(y)$, f is an embedding of P into a particular incidence structure (Y, ρ, Y) and therefore our encoding is a particular case of the coding notion as defined by Bouchet [4] see also Habib *et al* [13].

Well known encodings can be considered in this framework:

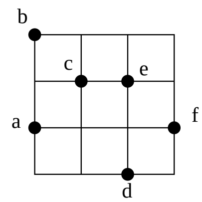
- When $P = (X, \leq_P)$ is an interval order, $f : X \to \mathcal{I}$, where $\mathcal{I}$ is a set of intervals of the real line, $I_x \rho I_y$ just says that I_x is totally to the left of I_y.



(a) A poset



(b) An embedding into a 6×6 grid



(c) Another embedding into a 4×4 grid

Fig. 1. Example of encoding

- When $P = (X, \leq_P)$ is some transitive orientation of a permutation graph (a 2-dimensional order), $f : X \rightarrow N^2$, where N is a chain of length $|X|$ and $f(x) = (x_1, x_2) \leq_\rho f(y) = (y_1, y_2)$ iff $x_1 \leq_N y_1$ and $x_2 \leq_N y_2$ (simple product ordering). Such a function f yields an encoding of P in $(N^2, <)$ see Figure 1(b).

Figure 1(c) show another embedding but in this case $f(x) = (x_1, x_2) \leq_\rho f(y) = (y_1, y_2)$ iff $x_1 \leq_N y_1$ and $x_2 \leq_N y_2$. Note that in this case the definition uses $\leq$. This can also be applied when the length of chains is bounded, it is known as the k-dimension.

Clearly the encoding yielded by Figure 1(b) need $2 * \lceil log_2 6 \rceil = 6$ bits per element, while that of Figure 1(c) gives a 4 bits encoding per element.

Standard dimension or Dushnik-Miller's dimension (see next section for a definition) or some variations as defined by Gambosi *et al* [11], can also be inserted in this framework.

Among all possible encodings of an order P, we are only interested in those in which for each element of P some memory storage is provided (sometimes called a *label* as Kannan *et al* [20]). Other encodings exist but are not studied in the following (for example there is no need of labels to encode the order $P = ([1, n], \leq_P)$ such that $i \leq_P j$ iff j is divisible by i).

They are several points to consider for encoding comparisons:

(i) Storage size.
(ii) Complexity of query ($x \leq y$?), related to the chosen representation for ρ.
(iii) Incremental aspects. In some cases (as for example in the analysis of distributed computations, Mattern [23]) new comparabilities are supposed to be given via some linear extension of P.

If P is a lattice and computations of $x \vee y$ and $x \wedge y$ are needed, very often $f(x \vee y)$ can easily be evaluated from $f(x)$ and $f(y)$ and therefore the complexity of these operations relies on the computation of the inverse function f^{-1}. For some systems (*e.g.* LIFE), decoding $f(x \vee y)$ is only performed when necessary. In such applications, the need for efficient decoding is less important.

In this first approach we focus on the two first points which are in fact closely related. But in order to avoid quicksands in our encoding comparisons, it is necessary to go back to bit operations and memory size computed in bits. Therefore let us denote by $s(f(x))$ the number of bits needed to store the label associated to an element x of P, and $s(f(P)) = \sum_{x \in P} s(f(x))$.

Such an encoding (f, ρ) is called optimal for a given class of orders $\mathcal{C}$, if:

(1) $max\,\{s(f(P))|P \in \mathcal{C}_n\} \in O\,(log_2|\mathcal{C}_n|)$ where $\mathcal{C}_n$ denotes the set of orders in $\mathcal{C}$ having exactly n elements.
(2) The query ($x \leq y$?) can be tested in $O\left(\frac{log_2|\mathcal{C}_n|}{n}\right)$ bits operations.

So for a given class of orders $\mathcal{C}$, there are two interesting questions:

– Does $\mathcal{C}$ admit an optimal encoding ?
– Can this optimal encoding be polynomially computed ?

To make easy reachability testing, most of the known encodings use fixed length labels, and therefore there is a constant number of bits associated to each element. In such a case, necessarily $s(f(x)) \geq \lceil log_2|P|\rceil$ for $x \in P$.

Previously defined encodings for interval orders or 2-dimensional orders are obviously optimal, since for every $x \in P$, $s(f(x)) = 2\lceil log_2|P|\rceil$. It should be noticed that such logarithmic sized labels are not available for all classes of orders, since the logarithm of the number of orders up to isomorphism with n elements is in $O(n^2)$ (see Kleitman and Rothschild [22]).

For nearest common ancestor (NCA) calculations in a tree, Harel and Tarjan [17], and also Schieber and Vishkin [25], propose encoding algorithms using $O(log_2 n)$ bits for each vertex which allow NCA computations and therefore reachability in $O(log_2 n)$ bits operations for a tree on n vertices.

Let us consider now an order whose covering graph is a tree (sometimes called tree-like order), it is well known that such orders are two dimensional and therefore as seen previously admit an optimal encoding. Unfortunately not much is known for the general case, and many different approaches for order encoding have been proposed in the literature.

One of these encodings uses in an heuristic way the previous optimal encoding for a spanning tree of the order, see Agrawal *et al* [1] and Talamo and Vocca

[27]. Other approaches are based on dimension approximation see Ellis [9] and Habib and Nourine [16, 15].

In the next section a new dimension parameter is introduced, namely the *encoding dimension* and we show how it can measure encoding into chain-products.

3 Encoding dimension

Let us start with the general definition of dimension of partially ordered sets.

Definition 1. The k-dimension of P, $k \geq 2$, denoted by $dim_k(P)$, is the smallest positive integer n for which P is isomorphic to a subposet of K^n (ie. K^n is the product of n chains of length k).

When $k = |P|$, using Hiraguchi's theorem [19], this definition is equivalent to the well-known Dushnik-Miller's dimension [7] simply denoted by $dim(P)$. For a survey on dimension theory the reader is referred to Trotter's book [29].

If $k = 2$ the *2-dimension* has many applications for taxonomies encoding [3, 5, 9, 15]. To each element of the taxonomy a $0 \perp 1$ configuration is associated and named *bit-vector*.

Equivalently the 2-dimension is the size of the smallest hypercube in which P can be embedded.

Obviously $dim(P) \leq dim_{n-1}(P) \leq \ldots \leq dim_2(P)$.

Several questions naturally arise:

- For a given partial order P, which k would give the best encoding in a computer?
- Determine the smallest integer k, $2 \leq k \leq |P|$, such that $dim(P) = dim_k(P)$?
- If $dim_k(P)$ is known, is there a polynomial time algorithm to compute $dim_{k-1}(P)$ or $dim_{k+1}(P)$?

Clearly any class of orders with bounded dimension has an optimal encoding. On the other hand there exist classes of orders with unbounded dimension (i.e. for any integer k there exists an order P in the class satisfying $dim(P)\, k$), for which an optimal encoding is known (see for example the interval orders, Rabinovitch [24]). But it is not a chain-product encoding.

Moreover when looking carefully at Ellis's work [9], in his search of the best encoding, he is looking for an embedding into some chain-product but in fact he allows chains to be of length 1 and 2 or more, such a coding is called *flat term encoding*.

So there is a need for new dimension parameters closer to encodings. A first formalization of the optimal encoding problem can be obtained with the following invariant which measures the efficiency of an embedding into a chain-product.

Definition 2. The *Encoding Dimension* of a partial order denoted by **edim(P)** is the least integer t such that $t = \sum_{i=1}^{i=p} \lceil log_2 k_i \rceil$ and P can be embedded into $K_1 \times K_2 \times \ldots \times K_p$, where K_i denotes a chain of length k_i.

Let us show how this new parameter captures various notions of encoding.

Any partial order $P = (X, \leq)$ can be stored using $|X|.edim(P)$ bits and the computation of any query $(x \leq y\ ?)$ requires $edim(P)$ bits operations. So for chain-product encodings $edim(P)$ captures criteria (i) and (ii).

Trivially $edim(P) \leq \lceil log_2 k \rceil dim_k(P)$, with two particular cases: $edim(P) \leq \lceil log_2|P| \rceil dim(P)$ and $edim(P) \leq dim_2(P)$. It should be noticed that these two bounds are incomparable as shown in the following examples.

- If P is a total order, then $\lceil log_2|P| \rceil dim(P) = \lceil log_2|P| \rceil$ which is smaller than $dim_2(P) = |P|$. $dim_k(P) = \lceil \frac{n-1}{k-1} \rceil$.
- Now consider a generalized crown $P = S_n^0$ defined by Trotter as a bipartite order with $min(P) = \{x_1, ..., x_n\}$, $max(P) = \{y_1, ..., y_n\}$ and for each $i \in [1..n]$, $x_i || y_i$ in P and $x_i <_P y_j$ for each $j \neq i$. Then $dim_2(P) = n$ which is smaller than $\lceil log_2|P| \rceil dim(P) = \lceil log_2 2n \rceil n$.
- Another standard example, if P is an antichain then $dim(P) = 2$ and from Sperner theorem [12] we have

$$dim_2(P) = Min\{k | \binom{k}{\lfloor \frac{k}{2} \rfloor} \geq |P|\} \leq\ 2log|P|$$

  Thus $edim(P) = 2log|P|$.
- Let P be the grid 2×8, i.e. the product of a chain with 2 elements by a chain with 8 elements, then $dim_2(P) = height(P) = 9$ and $dim(P) = 2$ therefore: $\lceil log_2|P| \rceil dim(P) = 8$, but $edim(P) = 1 + 3 = 4$. Such an example shows that $edim$ cannot be always reduced to some dim_k parameter. Therefore $edim(P)$ is associated to the most compact embedding of P.

A natural way to obtain bounds for $edim(P)$, is to compute some embedding in a product of k-chains and then try to compact the chains by deleting some coordinates (see de la Higuera and Nourine [18] for two dimensional orders).

Clearly $edim(P)$ is monotonic, i.e. $\forall P \subseteq Q, edim(P) \leq edim(Q)$.

A very interesting result can be easily derived from Bouchet's work [4], see also Habib *et al* [13]. Let us denote by $DM(P)$ the Dedekind MacNeille completion of an order P (i.e. the smallest lattice in which P can be embedded).

Theorem 3. *$edim(P) = edim(DM(P))$.*

Proof. Follows immediately from Bouchet's work [4], since $K_1 \times K_2 \times \ldots \times K_p$ is a complete lattice.

This implies that $\Pi_{i=1}^{i=edim(P)} k_i \geq |DM(P)|$. In other words this chain-product is big enough to contain $DM(P)$. It should be noticed that in some cases $|DM(P)|$ is exponential in $|P|$ (e.g. if P is a generalized crown).

Corollary 4. *Let L be a lattice and P be the suborder induced by its set of meet or join irreducible elements, then $edim(L) = edim(P)$.*

Proof. It suffices to remark that $DM(P) = L$, and to apply the above theorem.

Therefore only irreducible elements really matter for encoding dimension. It should be noticed that such a remark is still valid for any parameter defined via an embedding into a lattice, and therefore it is also true for dim_k for any k.

Property 1 *Let $\mathcal{C}$ be a class of orders. If for all $P = (X, \leq) \in \mathcal{C}$, $|X| = n$, $edim(P) \leq \phi(n)$ for some function $\phi : N \rightarrow R^+$, then $|\mathcal{C}_n| \leq 2^{n\phi(n)}$.*

Proof. Clearly, if $edim(P) \leq \phi(n)$ this implies the existence of a coding using $n\phi(n)$ bits. Therefore since two different labelled orders in $\mathcal{C}_n$ cannot have the same encoding, so $|\mathcal{C}_n| \leq 2^{n\phi(n)}$.

This property connects enumeration problems of labelled orders to encoding ones.

Corollary 5. *For a given class $\mathcal{C}$ of partial orders, $edim(P)_{P\in\mathcal{C}_n} \geq \lceil \frac{log_2|\mathcal{C}_n|}{n} \rceil$*

Therefore for any class $\mathcal{C}$ that contains the total orders, $|\mathcal{C}_n| \geq n!$ and $edim(P) \geq \lceil \frac{log_2 n!}{n} \rceil$. As already noticed for any coding, trivially $edim(P) \geq \lceil log_2|P| \rceil$. Hiraguchi's inequality yields the following inequality $\lceil log_2|P| \rceil \leq edim(P) \leq \lceil \frac{|P|}{2} \rceil \lceil log_2|P| \rceil$. Therefore it is natural to search for which classes of orders the ratio $\frac{edim(P)}{log_2|P|}$ is bounded by a constant.

Definition 6. A class $\mathcal{C}$ is said to admit an optimal chain-product encoding if $Max_{P\in\mathcal{C}_n}\{edim(P)\} \in O(\frac{log_2|\mathcal{C}_n|}{n})$.

Clearly optimal chain-product encoding implies optimal encoding as defined previously.

Property 2 *$edim(P) \leq \sum_{i=1}^{width(P)} \lceil log_2(|C_i| + 1) \rceil$, where $\{C_1, \ldots, C_{width(P)}\}$ is an optimal chain-partition of P.*

Proof. Let $\{C_1, \ldots, C_k\}$ be a chain-partition of P using $width(P) = k$ chains. Let us define $K_i = C_i \cup \{0\}$ (a zero element is added), and $x_i = max\{z \in K_i | z \leq x\}$. Then it is easy to check that $\phi : P \rightarrow K_1 \times K_2 \times \ldots \times K_k$, such that $\phi(x) = (x_1, \ldots, x_k)$ provides the required embedding. The inequality follows.

It should be noticed that Mattern [23] uses a similar idea for an incremental algorithm, but in this case he cannot know in advance the size of the chains of the partition and therefore of the size of the encoding.

For bounded width partial orders, this encoding yields obviously optimal encodings. But in general, it does not seem to provide optimal encodings. For a distributive lattice $L = 2^P$ the above result can be strengthened.

Property 3 *$edim(2^P) = min\{k | k = \sum_{i=1}^{h} \lceil log_2(|C_i| + 1) \rceil$, where $\{C_1, \ldots, C_h\}$ is a chain-partition of P $\}$.*

Proof. As already defined in the previous property the embedding ϕ can be enlarged to 2^P. Indeed P is the suborder of 2^P induced by the join irreducible elements, and therefore the extension is simply done using the usual lattice operations. It is also quite simple to check that such an extension is also an embedding from 2^P into $K_1 \times K_2 \times \ldots \times K_k$. As already noticed by Aigner [2], the converse is also true, i.e. from any embedding of 2^P into some chain-product it is easy to generate a chain-partition of P.

The above property is a slight variation of a classical Trotter's result on k-dimension, [28], and using the same correspondence between chain-partitions of P and embeddings of 2^P into chain-products, one can prove in particular that $dim(2^P) = width(P)$ and $dim_2(2^P) = |P|$.

Exact determination of $edim(P)$ seems to lead to very hard combinatorial problems (see [28], [12] for dim_2), but there is some hope to obtain relatively good approximation algorithms. Fortunately these algorithms seem to be easier to implement than for usual dimension contradicting somehow a Kelly and Trotter's claim in [21] p.176. The rest of the paper deals with the 2-dimension, or bit-vector encodings.

4 The Coatomic Tree of a Lattice

Let $f : L \rightarrow 2^k$ be an embedding of L into the hypercube of dimension k. Then $x <_L y$ iff for all $1 \leq i \leq k$, $f(x)(i) \leq f(y)(i)$ with $0 < 1$. By $f(x)(i)$, we mean the i^{th} bit in the bit-vector $f(x)$ associated to x. Sometimes, we use $f(x)$ as a subset of $\{1, ..., k\}$, therefore $x \leq_L y$ iff $f(x) \subseteq f(y)$.

In the following, a one element poset L is a lattice of 2-dimension 0, i.e. $f(L) = \{\}$.

First let us introduce a new parameter defined for any lattice L.

Let $\mu = [x_0{=}\bot, x_1, ..., x_n{=}\top]$ be a maximal chain in L. We denote by $Indegree(x_i)$ the number of immediate predecessors of x_i in the covering graph of L, and $C\Delta(\mu) = \sum_{x_i \in \mu} Indegree(x_i)$. Finally $C\Delta(L) = Max\{Degree(\mu)$ such that μ is a maximal chain in $L\}$.

Out of Caseau's work, we can produce the following conjecture.

Caseau's conjecture: For any lattice L, $dim_2(L) \leq C\Delta(L)$.

In fact, Caseau claimed that his algorithm produces in particular cases an encoding satisfying this inequality. But we were not able to give yet a simple proof for the general case (i.e. even finding an exponential algorithm which achieves this bound). Therefore we propose this bound as a challenge for bit-vectors encodings.

Only for special cases (namely series-parallel, tree-like or distributive lattices) we were able to build such encodings. To this aim, let us now consider a general decomposition theory for lattices.

Let us now introduce our basic lattice:

Definition 7. A lattice L is said to be coatomic (resp. atomic) iff every element of L is meet (resp. join) of coatoms (resp. atoms).

Property 4 *Let L be a coatomic lattice with k coatoms. Then* $dim_2(L) \leq k$*.*

Proof. It suffices to notice that $dim_2(L) \leq |M(L)|$ [15]. The idea of the encoding is to give to each coatom a_i, $f(a_i) = \{1, ..., k\} \setminus \{i\}$. Afterwards the codes are propagated downwards using intersection, so that $\forall x, f(x) = \{i \in [1, k] | x < a_i\}$. By convention $f(\top) = \{1, ..., k\}$.

The property 4 gives an encoding for coatomic lattices, but unfortunately such a code is not always optimal. Indeed let us consider a k-diamond with k coatoms (see Figure 2). By property 4 it is possible to encode it with k bits, but we can do much better as it is shown in the next property.

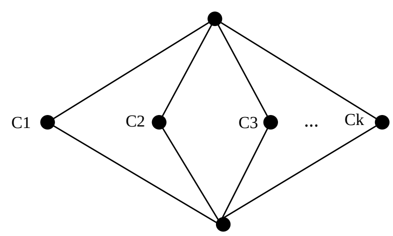


Fig. 2. A k-diamond with k coatoms

Definition 8. Let L be a coatomic lattice with k coatoms. We say that L has a **Sperner encoding** if there exists an embedding of L into an hypercube having $\lceil 2 * log\ k \rceil$ coatoms.

Property 5 *Let L be a lattice with k coatoms. If L is a k-diamond then it has the Sperner property. Moreover*

$$dim_2(L) = Min\{n | \binom{n}{\lfloor \frac{n}{2} \rfloor} \geq k\} \leq \ \lceil 2 * log\ k \rceil$$

Proof. Notice that all $\lceil log\ k \rceil$-elements subsets of an $\lceil 2 * log\ k \rceil$-elements set are incomparable for the inclusion order. It is easy to verify that $\binom{\lceil 2 * log\ k \rceil}{\lfloor log\ k \rfloor} \geq k$.

To generate k $\lceil log\ k \rceil$-elements subsets, a nice algorithm can be found in [8] (Algorithm 7, page 510).

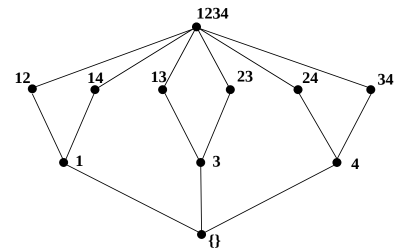


Fig. 3. A coatomic lattice having a Sperner encoding

Remark. The above example of a k-diamond lattice shows the hardness for dim_2 computations, since $dim_2(k \perp diamond) \in O(log\ k)$ and $C\Delta(k \perp diamond) = k+1$. Therefore in this case Caseau's bound is exponentional in dim_2, and such a bound is, as far as we know, still unproved ...

Problem: Determine which coatomic lattices admit a Sperner encoding ?

A unique tree will be associated to any lattice. This tree is based on coatomic lattices, and a dual tree is based on atomic lattices.

Definition 9. Let L be a lattice with k coatoms $\{a_1, ..., a_k\}$. We define

- a mapping $Mark : L \rightarrow 2^k$ as follows:
  1. $Mark(\top(L)) = \{\}$
  2. $\forall x \in L \quad Mark(x) = \{i \text{ such that } x \leq_L a_i\}$
- an equivalence relation $\sim$ on L as follows:
  Let $x, y \in L$ then $x \sim y$ iff $Mark(x) = Mark(y)$

To this equivalence relation $\sim$ corresponds a partition of L into disjoint non-empty subsets $\{C_1, ..., C_m\}$ called classes. To each class C_i we associate a single representative c_i. The quotient of L by the relation $\sim$ is denoted by $L/\sim$.

Property 6 *The quotient $L/\sim = \{c_1, ..., c_m\}$ ordered by the relation $<_\sim$ defined by :*

$$c_i <_\sim c_j \text{ iff } Mark(c_i) \supset Mark(c_j)$$

is a coatomic lattice.

Proof. Clearly $L/\sim$ is an order since its vertices are ordered by inclusion.

Let us show that $L/\sim$ is a lattice.

The class reduced to the top element of L is the unique maximal element in $L/\sim$, since $Mark(\top(L)) = \{\}$.

The class containing the bottom element of L has $Mark(\perp(L)) = \{1, ..., k\}$ and therefore it is the unique minimal element in $L/\sim$.

Let c_i and c_j be two incomparable elements. Suppose that c_i and c_j have two minimal upper bounds c_s and c_t. Then we have by definition

$$Mark(c_s) \subset Mark(c_i) \text{ and } Mark(c_s) \subset Mark(c_j)$$

and

$$Mark(c_t) \subset Mark(c_i) \;\; and \;\; Mark(c_t) \subset Mark(c_j)$$

Let c_l be the representative of the class containing $c_i \vee c_j$ in L. By definition of $\sim$, we have $Mark(c_l) \supseteq Mark(c_i) \cap Mark(c_j)$. Thus $Mark(c_l) \supset Mark(c_s) \cup Mark(c_t)$ implies $c_l <_\sim c_s$ and $c_l <_\sim c_t$, a contradiction. So $L/\sim$ is a lattice.

Let us show that $L/\sim$ is coatomic.

We must show that the only meet-irreducible elements are those classes containing the coatoms of L which are also coatoms of $L/\sim$.

Let c_j be a meet-irreducible (not a coatom) in $L/\sim$ and c_l its unique cover element. Clearly, we have $Mark(c_l) \subset Mark(c_j)$. As $Mark(c_l) \neq Mark(c_j)$ then there exists a_i such that $c_j <_L a_i$ and $c_l \,||_L\, a_i$. So there exists a class c_s covering c_j, such that $c_j \prec_\sim c_s \leq_\sim c_t$ where c_t is the class containing a_i. Then a contradiction.

Property 7 *A lattice L is coatomic iff each equivalence class of $\sim$ is reduced to a single element.*

Proof. Let L be a coatomic lattice. Then each element is a meet of coatoms. So if $x \neq y$ then $Mark(x) \neq Mark(y)$.

Now suppose that L is not coatomic. Then there exists a meet-irreducible x covered by y and x is not a coatom. So, $Mark(x) = Mark(y)$, a contradiction with the fact that each class is reduced to one element.

Definition 10. Let $P = (X, <_P)$ be an order and let $Q = (Y, <_Q)$ be a suborder of P. Q is said convex if $\forall x, y \in Y$, $\forall z \in X$ such that $x <_P z <_P y$, we have $z \in Y$.

Property 8 *Each equivalence class C_i is a convex suborder of L. Moreover either C_i or $C_i \cup \{\bot\}$ is a lattice. This lattice is denoted by L_i.*

Proof. Let $x, y \in C_i$, then $Mark(x) = Mark(y)$. Let $z \in L$ be such that $x <_L z <_L y$. Then $Mark(z) \supset Mark(x)$ and $Mark(y) \supset Mark(z)$, thus $Mark(x) = Mark(y) = Mark(z)$ and $z \in C_i$; thus C_i is a convex suborder of L.

Let $Mark(c_i) = \{a_{i_1}, ..., a_{i_l}\}$. Then the set $\{a_{i_1}, ..., a_{i_l}\}$ has a unique lower bound x with $Mark(x) = \{a_{i_1}, ..., a_{i_l}\}$ then $x \in C_i$ and it is maximal in C_i.

Since C_i is a convex suborder of L then C_i or $C_i \cup \{\bot\}$ is a lattice.

Clearly each lattice L can be partionned into smaller lattices $\{L_i | i \in [1..m]\}$ where m is the number of classes. Thus each lattice L_i can also be partionned into subclasses by the same relation $\sim$. And this process can be continued until each class is reduced to a single element.

In the following we will show that this process gives a unique tree that we call the **Coatomic Tree** of L denoted by $\mathcal{T}(L)$. Notice that a dual tree can be associated, we call the **Atomic Tree** of L.

Definition 11. Let L be a lattice with k coatoms $\{a_1, ..., a_k\}$. Then we associate to L its coatomic tree $\mathcal{T}(L)$ as follows:

1. If $L = \{x\}$ —i.e. a one element lattice— then $\mathcal{T}(L)$ is a leaf with label x.
2. Otherwise, $\mathcal{T}(L)$ contains the quotient $L/\sim$ in its root, and $\mathcal{T}(L_i)$ $i \in \{1, ..., m\}$ as subtrees, where L_i, $i \in \{1, ..., m\}$ are lattices corresponding to equivalence classes of L by $\sim$. Furthermore, we link $\mathcal{T}(L_i)$ as a son of its representative in $L/\sim$.

Remark. Clearly, by definition the coatomic tree is unique.

```
Procedure Compute-Tree(L: Lattice) ;
Data: L = (X, E) a covering graph of a lattice with k coatoms {a_1, ..., a_k}.
Result: The coatomic tree T of L.
begin
  if L = {x} then
    Create a leaf with label x
  else
    Let L/~ = {c_1, ..., c_m} be the coatomic lattice and {L_1, ..., L_m}
    the lattices induced by ~.
    Create a node N with L/~;
    for i = 1 to m do
      Son(c_i) <- Compute-Tree(L_i);
end
```

Theorem 12. *Let $L = (X, E)$ be the covering graph of a lattice. Then the algorithm Compute-Tree computes its unique coatomic tree in $O(|X| * |E|)$ time complexity. Moreover the space complexity of the coatomic tree is $O(|X| + |E|)$.*

Proof. Let us describe the algorithm.

- Each coatom of L is given a mark. This mark is added to its descendants via a depth-first-search traversal. When all coatoms are treated, the vertices of L are marked with lists automatically sorted by the initial sorting of the coatoms. Clearly, each depth-first-search costs $O(|E|)$, since $|E| > |X|$.
  Now, since each element is a coatom in at most one of the sublattices in the coatomic tree, there will be at most $O(|X|)$ depth-first searches in total (*i. e.* for all steps). Thus, the overall complexity for the marking is $O(|X| * |E|)$.
- We will delete all edges (x, y) such that $Mark(x) \neq Mark(y)$. This can be done by a unique traversal of L, comparing the marks of the ends of every edge. Each comparison between $Mark(x)$ and $Mark(y)$ can be done in $O(Max(|Mark(x)|, |Mark(y)|))$, which is less than the number of coatoms at this step. When considering all steps, an element is coatom at most for one step, so all the comparisons for the edge (x, y) can be done in $O(|X|)$.

The classes C_i and the corresponding lattices L_i are connected components obtained from L after edge-deletion. So the overall complexity of retrieving the L_i and $L/\sim$ will be $O(|X| * |E|)$.
– Using the deleted edges, one can obtain in $O(|E|)$ the lattice $L/\sim$.

It remains to notice that the size of all classes does not exceed the size of L, since edges deleted in a step are not further considered in next steps, and the fact that at each step at least one class (i.e. maximal class in $L/\sim$) is reduced to a single element. Thus the announced time complexity follows.

Let us show that the size of the tree is in $O(|X| + |E|)$. To each leaf with label x and father y, we can associate at most two vertices in the tree as follows:

– the pair (y, z) if y is the top element of the lattice of the node containing y, and z the father of y.
– the vertex y if y is not the top element of the lattice of its node.

This association allows to cover all vertices inside the nodes. Then the number of vertices of all the lattices in the coatomic tree is at most $3 * |X|$. Each edge in a lattice of a node corresponds to an edge of the covering graph of L. And the result follows.

Figure 5 shows the construction of the coatomic tree of the lattice in Figure 4.

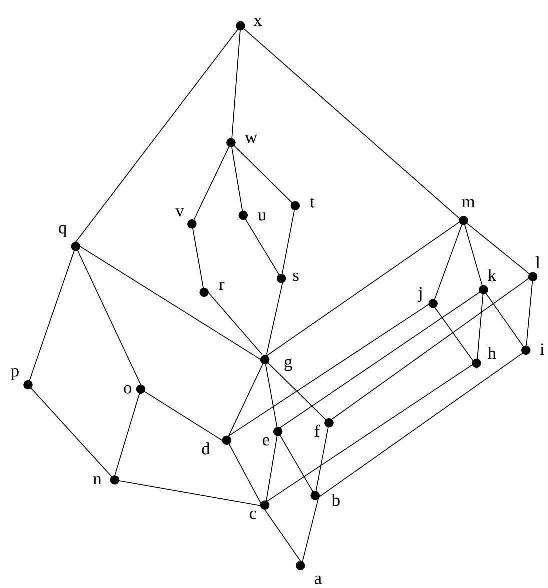


Fig. 4. An example of lattice

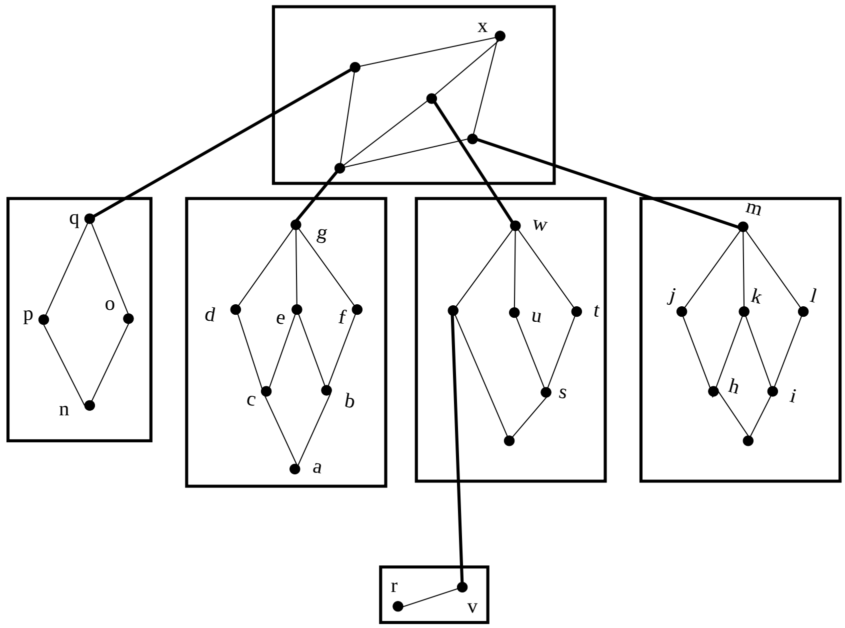


Fig. 5. The coatomic tree

5 Bit-vector encodings via coatomic tree

In this section we will show how this structure can be used to produce encoding algorithms. First let us see this structure in details throughout some classes of lattices as tree (i.e. $L \setminus \bot$ is a tree), series-parallel (i.e. L is a series-parallel order) or distributive lattice. Series parallel orders are based on two operations (series composition and parallel composition). Distributive lattices are based on factorization. Finally we give general ideas to obtain an encoding algorithm.

Let us state the following property which gives the relationship between the equivalence classes of $\sim$.

Property 9 *Let L be a lattice, let C_i, C_j be two equivalence classes of $\sim$. Then*

1. $C_i <_\sim C_j$ *implies* $\forall x \in C_i,\ \ \forall y \in C_j,\ \ x \not\geq_L y$.
2. $C_i ||_\sim C_j$ *implies* $\forall x \in C_i,\ \ \forall y \in C_j,\ \ x ||_L y$.

Proof. 1. Let $C_i <_\sim C_j$ then $\forall x \in C_i\ \ \forall y \in C_j,\ Mark(x) \supset Mark(y)$. Suppose that $x \geq_L y$. By definition of $\sim$ we have $Mark(y) \supset Mark(x)$, a contradiction.

2. The same proof as (1) can be applied.

Thus if two classes are incomparable in $L/\sim$ their encoding can be done separately using the same integers, and they will be distinguished by a code given to their representative.

But if two classes are comparable, it is difficult to encode them without extra computations. Before dealing with the general case, let us capture this idea through interesting classes of lattices.

5.1 Special classes of lattices

Definition 13. An order P is said to be series-parallel if it does not contain a suborder isomorphic to "N" (i.e. a, b, c, d with comparabilities $a < c$, $a < d$ and $b < d$).

Property 10 *Let L be a series-parallel lattice with $\{a_1, ..., a_k\}$ coatoms. Then $\forall i, j$ $C_i <_\sim C_j$ iff $\forall x \in C_i$ $\forall y \in C_j$ $x <_P y$. Moreover each class covers only one other class (i.e. $(L/\sim) \setminus \{\top\}$ is a tree).*

Proof. $\Longrightarrow$ Let $C_i <_\sim C_j$ and $x \in C_i$, $y \in C_j$. Suppose that x is incomparable to y. There exists a_s such that $x < a_s$ and $y < a_s$. Since $i \neq j$ then there exists a_t such that $x < a_t$ and $y||_L a_t$. Thus the induced order by x, y, a_t, a_s is isomorphic to "N".

Notice that if $y = a_s$ then we have $x <_L y$ since $Mark(x) \supset Mark(y)$.

$\Longleftarrow$ Assume that $\forall x \in C_i$ $\forall y \in C_j$ $x <_P y$. Clearly $Mark(x) \supset Mark(y)$ and therefore $C_i <_\sim C_j$.

Now let C_i and C_j be two classes covered by C_t. Since C_i is incomparable to C_j then there exists a_s with $a_s > c_i$ and a_s incomparable with c_j. Thus the induced order by c_i, c_j, a_s, c_t is isomorphic to "N".

Trees are particular class of series-parallel lattices, and we can obtain easily the following nice property:

Property 11 *If $L \setminus \{\bot\}$ is a tree with k coatoms. Then $L/\sim$ is a diamond with k coatoms.*

The last particular case is distributive lattices. This class of lattices is based on factorization (isomorphism property see Habib and Nourine [14]).

Property 12 *Let L be a distributive lattice. Then*

1. *$L/\sim$ is a boolean lattice.*
2. *Let $c_1, ..., c_n$ be a maximal chain of $L/\sim$. Then each class C_i $i \in [1..n]$ is isomorphic to a sublattice of $C_1 \cup ... \cup C_{i-1}$.*

Proof. 1. Trivial.

2. Clearly $C_1 \cup ... \cup C_{i-1}$ and C_i are distributive sublattices of L. Moreover $C_1 \cup ... \cup C_{i-1} \cup C_i$ is distributive. Thus C_i is isomorphic to a sublattice of $C_1 \cup ... \cup C_{i-1}$.

Now let us see how these properties can be used to have efficient encoding algorithms.

5.2 Encoding Algorithms

Let L be a lattice to be encoded, and $\mathcal{T}$ its coatomic tree. The idea is to encode the lattice associated with the root of $\mathcal{T}$, using the property that these lattices are coatomic. Notice that the code of a coatomic lattice L can be optimized when L has a Sperner encoding (see figure 3). Clearly the integers used to encode a node can not be used for encoding its sons. But the main problem is the following:

Can the sons of a node use the same integers in their codes?

The answer is trivially yes for sons which are incomparable, but is not the case for comparable sons. Let us captures this on the previous classes.

1. If L is a tree then all sons are incomparable (see property 11). Moreover the son associated with the bottom element of $L/\sim$ is empty. Thus we can encode them using the same integers. Notice that in this case a parallel algorithm can be used.
2. If L is series-parallel then we use the same integers for incomparable sons. But if $C_i <_\sim C_j$ then the code of every element in C_i is also in each element of C_j (see property 10), thus we must use other integers.
3. If L is a distributive lattice then we need to encode only the son associated with the bottom element of $L/\sim$. All other sons will be encoded by isomorphism property since they will be distinguished by the code of their father that is $L/\sim$ (see property 12).

From these, one can derive an algorithm for general case as follows:

Let N be a node with L its associated lattice. If N is a leaf then it is already encoded, otherwise:

- Encode the coatomic lattice L. For each element of L, propagate its code to its associated son.
- Take $\tau = \{c_1, ..., c_m\}$ a linear extension of L.
- Encode the son corresponding to c_1. Propagate the code of each element in C_1 to its successors in L.
- For each son $C_i \;\; i \in [2..m]$
  - Verify if C_i is already encoded by the propagated code from $\{C_1, ..., C_{i-1}\}$. (this is true for distributive lattices). Moreover if the sons of each maximal chain of L induce a distributive lattice this remains true.
  - Otherwise use other integers than those used by comparable sons already encoded. This step is crucial for optimization, since there exist some elements which are already encoded by propagation and it remains to complete this encoding by using other integers.
  - Propagate the code of each element in C_i to their successors in L.

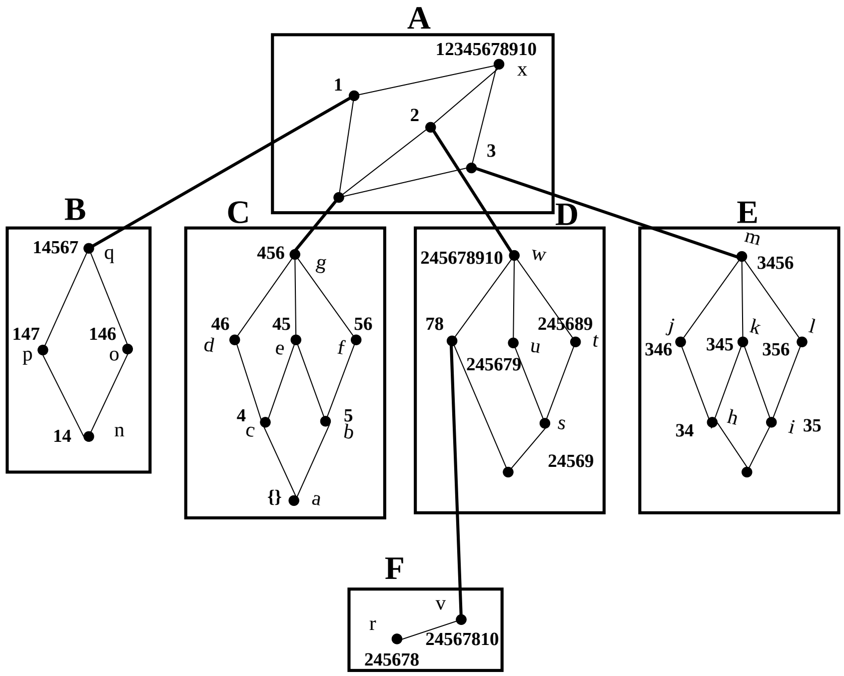


Fig. 6. An encoding via the coatomic tree

Example of encoding: Let us show an encoding of the lattice in figure 4 based on its coatomic tree. First we encode the coatomic lattice in node A using the integers $\{1, 2, 3\}$. Now we encode the sons of A through a linear extension without using $\{1, 2, 3\}$. First the node C is encoded using integers $\{4, 5, 6\}$. Notice that the node E is already encoded by propagation of codes from C (isomorphism property). Now the node D must use different integers than C (series operation), thus $\{7, 8, 9\}$ is used. The node F is the only son, so it is encoded using $\{10\}$. The last node B shows the general case. Indeed the vertices $\{n, o, q\}$ are already encoded by propagation, so we need only to encode p which receives the integer $\{7\}$.

5.3 Applications to partially ordered sets

It should be noticed that our algorithm can be easily generalized to encode not only lattices but also any partial order. Indeed let P be a partial order with a maximal element and a minimal element. The marking map applied to P leads to a partition of P into classes $C_1, ..., C_m$. The difference between orders and lattices is that for lattices each C_i has a maximal element. Thus to apply the marking map to classes, we add to each class a maximal and/or a minimal element.

The quotient $P/\sim$ is also an order but not a lattice. If P has k elements covered by the top element then $P/\sim$ can be encoded using k bits.

Thus the encoding algorithm for lattice can be also be used for partially ordered sets. Furthermore the coatomic tree associated with an order is unique and sometimes gives the Dedekind-MacNeille completion as for series-parallel orders.

6 Discussions

Finding near optimal bit-vectors encoding for lattices and orders seems to be an extremely hard problem. But there is still some hope to produce good polynomial heuristics.

Among the previous algorithms, the algorithm of Habib and Nourine [15] gives efficient encoding when the number of meet-irreducible or join-irreducible elements is small (proportionally to its height), whereas Caseau's algorithm is supposed to give good encoding when the input lattice L has a small $C\Delta(L)$. These two approaches are orthogonal since for boolean lattices our encoding is optimal and his encoding is the square of optimal encoding. For lattices with large number of meet-irreducible elements his algorithm is much better than ours.

In this paper, we have proposed a tree structure for lattices and orders which takes into account the two previous ideas. The proposed encoding algorithm uses and exploits the following properties: parallel, series, isomorphism.

The Sperner property allows us to optimize encoding for antichains. In section 4 we have extended this property to a Sperner encoding, which means encode a lattice in the low half of an hypercube. Let us end with an open question, related to improvement of the previous encoding heuristics.

Problem: When L is coatomic, does there exist a polynomial algorithm to compute the 2-dimension?

Acknowledgments; The authors wish to thank the anonymous referees for their valuable suggestions, and Andrew Fall for his remarks that lead to an improvement of the complexity for computing the coatomic tree.

References

1. Rakesh Agrawal, Alex Borgida, and H.V. Jagadish. Efficient management of transitive relationships in large data bases, including is-a hierarchies. *ACM SIGMOD*, 1989.
2. M. Aigner. *Combinatorial Theory*. Springer-Verlag, 1979.
3. Hassan Aït-Kaci, Robert Boyer, Patrick Lincoln, and Roger Nasr. Efficient implementation of lattice operations. *ACM Transactions on Programming Langages and Systems*, 11(1):115–146, january 1989.
4. A. Bouchet. Codages et dimensions de relations binaires. *Annals of Discrete Mathematics 23, Ordres: Description and Roles, (M. Pouzet, D. Richard eds)*, 1984.
5. Yves Caseau. Efficient handling of multiple inheritance hierarchies. In *OOPSLA'93*, pages 271–287, 1993.

6. B. A. Davey and H. A. Priestley. *Introduction to lattices and orders*. Cambridge University Press, second edition, 1991.
7. B. Dushnik and E.W. Miller. Partially ordered sets. *Amer. J. Math.*, 63:600–610, 1941.
8. Gideon Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. of ACM*, 20(3):500–513, july 1973.
9. Gerard Ellis. *Managing Complex Objects*. PhD thesis, University of Queensland, in preparation, PHD thesis 1995.
10. A. Fall. The foundations of taxomic encoding. submitted to ACM Trans. on Programming Languages and Systems. Also available as Simon Fraser University Technical report CSS/LCCR 94-20., 1995.
11. G. Gambosi, J. Nesetril, and M. Talamo. Posets, boolean representations and quick path searching. In *Proc. of the 14th International colloque on Automata, Languages and Programming, Lecture Notes in Computer Science 267, Springer-Verlag, Berlin*, 1987.
12. J.R. Griggs, J. Stahl, and W.T. Trotter. A sperner theorem on unrelated chains of subsets. *J. Comb. theory. (A)*, pages 124–127, 1984.
13. M. Habib, M. Morvan, M. Pouzet, and J.-X. Rampon. Incidence structure, coding and lattice of maximal antichains. Technial report., 1991.
14. M. Habib and L. Nourine. A linear time algorithm to recognize distributive lattices. Research Report 92-012, to appear in *Order*, LIRMM, Montpellier, France, March 1993.
15. M. Habib and L. Nourine. Bit-vector encoding for partially ordered sets. In Proc. of International Workshop ORDAL'94, editor, *Orders, Algorithms and Applications*, number 831, pages 1–12, Lyon, France, July 1994. Springer Verlag.
16. M. Habib and L. Nourine. Tree structure for distributive lattices and its applications. Research report, LIRMM, Montpellier, France, Avril 1994. to appear in Theoretical Computer Science.
17. D. Harel and R.E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Computing*, 13:338–355, 1984.
18. C. De La Higuera and L. Nourine. Embedding two dimensional posets. Research report, LIRMM, Montpellier, France, January 1995. to appear in Theoretical Computer Science.
19. T. Hiraguchi. On the dimension of orders. *Sci. Rep. Kanazawa Univ.*, 4:1–20, 1955.
20. S. Kannan, M. Naor, and S. Rudich. Implicit representation of graphs. *SIAM J. Disc. Math.*, 5(4):596–603, November 1992.
21. D. Kelly and W.T. Trotter. Dimension theory for ordered sets. In I. Rival, editor, *Ordered Sets*, pages 171–211. D. Reidel Publishing Company, 1982.
22. D.J. Kleitman and B.L. Rothschild. Asymptotic enumeration of partial orders on a finite set. *Transactions of the American Mathematical Society*, 205:205–220, 1975.
23. F. Mattern. Virtual time and global states of distributed systems. In M. Cosnard et al., editor, *Parallel and Distributed Algorithms*, pages 215–226. Elsevier / North-Holland, 1989.
24. I. Rabinovitch. An upper bound on the dimension of interval orders. *J. Combinatorial Theory*, Ser. A, 25:68–71, 1955.
25. B. Schieber and U. Vishkin. On finding lowest common ancestors: simplification and parallelization. *SIAM J. Computing*, 17:1253–1262, 1988.
26. J. Stahl and R. Wille. Preconcepts of contexts. *in Proc. Universal Algebra (Sienna)*, 1984.

27. M. Talamo and P. Vocca. Fast lattice browsing on sparse representations. In Proc. of International Workshop ORDAL'94, editor, *Orders, Algorithms and Applications*, number 831, pages 186–204, Lyon, France, July 1994. Springer Verlag.
28. William T. Trotter. Embedding finite posets in cubes. *Discrete Mathematics*, 12:165–172, 1975.
29. W.T. Trotter. *Combinatorics and Partially Ordered Sets: Dimension Theory*. Press, Baltimore. John Hopkins University, 1991.
30. M. Yannakakis. The comlexity of the partial order dimension problem. *SIAM J. Alg. Disc. Meth.*, 3:351–358, 1982.