An Algorithm for Dualization in Products of Lattices and its Applications*

Khaled M. Elbassioni

Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway NJ 08854-8003; elbassio@paul.rutgers.edu

Abstract. Let $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ be the product of n lattices, each of which has a bounded width. Given a subset $\mathcal{A} \subseteq \mathcal{L}$, we show that the problem of extending a given partial list of maximal independent elements of $\mathcal{A}$ in $\mathcal{L}$ can be solved in quasi-polynomial time. This result implies, in particular, that the problem of generating all minimal infrequent elements for a database with semi-lattice attributes, and the problem of generating all maximal boxes that contain at most a specified number of points from a given n-dimensional point set, can both be solved in incremental quasi-polynomial time.

1 Introduction

Let $\mathcal{L} \stackrel{\text{def}}{=} \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ be the product of n partially ordered sets (posets). We assume that each poset $\mathcal{L}_i$ is a lattice, i.e., for every pair of elements $x, y \in \mathcal{L}_i$, there exists a unique minimum element called the *meet* and a unique maximum element called the *join*. Throughout we shall denote by $\preceq$ the precedence relation in $\mathcal{L}$, and use, as customary, $\vee$ and $\wedge$ to denote the join and meet operators over $\mathcal{L}$. For $\mathcal{A} \subseteq \mathcal{L}$, denote by $\mathcal{A}^+ = \{x \in \mathcal{L} \mid x \succeq a, \text{ for some } a \in \mathcal{A}\}$ and $\mathcal{A}^- = \{x \in \mathcal{L} \mid x \preceq a, \text{ for some } a \in \mathcal{A}\}$, the ideal and filter generated by $\mathcal{A}$. Any element in $\mathcal{L} \setminus \mathcal{A}^+$ is called *independent of* $\mathcal{A}$. Let $\mathcal{I}(\mathcal{A})$ be the set of all maximal independent elements for $\mathcal{A}$ (also referred to as the *dual* of $\mathcal{A}$):

$$\mathcal{I}(\mathcal{A}) \stackrel{\text{def}}{=} \{p \in \mathcal{L} \mid p \notin \mathcal{A}^+ \text{ and } (q \in \mathcal{L}, q \succeq p, q \neq p \;\Rightarrow\; q \in \mathcal{A}^+)\}.$$

Then for any $\mathcal{A} \subseteq \mathcal{L}$, we have the following decomposition of $\mathcal{L}$:

$$\mathcal{A}^+ \cap \mathcal{I}(\mathcal{A})^- = \emptyset, \qquad \mathcal{A}^+ \cup \mathcal{I}(\mathcal{A})^- = \mathcal{L}. \tag{1}$$

Call a subset $\mathcal{A} \subseteq \mathcal{L}$ an antichain if no two elements of $\mathcal{A}$ are comparable. Given $\mathcal{A} \subseteq \mathcal{L}$, we consider the problem of incrementally generating all elements of $\mathcal{I}(\mathcal{A})$:

* This research was supported in part by the National Science Foundation Grant IIS-0118635, and by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science.

DUAL$(\mathcal{L},\mathcal{A},\mathcal{B})$: *Given an antichain* $\mathcal{A} \subseteq \mathcal{L}$ *in a lattice* $\mathcal{L}$ *and a partial list of maximal independent elements* $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$*, either find a new maximal independent element* $x \in \mathcal{I}(\mathcal{A}) \setminus \mathcal{B}$*, or prove that* $\mathcal{A}$ *and* $\mathcal{B}$ *form a dual pair:* $\mathcal{B} = \mathcal{I}(\mathcal{A})$.

Clearly, the entire set $\mathcal{I}(\mathcal{A})$ can be generated by initializing $\mathcal{B} = \emptyset$ and iteratively solving the above problem $|\mathcal{I}(\mathcal{A})| + 1$ times. If $\mathcal{L}$ is the Boolean cube, i.e., $\mathcal{L}_i = \{0, 1\}$ for all $i = 1, \dots, n$, the above dualization problem reduces to the well known *hypergraph transversal* problem, which calls for enumerating all minimal subsets that intersect all edges of a given hypergraph. The complexity of the hypergraph transversal problem is still an important open question. The best known algorithm runs in quasi-polynomial time $poly(n, m) + m^{o(\log m)}$, where $m = |\mathcal{A}| + |\mathcal{B}|$ (see [12]), providing partial evidence that the problem is not NP-hard. In this note, we will extend this result by showing that the problem is still unlikely to be NP-hard when each $\mathcal{L}_i$ is a lattice with bounded width. Specifically, for $x \in \mathcal{L}_i$, denote by $x^{\perp}$ the set of immediate predecessors of x, i.e., $x^{\perp} = \{y \in \mathcal{L}_i \mid y \prec x, \nexists z \in \mathcal{L}_i : y \prec z \prec x\}$, and let in-deg$(\mathcal{L}_i) = \max\{|x^{\perp}| : x \in \mathcal{L}_i\}$. Similarly, denote by $x^{\top}$ the set of immediate successors of x, and let out-deg$(\mathcal{L}_i) = \max\{|x^{\top}| : x \in \mathcal{L}_i\}$. Let $d = \max_{i \in [n]} \max\{\text{in-deg}(\mathcal{L}_i), \text{out-deg}(\mathcal{L}_i)\}$, and $\mu = \mu(\mathcal{L}) \stackrel{\text{def}}{=} \max\{|\mathcal{L}_i| : i \in [n]\}$. Finally, denote by $W(\mathcal{L}_i)$ the *width* of $\mathcal{L}_i$, i.e. the maximum size of an antichain in $\mathcal{L}_i$ and let $W \stackrel{\text{def}}{=} \max_{i \in [n]}\{W(\mathcal{L}_i)\}$ be the maximum width of the n lattices. The main result of this paper is the following.

Theorem 1. *Problem DUAL*$(\mathcal{L},\mathcal{A},\mathcal{B})$ *can be solved in poly*(n, μ)+ $m^{d \cdot \rho \cdot o(\log m)}$ *time, if* $\mathcal{L}$ *is a product of lattices, where* $m = |\mathcal{A}| + |\mathcal{B}|$ *and* $\rho \stackrel{\text{def}}{=} 2W \ln(W + 1)$.

Let us remark that the special case of Theorem 1 (with a slightly stronger bound) for integer lattices $\mathcal{L}_i = \{0, 1, 2, \dots\}$ (where $W = 1$) appears in [3], and implies that the set of minimal integer solutions for a monotone system of linear inequalities can be incrementally generated in quasi-polynomial time.

Let us also note that the result of Theorem 1 can be immediately extended to products of *join semi*-lattices, i.e. under the relaxed requirement that every two elements of $\mathcal{L}_i$ have only a unique maximum element. If, moreover, each semi-lattice $\mathcal{L}_i$ has an acyclic precedence graph (i.e., the underling graph is a rooted tree), then the bound on the running time in Theorem 1 can be further improved to $poly(n, \mu) + m^{o(\log m)}$, see [9] for more details.

In the next section, we motivate Theorem 1 by some applications related to the generation of combinatorial structures described by systems of polymatroid inequalities. In particular, we illustrate that the problem of generating infrequent elements in databases with lattice attributes (data mining) and the generation of maximal boxes that contain at most a certain number of points from a given set of points (geometry), can be reduced in quasi-polynomial time to dualization in products of bounded-width lattices. The proof of Theorem 1 will be given in Section 3.

2 Generating minimal feasible solutions for systems of polymatroid inequalities

Let $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ be the product of n lattices, and consider a system of inequalities

$$f_i(x) \geq t_i, \quad i = 1, \ldots, r, \tag{2}$$

over the elements $x \in \mathcal{L}$. We assume that each function f_i, $i = 1, \ldots, r$, is polymatroid, that is, f_i is integer-valued, monotone (i.e., $f(x) \leq f(y)$ whenever $x \preceq y$), submodular:

$$f(x \vee y) + f(x \wedge y) \leq f(x) + f(y) \qquad \text{for all } x, y \in \mathcal{L},$$

and $f(l) = 0$, where l is the minimum element of $\mathcal{L}$. Let $\mathcal{F} \subseteq \mathcal{L}$ be the set of minimal elements of $\mathcal{L}$ satisfying (2). Given a set of polynomially computable functions $f_1, \ldots, f_r$, and a set of integer thresholds $t_1, \ldots, t_r$, an interesting problem is to incrementally generate all elements of $\mathcal{F}$:

GEN($\mathcal{L}, \mathcal{F}, \mathcal{X}$): *Given an antichain* $\mathcal{X} \subseteq \mathcal{F}$*, either find a new element in* $\mathcal{F} \setminus \mathcal{X}$, or prove that no such element exists: $\mathcal{X} = \mathcal{F}$.

In the next two subsections, we consider two examples for such polymatroid systems. An important property of these systems, which enables us to generate their minimal feasible set $\mathcal{F}$ in quasi-polynomial time, is that the size of the dual set $\mathcal{I}(\mathcal{F})$ is upper-bounded by a (quasi-) polynomial $p(\cdot)$ in $|\mathcal{F}|$, and the size of the input description of (2). More precisely, the following result holds (see [4] and [7]).

Theorem 2. *Let* $\mathcal{F}$ *be the set of all minimal feasible solutions for a system of polymatroid inequalities (2) over a lattice* $\mathcal{L}$ *and let* $\mathcal{X} \subseteq \mathcal{F}$ *be an arbitrary subset of* $\mathcal{F}$ *of size* $|\mathcal{X}| \geq 2$. *Then*

$$|\mathcal{I}(\mathcal{F}) \cap \mathcal{I}(\mathcal{X})| \leq r|\mathcal{X}|^{(\log t)/c(2\mu(\mathcal{L})n, |\mathcal{X}|)}, \tag{3}$$

where $t = \max\{t_1, \ldots, t_r\}$*, and* $c = c(\alpha, \beta) \sim \log\log_\alpha \beta$ *is the unique positive root of the equation* $2^c(\alpha^{c/\log\beta} - 1) = 1$.

Consequently, it is enough for such examples to consider the following problem GEN($\mathcal{L}, \mathcal{F}, \mathcal{I}(\mathcal{F}), \mathcal{A}, \mathcal{B}$) of jointly generating all elements of $\mathcal{F}$ and $\mathcal{I}(\mathcal{F})$:

GEN($\mathcal{L}, \mathcal{F}, \mathcal{I}(\mathcal{F}), \mathcal{A}, \mathcal{B}$): *Given two explicitly listed collections* $\mathcal{A} \subseteq \mathcal{F}$ *and* $\mathcal{B} \subseteq \mathcal{I}(\mathcal{F})$*, either find a new element in* $(\mathcal{F} \setminus \mathcal{A}) \cup (\mathcal{I}(\mathcal{F}) \setminus \mathcal{B})$*, or prove that these collections are complete:* $(\mathcal{A}, \mathcal{B}) = (\mathcal{F}, \mathcal{I}(\mathcal{F}))$.

It was observed in [2, 13] that, if $\mathcal{L} = \{0, 1\}^n$ is the Boolean cube, then the above joint generation problem GEN($\mathcal{L}, \mathcal{F}, \mathcal{I}(\mathcal{F}), \mathcal{A}, \mathcal{B}$) can be reduced in polynomial time to problem DUAL($\mathcal{L}, \mathcal{A}, \mathcal{B}$). In fact, it is straightforward to see that the same observation holds for any lattice $\mathcal{L}$. Clearly this observation, together with inequality (3), implies that, for systems of polymatroid inequalities (2), problem $\mathrm{GEN}(\mathcal{L},\mathcal{F},\mathcal{X})$ is quasi-polynomial time reducible to dualization since, in the process of jointly generating elements of $\mathcal{F}$ and $\mathcal{I}(\mathcal{F})$, the number of intermediate elements generated from $\mathcal{I}(\mathcal{F})$ cannot be very large. On the other hand, it is interesting to note that, for many examples of polymatroid inequalities, the dual problem of incrementally generating $\mathcal{I}(\mathcal{F})$ turns out to be NP-hard [6, 13, 16].

Corollary 1. *All minimal feasible solutions to a monotone system of polymatroid inequalities (2), over products of bounded-width lattices, can be incrementally generated in quasi-polynomial time.*

Examples of combinatorial structures described by polymatroid inequalities can be found in [4, 15, 17]. The next two subsections also describe two more such examples.

2.1 Maximal frequent and minimal infrequent elements in databases with semi-lattice attributes

The notion of frequent sets in data mining [1] has a natural generalization over products of semi-lattices. Formally, consider a database $\mathcal{D} \subseteq \mathcal{L}$ of transactions, each of which is an n-dimensional vector of attributes over $\mathcal{L}$. For an element $x \in \mathcal{L}$, denote by

$$S(x) = S_{\mathcal{D}}(x) \stackrel{\mathrm{def}}{=} \{p \in \mathcal{D} \mid p \succeq x\},$$

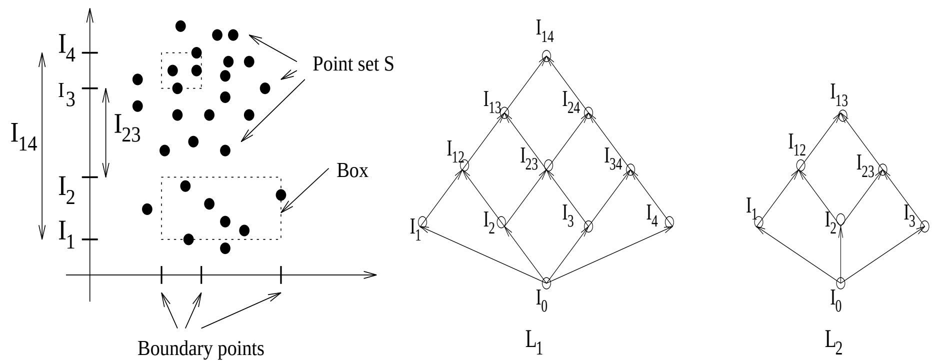
the set of transactions in $\mathcal{D}$ that *support* x. Note that, by this definition, the function $|S(\cdot)| : \mathcal{L} \mapsto \mathbb{Z}_+$ is an anti-monotone *supermodular* function, and hence, the function $f : \mathcal{L} \mapsto \mathbb{Z}_+$, defined by $f(x) = |\mathcal{D}| - |S(x)|$ is polymatroid.

Given $\mathcal{D} \subseteq \mathcal{L}$ and an integer threshold t, an element $x \in \mathcal{L}$ is called t-frequent if it is supported by at least t transactions in the database, i.e. if $|S_{\mathcal{D}}(x)| \geq t$. Conversely, $x \in \mathcal{L}$ is said to be t-infrequent if $|S_{\mathcal{D}}(x)| < t$. Denote by $\mathcal{F}_{\mathcal{D},t}$ the set of all minimal t-infrequent elements of $\mathcal{L}$ with respect to the database $\mathcal{D}$. Then $\mathcal{I}(\mathcal{F}_{\mathcal{D},t})$ is the set of all maximal t-frequent elements. It is clear that $\mathcal{F}_{\mathcal{D},t}$ is the set of minimal feasible solutions of the polymatroid inequality $f(x) \geq |\mathcal{D}| - t + 1$, and therefore can be generated in incremental quasi-polynomial time by Corollary 1.

The separate and joint generation of maximal frequent and minimal infrequent elements of a database are important tasks in knowledge discovery and data mining [1, 14, 18]. In many applications, the data attributes assume values ranging over products of semi-lattices of small width, e.g. quantitative attributes [19], taxonomies [18], and lattices of small sizes in logical analysis of data [8]. The special case of the above result for databases $\mathcal{D}$ of binary attributes can be found in [5, 6].

2.2 Packing points into boxes

Given a set of n-dimensional points $\mathcal{S} \subseteq \mathbb{R}^n$, suppose that we want to generate all maximal n-dimensional boxes that contain at most t points of $\mathcal{S}$, where



a: A 2-dimensional example. b: The corresponding lattice of intervals $\mathcal{L}_1 \times \mathcal{L}_2$.

Fig. 1. Packing points into boxes: each box has at most $t = 6$ points.

$1 \leq t \leq |\mathcal{S}|$ is a given integer threshold. Suppose further that the end points of each candidate box in each dimension, are to be selected from a small set of *boundary* points $\mathcal{P}_i$, for $i = 1, \ldots, n$. This problem has important applications in data mining, see, e.g., [11]. Interestingly, this problem can be cast as of generating minimal feasible solutions of a polymatroid inequality over a product of lattices of bounded width. Indeed, for each set of boundary points $\mathcal{P}_i$, let us construct the corresponding *lattice of intervals* $\mathcal{L}_i$ whose elements are the different intervals defined by these boundary points. The meet of any two intervals is their intersection, and the join is their span, i.e. the minimum interval containing both of them. The minimum element of $\mathcal{L}_i$ is the empty interval I_0. A 2-dimensional example is shown in Figure 1, where we denote the interval between two boundary points $I_j, I_k \in \mathcal{P}_i$ by I_{jk}. Clearly, the product $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ is the set of all possible boxes, and the problem is to generate the maximal elements of this product that contain at most t points from the set $\mathcal{S}$. Now consider the function

$$f(x) = |\{q \in \mathcal{S} \mid \text{point } q \text{ is contained inside the box } x\}|,$$

over the elements of $\mathcal{L}$, and observe that this function is supermodular. It follows then that the function $|\mathcal{S}| - f(x)$ is polymatroid over the elements x of the *dual lattice* $\mathcal{L}^*$ (that is, the lattice $\mathcal{L}^*$ with the same set of elements as $\mathcal{L}$, but such that $x \prec y$ in $\mathcal{L}^*$ whenever $x \succ y$ in $\mathcal{L}$). We conclude, therefore, that all maximal boxes that contain at most t points from $\mathcal{S}$, are the minimal solutions $x \in \mathcal{L}^*$ of the polymatroid inequality $|\mathcal{S}| - f(x) \geq |\mathcal{S}| - t$, and hence can be generated in quasi-polynomial time.

3 Dualization in products of lattices

In this section we prove Theorem 1. Let $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ where each $\mathcal{L}_i$ is a lattice with minimum element l_i and maximum element u_i. Given two antichains $\mathcal{A} \subseteq \mathcal{L}$, and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, we say that $\mathcal{B}$ is *dual to* $\mathcal{A}$ if $\mathcal{B} = \mathcal{I}(\mathcal{A})$, i.e., if $\mathcal{B}$ contains all the maximal elements of $\mathcal{L} \setminus \mathcal{A}^+$. Let us remark that, by (1), the latter condition is equivalent to $\mathcal{A}^+ \cup \mathcal{B}^- = \mathcal{L}$.

Given any $\mathcal{Q} \subseteq \mathcal{L}$, let us denote by

$$\mathcal{A}(\mathcal{Q}) = \{a \in \mathcal{A} \mid a^+ \cap \mathcal{Q} \neq \emptyset\}, \qquad \mathcal{B}(\mathcal{Q}) = \{b \in \mathcal{B} \mid b^- \cap \mathcal{Q} \neq \emptyset\},$$

the subsets of $\mathcal{A}, \mathcal{B}$ whose ideal and filter respectively intersect $\mathcal{Q}$. To solve problem DUAL$(\mathcal{L}, \mathcal{A}, \mathcal{B})$, we shall use the same general approach used in [12] to solve the hypergraph transversal problem, by decomposing it into a number of smaller subproblems which are solved recursively. In each such subproblem, we start with a sub-lattice $\mathcal{Q} = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_n \subseteq \mathcal{L}$ (initially $\mathcal{Q} = \mathcal{L}$), and two subsets $\mathcal{A}(\mathcal{Q}) \subseteq \mathcal{A}$ and $\mathcal{B}(\mathcal{Q}) \subseteq \mathcal{B}$, and we want to check whether $\mathcal{A}(\mathcal{Q})$ and $\mathcal{B}(\mathcal{Q})$ are dual in $\mathcal{Q}$. As mentioned before, the latter condition is equivalent to checking whether $\mathcal{Q} \subseteq \mathcal{A}(\mathcal{Q})^+ \cup \mathcal{B}(\mathcal{Q})^-$. To estimate the reduction in problem size from one level of the recursion to the next, we measure the change in the "volume" of the problem defined as $v = v(\mathcal{A}, \mathcal{B}) \stackrel{\text{def}}{=} |\mathcal{A}||\mathcal{B}|$. Since $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$ is assumed, the following condition holds, by (1) for the original problem and all subsequent subproblems:

$$a \not\preceq b, \quad \text{for all } a \in \mathcal{A}, b \in \mathcal{B}. \tag{4}$$

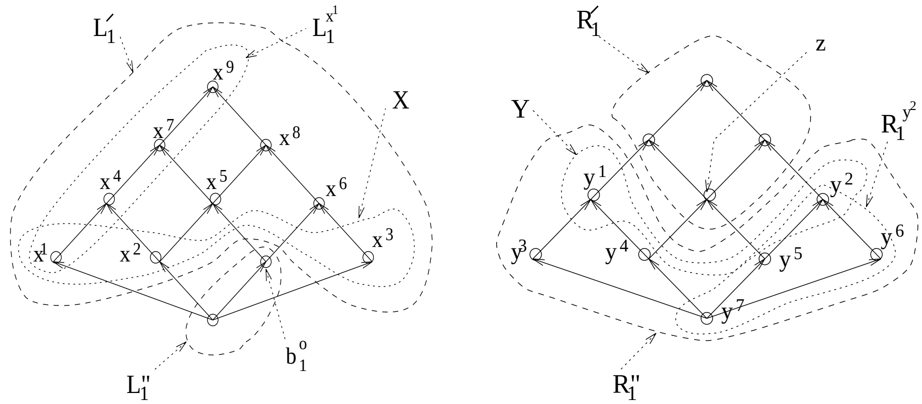
In the next section we develop several rules for decomposing a given dualization problem into smaller subproblems. The algorithm will select between these rules in such a way that the total volume is reduced from one iteration to the next. The base case for recursion is when one of the sets $\mathcal{A}$ or $\mathcal{B}$ becomes sufficiently small, in which case the problem is easily seen to be polynomially solvable.

Proposition 1. *Suppose* $\min\{|\mathcal{A}|, |\mathcal{B}|\} \leq$ *const, then problem DUAL*$(\mathcal{L}, \mathcal{A}, \mathcal{B})$ *can be solved in poly*$(n, m, \mu(\mathcal{L}))$ *time for any lattice product* $\mathcal{L}$.

Note that, after performing decomposition, we may end up with a subproblem DUAL$(\mathcal{L}', \mathcal{A}', \mathcal{B}')$, where $\mathcal{L}'$ is a sub-poset of $\mathcal{L}$, $\mathcal{A}' \subseteq \mathcal{A}$, $\mathcal{B}' \subseteq \mathcal{B}$, in which some elements of $\mathcal{A}'$, or $\mathcal{B}'$ do not belong to $\mathcal{L}'$. Clearly, if there is an $a \in \mathcal{A}'$ ($b \in \mathcal{B}'$) such that $a^+ \cap \mathcal{L}' = \emptyset$ (respectively, $b^- \cap \mathcal{L}' = \emptyset$) then this element a (respectively, b) can be *eliminated* from the subproblem. On the other hand, if $a \notin \mathcal{L}'$ but $a^+ \cap \mathcal{L}' \neq \emptyset$, then a cannot be eliminated. In such a case, it is necessary to *project* the element a to the poset $\mathcal{L}'$, by replacing it with the set of minimal elements in $a^+ \cap \mathcal{L}'$. In general this minimal set may be large, causing the number of elements of $\mathcal{A}$ or $\mathcal{B}$ to increase exponentially after a succession of decompositions. However, if the sub-poset $\mathcal{L}'$ is a lattice, then each element in $\mathcal{A}'$ or $\mathcal{B}'$ projects only to *a single* element in $\mathcal{L}'$. Thus, we shall always maintain the property that each new subproblem is defined over lattices.

3.1 Decomposition rules

In general, the algorithm will decompose a given problem by selecting an $i \in [n]$ and decomposing $\mathcal{L}_i$ into a number of sub-lattices, defining accordingly a number



a: Rules (i) and (ii). b: Rules (i') and (ii').

Fig. 2. Decomposing the lattice $\mathcal{L}_1$.

of lattice products. Specifically, let $a^o \in \mathcal{A}$, $b^o \in \mathcal{B}$ be arbitrary. By (4), there exists an $i \in [n]$, such that $a_i^o \not\preceq b_i^o$. Let us assume, without loss of generality, that $i = 1$ and set $\mathcal{L}_1' \leftarrow \mathcal{L}_1 \setminus (b_1^o)^-$, $\mathcal{L}_1'' \leftarrow \mathcal{L}_1 \cap (b_1^o)^-$. (Alternatively, we may set $\mathcal{L}_1' \leftarrow \mathcal{L}_1 \cap (a_1^o)^+$ and $\mathcal{L}_1'' \leftarrow \mathcal{L}_1 \setminus (a_1^o)^+$.) For brevity, we shall denote by $\overline{\mathcal{L}}$ the product $\mathcal{L}_2 \times \cdots \times \mathcal{L}_n$, and accordingly by $\overline{q}$ the vector $(q_2, \ldots, q_n)$, for an element $q = (q_1, q_2, \ldots, q_n) \in \mathcal{L}$.

Let $\mathcal{X}$ be the set of *minimal* elements in $\mathcal{L}_1'$ and define $\mathcal{L}_1^x = \mathcal{L}_1' \cap x^+$ for $x \in \mathcal{X}$ (see Figure 2-a). Let further

$$\mathcal{A}_1' = \{a \in \mathcal{A} \mid a_1 \not\preceq b_1^o\}, \qquad \mathcal{A}_1'' = \mathcal{A} \setminus \mathcal{A}_1',$$
$$\mathcal{B}_1' = \{b \in \mathcal{B} \mid b_1 \not\preceq b_1^o\}, \qquad \mathcal{B}_1'' = \mathcal{B} \setminus \mathcal{B}_1'.$$

Denoting by $\mathcal{L}^x = \mathcal{L}_1^x \times \mathcal{L}_2 \times \cdots \times \mathcal{L}_n$, for $x \in \mathcal{X}$, and by $\mathcal{L}'' = \mathcal{L}_1'' \times \mathcal{L}_2 \times \cdots \times \mathcal{L}_n$ the sub-lattices of $\mathcal{L}$ induced by the above decomposition, we conclude that $\mathcal{A}$ and $\mathcal{B}$ are dual in $\mathcal{L}$ *if and only if*

$$\mathcal{A}, \mathcal{B}_1' \text{ are dual in } \mathcal{L}^x, \quad \text{for all } x \in \mathcal{X}, \tag{5}$$
$$\mathcal{A}_1'', \mathcal{B} \text{ are dual in } \mathcal{L}'', \tag{6}$$

each of which is a dualization problem on lattices. Thus we obtain our first decomposition rule:

Rule (i) Solve $|\mathcal{X}|$ subproblems (5) together with subproblem (6).

Clearly, subproblems (5) and (6) are not independent. Once we know that (6) is satisfied, we gain some information about the solution of (5). The following lemma shows how to utilize such dependence to further decompose (5).

Lemma 1. *Given $z \in \mathcal{L}_1$, let $\mathcal{L}_1' = \mathcal{L}_1 \cap z^+$, $\mathcal{L}_1'' \subseteq \mathcal{L}_1 \cap z^- \setminus \{z\}$ be two disjoint subsets of $\mathcal{L}_1$. Define*

$$\mathcal{A}'' = \{a \in \mathcal{A} \mid a_1^+ \cap \mathcal{L}_1'' \neq \emptyset\}, \qquad \mathcal{A}' = \{a \in \mathcal{A} \setminus \mathcal{A}'' \mid a_1^+ \cap \mathcal{L}_1' \neq \emptyset\},$$
$$\mathcal{B}' = \{b \in \mathcal{B} \mid b_1^- \cap \mathcal{L}_1' \neq \emptyset\}, \qquad \mathcal{B}'' = \{b \in \mathcal{B} \setminus \mathcal{B}' \mid b_1^- \cap \mathcal{L}_1'' \neq \emptyset\}.$$

Suppose further that we know that $\mathcal{L}_1' \times \overline{\mathcal{L}} \subseteq (\mathcal{A}' \cup \mathcal{A}'')^+ \cup (\mathcal{B}')^-$, then

$$\mathcal{L}_1'' \times \overline{\mathcal{L}} \subseteq (\mathcal{A}'')^+ \cup (\mathcal{B}' \cup \mathcal{B}'')^- \iff \forall a \in \widetilde{\mathcal{A}} : \mathcal{L}_1'' \times (\overline{\mathcal{L}} \cap \overline{a}^+) \subseteq (\mathcal{A}'')^+ \cup (\mathcal{B}'')^-,$$

where $\widetilde{\mathcal{A}} = \{a \in \mathcal{A}' \cup \mathcal{A}'' \mid a_1 \preceq z\}$.

By considering the dual lattice of $\mathcal{L}$ (that is, the lattice $\mathcal{L}^*$ with the same set of elements as $\mathcal{L}$, but such that $x \prec y$ in $\mathcal{L}^*$ whenever $x \succ y$ in $\mathcal{L}$), and exchanging the roles of $\mathcal{A}$ and $\mathcal{B}$, we get the following symmetric version of Lemma 1.

Lemma 2. *Let $\mathcal{L}_1'' = \mathcal{L}_1 \cap z^-$, $\mathcal{L}_1' \subseteq \mathcal{L}_1 \cap z^+ \setminus \{z\}$ be two disjoint subsets of $\mathcal{L}_1$ where $z \in \mathcal{L}_1$. Let $\mathcal{A}'', \mathcal{A}', \mathcal{B}'', \mathcal{B}'$ be defined as in Lemma 1, and let $\widetilde{\mathcal{B}} = \{b \in \mathcal{B}' \cup \mathcal{B}'' \mid b_1 \succeq z\}$. Suppose we know that $\mathcal{L}_1'' \times \overline{\mathcal{L}} \subseteq (\mathcal{A}'')^+ \cup (\mathcal{B}' \cup \mathcal{B}'')^-$, then*

$$\mathcal{L}_1' \times \overline{\mathcal{L}} \subseteq (\mathcal{A}' \cup \mathcal{A}'')^+ \cup (\mathcal{B}')^- \iff \forall b \in \widetilde{\mathcal{B}} : \mathcal{L}_1' \times (\overline{\mathcal{L}} \cap \overline{b}^-) \subseteq (\mathcal{A}')^+ \cup (\mathcal{B}')^-.$$

To use Lemma 2, suppose that subproblem (6) has no solution (i.e. there is no $q \in \mathcal{L}'' \setminus [(\mathcal{A}_1'')^+ \cup (\mathcal{B}_1' \cup \mathcal{B}_1'')^-]$). We proceed in this case as follows. For $x \in \mathcal{L}_1$, let $\widetilde{\mathcal{A}}(x) = \{a \in \mathcal{A} \mid a_1 \preceq x\}$, $\widetilde{\mathcal{B}}(x) = \{b \in \mathcal{B} \mid b_1 \succeq x\}$, and $\mathcal{A}_1'(x) = \{a \in \mathcal{A}_1' \mid a_1 = x\}$. Let us use $x^1, \ldots, x^k$ to denote the elements of $\mathcal{L}_1'$ and assume, without loss of generality, that they are topologically sorted in this order, that is, $x^j \prec x^h$ implies $j < h$ (see Figure 2-a). Let us decompose (5) (which is equivalent to checking whether $\mathcal{L}_1' \times \overline{\mathcal{L}} \subseteq \mathcal{A}^+ \cup (\mathcal{B}_1')^-$) further into the k subproblems

$$\{x^j\} \times \overline{\mathcal{L}} \subseteq \left[\left(\bigcup_{y \in (x^j)^\perp} \widetilde{\mathcal{A}}(y)\right) \cup \mathcal{A}_1'(x^j)\right]^+ \cup (\mathcal{B}_1')^-, \qquad j = 1, \ldots, k. \tag{7}$$

The following lemma, which follows inductively from Lemma 2, will allow us to eliminate the contribution of the set $\mathcal{A}_1''$ in subproblems (7) at the expense of possibly introducing at most $|\mathcal{B}|^d$ additional subproblems.

Lemma 3. *Given $x^j \in \mathcal{L}_i'$, suppose we know that $(y^- \cap \mathcal{L}_1) \times \overline{\mathcal{L}} \subseteq \widetilde{\mathcal{A}}(y)^+ \cup \mathcal{B}^-$ for all $y \in (x^j)^\perp$. Then (7) is equivalent to*

$$\{x^j\} \times \left[\overline{\mathcal{L}} \cap \left(\bigwedge_{y \in (x^j)^\perp} \overline{b}(y)\right)^-\right] \subseteq \mathcal{A}_1'(x^j)^+ \cup (\mathcal{B}_1')^-, \tag{8}$$

for all collections $\{b(y) \in \widetilde{\mathcal{B}}(y) \mid y \in (x^j)^\perp\}$.

Informally, Lemma 3 says that, given $x^j \in \mathcal{L}_1'$, if the dualization subproblems for all sub-lattices that lie below x^j have been already verified to have no solution, then we can replace the solution to subproblem (7) by solving at most $\prod_{y\in(x^j)^\perp} |\widetilde{\mathcal{B}}(y)|$ subproblems of the form (8). Observe that it is important to check subproblems (7) in the topological order $j = 1, \ldots, k$ in order to be able to use Lemma 3. Thus we get

Rule (ii) Solve subproblem (6). If it has a solution then we get a point $q \in \mathcal{L} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$. Otherwise, we solve subproblems (8), for all collections $\{b(y) \in \widetilde{\mathcal{B}}(y) \mid y \in (x^j)^\perp\}$, for $j = 1, \ldots, k$ (in the topological order).

Suppose finally that we decompose $\mathcal{L}_1$ by selecting an element $z \in \mathcal{L}_1$, letting $\mathcal{R}_1' \leftarrow \mathcal{R}_1 \cap z^+$, $\mathcal{R}_1'' \leftarrow \mathcal{L}_1 \setminus z^+$, $\mathcal{R}' = \mathcal{R}_1' \times \overline{\mathcal{L}}$, and $\mathcal{R}'' = \mathcal{R}_1'' \times \overline{\mathcal{L}}$. Let $\mathcal{Y}$ denote the set of *maximal* elements in $\mathcal{R}_1''$ and define $\mathcal{R}_1^y = \mathcal{R}_1'' \cap y^-$ for $y \in \mathcal{Y}$ (see Figure 2-b). Let further

$$\mathcal{A}_2' = \{a \in \mathcal{A} \mid a_1 \succeq z\}, \qquad \mathcal{A}_2'' = \mathcal{A} \setminus \mathcal{A}_2',$$
$$\mathcal{B}_2' = \{b \in \mathcal{B} \mid b_1 \succeq z\}, \qquad \mathcal{B}_2'' = \mathcal{B} \setminus \mathcal{B}_2'.$$

By exchanging the roles of $\mathcal{A}$ and $\mathcal{B}$ and replacing $\mathcal{L}$ by its dual lattice $\mathcal{L}^*$ in rules (i), (ii) above, we can also derive the following symmetric versions of these rules:

Rule (i′) Solve the subproblem

$$\mathcal{A}, \mathcal{B}_2' \text{ are dual in } \mathcal{R}', \tag{9}$$

and the $|\mathcal{Y}|$ subproblems

$$\mathcal{A}_2'', \mathcal{B} \text{ are dual in } \mathcal{R}_1^y \times \overline{\mathcal{L}}, \quad \text{for all } y \in \mathcal{Y}. \tag{10}$$

Rule (ii′) Solve subproblem (9), and if it does not have a solution, then solve the subproblems

$$\{y^j\} \times \left[\overline{\mathcal{L}} \cap \left(\bigvee_{x\in(y^j)^\top} \overline{a}(x)\right)^+\right] \subseteq (\mathcal{A}_2'')^+ \cup (\mathcal{B}_2''(y^j))^-, \tag{11}$$

for all collections $\{a(x) \in \widetilde{\mathcal{A}}(x) \mid x \in (y^j)^\top\}$, for $j = 1, \ldots, h$, where $y^1, \ldots, y^h$ denote the elements of $\mathcal{R}_1''$ in *reverse* topological order, and $\mathcal{B}_2''(x) = \{a \in \mathcal{B}_2'' \mid b_1 = x\}$.

Finally it remains to remark that all the decomposition rules described above result, indeed, in dualization subproblems over lattices.

3.2 The algorithm

Given antichains $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ that satisfy the necessary duality condition (4), we proceed as follows:

Step 1. For each $k \in [n]$:

1. (*eliminate:*) if $a_k^+ \cap \mathcal{L}_k = \emptyset$ for some $a \in \mathcal{A}$ ($b_k^- \cap \mathcal{L}_k = \emptyset$ for some $b \in \mathcal{B}$), then a (respectively, b) can be discarded from further consideration;

2. (*project:*) if $a_k \notin \mathcal{L}_k$ for some $a \in \mathcal{A}$ ($b_k \notin \mathcal{L}_k$ for some $b \in \mathcal{B}$), set $a_k \leftarrow \bigwedge\{x \mid x \in a_k^+ \cap \mathcal{L}_k\}$ (respectively, set $b_k \leftarrow \bigvee\{x \mid x \in b_k^- \cap \mathcal{L}_k\}$).

Thus we may assume for next steps that $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L}$.

Step 2. If $\min\{|\mathcal{A}|, |\mathcal{B}|\} < \delta = 2$, then dualization can be solved in $poly(n, m, \mu)$ time.

Step 3. Let $a^o \in \mathcal{A}$, $b^o \in \mathcal{B}$. Find an $i \in [n]$ such that $a_i^o \not\preceq b_i^o$. Assume, without loss of generality, that $i = 1$ and set $\mathcal{L}_1' \leftarrow \mathcal{L}_1 \setminus (b_1^o)^-$, $\mathcal{L}_1'' \leftarrow \mathcal{L}_1 \cap (b_1^o)^-$. Let $\mathcal{X}, \mathcal{A}_1', \mathcal{A}_1'', \mathcal{B}_1', \mathcal{B}_1''$, and $\widetilde{\mathcal{B}}$ be as defined in the previous subsection, and let

$$\epsilon_1^{\mathcal{A}} = \frac{|\mathcal{A}_1'|}{|\mathcal{A}|}, \qquad \epsilon_1^{\mathcal{B}} = \frac{|\mathcal{B}_1''|}{|\mathcal{B}|}.$$

Observe that $\epsilon_1^{\mathcal{A}} > 0$ and $\epsilon_1^{\mathcal{B}} > 0$ since $a^o \in \mathcal{A}_1'$ and $b^o \in \mathcal{B}_1''$.

Step 4. Define $\epsilon(v) = \rho(W)/\chi(v)$, where $v = v(\mathcal{A}, \mathcal{B})$, $\rho(W) = 2W\ln(W+1)$, and $\chi(v)$ is defined to be the unique positive root of the equation

$$\left(\frac{\chi(v)}{\rho(W)}\right)^{\chi(v)} = \frac{v^d}{(1 - e^{-\rho(W)})(\delta^d - 1)}$$

and observe that $\epsilon(v) < 1$ for $v \geq \delta^2$, $\delta \geq 2$.

If $\min\{\epsilon_1^{\mathcal{A}}, \epsilon_1^{\mathcal{B}}\} > \epsilon(v)$, we use decomposition rule (i), which amounts to solving recursively $|\mathcal{X}|$ subproblems (5), each of which has a volume of at most $v(\mathcal{A}, \mathcal{B}_1') = |\mathcal{A}||\mathcal{B}_1'|$, and subproblem (6) of volume $v(\mathcal{A}_1'', \mathcal{B}) = |\mathcal{A}_1''||\mathcal{B}|$. This gives rise to the recurrence

$$\begin{aligned} C(v) &\leq 1 + |\mathcal{X}| C(|\mathcal{A}||\mathcal{B}_1'|) + C(|\mathcal{A}_1''||\mathcal{B}|) \\ &\leq 1 + W \cdot C((1 - \epsilon_1^{\mathcal{B}})v) + C((1 - \epsilon_1^{\mathcal{A}})v) \\ &\leq 1 + (W+1) C((1 - \epsilon(v))v), \end{aligned} \tag{12}$$

where $C(v)$ denotes the number of recursive calls on a subproblem of volume at most v.

Step 5. If $\epsilon_1^{\mathcal{A}} \leq \epsilon(v)$, we apply rule ($ii$) and get the recurrence

$$\begin{aligned} C(v) &\leq 1 + C(|\mathcal{A}_1''||\mathcal{B}|) + \sum_{j=1}^{k} \left(\prod_{y \in (x^j)^\perp} |\widetilde{\mathcal{B}}(y)| \right) C(|\mathcal{A}_1'(x^j)||\mathcal{B}_1'|) \\ &\leq 1 + C(|\mathcal{A}_1''||\mathcal{B}|) + |\widetilde{\mathcal{B}}|^d \sum_{j=1}^{k} C(|\mathcal{A}_1'(x^j)||\mathcal{B}_1'|) \\ &\leq 1 + C((1 - \epsilon_1^{\mathcal{A}})v) + |\mathcal{B}|^d C(\epsilon_1^{\mathcal{A}} v) \\ &\leq 1 + C((1 - \epsilon_1^{\mathcal{A}})v) + \frac{v^d}{\delta^d} C(\epsilon_1^{\mathcal{A}} v) \\ &\leq C((1 - \epsilon)v) + \frac{v^d}{\delta^d - 1} C(\epsilon v), \quad \text{for some } \epsilon \in (0, \epsilon(v)], \end{aligned} \tag{13}$$

where the second inequality follows from the fact that $|(x^j)^\perp| \leq d$, the third inequality follows from $\sum_{j=1}^k C(|\mathcal{A}'_1(x^j)||\mathcal{B}'_1|) \leq C(\sum_{j=1}^k |\mathcal{A}'_1(x^j)||\mathcal{B}'_1|) = C(|\mathcal{A}'_1||\mathcal{B}'_1|)$ since $\{\mathcal{A}'_1(x^j) \mid j = 1, \ldots, k\}$ is a partition of $\mathcal{A}'_1$ and the function $C(\cdot)$ is super-additive, the forth inequality follows from $|\mathcal{B}|^d \leq v(|\mathcal{A}|, |\mathcal{B}|)^d/\delta^d$, and the last inequality follows from the fact that $v \geq \delta^2$ and $\delta \geq 2$.

Step 6. We assume for next steps that $\epsilon_1^{\mathcal{A}} > \epsilon(v)$. Then there exists a point $z \in \mathcal{X}$, such that $|\{a \in \mathcal{A} \mid a_1 \succeq z\} \geq \epsilon_1^{\mathcal{A}}|\mathcal{A}|/|\mathcal{X}| > \epsilon(v)|\mathcal{A}|/W$. Let $\mathcal{R}'_1 \leftarrow \mathcal{R}_1 \cap z^+$, $\mathcal{R}''_1 \leftarrow \mathcal{L}_1 \setminus z^+$, and let $\mathcal{Y}, \mathcal{A}'_2, \mathcal{A}''_2, \mathcal{B}'_2, \mathcal{B}''_2, \widetilde{\mathcal{A}}$ be as defined in the previous subsection. Let also

$$\epsilon_2^{\mathcal{A}} = \frac{|\mathcal{A}'_2|}{|\mathcal{A}|}, \qquad \epsilon_2^{\mathcal{B}} = \frac{|\mathcal{B}''_2|}{|\mathcal{B}|},$$

and observe that $\epsilon_2^{\mathcal{A}} > \frac{\epsilon(v)}{W}$ by our selection of $z \in \mathcal{L}_1$, and that $\epsilon_2^{\mathcal{B}} > 0$ since $b^o \notin \mathcal{B}'_2$.

Step 7. If $\epsilon_2^{\mathcal{B}} > \epsilon(v)$, then we use decomposition rule (i') which gives

$$\begin{aligned} C(v) &\leq 1 + C(|\mathcal{A}||\mathcal{B}'_2|) + |\mathcal{Y}|C(|\mathcal{A}''_2||\mathcal{B}|) \\ &\leq 1 + C((1-\epsilon_2^{\mathcal{B}})v) + W \cdot C((1-\epsilon_2^{\mathcal{A}})v) \\ &\leq 1 + C((1-\epsilon(v))v) + W \cdot C((1-\frac{\epsilon(v)}{W})v), \end{aligned} \tag{14}$$

since $\epsilon_2^{\mathcal{A}} > \frac{\epsilon(v)}{W}$ and $\epsilon_2^{\mathcal{B}} > \epsilon(v)$.

Step 8. Finally if $\epsilon_2^{\mathcal{B}} \leq \epsilon(v)$, use rule (ii') and get thus the recurrence

$$\begin{aligned} C(v) &\leq 1 + C(|\mathcal{A}||\mathcal{B}'_2|) + \sum_{j=1}^{h} \left(\prod_{x \in (y^j)^\top} |\widetilde{\mathcal{A}}(x)| \right) C(|\mathcal{A}''_2||\mathcal{B}''_2(y^j)|) \\ &\leq C((1-\epsilon)v) + \frac{v^d}{\delta^d - 1} C(\epsilon v), \quad \text{for some } \epsilon \in (0, \epsilon(v)]. \end{aligned} \tag{15}$$

One can show by induction on $v = v(\mathcal{A}, \mathcal{B}) \geq 4$ that recurrences (12)–(15) imply that

$$C(v) \leq v^{\chi(v)},$$

see [10]. Note that, for $\delta \geq 2$, $d \geq 1$ and $W \geq 1$, we have $(\chi/\rho(W))^\chi < 3\,(v/\delta)^d$, and thus,

$$\chi(v) < \frac{d \log(v/\delta) + \log 3}{\log(\chi/\rho(W))} \sim \frac{d\rho(W)\log v}{\log d + \log\log v}.$$

As $v(\mathcal{A}, \mathcal{B}) < m^2$, we get $\chi(v) = d\rho(W)o(\log m)$. This establishes the bound stated in Theorem 1.

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, Fast discovery of association rules, in *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, AAAI Press, Menlo Park, California, 1996.

2. J. C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation* 123 (1995) 50–63.
3. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan and K. Makino, On generating all minimal integer solutions to a monotone system of linear inequalities, in *Automata, Languages and Programming, 28-th International Colloquium (ICALP 2001)*, *Lecture Notes in Computer Science* 2076, pp. 92–103, 2001. To appear in *SIAM Journal on Computing*.
4. E. Boros, K. Elbassioni, V. Gurvich and L. Khachiyan, An inequality for polymatroid functions and its applications, to appear in *Discrete Applied Mathematics*, 2002. DIMACS Technical Report 2001-14, Rutgers University.
5. E. Boros, V. Gurvich, L. Khachiyan and K. Makino, Dual bounded generation: partial and multiple transversals of a hypergraph, *SIAM Journal on Computing* **30** (6) (2001) 2036–2050.
6. E. Boros, V. Gurvich, L. Khachiyan and K. Makino, On the complexity of generating maximal frequent and minimal infrequent sets, in *Proc. 19th International Symposium on Theoretical Aspects of Computer Science, (STACS)*, Mrach 2002, *Lecture Notes in Computer Science* 2285, pp. 133–141.
7. E. Boros, K. Elbassioni, V. Gurvich and L. Khachiyan, Extending the Balas-Yu bounds on the number of maximal independent sets in graphs to hypergraphs and lattices, DIMACS Technical Report 2002-27, Rutgers University.
8. E. Boros, P.L. Hammer and J.N. Hooker, Predicting cause-effect relationships from incomplete discrete observations, *SIAM J. Discrete Math.* **7** (1994), 481–491.
9. K. Elbassioni, On dualization in products of forests, in *Proc. 19th International Symposium on Theoretical Aspects of Computer Science (STACS)*, March 2002, *Lecture Notes in Computer Science* 2285, pp. 142–153.
10. K. Elbassioni, An algorithm for dualization in products of lattices, DIMACS Technical Report 2002-26, Rutgers University.
11. J. Edmonds, J. Gryz, D. Liang and R. J. Miller, Mining for empty rectangles in large data sets, in *Proc. 8th Int. Conf. on Database Theory (ICDT)*, Jan. 2001, *Lecture Notes in Computer Science* 1973, pp. 174–188.
12. M. L. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, *Journal of Algorithms*, 21 (1996) 618–628.
13. V. Gurvich and L. Khachiyan, On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions, *Discrete Applied Mathematics*, 96-97 (1999) 363–373.
14. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen, Data mining, hypergraph transversals and machine learning, in *Proc. 16th ACM-SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, (1997) pp. 12–15.
15. E. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Generating all maximal independent sets: NP-hardness and polynomial-time algorithms, *SIAM Journal on Computing*, 9 (1980) 558–565.
16. K. Makino and T. Ibaraki, Interior and exterior functions of Boolean functions, *Discrete Applied Mathematics*, 69 (1996) 209–231.
17. R. C. Read and R. E. Tarjan, Bounds on backtrack algorithms for listing cycles, paths, and spanning trees, *Networks* 5 (1975) 237–252.
18. R. Srikant and R. Agrawal, Mining generalized association rules, in *Proc. 21st Int. Conf. on Very Large Data Bases*, pp. 407–419, 1995.
19. R. Srikant and R. Agrawal, Mining quantitative association rules in large relational tables, in *Proc. ACM-SIGMOD 1996 Conf. on Management of Data*, pp. 1–12, 1996.