Algorithms for Dualization over Products of Partially Ordered Sets*

Khaled M. Elbassioni †

Abstract

Let $\mathcal{P} = \mathcal{P}_1 \times \cdots \times \mathcal{P}_n$ be the product of n partially ordered sets. Given a subset $\mathcal{A} \subseteq \mathcal{P}$, we consider problem $DUAL(\mathcal{P}, \mathcal{A}, \mathcal{B})$ of extending a given partial list $\mathcal{B}$ of maximal independent elements of $\mathcal{A}$ in $\mathcal{P}$. We give quasi-polynomial time algorithms for solving problem $DUAL(\mathcal{P}, \mathcal{A}, \mathcal{B})$ when each poset $\mathcal{P}_i$ belongs to one of the following classes: (i) semi-lattices of bounded width, (ii) forests, that is, posets with acyclic underlying graphs, with either bounded in-degrees or out-degrees, or (iii) lattices defined by a set of real closed intervals.

1 Introduction

Let $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$ be the product of n partially ordered sets (posets). Denote by $\preceq$ the precedence relation in $\mathcal{P}$ and also in $\mathcal{P}_1, \ldots, \mathcal{P}_n$, i.e., if $p = (p_1, \ldots, p_n) \in \mathcal{P}$ and $q = (q_1, \ldots, q_n) \in \mathcal{P}$, then $p \preceq q$ in $\mathcal{P}$ if and only if $p_1 \preceq q_1$ in $\mathcal{P}_1$, $p_2 \preceq q_2$ in $\mathcal{P}_2, \ldots$, and $p_n \preceq q_n$ in $\mathcal{P}_n$. For a set $\mathcal{A} \subseteq \mathcal{P}$, let $\mathcal{A}^+ = \{x \in \mathcal{P} \mid x \succeq a,$ for some $a \in \mathcal{A}\}$ and $\mathcal{A}^- = \{x \in \mathcal{P} \mid x \preceq a,$ for some $a \in \mathcal{A}\}$ denote respectively the ideal and filter generated by $\mathcal{A}$. For simplicity, we shall use p^+ and p^- to denote $\{p\}^+$ and $\{p\}^-$ for any $p \in \mathcal{P}$. Any element in $\mathcal{P} \setminus \mathcal{A}^+$ is called *independent of* $\mathcal{A}$. Let $\mathcal{I}(\mathcal{A})$ be the set of all maximal independent elements for $\mathcal{A}$, also called the *dual* of $\mathcal{A}$ in $\mathcal{P}$:

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

Then we have the following decomposition of $\mathcal{P}$

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

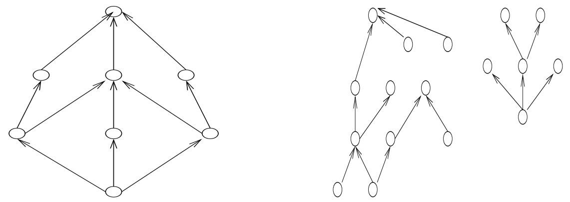
Call $\mathcal{A}$ an *antichain* if no two elements are comparable in $\mathcal{P}$. In this paper, we are concerned with the following *dualization* problem:

DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$**:** *Given an antichain $\mathcal{A} \subseteq \mathcal{P}$ in a poset $\mathcal{P}$ and a collection 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 state that the given collection is complete: $\mathcal{B} = \mathcal{I}(\mathcal{A})$.*

If $\mathcal{P}$ is the Boolean cube, i.e., $\mathcal{P}_i = \{0, 1\}$ for all $i = 1, \ldots, n$, the above dualization problem reduces to the well-known *hypergraph transversal* problem, which calls for enumerating all minimal subsets $X \subseteq V$ that intersect all edges of a given hypergraph $\mathcal{H} \subseteq 2^V$. The complexity of the dualization problem is still an important open question. For the Boolean

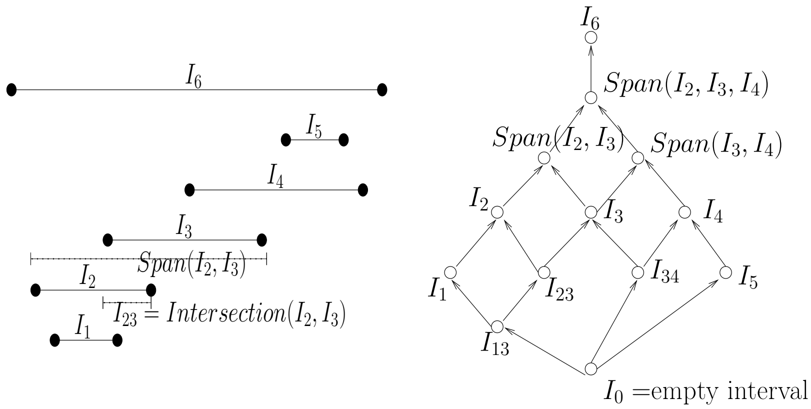
*This research was supported in part by the National Science Foundation Grant IIS-0118635. A preliminary version of this paper, containing some of the results, appeared in [Elb02b, Elb02a, Elb06]

†Max-Planck-Institut für Informatik, Saarbrücken, Germany; (elbassio@mpi-sb.mpg.de).



a: A lattice with $W = 3$. b: A forest with $d = 3$.

Figure 1: Lattices and forests.



a: A set of intervals $\mathbb{I}_1$. b: The corresponding lattice of intervals $\mathcal{L}_1$.

Figure 2: The lattice of intervals.

case, the best known algorithm runs in quasi-polynomial time $poly(n) + m^{o(\log m)}$, where $m = |\mathcal{A}| + |\mathcal{B}|$, see [FK96], providing strong evidence that the problem is unlikely to be NP-hard. More generally, when each $\mathcal{P}_i$ is a chain, that is, a totally ordered set, the problem was considered in [BEG$^+$02], where it was shown that the above algorithm can be extended to work in quasi-polynomial time, regardless of the chains sizes. It is natural to investigate whether these results can be extended further to wider classes of partially ordered sets. In this paper, we achieve this for the cases when each $\mathcal{P}_i$ is

(i) a join (or meet) semi-lattice with bounded width (see Figure 1–a),

(ii) a forest, that is a poset in which the underlying undirected graph of the precedence graph is acyclic (see Figure 1–b), and in which either the in-degree or the out-degree of each element is bounded, and

(iii) the lattice of intervals defined by a set of intervals on the real line $\mathbb{R}$ (see Figure 2): Let $\mathbb{I}_i$ be a set of intervals in $\mathbb{R}$, and let $\mathcal{L}_i$ be the *lattice of intervals* whose elements are all possible intersections and spans defined by the intervals in $\mathbb{I}_i$, and ordered by containment. The meet of any two intervals in $\mathcal{L}_i$ is their *intersection*, and the join is their *span*, i.e. the minimum interval containing both of them.

We remark that for case (i), all posets $\mathcal{P}_i$ must be of the same type: either all posets are join semi-lattices, or all of them are meet semi-lattices. Without loss of generality we will only consider join semi-lattices.

1.1 Main results

Here is a more formal description of the results in this paper. For $x \in \mathcal{P}_i$, denote by $x^\perp$ the set of immediate predecessors of x, i.e.

$$x^\perp = \{y \in \mathcal{P}_i \mid y \prec x, \ (\nexists z \in \mathcal{P}_i : y \prec z \prec x)\},$$

and let in-deg$(\mathcal{P}_i) = \max\{|x^\perp| : \ x \in \mathcal{P}_i\}$. Similarly, denote by $x^\top$ the set of immediate successors of x, and let out-deg$(\mathcal{P}_i) = \max\{|x^\top| : \ x \in \mathcal{P}_i\}$. Throughout the paper, we shall use the notation $m \stackrel{\text{def}}{=} |\mathcal{A}| + |\mathcal{B}|$, $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$, $d \stackrel{\text{def}}{=} \max_{i \in [n]} \min\{\text{in-deg}(\mathcal{P}_i), \text{out-deg}(\mathcal{P}_i)\}$, and $\mu = \mu(\mathcal{P}) \stackrel{\text{def}}{=} \max\{|\mathcal{P}_i| \ : \ i \in [n]\}$. Finally, denote by $W(\mathcal{P}_i)$ the *width* of $\mathcal{P}_i$, i.e. the maximum size of an antichain in $\mathcal{P}_i$, let $W = W(\mathcal{P}) \stackrel{\text{def}}{=} \max_{i \in [n]}\{W(\mathcal{P}_i)\}$ be the maximum width of the n posets, and write $\gamma(W) \stackrel{\text{def}}{=} 2W^2 \ln(W+1)$. A join (respectively, meet) semi-lattice is a poset $\mathcal{P}$ in which every two elements $x, y \in \mathcal{P}$ have a unique minimum upper-bound, called the *join* $x \vee y$ (respectively, a unique maximum lower-bound, called the *meet* $x \wedge y$).

Theorem 1 *Problem DUAL$(\mathcal{L}, \mathcal{A}, \mathcal{B})$ can be solved in poly$(n, \mu(\mathcal{L}))+m^{\gamma(W(\mathcal{L}))\cdot o(\log m)}$ time, if $\mathcal{L}$ is a product of join semi-lattices.*

Theorem 2 *Problem DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$ can be solved in poly$(n, \mu(\mathcal{P})) + m^{d\cdot o(\log m)}$ time, if $\mathcal{P}$ is a product of forests.*

Theorem 3 *Problem DUAL$(\mathcal{L}, \mathcal{A}, \mathcal{B})$ can be solved in poly$(n, \mu(\mathcal{L}))+\min\{k^{O(\log^2 k)}, m^{\gamma(W(\mathcal{L}))\cdot o(\log m)}\}$ time, if $\mathcal{L}$ is a product of lattices of intervals, where $k = |\mathcal{A}| + |\mathcal{B}| + \sum_{i=1}^n |\mathcal{L}_i|$.*

Note that Theorem 3 strengthens Theorem 1 for the special case of the product of lattices of intervals. Indeed, for the lattice of intervals $\mathcal{L}_i$, defined by a set of intervals $\mathbb{I}_i$, we have $W(\mathcal{L}_i) = O(|\mathbb{I}_i|)$ and $|\mathcal{L}_i| = O(|\mathbb{I}_i|^2)$. Thus, for this special case, the result of Theorem 1 gives an exponential algorithm in the total number of intervals $\sum_{i=1}^n |\mathbb{I}_i|$, while Theorem 3 gives a quasi-polynomial bound.

In the next section, we consider some practical applications that motivate our consideration of problem DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$. In Section 3, we describe the general approach we use for solving the dualization problem. We prove Theorems 1, 2 and 3 in Sections 4, 5 and 6 respectively.

2 Some applications

Let $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$ be a partially ordered set. Consider a monotone property $\pi : \mathcal{P} \mapsto \{0, 1\}$ defined over the elements of $\mathcal{P}$: if $x \in \mathcal{P}$ satisfies π, i.e. $\pi(x) = 1$, then any $y \succeq x$ satisfies π. We assume that π is described by a polynomial satisfiability oracle $\mathcal{O}_\pi$, i.e. an algorithm that can decide whether a given vector $x \in \mathcal{P}$ satisfies π, in time polynomial in n and the size $|\pi|$ of the input description of π. Denote respectively by $\mathcal{F}_\pi$ and $\mathcal{G}_\pi$ the families of minimal elements satisfying property π, and maximal elements not satisfying property π. Then it is clear that $\mathcal{G}_\pi = \mathcal{I}(\mathcal{F}_\pi)$ for any monotone property π. Given a monotone property π, we consider the problem of jointly generating the families $\mathcal{F}_\pi$ and $\mathcal{G}_\pi$:

GEN$(\mathcal{P}, \mathcal{F}_\pi, \mathcal{G}_\pi, \mathcal{X}, \mathcal{Y})$**:** *Given a monotone property π, represented by a satisfiability oracle $\mathcal{O}_\pi$, and two explicitly listed vector families $\mathcal{X} \subseteq \mathcal{F}_\pi \subseteq \mathcal{P}$ and $\mathcal{Y} \subseteq \mathcal{G}_\pi \subseteq \mathcal{P}$, either find a new element in $(\mathcal{F}_\pi \setminus \mathcal{X}) \cup (\mathcal{G}_\pi \setminus \mathcal{Y})$, or state that these families are complete: $(\mathcal{X}, \mathcal{Y}) = (\mathcal{F}_\pi, \mathcal{G}_\pi)$.*

For a given monotone property π, described by a satisfiability oracle $\mathcal{O}_\pi$, we can generate both $\mathcal{F}_\pi$ and $\mathcal{G}_\pi$ simultaneously by starting with $\mathcal{X} = \mathcal{Y} = \emptyset$ and solving problem GEN$(\mathcal{P}, \mathcal{F}_\pi, \mathcal{G}_\pi, \mathcal{X}, \mathcal{Y})$ for a total of $|\mathcal{F}_\pi| + |\mathcal{G}_\pi| + 1$ times, incrementing in each iteration either $\mathcal{X}$ or $\mathcal{Y}$ by the newly found vector $x \in (\mathcal{F}_\pi \setminus \mathcal{X}) \cup (\mathcal{G}_\pi \setminus \mathcal{Y})$, according to the answer of the oracle $\mathcal{O}_\pi$, until we have $(\mathcal{X}, \mathcal{Y}) = (\mathcal{F}_\pi, \mathcal{G}_\pi)$.

The following result, relating the time complexity of joint generation to that of dualization, is a straightforward generalization of a similar result known for the binary case [BI95, GK99].

Proposition 1 *Problem GEN*$(\mathcal{P}, \mathcal{F}_\pi, \mathcal{I}(\mathcal{F}_\pi), \mathcal{X}, \mathcal{Y})$ *can be solved in time* $\sum_{i=1}^n |\mathcal{P}_i|(poly(|\mathcal{A}|,\ |\mathcal{B}|) + T_\pi) + T_{dual}$ *for any monotone property* π *defined by a satisfiability oracle* $\mathcal{O}_\pi$*, where* T_π *is the worst-case running time of the oracle on any* $x \in \mathcal{P}$*, and* T_{dual} *denotes the time required to solve problem DUAL*$(\mathcal{P}, \mathcal{A}, \mathcal{B})$.

When one the two families, say $\mathcal{I}(\mathcal{F}_\pi)$, is bounded polynomially (or quasi-polynomially) in size by the other:

$$\mathcal{I}(\mathcal{F}_\pi) \leq \text{poly}(|\pi|, |\mathcal{F}_\pi), \tag{2}$$

then, it follows from the above proposition that all the elements of the family $\mathcal{F}_\pi$ can be generated in *total quasi-polynomial-time* quasi-poly$(|\pi|, |\mathcal{F}_\pi|)$.

Problem GEN$(\mathcal{P}, \mathcal{F}_\pi, \mathcal{I}(\mathcal{F}_\pi), \mathcal{X}, \mathcal{Y})$ arise in many practical applications and in a variety of fields, including artificial intelligence [EG95], game theory [Gur75], reliability theory [BEGK04, Col87], database theory [BGKM03, EG95, GMKT97], integer programming [BEG+02, KBE+07, LLK80], learning theory [AB92], and data mining [AIS93, KBE+07, BGKM03]. In the next subsections we consider three such applications.

2.1 Monotone systems of linear inequalities

Let $A \in \mathbb{R}^{r \times n}$ be a given non-negative real matrix, $b \in \mathbb{R}^r$ be a given r-vector, $c \in \mathbb{R}^n_+$ be a given non-negative n-vector, and consider the system of linear inequalities:

$$Ax \geq b, \quad x \in \mathcal{C} = \{x \in \mathbb{Z}^n \mid 0 \leq x \leq c\}. \tag{3}$$

For $x \in \mathcal{C}$, let $\pi(x)$ be the property that x satisfies (3). Then the families $\mathcal{F}_\pi$ and $\mathcal{G}_\pi$ correspond respectively to the minimal feasible and maximal infeasible vectors for (3). Proposition 1 implies that problem GEN$(\mathcal{C}, \mathcal{F}_\pi, \mathcal{I}(\mathcal{F}_\pi), \mathcal{X}, \mathcal{Y})$ is polynomially equivalent to dualization over the chain product $\mathcal{C}$. Furthermore, it was shown in [BEG+02] that an inequality of the form (2) holds, namely, $|\mathcal{I}(\mathcal{F}_\pi)| \leq rn|\mathcal{F}_\pi|$. Thus, all minimal feasible solutions for (3) can be generated in quasi-polynomial time (see [BEG+02] for more details).

2.2 Maximal frequent and minimal infrequent elements in products of partially ordered sets

Let $\mathcal{P} \stackrel{\text{def}}{=} \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$ be the product of n explicitly given posets. Consider a database $\mathcal{D} \subseteq \mathcal{P}$ of transactions, each of which is an n-dimensional vector of attribute values over $\mathcal{P}$. For an element $p \in \mathcal{P}$, let us denote by

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

the set of transactions in $\mathcal{D}$ that *support* $p \in \mathcal{P}$. Note that, by this definition, the function $|S(\cdot)| : \mathcal{P} \mapsto \{0, 1, \ldots, |\mathcal{D}|\}$ is an anti-monotone function, i.e., $|S(p)| \leq |S(q)|$, whenever

$p \succeq q$. Given $\mathcal{D} \subseteq \mathcal{P}$ and an integer threshold t, let us say that an element $p \in \mathcal{P}$ is t-frequent if it is supported by at least t transactions in the database, i.e., if $|S_{\mathcal{D}}(p)| \geq t$. Conversely, $p \in \mathcal{P}$ is said to be t-infrequent if $|S_{\mathcal{D}}(p)| < t$. For each $x \in \mathcal{P}$, let $\pi(x)$ be the property that x is t-infrequent. Then π is a monotone property and the families $\mathcal{F}_\pi$ and $\mathcal{G}_\pi$ correspond respectively to minimal t-infrequent and maximal t-frequent elements for $\mathcal{D}$.

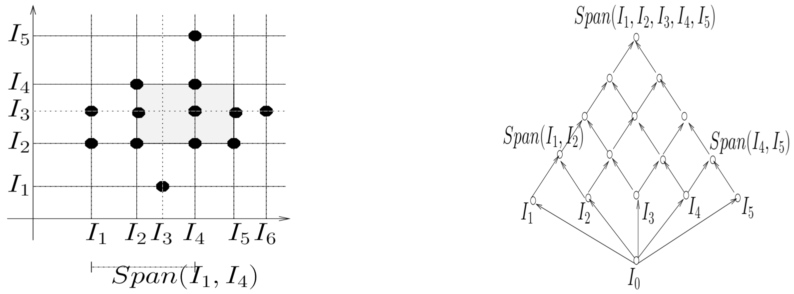
The joint generation of maximal frequent and minimal infrequent elements of a database can be used for finding the so-called *association rules* in data mining applications [AIS93]. If the database $\mathcal{D}$ contains categorical (e.g., zip code, make of car), or quantitative (e.g., age, income) attributes, and the corresponding posets $\mathcal{P}_i$ are total orders, then the above generation problems can be used to mine the so called *quantitative* association rules [SA96]. More generally, each attribute a_i in the database can assume values belonging to some partially ordered set $\mathcal{P}_i$. For example, [SA95] describes applications where items in the database belong to sets of *taxonomies* (or *is-a hierarchies*), and proposes several algorithms for mining association rules among these hierarchical data (see also [HCC93, HF95]). Proposition 1 and Theorems 1 and 2 imply that, for databases $\mathcal{D} \subseteq \mathcal{P}$ where the underlying precedence graph of each poset $\mathcal{P}_i$ is a rooted tree (is-a hierarchy), or where each poset $\mathcal{P}_i$ a join semi-lattice of bounded width, and for any integer threshold t, all maximal frequent elements and all minimal infrequent elements can be jointly generated in quasi-polynomial time (the binary case was considered in [BGKM03]).

2.3 Sparse boxes for multi-dimensional data

Let $\mathcal{S}$ be a set of points in $\mathbb{R}^n$, and $k \leq |\mathcal{S}|$ be a given integer. A *maximal k-box* is a closed n-dimensional rectangle which contains at most k points of $\mathcal{S}$ in its interior, and which is maximal with respect to this property (i.e., cannot be extended in any direction without strictly enclosing more points of $\mathcal{S}$). Suppose we are interested in generating the family $\mathcal{F}_{\mathcal{S},k}$ of maximal k-boxes, defined by the set of points $\mathcal{S}$. Then, without any loss of generality, we may consider the generation of maximal k-boxes contained in a fixed bounded box D containing all points of $\mathcal{S}$ in its interior. Let us further note that the ith coordinate of each vertex of such a box is the same as p_i for some $p \in \mathcal{S}$, or the ith coordinate of a vertex of D, hence all these coordinates belong to a finite set of cardinality at most $|\mathcal{S}| + 2$. In other words, we can view $\mathcal{F}_{\mathcal{S},k}$ as a set of boxes with vertices belonging to such a finite grid.

For $i = 1, \ldots, n$, consider the set of projection points $\mathcal{S}_i \stackrel{\text{def}}{=} \{p_i \in \mathbb{R} \mid p \in \mathcal{S}\}$, and let $\mathcal{L}_i$ be the *lattice of intervals* whose elements are the different intervals defined by the projection points $\mathcal{S}_i$, and ordered by containment. The minimum element l_i of $\mathcal{L}_i$ corresponds to the empty interval I_0. A 2-dimensional example is shown in Figure 3. Let $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$, then each element of $\mathcal{L}$ represents a box, containing some points of $\mathcal{S}$. For $x \in \mathcal{L}$, let $\pi(x)$ be the property that the box defined by x contains at most t points of S in its interior. Then the sets $\mathcal{F}_\pi$ and $\mathcal{G}_\pi$ can be identified respectively with the set of maximal k-boxes, and the set of minimal boxes of $x \in \mathcal{L}$ which contain at least $k+1$ points of $\mathcal{S}$ in their interior. Furthermore, it can be shown [KBE$^+$07] that $|\mathcal{I}(\mathcal{F}_{\mathcal{S},k})| \leq |\mathcal{S}||\mathcal{F}_{\mathcal{S},k}|$. Thus, Proposition 1 and Theorem 3 imply that the family $\mathcal{F}_{\mathcal{S},k}$ can be generated in quasi-polynomial time (see [KBE$^+$07] for more details).

The problem of generating all elements of $\mathcal{F}_{\mathcal{S},0}$ has been studied in the machine learning and computational geometry literatures (see [CDL86, EGLM03, Orl90]), and is motivated by the discovery of missing associations or "holes" in data mining applications (see [AMS$^+$96, LKH97, BLQ98]).



a: A 2-dim. pointset and a maximal 1-box. b: The lattice of intervals $\mathcal{L}_1$.

Figure 3: Maximal sparse boxes: the shaded box has at most $t = 1$ point in its interior.

3 General approach

3.1 Preliminaries

Given two subsets $\mathcal{A} \subseteq \mathcal{P}$, and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, we say that $\mathcal{B}$ is *dual to* $\mathcal{A}$ if $\mathcal{B} = \mathcal{I}(\mathcal{A})$. By (1), this condition is equivalent to $\mathcal{A}^+ \cup \mathcal{B}^- = \mathcal{P}$, and

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

Thus problem DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$ can be equivalently stated as follows:

DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$**:** *Given antichains $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$ satisfying (4), check if there an $x \in \mathcal{P} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$.*

Having found a *solution* to DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$, i.e., an element $x \in \mathcal{P} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$, it can be extended to a maximal element x^* with the same property in $O(n\mu|\mathcal{A}|)$ time. This can be done by initializing $c(a) = |\{i \in [n] : a_i \not\preceq x_i\}|$ for all $a \in \mathcal{A}$, and repeating, for $i = 1, \ldots, n$, the following two steps: (i) $x_i^* \leftarrow$ a maximal $y \in \mathcal{P}_i$ such that $y \not\succeq a_i$ for all $a \in \mathcal{A}$ with $c(a) = 1$ and $a_i \not\preceq x_i$; (ii) $c(a) \leftarrow c(a) - 1$ for each $a \in \mathcal{A}$ such that $a_i \preceq x_i^*$.
It is also worth noting that, if condition (4) does not hold, then the problem becomes NP-hard in general, even if $\mathcal{P} = \{0, 1\}^n$ is just the Boolean cube [EG95].

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

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

Note that, for $a \in \mathcal{A}$ and $\mathcal{Q} = \mathcal{Q}_1 \times \ldots \times \mathcal{Q}_n$, $a^+ \cap \mathcal{Q} \neq \emptyset$ if and only if $a_i^+ \cap \mathcal{Q}_i \neq \emptyset$, for all $i \in [n]$. Thus, the sets $\mathcal{A}(\mathcal{Q})$ and $\mathcal{B}(\mathcal{Q})$ can be found in $O(nm\mu)$ time[1] A simple but important observation, which will be used frequently in the algorithms below, is that

$$\mathcal{Q} \subseteq \mathcal{A}^+ \cup \mathcal{B}^- \Longleftrightarrow \mathcal{Q} \subseteq \mathcal{A}(\mathcal{Q})^+ \cup \mathcal{B}(\mathcal{Q})^-. \tag{5}$$

To solve problem $DUAL(\mathcal{P}, \mathcal{A}, \mathcal{B})$, we shall use the same general approach used in [FK96] to solve the hypergraph dualization problem, by decomposing it into a number of smaller subproblems which are solved recursively. In each such subproblem, we start with a subposet $\mathcal{Q} = \mathcal{Q}_1 \times \ldots \times \mathcal{Q}_n \subseteq \mathcal{P}$ (initially $\mathcal{Q} = \mathcal{P}$), 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 $\mathcal{Q}$, i.e. whether $\mathcal{Q} \subseteq \mathcal{A}(\mathcal{Q})^+ \cup \mathcal{B}(\mathcal{Q})^-$. Note that since $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$ is assumed, (4) continues to hold for the recursive subproblems.

[1] In fact, by the way $\mathcal{Q}$ is chosen in our algorithms, these sets can be found in $O(nm)$ time

Procedure PD$(\mathcal{Q},\mathcal{A},\mathcal{B})$**:**
Input: A subposet $\mathcal{Q}=\mathcal{Q}_1\times\cdots\times\mathcal{Q}_n\subseteq\mathcal{P}$ and two anti-chains $\mathcal{A},\mathcal{B}\subseteq\mathcal{P}$
Output: **true** if $\mathcal{Q}\subseteq(\mathcal{A}^+\cup\mathcal{B}^-)$ and **false** otherwise

1. $\mathcal{A}\leftarrow\mathcal{A}(\mathcal{Q})$, $\mathcal{B}\leftarrow\mathcal{B}(\mathcal{Q})$
2. $\mathcal{A}\leftarrow$ PROJECT$(\mathcal{Q},\mathcal{A})$, $\mathcal{B}\leftarrow$ PROJECT$(\mathcal{Q},\mathcal{B})$
3. **if** $\min\{\mathcal{A},\mathcal{B}\}\le Const.$ **then**
4. **return** POLY-DUAL$(\mathcal{Q},\mathcal{A},\mathcal{B})$
5. Using the appropriate decomposition rule, select $i\in[n]$, and decompose $\mathcal{Q}_i$ into $\mathcal{Q}_i^1,\ldots,\mathcal{Q}_i^r$,
6. **return** $\bigwedge_{j=1}^r$ PD$(\mathcal{Q}_i^j\times\overline{\mathcal{Q}},\mathcal{A},\mathcal{B})$

Figure 4: The general dualization procedure.

The decomposition of $\mathcal{Q}$ is done by decomposing one factor poset, say $\mathcal{Q}_i$, into a number of (not necessarily disjoint) subposets $\mathcal{Q}_i^1,\ldots,\mathcal{Q}_i^r$, and solving r subproblems on the r different posets $\mathcal{Q}_1\times\cdots\times\mathcal{Q}_{i-1}\times\mathcal{Q}_i^j\times\mathcal{Q}_{i+1}\times\cdots\times\mathcal{Q}_n$, $j=1,\ldots,r$. In most of the cases, a number of decomposition rules may be followed, based on the sizes of certain subsets of $\mathcal{A}$ and $\mathcal{B}$, with the objective of reducing the problem size from one level of the recursion to the next. To estimate this reduction in size (only in the analysis of the running time), we measure the change in the "*volume*" of the problem defined as $v=v(\mathcal{A},\mathcal{B})\stackrel{\text{def}}{=}|\mathcal{A}||\mathcal{B}|$ (actually, in Section 6, we use $v(\mathcal{Q},\mathcal{A},\mathcal{B})=|\mathcal{A}||\mathcal{B}|\sum_{i=1}^n|\mathcal{Q}_i|$). For brevity, we shall denote by $\overline{\mathcal{Q}}^i$ the product $\mathcal{Q}_1\times\cdots\times\mathcal{Q}_{i-1}\times\mathcal{Q}_{i+1}\times\cdots\times\mathcal{Q}_n$,, and accordingly by $\overline{q}^i$ the vector $(q_1,\ldots,q_{i-1},q_{i+1},\ldots,q_n)$, for an element $q=(q_1,q_2,\ldots,q_n)\in\mathcal{Q}$. When the index i is understood from the context, we will use $\overline{\mathcal{Q}}$ and $\bar{q}$, for simplicity.

A general high-level dualization procedure is shown in Figure 4. In the procedure, we use 2 subroutines: PROJECT and POLY-DUAL. The second of these routines acts as the base case for recursion, while the first one is used to ensure that, at that base level, the subsets $\mathcal{A},\mathcal{B},\mathcal{Q}\subseteq\mathcal{P}$ satisfy $\mathcal{A},\mathcal{B}\subseteq\mathcal{Q}$. The reason that we need that last condition, and the details of these two routines will be given in subsections 3.4 and 3.5, respectively. In the next section, we derive some general decomposition rules that can be used in Step 5 of the procedure. The selection of which decomposition rule to use in the algorithm depends on the *frequencies* of the element, at which the decomposition is performed, with respect to $\mathcal{A}$ and $\mathcal{B}$, but does not otherwise assume anything about these frequencies. Assuming duality of $\mathcal{A}$ and $\mathcal{B}$, one can show that there exists a "high-frequency" element in one of the factor posets. Using this element for decomposition, at each recursion level, usually yields much simpler algorithms, but with worse running times with respect to m, although possibly better in terms of the other parameters (e.g. width). In fact, this is the only method we know for getting quasi-polynomial bounds in the width, in the case of products lattices of intervals (see Section 6). In Section 3.3, we give the arguments for the existence of such high frequency elements.

We assume that procedure PD returns either true or false depending on whether $\mathcal{A}$ and $\mathcal{B}$ are dual in $\mathcal{Q}$ or not. Returning a vector $x\in\mathcal{Q}\setminus(\mathcal{A}^+\cup\mathcal{B}^-)$ in the latter case is straightforward, as it can be obtained from any subproblem that failed the test for duality.

In the rest of the paper, we shall denote by $C(v(\mathcal{Q},\mathcal{A},\mathcal{B}))$ the total number of subproblems created by procedure PD$(\mathcal{Q},\mathcal{A},\mathcal{B})$ on a problem of size $v(\mathcal{Q},\mathcal{A},\mathcal{B})$. We assume that

$C(v) \leq R(v)$ where $R(v)$ is a *super-additive*[2] function of v (i.e., $R(v) + R(v') \leq R(v + v')$ for all $v, v' \geq 0$), monotone (i.e. $f(v) \geq f(v')$ for all $v \geq v'$), and $R(0) = 1$. Thus we may assume also without loss of generality that $C(v)$ is monotone and superadditive.

3.2 Decomposition

3.2.1 Independent decomposition

Let us call two subposets $\mathcal{Q}, \mathcal{R} \subseteq \mathcal{P}$ *independent* if $q \not\preceq r$ and $q \not\succeq r$ for all $q \in \mathcal{Q}, r \in \mathcal{R}$. The following decomposition rule can be used to reduce the problem on products of forests into one in which is each forest has exactly on connected component, i.e. a tree.

Proposition 2 *Let $\mathcal{Q} = \mathcal{Q}_1 \times \ldots \times \mathcal{Q}_n$ and $\mathcal{A}, \mathcal{B} \subseteq \mathcal{Q}$. Suppose that poset $\mathcal{Q}_i$, can be partitioned into two independent posets $\mathcal{Q}_i'$ and $\mathcal{Q}_i''$. Let $\mathcal{Q}' = \mathcal{Q}_i' \times \overline{\mathcal{Q}}$, $\mathcal{Q}'' = \mathcal{Q}_i'' \times \overline{\mathcal{Q}}$, $\mathcal{A}' = \mathcal{A}(\mathcal{Q}')$, $\mathcal{B}' = \mathcal{B}(\mathcal{Q}')$, $\mathcal{A}'' = \mathcal{A}(\mathcal{Q}'')$, and $\mathcal{B}'' = \mathcal{B}(\mathcal{Q}'')$. If $C(v(\mathcal{A}', \mathcal{B}')) \leq R(v(\mathcal{A}', \mathcal{B}'))$ and $C(v(\mathcal{A}'', \mathcal{B}'')) \leq R(v(\mathcal{A}'', \mathcal{B}''))$, then $C(v(\mathcal{A}, \mathcal{B})) \leq R(v(\mathcal{A}, \mathcal{B}))$.*

Proof. We observe by (5) and the independence of $\mathcal{Q}', \mathcal{Q}''$ that

$$\mathcal{Q} \subseteq \mathcal{A}^+ \cup \mathcal{B}^- \iff \mathcal{Q}' \subseteq (\mathcal{A}')^+ \cup (\mathcal{B}')^- \text{ and } \mathcal{Q}'' \subseteq (\mathcal{A}'')^+ \cup (\mathcal{B}'')^-.$$

Clearly, if $\mathcal{A}' \cup \mathcal{B}' = \emptyset$ (or $\mathcal{A}'' \cup \mathcal{B}'' = \emptyset$) then any element in $\mathcal{Q}'$ (respectively, in $\mathcal{Q}''$) does not belong to $\mathcal{A}^+ \cup \mathcal{B}^-$. On the other hand, if these unions are not empty, we can proceed by recursively solving the two subproblems DUAL$(\mathcal{Q}', \mathcal{A}', \mathcal{B}')$ and DUAL$(\mathcal{Q}'', \mathcal{A}'', \mathcal{B}'')$. This gives

$$C(v(\mathcal{A}, \mathcal{B})) = 1 + C(v(\mathcal{A}', \mathcal{B}')) + C(v(\mathcal{A}'', \mathcal{B}'')) \leq 1 + R(v(\mathcal{A}', \mathcal{B}')) + R(v(\mathcal{A}'', \mathcal{B}'')).$$

Note that $\{\mathcal{A}', \mathcal{A}''\}$ and $\{\mathcal{B}', \mathcal{B}''\}$ form partitions of $\mathcal{A}$ and $\mathcal{B}$ respectively and therefore, we get by the super-additivity and monotonicity of $R(\cdot)$,

$$\begin{aligned} R(v(\mathcal{A}, \mathcal{B})) &\geq R(v(\mathcal{A}', \mathcal{B}') + v(\mathcal{A}'', \mathcal{B}'') + v(\mathcal{A}', \mathcal{B}'') + v(\mathcal{A}'', \mathcal{B}')) \\ &\geq R(v(\mathcal{A}', \mathcal{B}')) + R(v(\mathcal{A}'', \mathcal{B}'')) + R(v(\mathcal{A}', \mathcal{B}'')) + R(v(\mathcal{A}'', \mathcal{B}')) \\ &\geq R(v(\mathcal{A}', \mathcal{B}')) + R(v(\mathcal{A}'', \mathcal{B}'')) + 1. \end{aligned}$$

implying the proposition. □

3.2.2 General decomposition

For an operator $\circ \in \{\preceq, \not\preceq, \succeq, \not\succeq\}$, a subset $\mathcal{X} \subseteq \mathcal{P}$, an $i \in [n]$, and an element $z \in \mathcal{P}_i$, we use the notation $\mathcal{X}_\circ(z) \stackrel{\text{def}}{=} \{x \in \mathcal{X} : x_i \circ z\}$. In general, the algorithms will decompose a given problem by selecting an $i \in [n]$ and partitioning $\mathcal{Q}_i$ into two subposets $\mathcal{Q}_i'$ and $\mathcal{Q}_i''$, defining accordingly two poset products $\mathcal{Q}'$ and $\mathcal{Q}''$. Specifically, let $a^o \in \mathcal{A}$, $b^o \in \mathcal{B}$ be arbitrary elements of $\mathcal{A}, \mathcal{B}$ (in fact the algorithm in Section 4.1 will select specific elements $a^o \in \mathcal{A}$ and $b^o \in \mathcal{B}$). By (4), there exists an $i \in [n]$, such that $a_i^o \not\preceq b_i^o$. Let $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i \cap (a_i^o)^+$, $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i'$ (we may alternatively set $\mathcal{Q}_i'' \leftarrow \mathcal{P}_i \cap (b_i^o)^-$, and $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i''$, see Section 5.1). Defining $\mathcal{Q}' = \mathcal{Q}_i' \times \overline{\mathcal{Q}}$ and $\mathcal{Q}'' = \mathcal{Q}_i'' \times \overline{\mathcal{Q}}$ to be the two subposets induced by this partitioning, and letting $\mathcal{A}'' \stackrel{\text{def}}{=} \mathcal{A}(\mathcal{Q}'') = \mathcal{A}_{\not\succeq}(a_i^o)$, $\mathcal{A}' \stackrel{\text{def}}{=} \mathcal{A}_{\succeq}(a_i^o)$, $\mathcal{B}' \stackrel{\text{def}}{=} \mathcal{B}(\mathcal{Q}') = \mathcal{B}_{\succeq}(a_i^o)$, $\mathcal{B}'' \stackrel{\text{def}}{=} \mathcal{B}_{\not\succeq}(a_i^o)$, we conclude by (5) that $\mathcal{Q} \subseteq \mathcal{A}^+ \cup \mathcal{B}^-$ *if and only if*

$$\mathcal{Q}' \subseteq \mathcal{A}^+ \cup (\mathcal{B}')^-, \quad \text{and} \tag{6}$$
$$\mathcal{Q}'' \subseteq (\mathcal{A}'')^+ \cup \mathcal{B}^-. \tag{7}$$

[2] which is naturally satisfied by any monotone superlinear function

Thus we have decomposed the original problem into two new subproblems. Note that the volumes of the resulting problems are strictly less than the volume of the original problem. For lattices and forests, it may be necessary to further decompose the subposet $\mathcal{Q}_i''$ in order to maintain a certain nice property (lattice property, connectedness of the precedence graph) which allows for the *projection step* described in the Section 3.5.

Clearly, there may exist precedence relations between the elements of $\mathcal{Q}_i'$ and $\mathcal{Q}_i''$ and, therefore, the two subproblems (6) and (7) may not be independent. Once we get an affirmative answer to one subproblem, we gain some information about the solution of the other. The following lemma shows how to utilize this dependence to further decompose the other subproblem in such a case.

Lemma 1 *Given $z \in \mathcal{Q}_i$, let $\mathcal{R}_i' = \mathcal{Q}_i \cap z^+$, $\mathcal{R}_i'' \subseteq \mathcal{Q}_i \cap z^- \setminus \{z\}$ be two disjoint subposets of $\mathcal{Q}_i$. Define*

$$\mathcal{A}^2 = \{a \in \mathcal{A} \mid a_i^+ \cap \mathcal{R}_i'' \neq \emptyset\}, \qquad \mathcal{A}^1 = \{a \in \mathcal{A} \setminus \mathcal{A}^1 | \ a_i^+ \cap \mathcal{R}_i' \neq \emptyset\},$$
$$\mathcal{B}^1 = \{b \in \mathcal{B} \mid b_i^- \cap \mathcal{R}_i' \neq \emptyset\}, \qquad \mathcal{B}^2 = \{b \in \mathcal{B} \setminus \mathcal{B}^1 | \ b_i^- \cap \mathcal{R}_i'' \neq \emptyset\}.$$

Suppose further that $\mathcal{R}_i' \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}^2 \cup \mathcal{A}^1)^+ \cup (\mathcal{B}^1)^-$, then

$$\mathcal{R}_i'' \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}^2)^+ \cup (\mathcal{B}^1 \cup \mathcal{B}^2)^- \iff \forall a \in \mathcal{A}_{\preceq}(z) : \mathcal{R}_i'' \times (\overline{\mathcal{Q}} \cap \overline{a}^+) \subseteq (\mathcal{A}^2)^+ \cup (\mathcal{B}^2)^-.$$

Proof. Suppose first that $\mathcal{R}_i'' \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}^2)^+ \cup (\mathcal{B}^1 \cup \mathcal{B}^2)^-$. Let $(q_i, \overline{q}^i) \in \mathcal{R}_i'' \times (\overline{\mathcal{Q}} \cap \overline{a}^+)$ for some $a \in \mathcal{A}_{\preceq}(z)$, then $(q_i, \overline{q}) \in (\mathcal{A}^2)^+ \cup (\mathcal{B}^1 \cup \mathcal{B}^2)^-$. If $(q_i, \overline{q}) \preceq (b_1, \overline{b}) \in \mathcal{B}^1$, then by the definition of $\mathcal{B}^1$, there is a $y \in \mathcal{R}_i'$ such that $y \preceq b_i$. But then, $a \in \mathcal{A}_{\preceq}(z)$, $\overline{q} \in \overline{\mathcal{Q}} \cap \overline{a}^+$ and $y \in \mathcal{R}_i'$ imply that $(a_i, \overline{a}) \preceq (z, \overline{q}) \preceq (y, \overline{q}) \preceq (b_i, \overline{b})$, which contradicts the assumed condition (4). This shows that $(q_i, \overline{q}) \in (\mathcal{A}^2)^+ \cup (\mathcal{B}^2)^-$.

For the other direction, let $(q_i, \overline{q}) \in (\mathcal{R}_i'' \times \overline{\mathcal{Q}}) \setminus (\mathcal{B}^1)^-$. Since $x \preceq y$ for all $x \in \mathcal{R}_i''$, $y \in \mathcal{R}_i'$, we must have $(y, \overline{q}) \notin (\mathcal{B}^1)^-$ for all $y \in \mathcal{R}_i'$, for otherwise we get the contradiction $(q_1, \overline{q}) \preceq (y, \overline{q}) \preceq (b_1, \overline{b})$ for some $b \in \mathcal{B}^1$. Now we use our assumption that $\mathcal{R}_i' \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}^1 \cup \mathcal{A}^2)^+ \cup (\mathcal{B}^1)^-$ to conclude that $(y, \overline{q}) \in (\mathcal{A}^1 \cup \mathcal{A}^2)^+$ for all $y \in \mathcal{R}_i'$. In particular, we have $(z, \overline{q}) \succeq (a_1, \overline{a})$ for some $(a_1, \overline{a}) \in \mathcal{A}^1 \cup \mathcal{A}^2$. But this implies that $a \in \mathcal{A}_{\preceq}(z)$ and hence that $(q_1, \overline{q}) \in \mathcal{R}_i'' \times (\overline{\mathcal{Q}} \cap \overline{a}^+)$ for some $a \in \mathcal{A}_{\preceq}(z)$. This gives $(q_i, \overline{q}) \in (\mathcal{A}^2)^+ \cup (\mathcal{B}^2)^-$. □

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

Lemma 2 *Let $\mathcal{R}_i'' = \mathcal{Q}_i \cap z^-$, $\mathcal{R}_i' \subseteq \mathcal{Q}_i \cap z^+ \setminus \{z\}$ be two disjoint subposets of $\mathcal{Q}_i$ where $z \in \mathcal{Q}_i$. Let $\mathcal{A}^1, \mathcal{A}^2, \mathcal{B}^1, \mathcal{B}^2$ be defined as in Lemma 1. Suppose that $\mathcal{R}_i'' \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}^2)^+ \cup (\mathcal{B}^1 \cup \mathcal{B}^2)^-$, then*

$$\mathcal{R}_i' \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}^1 \cup \mathcal{A}^2)^+ \cup (\mathcal{B}^1)^- \iff \forall b \in \mathcal{B}_{\succeq}(z) : \mathcal{R}_i' \times (\overline{\mathcal{Q}} \cap \overline{b}^-) \subseteq (\mathcal{A}^1)^+ \cup (\mathcal{B}^1)^-.$$

We now use Lemma 1 inductively to get a further decomposition of poset $\mathcal{Q}_i''$. Suppose that one of the subproblems, say (6) has no solution, i.e., $\mathcal{Q}' \subseteq \mathcal{A}^+ \cup (\mathcal{B}')^-$. Then we can proceed in this case as follows. Let us use $y^1, \ldots, y^k$ to denote the elements of $\mathcal{Q}_i''$ and assume, without loss of generality, that they are *inversely topologically sorted* in this order, that is, $y^j \prec y^r$ implies $j > r$ (see Figure 5–a). Let us decompose (7) further into the k subproblems

$$\{y^j\} \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}_{\preceq}''(y^j))^+ \cup \left[\mathcal{B}_{=}''(y^j) \cup \left(\bigcup_{x \in (y^j)^\top} \mathcal{B}_{\succeq}(x)\right)\right]^-, j = 1, \ldots, k. \tag{8}$$

The following lemma will allow us to eliminate the contribution of the set $\mathcal{B}'$ in subproblems (8) at the expense of possibly introducing at most $|\mathcal{A}|^{W(\mathcal{Q})}$ additional subproblems.

Lemma 3 *Given $y^j \in \mathcal{Q}_i''$, suppose we know that $(x^+ \cap \mathcal{Q}_i) \times \overline{\mathcal{Q}} \subseteq \mathcal{A}^+ \cup (\mathcal{B}^{\succeq}(x))^-$ for all $x \in (y^j)^\top$. Then (8) is equivalent to*

$$\{y^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\bigcap_{x \in (y^j)^\top} \overline{a}(x)^+\right)\right] \subseteq (\mathcal{A}''_{\preceq}(y^j))^+ \cup (\mathcal{B}''_{=}(y^j))^-, \tag{9}$$

for all collections $\{a(x) \in \mathcal{A}_{\preceq}(x) \mid x \in (y^j)^\top\}$, for $j = 1, \ldots, k$. (That is, if $(y^j)^\top = \{x^1, \ldots, x^s\}$, then we consider all collections of the form $\{a(x^1), \ldots, a(x^s)\}$, where $a(x^1) \in \mathcal{A}_{\preceq}(x^1), \ldots,$ $a(x^s) \in \mathcal{A}_{\preceq}(x^s)$.)

Proof. We prove by induction on $|X|$, where $X \subseteq (y^j)^\top$, that

$$\begin{aligned} \{y^j\} \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}''_{\preceq}(y^j))^+ \cup \left[\mathcal{B}''_{=}(y^j) \cup \left(\textstyle\bigcup_{x \in (y^j)^\top} \mathcal{B}_{\succeq}(x)\right)\right]^- \qquad \Longleftrightarrow \\ \{y^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\textstyle\bigcap_{x \in X} \overline{a}(x)^+\right)\right] \subseteq (\mathcal{A}''_{\preceq}(y^j))^+ \cup \left[\mathcal{B}''_{=}(y^j) \cup \left(\textstyle\bigcup_{x \in (y^j)^\top \setminus X} \mathcal{B}_{\succeq}(y)\right)\right]^-, \end{aligned} \tag{10}$$

for all collections $\{a(y) \in \mathcal{A}_{\preceq}(x) \mid x \in X\}$. This trivially holds for $X = \emptyset$ and will prove the lemma for $X = (y^j)^\top$. To show (10), assume that it holds for some $X \subset (y^j)^\top$ and let $u \in (y^j)^\top \setminus Y$. Consider a subproblem of the form

$$\{y^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\bigcap_{x \in X} \overline{a}(x)^+\right)\right] \subseteq (\mathcal{A}''_{\preceq}(y^j))^+ \cup \left[\mathcal{B}''_{=}(y^j) \cup \mathcal{B}_{\succeq}(u) \cup \left(\bigcup_{x \in (y^j)^\top \setminus (X \cup \{u\})} \mathcal{B}_{\succeq}(y)\right)\right]^-,$$

for some collection $\{a(y) \in \mathcal{A}_{\preceq}(x) \mid x \in X\}$. Now we apply Lemma 1 with $z \leftarrow u$, $\mathcal{R}_i' \leftarrow z^+ \cap \mathcal{Q}_i$, $\mathcal{R}_i'' \leftarrow \{y^j\}$, $\mathcal{A}^2 \leftarrow \mathcal{A}_{\preceq}(y^j)$, $\mathcal{B}^1 \leftarrow \mathcal{B}_{\succeq}(u)$, and $\mathcal{B}^2 \leftarrow \mathcal{B}''_{=}(y^j) \cup \left(\bigcup_{x \in (y^j)^\top \setminus (X \cup \{u\})} \mathcal{B}_{\succeq}(y)\right)$, to get the required result. $\square$

Informally, Lemma 3 says that, given $y^j \in \mathcal{Q}_i''$, if the dualization subproblems for all subposets that lie above y^j have been already verified to have no solution, then we can solve subproblem (8) by solving at most $\prod_{x \in (y^j)^\top} |\mathcal{A}_{\preceq(x)}|$ subproblems of the form (9). Observe that it is important to check subproblems (8) in the reverse topological order $j = 1, \ldots, h$ in order to be able to use Lemma 3.

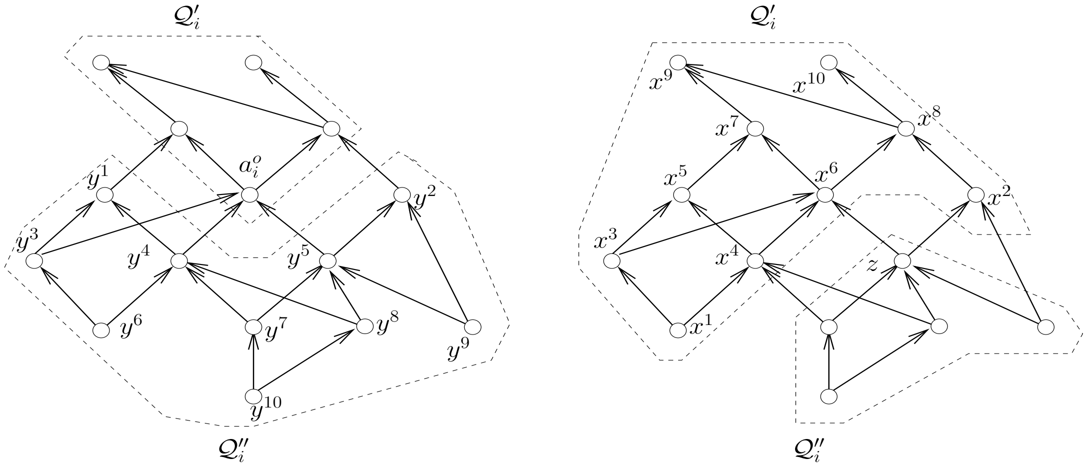
3.2.3 Decomposition rules

Using the decomposition lemmas stated in the previous section, we now derive some general decomposition rules that will be used later by the algorithms.

Rule (R1) Solve subproblems (6) and (7).

Rule (R2) Solve subproblem (6). If it has a solution then we get an element $x \in \mathcal{Q} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$. Otherwise, we solve subproblems (8), for all collections $\{a(x) \in \mathcal{A}_{\preceq}(x) \mid x \in (y^j)^\top\}$, for $j = 1, \ldots, k$, where where $y^1, \ldots, y^h$ denote the elements of $\mathcal{Q}_i''$ in *reverse* topological order (see Figure 5–a).

Suppose finally that we decompose $\mathcal{Q}_i$ by selecting an element $z \in \mathcal{Q}_i$, letting $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \cap z^-$, $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i \setminus z^-$, $\mathcal{A}'' = \mathcal{A}_{\preceq}(z)$, $\mathcal{A}' = \mathcal{A} \setminus \mathcal{A}''$, and $\mathcal{B}' = \mathcal{B}_{\not\preceq}(z)$. By exchanging the roles of $\mathcal{A}$ and $\mathcal{B}$ and replacing $\mathcal{P}$ by its dual poset $\mathcal{P}^*$ in rule $(R2)$ above, we can also arrive at the following symmetric version of this rule (see Figure 5–b):



a: Rules $(R1)$ and $(R2)$. b: Rule $(R2')$.

Figure 5: Decomposing the poset $\mathcal{Q}_i$.

Rule (R2′) Solve subproblem (7), and if it does not have a solution, then solve the subproblems

$$\{x^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\bigcap_{y \in (x^j)^\perp} \overline{b}(y)^- \right)\right] \subseteq (\mathcal{A}'_{=}(x^j))^+ \cup (\mathcal{B}'_{\succeq}(x^j))^-, \tag{11}$$

for all collections $\{b(y) \in \mathcal{B}_{\succeq}(y) \mid y \in (x^j)^\perp\}$, for $j = 1, \ldots, k$, where $x^1, \ldots, x^k$ denote the elements of $\mathcal{Q}'_i$ in *topological* order (that is, $x^j \prec x^r$ implies $j < r$).

In the sections 4, 5 and 6, we show how to use the above rules for decomposing a given dualization problem into smaller subproblems. The algorithms will select between these rules in such a way that the total volume is reduced significantly from one recursion level to the next.

3.3 High frequency-based decomposition

Assume that $\mathcal{A}, \mathcal{B}$ satisfy (4), and let us denote, respectively, by $\mathrm{Min}(\mathcal{Q}_i)$, $\mathrm{Max}(\mathcal{Q}_i)$, the sets of minimal and maximal elements of poset $\mathcal{Q}_i$. Define the *support* of $a \in \mathcal{A}$ (respectively, $b \in \mathcal{B}$) to be the set of all non-minimum coordinates of a (respectively, the set of all non-maximum coordinates of b):

$$\mathrm{Supp}(a) = \{i \in [n] \ : \ \mathrm{Min}(\mathcal{Q}_i) \neq \{a_i\}\}, \qquad \mathrm{Supp}(b) = \{i \in [n] \ : \ \mathrm{Max}(\mathcal{Q}_i) \neq \{b_i\}\}.$$

Let $\alpha = \alpha(\mathcal{Q}) \stackrel{\text{def}}{=} \max_{i \in [n]}\{|\mathrm{Min}(\mathcal{Q}_i) \cup \mathrm{Max}(\mathcal{Q}_i)|\}$. The following lemma generalizes a known fact for dual Boolean functions (cf. [FK96]).

Lemma 4 *If $\mathcal{A}, \mathcal{B}$ are dual in $\mathcal{Q}$, then there exists an element $x \in \mathcal{A} \cup \mathcal{B}$ with a logarithmically small support: $|\mathrm{Supp}(x)| \leq \alpha \ln m$, where $m = |\mathcal{A}| + |\mathcal{B}|$. Such an element can be found in $O(n^2 m \alpha)$ time.*

Proof. Let $z \in \mathcal{Q}$ be the vector obtained by picking each coordinate z_i randomly from $\mathcal{X}_i \stackrel{\text{def}}{=} \mathrm{Min}(\mathcal{Q}_i) \cup \mathrm{Max}(\mathcal{Q}_i)$, $i = 1, \ldots, n$, and consider the random variable $N(z) \stackrel{\text{def}}{=} |\{a \in$

$\mathcal{A} \mid z \succeq a\}| + |\{b \in \mathcal{B} \mid z \preceq b\}|$. Then the expected value of $N(z)$ is given by

$$\begin{aligned}
\mathbb{E}[N(z)] &= \sum_{a\in\mathcal{A}} \Pr\{z \succeq a\} + \sum_{b\in\mathcal{B}} \Pr\{z \preceq b\} \\
&= \sum_{a\in\mathcal{A}} \prod_{i\in\mathrm{Supp}(a)} \frac{|\mathcal{X}_i \cap a_i^+|}{|\mathcal{X}_i|} + \sum_{b\in\mathcal{B}} \prod_{i\in\mathrm{Supp}(b)} \frac{|\mathcal{X}_i \cap b_i^-|}{|\mathcal{X}_i|} \\
&\le \sum_{a\in\mathcal{A}} \prod_{i\in\mathrm{Supp}(a)} \left(1 - \frac{1}{|\mathcal{X}_i|}\right) + \sum_{b\in\mathcal{B}} \prod_{i\in\mathrm{Supp}(b)} \left(1 - \frac{1}{|\mathcal{X}_i|}\right) \\
&\le \sum_{a\in\mathcal{A}} \left(1 - \frac{1}{\alpha}\right)^{|\mathrm{Supp}(a)|} + \sum_{b\in\mathcal{B}} \left(1 - \frac{1}{\alpha}\right)^{|\mathrm{Supp}(b)|}.
\end{aligned} \tag{12}$$

Clearly if $\mathbb{E}[N(z)] < 1$, we can find an $x \in \mathcal{L} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ in $O(n^2 m\alpha)$ using the standard method of conditional expectation [MR95]. Let us therefore assume that $\mathbb{E}[N(z)] \ge 1$, and let $r = \min\{|\mathrm{Supp}(z)| \,:\, z \in \mathcal{A} \cup \mathcal{B}\}$, then (12) implies that

$$1 \le \mathbb{E}[N(z)] \le (|\mathcal{A}| + |\mathcal{B}|) \left(1 - \frac{1}{\alpha}\right)^r \le m e^{-r/\alpha}.$$

The lemma follows. □

Next we show that, for any dual pair $(\mathcal{A}, \mathcal{B})$, a high frequency element exists, with respect to either $\mathcal{A}$ or $\mathcal{B}$.

Lemma 5 *Let $\mathcal{A}, \mathcal{B}$ be a pair of dual subsets of $\mathcal{Q}$ with $|\mathcal{A}||\mathcal{B}| \ge 1$. Then there exist a coordinate $i \in [n]$ and an element $z \in \mathcal{Q}_i$, such that either:*

(i) $|\mathcal{A}_{\succeq}(z)| \ge 1$ *and* $|\mathcal{B}_{\not\succeq}(z)| \ge \frac{|\mathcal{B}|}{\alpha(\mathcal{Q}) \ln m}$, *or*

(ii) $|\mathcal{B}_{\preceq}(z)| \ge 1$ *and* $|\mathcal{A}_{\not\preceq}(z)| \ge \frac{|\mathcal{A}|}{\alpha(\mathcal{Q}) \ln m}$.

There also exist a coordinate $i \in [n]$ and a element $z \in \mathcal{Q}_i$, such that either:

(iii) $|\mathcal{A}_{\not\preceq}(z)| \ge 1$ *and* $|\mathcal{B}_{\preceq}(z)| \ge \frac{|\mathcal{B}|}{\alpha(\mathcal{Q}_i) W(\mathcal{Q}_i) \ln m}$, *or*

(iv) $|\mathcal{B}_{\not\succeq}(z)| \ge 1$ *and* $|\mathcal{A}_{\succeq}(z)| \ge \frac{|\mathcal{A}|}{\alpha(\mathcal{Q}_i) W(\mathcal{Q}_i) \ln m}$.

Proof. By Lemma 4, $\mathcal{A} \cup \mathcal{B}$ contains an element x with $|\mathrm{Supp}(x)| \le \alpha \ln m$. Suppose without loss of generality that $x \in \mathcal{A}$. From condition (4), we know that for every $b \in \mathcal{B}$, there is an $i \in \mathrm{Supp}(b) \cap \mathrm{Supp}(x)$ such that $b_i \not\succeq x_i$. Thus

$$|\mathcal{B}| = |\bigcup_{i\in\mathrm{Supp}(x)} \mathcal{B}_{\not\succeq}(x_i)| \le \sum_{i\in\mathrm{Supp}(x)} |\mathcal{B}_{\not\succeq}(x_i)|,$$

and therefore there is an $i \in [n]$ such that $|\mathcal{B}_{\not\succeq}(x_i)| \ge |\mathcal{B}|/|\mathrm{Supp}(x)| \ge |\mathcal{B}|/(\alpha \ln m)$, which implies (i) for $z = x_i$.

To show (iii), consider the set $\mathcal{Y} = \mathcal{I}(\{x_i\})$ of maximal independent elements in $\mathcal{Q}_i \setminus \{x_i\}^+$, and observe that

$$|\mathcal{B}_{\not\succeq}(x_i)| = |\bigcup_{z\in\mathcal{Y}} \mathcal{B}_{\preceq}(z)| \le \sum_{z\in\mathcal{Y}} |\mathcal{B}_{\preceq}(z)|.$$

Noting that $|\mathcal{Y}| \le W(\mathcal{Q}_i)$, we conclude that (iii) holds. If x actually belongs to $\mathcal{B}$, then by a similar argument we obtain (ii) and (iv). □

3.4 Polynomial dualization when one of the sets is small

When one of the sets $\mathcal{A}$ or $\mathcal{B}$ has constant size, the problem can be solved in polynomial time.

Proposition 3 *Suppose that* $\min\{|\mathcal{A}|, |\mathcal{B}|\} \le k$*,* $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$*, then problem DUAL(*$\mathcal{P}$*,* $\mathcal{A}, \mathcal{B}$*) is solvable in time* $O(n^{k+1}\ mW(\mathcal{P})^{k+1}\mu(\mathcal{P}))$*.*

Proof. Let us assume without loss of generality that $\mathcal{B} = \{b^1, \ldots, b^k\}$ for some constant k. Then problem DUAL($\mathcal{P}$, $\mathcal{A}, \mathcal{B}$) can be reduced to n^k subproblems of the form DUAL($\mathcal{P}', \mathcal{A}, \emptyset$), where $\mathcal{P}' = \mathcal{P}'_1 \times \ldots \times \mathcal{P}'_n$, is obtained from $\mathcal{P}$ by selecting, for each $j \in [k]$, a coordinate $i_j \in [n]$ and setting $\mathcal{P}'_i = \mathcal{P}_i \setminus \bigcup_{j\in[k]}(b^j_{i_j})^-$.

Clearly, $\mathcal{P}' \subseteq \mathcal{A}^+$ if and only if $\mathrm{Min}(\mathcal{P}') \subseteq \mathcal{A}^+$ where $\mathrm{Min}(\mathcal{P}') = \mathrm{Min}(\mathcal{P}'_1) \times \ldots \times \mathrm{Min}(\mathcal{P}'_n)$ and $\mathrm{Min}(\mathcal{P}'_i)$ is the set of minimal elements of $\mathcal{P}'_i$. Now the latter problem is easily seen to be polynomially solvable as follows. Let $\mathrm{Min}(\mathcal{P}'_i) = \{q_i^1, \ldots, q_i^{k_i}\}$, for $i \in [n]$ where $k_i = |\mathcal{P}'_i|$. By construction, only $l \le k$ of the posets $\mathcal{P}'_i$ satisfy $\mathcal{P}'_i \ne \mathcal{P}_i$. Assume without loss of generality that these posets are $\mathcal{P}'_1 \times \ldots \times \mathcal{P}'_l$, then our problem reduces to finding whether $\{q_1^{i_1}\} \times \ldots \times \{q_l^{i_l}\} \times \mathrm{Min}(\mathcal{P}_{l+1}) \times \ldots \times \mathrm{Min}(\mathcal{P}_n) \subseteq \mathcal{A}^+$ for all $(i_1, \ldots, i_l) \in [k_1] \times \ldots \times [k_l]$. Each such problem is equivalent to determining whether $\mathrm{Min}(\mathcal{P}_{l+1}) \times \ldots \times \mathrm{Min}(\mathcal{P}_n) \subseteq (\mathcal{A}^{i_1,\ldots,i_l})^+$, where $\mathcal{A}^{i_1,\ldots,i_l} = \{(a_{l+1}, \ldots, a_n) \mid a \in \mathcal{A},\ \ a_j \preceq q_j^{i_j}$ for $j = 1, \ldots, l$, and $a_j^+ \cap \mathrm{Min}(\mathcal{P}_j) \ne \emptyset$ for $j = l+1, \ldots, n\}$. Note that $\mathcal{A}^{i_1,\ldots,i_l} \subseteq \mathrm{Min}(\mathcal{P}_{l+1}) \times \ldots \times \mathrm{Min}(\mathcal{P}_n)$ since $\mathcal{A} \subseteq \mathcal{P}$ was assumed, and hence, each subproblem of the form $\mathrm{Min}(\mathcal{P}_{l+1}) \times \ldots \times \mathrm{Min}(\mathcal{P}_n) \subseteq (\mathcal{A}^{i_1,\ldots,i_l})^+$ can be solved in $O(W(\mathcal{P})nm)$ as a special case of Proposition 2 (since each of the posets $\mathrm{Min}(\mathcal{P}_{l+1}), \ldots, \mathrm{Min}(\mathcal{P}_n)$ can be decomposed into independent posets of size 1 each). $\square$

On the negative side, if we do not insist on the condition $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$ in Proposition 3, the problem becomes NP-hard even for $\mathcal{B} = \emptyset$:

Proposition 4 *Given a subposet* $\mathcal{Q}$ *of a poset* $\mathcal{P}$ *and a subset* $\mathcal{A} \subseteq \mathcal{P}$*, it is coNP-complete to decide if* $\mathcal{Q} \subseteq \mathcal{A}^+$*.*

Proof. We use a polynomial transformation from the satisfiability problem. Let $C = C_1 \wedge \ldots \wedge C_m$ be a conjunctive normal form in n variables $x_1, \ldots, x_n$, and let us consider the poset $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$, where $\mathcal{P}_i = \mathcal{Q}_i \cup \{a_i^1, \ldots, a_i^m\}$, $\mathcal{Q}_i = \{x_i, \overline{x}_i\}$, and where we associate a vector $a^j = (a_1^j, \ldots, a_n^j)$ with each clause C_j, $j = 1, \ldots, m$. The relations in the poset $\mathcal{P}$ are defined as follows: for a literal $l_i \in \mathcal{Q}_i$ and an element $a_i^j \in \mathcal{P}_i \setminus \mathcal{Q}_i$, let $a_i^j \prec l_i$ in $\mathcal{P}_i$ if and only if l_i does not appear in clause C_j in C. Finally, Let $\mathcal{Q} = \mathcal{Q}_1 \times \ldots \times \mathcal{Q}_n$, and $\mathcal{A} = \{a^1, \ldots, a^m\}$. Then $\mathcal{Q} \not\subseteq \mathcal{A}^+$ if and only if C is satisfiable. $\square$

3.5 Projection

As seen above, it is necessary throughout the algorithm, to maintain the condition $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$, so that when we arrive at the base case, we can apply Proposition 3. Clearly, $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$ holds initially, but might not hold after decomposing $\mathcal{P}$. To solve this problem, we project the elements of $\mathcal{A}$ and $\mathcal{B}$ on the poset $\mathcal{P}$, for each newly created subproblem DUAL($\mathcal{P}, \mathcal{A}, \mathcal{B}$). More precisely, if there is an $a \in \mathcal{A}$, $k \in [n]$ such that $a_k^+ \cap \mathcal{P}_k \ne \emptyset$, but $a_k \notin \mathcal{P}_k$, we replace a by the set of elements $\{(a_1, \ldots, a_{k-1}, x, a_{k+1}, \ldots, a_n) \mid x \in \mathrm{Min}(a_k^+ \cap \mathcal{P}_k)\}$, where $\mathrm{Min}(\cdot)$ is the set of minimal elements of $(\cdot)$. Similarly, if there is an element $b \in \mathcal{B}$, and an index $k \in [n]$ such that $b_k^- \cap \mathcal{P}_k \ne \emptyset$, but $b_k \notin \mathcal{P}_k$, we replace b by the set of elements $\{(b_1, \ldots, b_{k-1}, x, b_{k+1}, \ldots, b_n) \mid x \in \mathrm{Max}(b_k^- \cap \mathcal{P}_k)\}$. Note that condition (4) continues to hold after such replacements.

In general, an element of $\mathcal{A}$ or $\mathcal{B}$ may project to a number of elements in $\mathcal{P}$. Thus performing a large number of projection steps, we may end up with an exponential increase in the sizes of $\mathcal{A}, \mathcal{B}$. However, for certain classes of posets, such as lattices and forests with connected precedence graphs (i.e., trees), each element of $\mathcal{A}, \mathcal{B}$ projects on a single element in $\mathcal{P}$, i.e., $|\operatorname{Min}(a_k^+ \cap \mathcal{P}_k)| = |\operatorname{Max}(b_k^- \cap \mathcal{P}_k)| = 1$, for all $a \in \mathcal{A}$, $b \in \mathcal{B}$, and $k \in [n]$. Indeed, if $\mathcal{P}_k$ is a lattice then $\operatorname{Min}(a_k^+ \cap \mathcal{P}_k) = \{a_k \vee \min(\mathcal{P}_k)\}$ and $\operatorname{Max}(b_k^- \cap \mathcal{P}_k) = \{b_k \wedge \max(\mathcal{P}_k)\}$, where $\min(\mathcal{P}_k)$ and $\max(\mathcal{P}_k)$ are respectively the minimum and maximum elements of $\mathcal{P}_k$. Similarly, if $\mathcal{P}_k$ is has a tree precedence graph, then $\operatorname{Min}(a_k^+ \cap \mathcal{P}_k) = \{\min(a_k^+ \cap \mathcal{P}_k)\}$ and $\operatorname{Max}(b_k^- \cap \mathcal{P}_k) = \{\max(b_k^- \cap \mathcal{P}_k)\}$ (if the precedence graph of $\mathcal{P}_k$ is a tree, and there are two distinct minimal elements $y, z \in \mathcal{P}_k$ with the property that $y \succ a_k$ and $z \succ a_k$, then there exists an undirected path between y and z in the precedence graph of $\mathcal{P}_k$, and another path through a_k, forming a cycle, in contradiction to the fact that the original poset (of which $\mathcal{P}_k$ is subposet) is a forest).

Thus, in conclusion, when decomposing a given dualization problem into a number of subproblems, we need to make sure that, in each resulting subproblem DUAL$(\mathcal{P}, \mathcal{A}, \mathcal{B})$, the poset $\mathcal{P}$ is still the product of lattices, or the product of forests with connected precedence graphs. In fact, this is the only place where the algorithms, described below, fail to work for products of general posets.

4 Dualization in products of join semi-lattices

Let $\mathcal{L} = \mathcal{L}_1 \times \ldots \times \mathcal{L}_n$ where each $\mathcal{L}_i$ is a join semi-lattice with maximum element u_i, and let $\mathcal{A} \subseteq \mathcal{L}$ and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$.

We begin with the observation that dualization on products of join semi-lattices can be reduced in polynomial time to dualization on products of lattices. Indeed, for each join semi-lattice $\mathcal{L}_i$, let us add a minimum element l_i that precedes every element in $\mathcal{L}_i$. Then it is easy to see that the resulting poset $\mathcal{L}'_i \stackrel{\text{def}}{=} \mathcal{L}_i \cup \{l_i\}$ is a lattice. Given $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L}$ satisfying (4), let us obtain a new set $\mathcal{B}' \subseteq \mathcal{L}' \stackrel{\text{def}}{=} \mathcal{L}'_1 \times \ldots \times \mathcal{L}'_n$ by extending $\mathcal{B}$ as follows. For each added minimum element l_i, we define a new element $b \in \mathcal{B}'$ by setting $b_i = l_i$, and $b_j = u_j$ for $j \neq i$. Clearly, condition (4) still holds for the pair $(\mathcal{A}, \mathcal{B} \cup \mathcal{B}')$, and $\mathcal{A}^+ \cup \mathcal{B}^- = \mathcal{L}$ if and only if $\mathcal{A}^+ \cup (\mathcal{B} \cup \mathcal{B}')^- = \mathcal{L}'$ by construction. Thus, for the rest of this section, we shall assume without loss of generality that each poset $\mathcal{L}_i$ is a lattice.

Before we prove Theorem 1, we show that the simpler (high-frequency based) algorithm of [FK96] can also be generalized for lattices to get a weaker bound than that of Theorem 1 (in fact, with an exponent linear in W, in contrast to the super-quadratic bound in Theorem 1).

4.1 Algorithm A

The first dualization algorithm for lattices is given in Figure 6. In the algorithm, we use $\delta = \delta(W) = \sqrt{(W+3)\log(W+2)}$, where $W = W(\mathcal{L})$. As usual, the algorithm is called initially with $\mathcal{Q} = \mathcal{L}$. In a general step, we check if there is a frequent element $z \in \cup_{i=1}^n \mathcal{Q}_i$, satisfying Lemma 5 (iii)-(iv) (where $\alpha = 2$). If no such z can be found, a new element in $\mathcal{Q} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ can be obtained as described in the proof of Lemma 4. Otherwise, a decomposition of $\mathcal{Q}_i$ into a set of lattices can be obtained and the algorithm is called recursively as in Steps 8 and 10.

Procedure LD–A($\mathcal{Q},\mathcal{A},\mathcal{B}$)**:**

Input: A sublattice $\mathcal{Q}=\mathcal{Q}_1\times\cdots\times\mathcal{Q}_n\subseteq\mathcal{L}$ and two anti-chains $\mathcal{A},\mathcal{B}\subseteq\mathcal{L}$

Output: **true** if $\mathcal{Q}\subseteq(\mathcal{A}^+\cup\mathcal{B}^-)$ and **false** otherwise

1. $\mathcal{A}\leftarrow\mathcal{A}(\mathcal{Q})$, $\mathcal{B}\leftarrow\mathcal{B}(\mathcal{Q})$
2. $\mathcal{A}\leftarrow$ PROJECT($\mathcal{Q},\mathcal{A}$), $\mathcal{B}\leftarrow$ PROJECT($\mathcal{Q},\mathcal{B}$)
3. **if** $\min\{\mathcal{A},\mathcal{B}\}\leq\delta$ **then**
4.   **return** POLY-DUAL($\mathcal{Q},\mathcal{A},\mathcal{B}$)
5. Find $i\in[n]$ and $z\in\mathcal{Q}_i$ that satisfy Lemma 5 (iii)-(iv); **if** no such elements exist **then**
6.   **return false**
7. **if** z satisfies Lemma 5(iii) **then**
8.   **return** LD–A($(\mathcal{Q}_i\cap z^-)\times\overline{\mathcal{Q}},\mathcal{A},\mathcal{B}$) $\wedge$ ($\bigwedge_{x\in\mathrm{Min}(\mathcal{Q}_i\setminus z^-)}$ LD–A($(\mathcal{Q}_i\cap x^+)\times\overline{\mathcal{Q}},\mathcal{A},\mathcal{B}$))
9. **else**
10.   **return** LD–A($(\mathcal{Q}_i\cap z^+)\times\overline{\mathcal{Q}},\mathcal{A},\mathcal{B}$) $\wedge$ ($\bigwedge_{x\in\mathrm{Max}(\mathcal{Q}_i\setminus z^+)}$ LD–A($(\mathcal{Q}_i\cap x^-)\times\overline{\mathcal{Q}},\mathcal{A},\mathcal{B}$))

Figure 6: The first dualization procedure for lattices.

4.1.1 Analysis of algorithm LD–A

Lemma 6 *Let $C(v)$ be the total number of recursive calls of procedure* LD–A($\mathcal{Q},\mathcal{A},\mathcal{B}$) *on a problem of size $v=|\mathcal{A}(\mathcal{Q})||\mathcal{B}(\mathcal{Q})|\geq 1$. Then $C(v)\leq R(v)\stackrel{\mathrm{def}}{=}v^{\log v/\epsilon}$, where $\epsilon=1/(2W\ln m)$.*

Proof. If $v\geq 1$ but $\min\{\mathcal{A},\mathcal{B}\}\leq\delta$, then Step 4 implies that $c(v)=1\leq R(v)$. Suppose now that the algorithm proceeds to Step 8, and let $\mathcal{Q}'=(\mathcal{Q}_i\cap z^-)\times\overline{\mathcal{Q}}$ and $\mathcal{Q}^x=(\mathcal{Q}_i\cap x^+)\times\overline{\mathcal{Q}}$ for $x\in\mathrm{Min}(\mathcal{Q}_i\setminus z^-)$ be the subposets constructed at that step. Then it follows from Lemma 5(i) that $|\mathcal{A}(\mathcal{Q}')|\leq|\mathcal{A}|-1$ and $|\mathcal{B}(\mathcal{Q}')|\geq\epsilon|\mathcal{B}|$, and thus

$$\begin{aligned}v(\mathcal{A}(\mathcal{Q}'),\mathcal{A}(\mathcal{Q}'))&\leq(|\mathcal{A}|-1)|\mathcal{B}|\leq v-\delta,\\ v(\mathcal{A}(\mathcal{Q}^x),\mathcal{A}(\mathcal{Q}^x))&\leq|\mathcal{A}|(1-\epsilon)|\mathcal{B}|=(1-\epsilon)v.\end{aligned}$$

Combined with the fact that $|\mathrm{Min}(\mathcal{Q}_i\setminus z^-)|\leq W$, this leads to the recurrence

$$C(v)\leq 1+W\cdot C((1-\epsilon)v)+C(v-\delta).$$

We get also a similar recurrence if the algorithm proceeds to Step 10. To evaluate this recurrence, we first apply it k times to get $C(v)\leq k+kW\cdot C((1-\epsilon)v)+C(v-k\delta)$. Letting $k=\lceil\frac{v\epsilon}{\delta}\rceil$ yields $C(v)\leq(1+(W+1)(\frac{v\epsilon}{\delta}+1))C((1-\epsilon)v)$, and hence $C(v)\leq(1+(W+1)(\frac{v\epsilon}{\delta}+1))^{\log v/\epsilon}=(W+2+\frac{W+1}{\delta}v\epsilon)^{\log v/\epsilon}$. Since $\min\{|\mathcal{A}|,|\mathcal{B}|\}\geq\delta$, we have $v\geq\delta^2$ and thus

$$v\left(1-\frac{W+1}{\delta}\epsilon\right)\geq\delta^2\left(1-\frac{W+1}{\delta W(1+\log\delta)}\right)\geq W+2,\tag{13}$$

for all $W\geq 1$, by our selection of $\delta(W)$, implying that $C(v)\leq v^{\log v/\epsilon}$. □

Since $v\leq m^2$, we get by combining Proposition 3 and Lemma 6 that the running time of the algorithm is $O(m^{4W\log^2 m+1}(nW)^{\sqrt{(W+3)\log(W+2)}}\mu)$.

4.2 Algorithm B

This algorithm, shown in Figure 7, does not use high-frequency decomposition; any $a \in \mathcal{A}$, $b \in \mathcal{B}$, and $i \in [n]$ such that $a_i \not\preceq b_i$ can be used as explained in Section 3.2.2 (see Step 5 of the algorithm). The algorithm chooses between rules $(R1)$, $(R2)$ and $(R2')$ according to the sizes of the relevant subsets of $\mathcal{A}$ and $\mathcal{B}$. More precisely, define $\epsilon(v) = \rho(W)/\chi(v)$, where $v = v(\mathcal{A}, \mathcal{B})$, $\rho(W) \stackrel{\text{def}}{=} \gamma(W)/W = 2W\ln(W+1)$, and $\chi(v)$ is defined to be the unique positive root of the equation

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

and observe that $\epsilon(v) < 1$ for $v \geq \delta^2$, $\delta \geq 2$. If the both $\epsilon_1^{\mathcal{A}} \stackrel{\text{def}}{=} |\mathcal{A}_{\not\preceq}(b_i)|/|\mathcal{A}|$ and $\epsilon_1^{\mathcal{B}} \stackrel{\text{def}}{=} |\mathcal{B}_{\preceq}(b_i)|/|\mathcal{B}|$ are greater than $\epsilon(v)$, then the algorithm uses rule $(R1)$, but with the further decomposition of $\mathcal{Q}_i \setminus b_i^-$, to ensure the lattice property. Otherwise, if $\epsilon_1^{\mathcal{A}} \leq \epsilon(v)$, the algorithm uses rule $(R2')$. If $\epsilon_1^{\mathcal{A}} > \epsilon(v)$, then there exists a element $z \in \mathcal{Q}_i$, such that $|\mathcal{A}_{\succeq}(z) \geq \epsilon_1^{\mathcal{A}}|\mathcal{A}|/|\operatorname{Min}(\mathcal{Q}_i \setminus b_i^-)| > \epsilon(v)|\mathcal{A}|/W$. Then again the algorithm chooses between rules $(R1)$ and $(R2)$ according to the sizes of the sets $\mathcal{A}_{\succeq}(z)$ and $\mathcal{B}_{\not\succeq}(z)$ (see steps 15-20).

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

4.2.1 Analysis of algorithm LD–B

Again, we measure the reduction in the "effective" volume at each recursion level.

Step 7: from the condition $\min\{\epsilon_1^{\mathcal{A}}, \epsilon_1^{\mathcal{B}}\} > \epsilon(v)$, and $|\operatorname{Min}(\mathcal{Q}_i \setminus b_i^-)| \leq W$, we get the recurrence

$$\begin{aligned} C(v) &\leq 1 + C(|\mathcal{A}_{\preceq}(b_i)||\mathcal{B}|) + |\operatorname{Min}(\mathcal{Q}_i \setminus b_i^-)|C(|\mathcal{A}||\mathcal{B}_{\not\preceq}(b_i)|) \\ &\leq 1 + C((1-\epsilon_1^{\mathcal{A}})v) + W \cdot C((1-\epsilon_1^{\mathcal{B}})v) \\ &\leq 1 + (W+1)C((1-\epsilon(v))v). \end{aligned} \tag{14}$$

Steps 11-12: from $\epsilon_1^{\mathcal{A}} \leq \epsilon(v)$ and (11), we get the recurrence

$$\begin{aligned} C(v) &\leq 1 + C(|\mathcal{A}_{\preceq}(b_i)||\mathcal{B}|) + \sum_{j=1}^{k}\left(\prod_{y \in (x^j)^\perp} |\mathcal{B}_{\succeq}(y)|\right) C(|\mathcal{A}_{=}(x^j)||\mathcal{B}_{\succeq}(x^j)|) \\ &\leq 1 + C(|\mathcal{A}_{\preceq}(b_i)||\mathcal{B}|) + |\mathcal{B}|^W \sum_{j=1}^{k} C(|\mathcal{A}_{=}(x^j)||\mathcal{B}_{\succeq}(x^j)|) \\ &\leq 1 + C((1-\epsilon_1^{\mathcal{A}})v) + |\mathcal{B}|^W C(\epsilon_1^{\mathcal{A}} v) \\ &\leq 1 + C((1-\epsilon_1^{\mathcal{A}})v) + \frac{v^W}{\delta^W} C(\epsilon_1^{\mathcal{A}} v) \\ &\leq C((1-\epsilon)v) + \frac{v^W}{\delta^W - 1} C(\epsilon v), \quad \text{for some } \epsilon \in (0, \epsilon(v)] \end{aligned} \tag{15}$$

where the second inequality follows from the fact that $|(x^j)^\perp| \leq W$, the third inequality follows from $\sum_{j=1}^k C(|\mathcal{A}_{=}(x^j)||\mathcal{B}_{\succeq}(x^j)|) \leq C(\sum_{j=1}^k |\mathcal{A}_{=}(x^j)||\mathcal{B}_{\succeq}(x^j)|) = C(|\mathcal{A}_{\not\preceq}(b_i)||\mathcal{B}_{\succeq}(x^j)|)$ since $\{\mathcal{A}_{=}(x^j) \mid j = 1, \ldots, k\}$ is a partition of $\mathcal{A}_{\not\preceq}(b_i)$ and the function $C(\cdot)$ is super-additive, the forth inequality follows from $|\mathcal{B}|^W \leq v(|\mathcal{A}|, |\mathcal{B}|)^W/\delta^W$, and the last inequality follows from the fact that $v \geq \delta^2$ and $\delta \geq 2$.

Procedure LD–B$(\mathcal{Q}, \mathcal{A}, \mathcal{B})$**:**

Input: A sublattice $\mathcal{Q} = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_n \subseteq \mathcal{L}$ and two anti-chains $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L}$

Output: **true** if $\mathcal{Q} \subseteq (\mathcal{A}^+ \cup \mathcal{B}^-)$ and **false** otherwise

$\mathcal{A} \leftarrow \mathcal{A}(\mathcal{Q})$, $\mathcal{B} \leftarrow \mathcal{B}(\mathcal{Q})$

$\mathcal{A} \leftarrow$ PROJECT$(\mathcal{Q}, \mathcal{A})$, $\mathcal{B} \leftarrow$ PROJECT$(\mathcal{Q}, \mathcal{B})$

if $\min\{\mathcal{A}, \mathcal{B}\} \leq \delta = 2$ **then**

  return POLY-DUAL$(\mathcal{Q}, \mathcal{A}, \mathcal{B})$

Let $a \in \mathcal{A}$, $b \in \mathcal{B}$, and $i \in [n]$ be such that $a_i \not\preceq b_i$

$\epsilon_1^{\mathcal{A}} \leftarrow \frac{|\mathcal{A}_{\not\preceq}(b_i)|}{|\mathcal{A}|}$ and $\epsilon_1^{\mathcal{B}} \leftarrow \frac{|\mathcal{B}_{\preceq}(b_i)|}{|\mathcal{B}|}$

if $\min\{\epsilon_1^{\mathcal{A}}, \epsilon_1^{\mathcal{B}}\} > \epsilon(v(\mathcal{A}, \mathcal{B}))$ **then**

  return LD–B$((\mathcal{Q}_i \cap b_i^-) \times \overline{\mathcal{Q}}, \mathcal{A}, \mathcal{B}) \wedge (\bigwedge_{x \in \mathrm{Min}(\mathcal{Q}_i \setminus b_i^-)}$ LD–B$((\mathcal{Q}_i \cap x^+) \times \overline{\mathcal{Q}}, \mathcal{A}, \mathcal{B}))$

if $\epsilon_1^{\mathcal{A}} \leq \epsilon(v(\mathcal{A}, \mathcal{B}))$ **then**

  Let $x^1, \ldots, x^k$ be the elements of $\mathcal{Q}_i \setminus b_i^-$ in topologically non-decreasing order

  $d \leftarrow$ LD–B$((\mathcal{Q}_i \cap b_i^-) \times \overline{\mathcal{Q}}, \mathcal{A}, \mathcal{B})$

  return $d \wedge (\bigwedge_{j \in [k]} \bigwedge_{(b(y) \in \mathcal{B}_{\succeq}(y) : y \in (x^j)^\perp)}$ LD–B$\left(\{x^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\bigwedge_{y \in (x^j)^\perp} \overline{b}(y)\right)^-\right], \mathcal{A}, \mathcal{B}\right))$

Let $z \in \mathcal{Q}_i$ be such that $|\mathcal{A}_{\succeq}(z)| > \epsilon(v(\mathcal{A}, \mathcal{B}))|\mathcal{A}|/W$

$\epsilon_2^{\mathcal{A}} \leftarrow \frac{|\mathcal{A}_{\succeq}(z)|}{|\mathcal{A}|}$ and $\epsilon_2^{\mathcal{B}} \leftarrow \frac{|\mathcal{B}_{\not\succeq}(z)|}{|\mathcal{B}|}$

if $\epsilon_2^{\mathcal{B}} > \epsilon(v(\mathcal{A}, \mathcal{B}))$ **then**

  return LD–B$((\mathcal{Q}_i \cap z^+) \times \overline{\mathcal{Q}}^i, \mathcal{A}, \mathcal{B}) \wedge (\bigwedge_{x \in \mathrm{Max}(\mathcal{Q}_i \setminus z^+)}$ LD–B$((\mathcal{Q}_i \cap x^-) \times \overline{\mathcal{Q}}, \mathcal{A}, \mathcal{B}))$

else

  Let $y^1, \ldots, y^h$ be the elements of $\mathcal{Q}_i \setminus z^+$ in topologically non-increasing order

  $d \leftarrow$ LD–B$((\mathcal{Q}_i \cap z^+) \times \overline{\mathcal{Q}}, \mathcal{A}, \mathcal{B})$

  return $d \wedge (\bigwedge_{j \in [k]} \bigwedge_{(a(y) \in \mathcal{A}_{\preceq}(x) : x \in (y^j)^\top)}$ LD–B$\left(\{y^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\bigvee_{x \in (y^j)^\perp} \overline{a}(x)\right)^-\right], \mathcal{A}, \mathcal{B}\right))$

Figure 7: The second dualization procedure for lattices.

Step 16: since $\epsilon_2^{\mathcal{A}} > \frac{\epsilon(v)}{W}$ by our selection of $z \in \mathcal{Q}_i$, and $\epsilon_2^{\mathcal{B}} > \epsilon(v)$, we get

$$\begin{aligned} C(v) &\leq 1 + C(|\mathcal{A}||\mathcal{B}_{\succeq}(z)|) + |\operatorname{Max}(\mathcal{Q}_i \setminus z^+)|C(|\mathcal{A}_{\not\succeq}(z)||\mathcal{B}|) \\ &\leq 1 + C((1-\epsilon_2^{\mathcal{B}})v) + W \cdot C((1-\epsilon_2^{\mathcal{A}})v) \\ &\leq 1 + C((1-\epsilon(v))v) + W \cdot C((1-\frac{\epsilon(v)}{W})v). \end{aligned} \tag{16}$$

Steps 19-20: symmetric to steps 11-12, we get again (15).

4.2.2 Proof of Theorem 1

We show by induction on $v = v(\mathcal{A}, \mathcal{B})$, that recurrences (14)–(16) imply that $C(v) \leq R(v) \stackrel{\text{def}}{=} v^{\chi(v)}$. Since, for $\min\{|\mathcal{A}|, |\mathcal{B}|\} < \delta$, Step 2 of the algorithm implies that $C(v) = 1$, we may assume that $\min\{|\mathcal{A}|, |\mathcal{B}|\} \geq \delta$, i.e., $v \geq \delta^2 = 4$.

Let us consider first recurrence (16). Using the induction hypothesis and the monotonicity of $\mathcal{X}(v)$, we obtain

$$\begin{aligned} C(v) &\leq 1 + [(1-\epsilon(v))v]^{\chi(v)} + W[(1-\frac{\epsilon(v)}{W})v]^{\chi(v)} \\ &\leq 1 + \left(e^{-\rho(W)} + We^{-\rho(W)/W}\right) v^{\chi(v)} \leq v^{\chi(v)}, \end{aligned} \tag{17}$$

since $1 - e^{-\rho(W)} - We^{-\rho(W)/W} \geq 1/2$ for all $W \geq 1$.

Let us consider next (14) and note that the monotonicity of $C(v)$ implies that $C((1 - \epsilon(v))v) \leq C((1 - \frac{\epsilon(v)}{W})v)$, concluding by (14) and (17) that $C(v) \leq R(v)$ for this case too.

Let us now consider (15) and apply induction to get

$$C(v) \leq [(1-\epsilon)v]^{\chi(v)} + \frac{v^W}{\delta^W - 1}[\epsilon v]^{\chi(v)} = \psi(\epsilon)v^{\chi(v)},$$

where $\psi(\epsilon) \stackrel{\text{def}}{=} (1-\epsilon)^{\chi(v)} + \frac{v^W}{\delta^W - 1}\epsilon^{\chi(v)}$. Since $\psi(\epsilon)$ is convex in ϵ, $\psi(0) = 1$, $\epsilon \leq \epsilon(v)$, and

$$\psi(\epsilon(v)) = \left(1 - \frac{\rho(W)}{\chi(v)}\right)^{\chi(v)} + \frac{v^W}{\delta^W - 1}\left(\frac{\rho(W)}{\chi(v)}\right)^{\chi(v)} \leq e^{-\rho(W)} + \frac{v^W}{\delta^W - 1}\left(\frac{\rho(W)}{\chi(v)}\right)^{\chi(v)} = 1,$$

by the definition of $\chi(v)$, it follows that $\psi(\epsilon) \leq 1$ and hence, $C(v) \leq v^{\chi(v)}$.

Note that, for $\delta \geq 2$ and $W \geq 1$, we have $(\chi/\rho(W))^{\chi} < 3\,(v/\delta)^W$, and thus,

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

As $v(\mathcal{A}, \mathcal{B}) < m^2$, we get $\chi(v) = o(W\rho(W)\log m)$, concluding the proof of the theorem.

5 Dualization in products of forests

5.1 The algorithm

Let $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$, where the precedence graph of each poset $\mathcal{P}_i$ is a forest, and let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}$ two antichains satisfying (4). The algorithm is shown in Figure 9.

If the precedence graph of $\mathcal{P}_i$ is not connected, for some $i \in [n]$, we use Proposition 2 to decompose the problem into a number of subproblems over posets with connected precedence graphs (steps 3-4).

Starting from step 7, we decompose $\mathcal{Q} \subseteq \mathcal{P}$ by picking $a \in \mathcal{A}$, $b \in \mathcal{B}$ and an $i \in [n]$, such that $a_i \not\preceq b_i$. If in-deg$(\mathcal{Q}_i) \leq$ out-deg$(\mathcal{Q}_i)$, we set $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i \cap a_i^+$ and $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i'$, otherwise, we set $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \cap b_i^-$, and $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i''$. In the latter case, we should use the symmetric versions of the decomposition rules used for the former case, and a brief way to describe this is to replace $\mathcal{P}$ by its dual poset $\mathcal{P}^*$, and exchange the roles of $\mathcal{A}$ and $\mathcal{B}$ in these rules (step 9). Assume without loss of generality in the following that the former case holds.

As in the case of lattices, the algorithm uses the effective volume $v = v(\mathcal{A}, \mathcal{B})$ to compute the threshold

$$\epsilon(v) = \frac{1}{\chi(v)}, \qquad \text{where } \chi(v)^{\chi(v)} = v^d,\ v = v(\mathcal{A}, \mathcal{B}).$$

If the minimum of $\epsilon^{\mathcal{A}} \stackrel{\text{def}}{=} |\mathcal{A}_{\succeq}(a_i)|/|\mathcal{A}|$ and $\epsilon^{\mathcal{B}} \stackrel{\text{def}}{=} |\mathcal{B}_{\not\succeq}(a_i)|/|\mathcal{B}|$ is bigger than $\epsilon(v)$, then rule $(R1)$ is used for decomposition (step 13). Otherwise we proceed as follows. Let $\mathcal{Q}_i^e = \{x \in \mathcal{Q}_i' \mid x^\perp \cap \mathcal{Q}_i'' \neq \emptyset\}$ be the set of elements in $\mathcal{Q}_i'$ with immediate predecessors in $\mathcal{Q}_i''$ (see Figure 8). Let, for each $x \in \mathcal{Q}_i^e$, $\mathcal{Q}_i(x) = \mathcal{Q}_i'' \cap x^-$, $\mathcal{A}(x) = \mathcal{A}(\mathcal{Q}_i(x) \times \overline{\mathcal{Q}})$ and $\mathcal{B}(x) = \mathcal{B}(\mathcal{Q}_i(x) \times \overline{\mathcal{Q}})$. Observe that $\mathcal{Q}_i(x)$ and $\mathcal{Q}_i(y)$ are independent posets for $x \neq y$, $x, y \in \mathcal{Q}_i^e$, and that $\mathcal{B}(x) \cap (\mathcal{Q}_i' \times \overline{\mathcal{Q}}) = \mathcal{B}_{\succeq}(x)$ for all $x \in \mathcal{Q}_i^e$, since the precedence graph of $\mathcal{Q}_i$ is a tree.

Letting further $\mathcal{Q}_i^r = \mathcal{Q}_i'' \setminus \left(\bigcup_{x \in \mathcal{Q}_i^e} \mathcal{Q}_i(x)\right)$, we can apply rule $(R2)$ but stop the decomposition after processing the first layer $\{x^\perp \mid x \in \mathcal{Q}_i^e\}$. This gives the following set of duality testing subproblems (steps 15-17):

$$\begin{array}{l} \mathcal{Q}_i' \times \overline{\mathcal{Q}} \subseteq \mathcal{A}^+ \cup (\mathcal{B}_{\succeq}(a_i))^- \\ \mathcal{Q}_i^r \times \overline{\mathcal{Q}} \subseteq (\mathcal{A}_{\not\succeq}(a_i))^+ \cup (\mathcal{B}_{\not\succeq}(a_i))^- \\ \mathcal{Q}_i(x) \times (\overline{\mathcal{Q}} \cap \overline{a}^+) \subseteq (\mathcal{A}(x))^+ \cup (\mathcal{B}(x) \setminus \mathcal{B}_{\succeq}(a_i))^-,\ \forall a \in \mathcal{A}_{\preceq}(x), x \in \mathcal{Q}_i^e. \end{array} \tag{18}$$

To see the last decomposition in (18), fix an $x \in \mathcal{Q}_i^e$, and make use of Lemma 1 by taking $z \leftarrow x$ and $\mathcal{R}_i'' \leftarrow \mathcal{Q}_i(x)$.

Finally, if $\epsilon^{\mathcal{A}} \leq \epsilon(v) < \epsilon^{\mathcal{B}}$, we use rule $(R2')$, see steps 19-21.

5.2 Analysis of algorithm FD

As before, we first write the recurrences corresponding to the different recursive calls. By steps 3-4 and Proposition 2, we may assume that the precedence graph of each poset $\mathcal{Q}_i$ is connected.

Step 13: Suppose that the connected components of $\mathcal{Q}_i''$ are $\mathcal{Q}_i^1, \ldots, \mathcal{Q}_i^h$. Then

$$\begin{aligned} C(v(\mathcal{A}, \mathcal{B})) &\leq 1 + C(|\mathcal{A}||\mathcal{B}_{\succeq}(a_i)|) + \sum_{j=1}^{h} C(|\mathcal{A}(\mathcal{Q}_i^j \times \overline{\mathcal{Q}})||\mathcal{B}|) \\ &\leq 1 + C(|\mathcal{A}||\mathcal{B}_{\succeq}(a_i)|) + C(|\mathcal{A}_{\not\succeq}(a_i)||\mathcal{B}|) \\ &= 1 + C((1 - \epsilon^{\mathcal{B}})v) + C((1 - \epsilon^{\mathcal{A}})v) \leq 1 + 2C((1 - \epsilon(v))v), \end{aligned} \tag{19}$$

since $\mathcal{A}(\mathcal{Q}_i^j \times \overline{\mathcal{Q}})$, for $j = 1, \ldots, h$ partition $\mathcal{A}_{\not\succeq}(a_i)$.

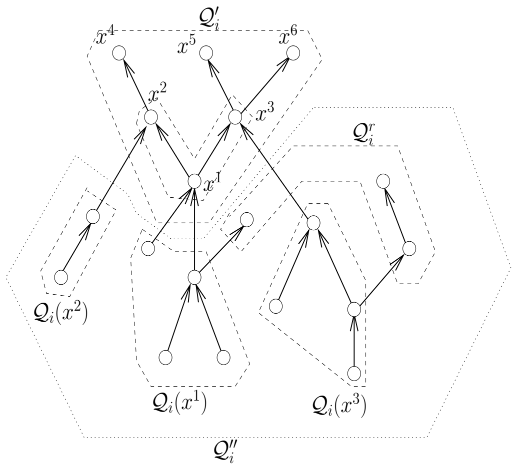


Figure 8: Decomposing the forest $\mathcal{Q}_i$.

Procedure FD($\mathcal{Q},\mathcal{A},\mathcal{B}$)**:**

Input: A subposet of a product of forests $\mathcal{Q} = \mathcal{Q}_1 \times \cdots \times \mathcal{Q}_n \subseteq \mathcal{P}$ and two anti-chains $\mathcal{A},\mathcal{B} \subseteq \mathcal{P}$

Output: **true** if $\mathcal{Q} \subseteq (\mathcal{A}^+ \cup \mathcal{B}^-)$ and **false** otherwise

$\mathcal{A} \leftarrow \mathcal{A}(\mathcal{Q})$, $\mathcal{B} \leftarrow \mathcal{B}(\mathcal{Q})$
$\mathcal{A} \leftarrow$ PROJECT($\mathcal{Q},\mathcal{A}$), $\mathcal{B} \leftarrow$ PROJECT($\mathcal{Q},\mathcal{B}$)
if there is an $i \in [n]$ such that $\mathcal{Q}_i$ can be decomposed into independent posets $\mathcal{Q}_i^1,\ldots,\mathcal{Q}_i^r$, **then**
    return $\bigwedge_{j=1}^r \mathrm{FD}(\mathcal{Q}_i^j \times \overline{\mathcal{Q}},\mathcal{A},\mathcal{B})$.
if $\min\{\mathcal{A},\mathcal{B}\} \le \delta = 3$ **then**
    return POLY-DUAL($\mathcal{Q},\mathcal{A},\mathcal{B}$)
Let $a \in \mathcal{A}$, $b \in \mathcal{B}$, and $i \in [n]$ be such that $a_i \not\preceq b_i$
    if in-deg($\mathcal{Q}_i$) $>$ out-deg($\mathcal{Q}_i$) **then**
    $\mathcal{P} \leftarrow \mathcal{P}^*$, exchange $\mathcal{A}$ and $\mathcal{B}$
$\epsilon^{\mathcal{A}} \leftarrow \frac{|\mathcal{A}_{\succeq}(a_i)|}{|\mathcal{A}|}$ and $\epsilon^{\mathcal{B}} \leftarrow \frac{|\mathcal{B}_{\not\succeq}(a_i)|}{|\mathcal{B}|}$
Let $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i \cap a_i^+$, $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i'$
if $\min\{\epsilon^{\mathcal{A}},\epsilon^{\mathcal{B}}\} > \epsilon(v(\mathcal{A},\mathcal{B}))$ **then**
    return $\mathrm{FD}(\mathcal{Q}_i' \times \overline{\mathcal{Q}},\mathcal{A},\mathcal{B}) \wedge \mathrm{FD}(\mathcal{Q}_i'' \times \overline{\mathcal{Q}},\mathcal{A},\mathcal{B})$
if $\epsilon^{\mathcal{B}} \le \epsilon(v(\mathcal{A},\mathcal{B}))$ **then**
Let $\mathcal{Q}_i^e = \{x \in \mathcal{Q}_i' \mid x^\perp \cap \mathcal{Q}_i'' \ne \emptyset\}$, $\mathcal{Q}_i(x) = \mathcal{Q}_i'' \cap x^-$, for $x \in \mathcal{Q}_i$, and $\mathcal{Q}_i^r = \mathcal{Q}_i'' \setminus \left(\bigcup_{x\in\mathcal{Q}_i^e} \mathcal{Q}_i(x)\right)$
    $d_1 \leftarrow \mathrm{FD}(\mathcal{Q}_i' \times \overline{\mathcal{Q}},\mathcal{A},\mathcal{B})$; $d_2 \leftarrow \mathrm{FD}(\mathcal{Q}_i^r \times \overline{\mathcal{Q}},\mathcal{A},\mathcal{B})$
    return $d_1 \wedge d_2 \wedge (\bigwedge_{x\in\mathcal{Q}_i^e} \bigwedge_{a\in\mathcal{A}_{\preceq}(x)} \mathrm{FD}(\mathcal{Q}_i(x) \times (\overline{\mathcal{Q}} \cap \overline{a}^+),\mathcal{A},\mathcal{B}))$
else
    Let $x^1,\ldots,x^k$ be the elements of $\mathcal{Q}_i'$ in topologically non-decreasing order
    $d \leftarrow$ LD–A($\mathcal{Q}_i'' \times \overline{\mathcal{Q}},\mathcal{A},\mathcal{B}$)
    return $d \wedge (\bigwedge_{j\in[k]} \bigwedge_{(b(y)\in\mathcal{B}_{\succeq}(y):y\in(x^j)^\perp)} \mathrm{FD}(\{x^j\} \times \left[\overline{\mathcal{Q}} \cap \left(\bigcap_{y\in(x^j)^\perp} \overline{b}(y)^-\right)\right],\mathcal{A},\mathcal{B}))$

Figure 9: The dualization procedure for forests.

Steps 16-17: from (18) and the fact that $\{\mathcal{A}(x) \mid x \in \mathcal{Q}_i^e\}$ is a partition of $\mathcal{A}$, we get the recurrence

$$\begin{aligned}
C(v) &\leq 1 + C(|\mathcal{A}||\mathcal{B}_{\succeq}(a_i)|) + C(|\mathcal{A}_{\not\succeq}(a_i)||\mathcal{B}_{\not\succeq}(a_i)|) + \sum_{x \in \mathcal{Q}_i^e} |\mathcal{A}_{\preceq}(x)| C(|\mathcal{A}(x)||\mathcal{B}(x) \setminus \mathcal{B}_{\succeq}(a_i)|) \\
&\leq 1 + C((1 - \epsilon^{\mathcal{B}})v) + (|\mathcal{A}| + 1) C(\epsilon^{\mathcal{B}} v) \\
&\leq C((1-\epsilon)v) + \frac{v}{2} C(\epsilon v), \quad \text{for some } \epsilon \in (0, \epsilon(v)], \qquad (20)
\end{aligned}$$

where the last inequality follows from the assumption that $\min\{|\mathcal{A}|, |\mathcal{B}|\} \geq 4$ and hence $|\mathcal{A}| + 1 \leq |\mathcal{A}||\mathcal{B}|/3 = v/3$.

Steps 20-21: since $|(x^j)^{\perp}| \leq d$ for every $x^j \in \mathcal{Q}_i'$, by our assumption that in-deg$(\mathcal{Q}_i) \leq$ out-deg$(\mathcal{Q}_i)$ (see steps 8-9 of the algorithm), we get

$$\begin{aligned}
C(v(\mathcal{A}, \mathcal{B})) &\leq 1 + C(|\mathcal{A}_{\not\succeq}(a_i)||\mathcal{B}|) + \sum_{j=1}^{k} \left(\prod_{y \in (x^j)^{\perp}} |\mathcal{B}_{\succeq}(y)| \right) C(|\mathcal{A}_{=}(x^j)||\mathcal{B}_{\succeq}(x^j)|) \\
&\leq 1 + C(|\mathcal{A}_{\not\succeq}(a_i)||\mathcal{B}|) + |\mathcal{B}|^d C(|\mathcal{A}_{\succeq}(a_i)||\mathcal{B}_{\succeq}(a_i)|) \\
&\leq 1 + C((1 - \epsilon^{\mathcal{A}})v) + \frac{v^d}{3} R(\epsilon^{\mathcal{A}} v) \\
&\leq C((1-\epsilon)v) + \frac{v^d}{2} C(\epsilon v), \quad \text{for some } \epsilon \in (0, \epsilon(v)]. \qquad (21)
\end{aligned}$$

Note that this the only place in which the bound d on the degrees appears.

As in Section 4.2.2, we can show by induction on v, that recurrences (19)–(21) imply $C(v) \leq R(v) \stackrel{\text{def}}{=} v^{\chi(v)}$. Noting that $\chi(v) < 2\chi(m) \sim 2d \log m / \log\log m$, and we get the bound stated in Theorem 2.

6 Dualization Algorithm in products of lattices of intervals

Let $\mathcal{L} = \mathcal{L}_1 \times \cdots \times \mathcal{L}_n$ be a product of n lattices of intervals, defined respectively by sets of intervals $\mathbb{I}_1, \ldots, \mathbb{I}_n$, and denote by l_i the minimum element of $\mathcal{L}_i$. In this section we prove Theorem 3. We fix fix $\epsilon = 1/(2 \ln m)$, and use $v(\mathcal{A}, \mathcal{B}, \mathcal{L}) = |\mathcal{A}||\mathcal{B}| \sum_{i=1}^{n} |\mathcal{L}_i|$ as a measure of the volume of the problem.

We begin with the following simple property satisfied by any lattice of intervals.

Proposition 5 *Let $\mathcal{L}_i$ be a lattice of intervals. Then (i) $|x^{\top}| \leq 2$ for all $x \neq l_i$ in $\mathcal{L}_i$, and (ii) $|x^{\perp}| \leq 2$ for all $x \in \mathcal{L}_i$.*

Proof. (i) Assume non-minimum $x \in \mathcal{L}_i$ has $|x^{\top}| \geq 3$. Let I_1, I_2, and I_3 be 3 immediate successors of x in $\mathcal{L}_i$. Let $I_1 = [a, b]$, $I_2 = [c, d]$, where $a, b, c, d \in \mathbb{R}$, and $a < c < b < d$. Then $x = I_1 \cap I_2 = [c, b]$. Let $I_3 = [e, f]$. Now, $I_1 \cap I_3 = x$ implies that $e = c$, and $I_2 \cap I_3 = x$ implies that $f = b$. This gives the contradiction $I_3 = x$.

(ii) Assume $x \in \mathcal{L}_i$ has $|x^{\perp}| \geq 3$. Let I_1, I_2, and I_3 be 3 immediate predecessors of x in $\mathcal{L}_i$. Let $I_1 = [a, b]$, $I_2 = [c, d]$, where $a, b, c, d \in \mathbb{R}$, and $a < b$, $c < d$, $a < c$ and $b < d$. Then $x = Span(I_1, I_2) = [a, d]$. Let $I_3 = [e, f]$. Now, $Span(I_1, I_3) = x$ implies that $f = d$, and $Span(I_2, I_3) = x$ implies that $e = a$. This gives the contradiction $I_3 = x$. □

Given subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{L}$ that satisfy (4), and a product of lattices of intervals $\mathcal{Q} \subseteq \mathcal{L}$, we follow the general framework as in Figure 4, but use a high-frequency based decomposition. More precisely, assuming $v(\mathcal{A}, \mathcal{B}, \mathcal{Q}) \geq 2$ at a general recursion level, we check if either condition (i) nor (ii) of Lemma 5 is satisfied. If neither is satisfied, then we can find an element $x \in \mathcal{Q} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$. Otherwise, we consider the following cases:

Case 1. If $i \in [n]$ and $z \in \mathcal{Q}_i$ satisfy condition (i) of Lemma 5 (with $\alpha = 2$), then we consider two subcases:

Case 1.1. If $\mathcal{Q}_i$ is a total order (chain), then use the following decomposition of $\mathcal{Q}_i$: $\mathcal{Q}'_i \leftarrow z^+ \cap \mathcal{Q}_i$, $\mathcal{Q}''_i \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}'_i$. Then $|\mathcal{B}(\mathcal{Q}'_i \times \overline{\mathcal{Q}})| \leq (1-\epsilon)|\mathcal{B}|$ and $|\mathcal{A}(\mathcal{Q}''_i \times \overline{\mathcal{Q}})| \leq |\mathcal{A}| - 1$. This reduces the original problem, of volume $v = |\mathcal{A}||\mathcal{B}| \sum_{i=1}^n |\mathcal{Q}_i|$, into two subproblems of volumes

$$
\begin{aligned}
v' &\leq |\mathcal{A}||\mathcal{B}|(1-\epsilon)(\sum_{i=1}^n |\mathcal{Q}_i| - 1) \leq (1-\epsilon)v, \\
v'' &\leq (|\mathcal{A}| - 1|)|\mathcal{B}|(\sum_{i=1}^n |\mathcal{Q}_i| - 1) \leq v - 1.
\end{aligned}
$$

Case 1.2. Otherwise ($\mathcal{Q}_i$ is not a chain), let w be the *largest* element, with respect to the precedence relation on the lattice $\mathcal{Q}_i$, such that $|w^\perp| = 2$ (see Figure 10–a). Denote respectively by q and y the two immediate predecessors of w in $\mathcal{Q}_i$, and assume that without loss of generality that $|\mathcal{B}_\preceq(y)| \geq |\mathcal{B}_\preceq(q)|$. It is not hard to see that $\mathcal{Q}_i \cap y^-$ is a lattice of intervals and that $\mathcal{Q}_i \setminus y^-$ is a chain. (Indeed, let $I_q = [a, b]$ and $I_y = [c, d]$ be the two intervals represented by q and y respectively, and assume that $a < c$ (and therefore $b < d$). Then the former claim follows from the fact that every element in $y^- \subseteq \mathcal{Q}_i$ is associated with an interval, which is the intersection or span of some intervals in $\mathbb{I}_i$, each of which is a subinterval in I_y. The latter claim follows from the fact that if an element $p \in q^- \setminus y^-$ has two immediate predecessors p' and p'' representing intervals $I_{p'} = [e, f]$ and $I_{p''} = [g, h]$, where $e < g$, then we must have $p'' \in y^-$, for otherwise $I_y \subset Span(I_{p''}, I_y) \subset Span(I_p, I_y)$, giving a contradiction with the fact that y is an immediate predecessor of w.)

Now we consider two cases:

(i) if $z \succ w$, we use the decomposition $\mathcal{Q}'_i \leftarrow z^+ \cap \mathcal{Q}_i$, $\mathcal{Q}''_i \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}'_i$. Otherwise, the choice of z implies that either cases (ii) or (iii) hold.

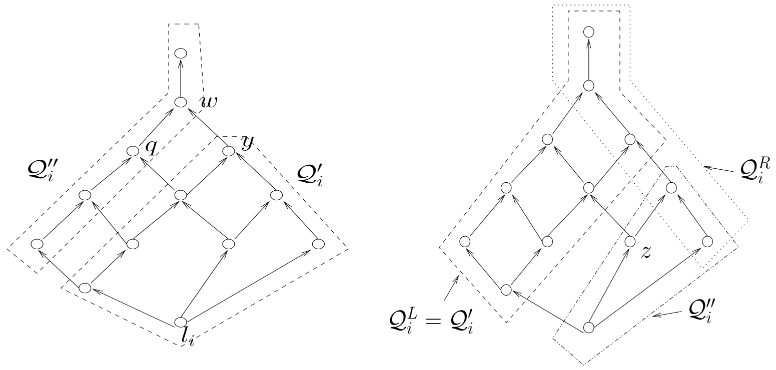
(ii) $z \not\succ w$: in this case, we decompose $\mathcal{Q}_i$ as $\mathcal{Q}'_i \leftarrow \mathcal{Q}_i \cap y^-$, $\mathcal{Q}''_i \leftarrow \mathcal{Q}_i \setminus y^-$.

In case (i), we get again that $|\mathcal{B}(\mathcal{Q}'_i \times \overline{\mathcal{Q}})| \leq (1-\epsilon)|\mathcal{B}|$ and $|\mathcal{A}(\mathcal{Q}''_i \times \overline{\mathcal{Q}})| \leq |\mathcal{A}| - 1$, and consequently, the resulting problems are of respective volumes $v' \leq (1-\epsilon)v$ and $v'' \leq v - 1$. In case (ii), we know that $|\mathcal{B}_\preceq(y)| \geq \frac{\epsilon}{2}|\mathcal{B}|$ and thus get $|\mathcal{B}(\mathcal{Q}''_i \times \overline{\mathcal{Q}})| \leq (1-\epsilon/2)|\mathcal{B}|$ and $|\mathcal{Q}'_i| \leq |\mathcal{Q}_i| - 1$, and therefore, the resulting two problems have volumes $v' \leq v - 1$ and $v'' \leq (1-\epsilon/2)v$.

Case 2. Now assume that $i \in [n]$, and $z \in \mathcal{Q}_i$ satisfy condition (ii) of Lemma 5. Consider further two subcases:

Case 2.1. If $|z^\top| \leq 2$, then let $I_z = [a, b]$ be the interval corresponding to z, and let $\mathcal{Q}_i^L \subseteq \mathcal{Q}_i$ be the lattice of intervals $I = [c, d]$ for which $c < a$, and likewise, $\mathcal{Q}_i^R \subseteq \mathcal{Q}_i$ be the lattice of intervals $I = [e, f]$ for which $f > b$ (see Figure 10–b). Note that these definitions imply that $(\mathcal{Q}_i^L \cup \{l_i\}) \cap z^- = \{l_i\}$, $(\mathcal{Q}_i^R \cup \{l_i\}) \cap z^- = \{l_i\}$, and $\mathcal{Q}_i^L \cup z^- \cup \mathcal{Q}_i^R = \mathcal{Q}$, where $l_i = \min(\mathcal{Q}_i)$. Note also that $\mathcal{Q}_i^L \cup \mathcal{Q}_i^R \neq \emptyset$ since $z \neq \max(\mathcal{Q}_i)$. By our selection of z, either

$$
\text{(i) } |\{a \in \mathcal{A} \mid a_i \in \mathcal{Q}_i^L \setminus \{l_i\}\} \geq \frac{\epsilon}{2}|\mathcal{A}|, \text{ or (ii) } |\{a \in \mathcal{A} \mid a_i \in \mathcal{Q}_i^R \setminus \{l_i\}\} \geq \frac{\epsilon}{2}|\mathcal{A}|.
$$



a: Decomposition rule used in case 1.2 (ii). b: Decomposition rule used in case 2.1.

Figure 10: Decomposing the lattice $\mathcal{L}_i$.

(Note that it is possible that $l_i \in \mathcal{Q}_i^L$ if there are two disjoint intervals in $\mathbb{I}_i$ whose left end-points are strictly to the left of the left end-point of z.) In case (i), we decompose $\mathcal{Q}_i$ as follows: $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i^L$, $\mathcal{Q}_i'' \leftarrow (\mathcal{Q}_i \setminus \mathcal{Q}_i') \cup \{l_i\}$. Note that both $\mathcal{Q}_i'$ and $\mathcal{Q}_i''$ are also lattices of intervals, that $|\mathcal{Q}_i'| \leq |\mathcal{Q}_i| - 1$ since $z \notin \mathcal{Q}_i'$, and that $\mathcal{A}(\mathcal{Q}_i'' \times \overline{\mathcal{Q}}) \leq (1-\epsilon/2)|\mathcal{A}|$, since $w \not\preceq y$ for all $w \in \mathcal{Q}_i' \setminus \{l_i\}$ and $y \in \mathcal{Q}_i'' \setminus \{l_i\}$ (indeed, if $I_w = [c,d]$ is the interval corresponding to $w \in \mathcal{Q}_i' \setminus \{l_i\}$ and $I_y = [e,f]$ is the interval corresponding to $y \in \mathcal{Q}_i'' \setminus \{l_i\}$, then $c < a$ while $e \geq a$ and thus $I_w \not\subseteq I_y$). Therefore, we get, in this case, two subproblems of volumes $v' \leq v - 1$ and $v'' \leq (1-\epsilon/2)v$. In case (ii), we let similarly $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i^R$ and $\mathcal{Q}_i'' \leftarrow (\mathcal{Q}_i \setminus \mathcal{Q}_i') \cup \{l_i\}$, and we decompose the original problem into two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon/2)v$, respectively.

Case 2.2. $|z^\top| > 2$ (which is possible only if $z = \min(\mathcal{Q}_i) = l_i$ represents the empty interval of $\mathcal{L}_i$), then we let z' be any immediate successor of z, and let $\mathcal{Q}_i^L$ and $\mathcal{Q}_i^R$ be the lattices of intervals as defined in case 2.1, but with respect to $I_{z'} = [a,b]$ instead of I_z. Note in this case that any interval $[c,d]$ in $\mathcal{Q}_i^L$ either must be strictly to the left of $I_{z'}$, i.e. with $d < a$, or must contain $I_{z'}$. Similarly, any interval $[e,f]$ in $\mathcal{Q}_i^R$ either must be strictly to the right of $I_{z'}$, i.e. with $e > b$, or must contain $I_{z'}$. We consider four cases:

(i) No interval of $\mathbb{I}_i$, corresponding to an element of $\mathcal{Q}_i$, lies strictly to the left or strictly to the right of $I_{z'}$: the choice of z, in this case, implies that $|\mathcal{A}_{\succeq}(z')| \geq \epsilon|\mathcal{A}|$. Thus using the decomposition $\mathcal{Q}_i' \leftarrow (z')^+ \cap \mathcal{Q}_i$ and $\mathcal{Q}_i'' \leftarrow \{z\}$ results in two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon)v$.

(ii) No interval lies strictly on the right of $I_{z'}$, but there is at least one that lies strictly to its left: by our choice of z, one of the sets $\{a \in \mathcal{A} \ : \ a_i \in \mathcal{Q}_i^L\}$ or $\mathcal{A}_{\succeq}(z')$ has size at least $\frac{\epsilon}{2}|\mathcal{A}|$. In the former case we use the decomposition $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i^L$, $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i'$, and get two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon)v$. In the latter case, we let $\mathcal{Q}_i''$ be the lattice of intervals lying strictly to the left of $I_{z'}$ and $\mathcal{Q}_i' \leftarrow ((z')^+ \cap \mathcal{Q}_i) \cup \{z\}$, and get two subproblems of volumes $v' \leq v-1$ and $v'' \leq (1-\epsilon)v$.

(iii) No interval lies strictly on the left of $I_{z'}$, but there is at least one that lies strictly to its right: we use a similar decomposition as in case (ii) above.

(iv) There is at least one interval that lies strictly to the left of $I_{z'}$, and at least one interval strictly to its right: in this case, we know that either $|\{a \in \mathcal{A} \mid a_i \in \mathcal{Q}_i^L \cup (z')^+\}| \geq \epsilon|\mathcal{A}|/2$ or $|\{a \in \mathcal{A} \mid a_i \in \mathcal{Q}_i^R \cup (z')^+\}| \geq \epsilon|\mathcal{A}|/2$. In the former case, we use the decomposition $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i^L \cup ((z')^+ \cap \mathcal{Q}_i) \cup \{z\}$, $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i' \cup \{z\}$, and in the latter case, we use the decomposition $\mathcal{Q}_i' \leftarrow \mathcal{Q}_i^R \cup ((z')^+ \cap \mathcal{Q}_i) \cup \{z\}$, $\mathcal{Q}_i'' \leftarrow \mathcal{Q}_i \setminus \mathcal{Q}_i' \cup \{z\}$. In both cases, we get two subproblems of volumes $v' \leq v-1$ and $v' \leq (1-\epsilon/2)v$.

Thus, in all cases, we apply the algorithm recursively to the resulting subproblems, and obtain the recurrence

$$C(v) \leq 1 + C((1-\epsilon/2)v) + C(v-1),$$

where $C(v)$ is the number of recursive calls required to solve a problem of volume v. Together with $C(v) = 1$, this recurrence evaluates to $C(v) \leq v^{2\log v/\epsilon}$. Since $v \leq m^2 n\mu$, we get that the running time of the algorithm is $O((m^2 n\mu)^{4\ln m \log(m^2 n\mu)})$.

7 Concluding remarks

It is worth mentioning that each poset $\mathcal{P}_i$ belonging to any of the classes of posets, considered in this paper, has constant dimension, i.e. $\mathcal{P}_i$ is isomorphic to a subposet of the product of a constant number of chains. In particular, the poset product $\mathcal{P} = \mathcal{P}_1 \times \ldots \times \mathcal{P}_n$, over which we want to solve the dualization problem, can be considered as a subposet of a chain product $\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_{n'}$, where n' is bounded by a constant in n. Although we know from [BEG+02] how to solve the dualization problem on products of chains, it is not clear how such a result can be used to solve the original dualization problem on $\mathcal{P}$, since the solution we obtain on $\mathcal{C} \supseteq \mathcal{P}$ (that is, the element $x \in \mathcal{I}(\mathcal{A}) \setminus \mathcal{B}$) might not be an element of $\mathcal{P}$. In fact, as we have seen, the algorithms presented for these classes of posets, depend heavily on the type of poset under consideration. This naturally raises the question whether a more general approach can unify these results for posets $\mathcal{P}_i$ of bounded dimension.

It is also not clear whether it is possible to solve the dualization problem in the products of lattices of intervals in time $k^{o(\log k)}$, where $k = |\mathcal{A}| + |\mathcal{B}| + \sum_{i=1}^n |\mathcal{L}_i|$, by following a set of decomposition rules, as those used in Section 4.2 to solve the problem for general lattices. It seems that, if this is to be achieved, some new decomposition rules are needed, since the current rules in section 4.2 depend exponentially on the maximum out-degree of the lattice $\mathcal{L}_i$, which is $O(|\mathbb{I}_i|)$ in the case of a lattice of a set of intervals $\mathbb{I}_i$.

Finally, we note that, for the more general case of products of arbitrary posets, it remains open whether the problem can be solved in quasi-polynomial time, even for posets $\mathcal{P}_i$ of small size.

Acknowledgments

The author is grateful to E. Boros, V. Gurvich, L. Khachiyan and K. Makino for helpful discussions, and to two anonymous reviewers for useful comments.

References

[AB92] M. Anthony and N. Biggs. *Computational learning theory: an introduction.* Cambridge University Press, New York, NY, USA, 1992.

[AIS93] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

[AMS+96] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. pages 307–328, 1996.

[BEG+02] E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. Dual-bounded generating problems: All minimal integer solutions for a monotone system of linear inequalities. *SIAM Journal on Computing*, 31(5):1624–1643, 2002.

[BEGK04] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. Enumerating minimal dicuts and strongly connected subgraphs and related geometric problems. In D. Bienstock and G. Nemhauser, editors, *Integer Programming and Combinatorial Optimization, 10th International IPCO Conference, New York, NY, USA*, volume 3064 of *Lecture Notes in Computer Science*, pages 152–162, Berlin, Heidelberg, New York, June 7-11 2004. Springer. (RRR 36-2003).

[BGKM03] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. On maximal frequent and minimal infrequent sets in binary matrices. *Ann. Math. Artif. Intell.*, 39(3):211–221, 2003.

[BI95] J. C. Bioch and T. Ibaraki. Complexity of identification and dualization of positive boolean functions. *Information and Computation*, 123(1):50–63, 1995.

[BLQ98] L.-F. Mun B. Liu, K. Wang and X.-Z. Qi. Using decision tree induction for discovering holes in data. In *PRICAI '98: Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence*, pages 182–193, London, UK, 1998. Springer-Verlag.

[CDL86] B. Chazelle, R. L. Drysdale, and D. T. Lee. Computing the largest empty rectangle. *SIAM Journal on Computing*, 15(1):300–315, 1986.

[Col87] C. J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, Inc., New York, NY, USA, 1987.

[EG95] T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.

[EGLM03] J. Edmonds, J. Gryz, D. Liang, and R. J. Miller. Mining for empty spaces in large data sets. *Theor. Comput. Sci.*, 296(3):435–452, 2003.

[Elb02a] Khaled M. Elbassioni. An algorithm for dualization in products of lattices and its applications. In *ESA*, pages 424–435, 2002.

[Elb02b] Khaled M. Elbassioni. On dualization in products of forests. In *STACS*, pages 142–153, 2002.

[Elb06] Khaled M. Elbassioni. Finding all minimal infrequent multi-dimensional intervals. In *LATIN*, pages 423–434, 2006.

[FK96] M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.

[GK99] V. Gurvich and L. Khachiyan. On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions. *Discrete Applied Mathematics*, 96-97(1):363–373, 1999.

[GMKT97] D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen. Data mining, hypergraph transversals, and machine learning (extended abstract). In *PODS '97: Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 209–216, New York, NY, USA, 1997. ACM Press.

[Gur75] V. Gurvich. Nash-solvability of games in pure strategies. *USSR Comput. Math and Math. Phys.*, 15(2):357–371, 1975.

[HCC93] J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. *IEEE Transactions on Knowledge and Data Engineering*, 05(1):29–40, 1993.

[HF95] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*, pages 420–431, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[KBE+07] L. Khachiyan, E. Boros, K. Elbassioni, V. Gurvich, and K. Makino. Dual-bounded generating problems: Efficient and inefficient points for discrete probability distributions and sparse boxes for multidimensional data. *Theor. Comput. Sci.*, 379(3):361–376, 2007.

[LKH97] B. Liu, L.-P. Ku, and Wynne Hsu. Discovering interesting holes in data. In *IJCAI (2)*, pages 930–935, 1997.

[LLK80] 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*, 9:558–565, 1980.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Orl90] M. Orlowski. A new algorithm for the largest empty rectangle problem. *Algorithmica*, 5(1):65–73, 1990.

[SA95] R. Srikant and R. Agrawal. Mining generalized association rules. In *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*, pages 407–419, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[SA96] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, NY, USA, 1996. ACM.