CLAUDE JARD

GUY-VINCENT JOURDAN

JEAN-XAVIER RAMPON

On-line computations of the ideal lattice of posets

ON-LINE COMPUTATIONS
OF THE IDEAL LATTICE OF POSETS (*)

by Claude JARD ([1]), Guy-Vincent JOURDAN ([1])
and Jean-Xavier RAMPON ([2])

Communicated by A. ARNOLD

Abstract. – *Partially ordered sets appear in many branches of computer science. In some cases, people are interested in manipulating the associated set of all linear extensions. This leads directly to the problem of building the ideal lattice of posets. Furthermore, we require that the poset be only known dynamically, element by element, in order to obtain an "on-line" construction of the lattice. This has practical interest notably in the area of "on-the-fly" testing (i.e. during the running time) of distributed systems. We introduce an algorithmic principle to compute this "on-line" construction of the lattice. From this, we first directly deduce an algorithm performing this construction under some wide assumptions and with a time complexity close to that of the best known algorithms. Secondly, we detail two efficient specific cases: the case in which the covering relations of the poset are known and the case in which the order elements are given with respect to a linear extension. In this last case, we obtain a time complexity comparable to that of the best known algorithms in spite of the fact of using a more general approach.*

Résumé. – *De nombreuses branches de l'informatique sont aujourd'hui concernées par la théorie des ensembles ordonnés. Lors de l'utilisation d'un ensemble ordonné comme outil de modélisation, un des objets fréquemment considéré est l'ensemble de ses extensions linéaires et par là même le treillis de ses idéaux. Les objets ainsi modélisés ont bien souvent un caractère dynamique important comme dans le cadre de la vérification des exécutions réparties. Nous nous sommes donc intéressés à la construction "on-line" du treillis des idéaux d'un ensemble ordonné (i.e. les éléments de l'ensemble ordonné ainsi que leurs relations ne sont connus que les uns après les autres). Nous introduisons une structure d'algorithme permettant la construction "on-line" de ce treillis. En un premier temps, nous en déduisons directement un algorithme qui, sous des conditions très générales, permet cette construction en une complexité temporelle proche de celles des meilleurs algorithmes connus. Puis nous détaillons deux cas particuliers qui nous permettent d'améliorer la complexité précédente, sous réserve de légères modifications de l'algorithme. Dans le premier cas étudié les relations de couvertures de l'ensemble ordonné sont connues, alors que dans le deuxième cas les éléments de l'ensemble ordonné sont donnés suivant une de ses extensions linéaires. Dans ce dernier cas nous obtenons une complexité temporelle comparable à celle des algorithmes connus tout en étant dans un cadre plus général.*

1. INTRODUCTION

1.1. Problem Statement

Designing "on-line" algorithms is a natural idea for processing dynamic and time-evolving information. In an "on-line" approach, data is only available piece by piece. With each newly added part, the algorithm carries on its computation, accumulates the result provided by the new piece of data, and adds it to the already computed results. We present herewith an "on-line" method for computing the covering digraph of the ideal lattice of posets.

This research has been motivated by practical problems in the context of parallel program debugging where the correct evaluation of global properties requires a careful analysis of the causal structure of the execution. The causal structure, induced by the message exchanges between processes in the distributed system, forms a partial order, as Lamport [18] remarked in 1978. Since then, "on-line" time-stamping algorithms which encode this causal poset have been proposed by Fidge [13] and Mattern [19]. As a consequence, numerous questions about distributed executions refer to the notions of linear extensions and order ideals (also called consistent cuts). At present, some testing methods are based on a kind of reachability analysis which builds the covering digraph of the ideal lattice of the causality relation (e.g. Cooper and Marzullo [9], Diehl [10], Diehl *et al.* [11] or Babaoglu and Raynal [2]). In this context, testing must be performed "on-the-fly", *i.e.* in parallel with the execution of the application under test.

In general, the problem of generating a set of ideals, say $I(P)$, of a given poset, say $\tilde{P} = (P, \leq_{\tilde{P}})$, occurs in combinatorial optimization as well as in operations research. One of the first efficient algorithms to solve this problem was proposed by Steiner in [20], and has a time complexity in $\mathcal{O}(|P| \, |I(P)|)$. This result was improved by Bordat in [5] with a time complexity in $\mathcal{O}(\omega(\tilde{P}) \, |I(P)|)$ (where $\omega(\tilde{P})$ is the width of $\tilde{P}$). Moreover, Bordat extends this generation to a construction of the covering digraph of the ideal lattice of $\tilde{P}$, say $\widetilde{I(P)}$, in the same time complexity, providing the computation of some poset invariants (*see* Bouchitté and Habib [7] for a survey on such invariants). Indeed, the covering digraph of the ideal lattice of $\tilde{P}$ allows the computation of the number of linear extensions of $\tilde{P}$ (the linear extensions of $\tilde{P}$ are in one-to-one correspondence with the maximal chains of the ideal lattice, *see* Bonnet and Pouzet [4]) as well as the jump number of $\tilde{P}$ (*see* Chaty and Chein [8]). A profound study of the combinatorial behavior of posets and lattices was achieved by Bordat in [6]. The drawback of these

algorithms is their intrinsic "off-line" structure. Motivated by our practical problems, we investigated an "on-line" approach in [10] and [11], and began a first generalization in [12]. Dealing with the recognition of distributive lattices, Habib and Nourine also investigated in [14] the construction of such covering digraphs. In order to perform this construction, they reused the algorithmic principle introduced in [10] and [11]. The case they considered, included in the second specific case that we present herein, shows equivalent time complexity. Note that an "on-line" approach was first considered by Cooper and Marzullo in [9] within the applied context of detecting global predicates in distributed computations. Their algorithm, however, requiring a level-by-level processing of the elements of the order, is based on a naive (and costly) enumeration.

This paper presents a complete generalization of [12] together with an improvement in time complexity. In the general case of a "subposet" hypothesis, our algorithm builds the covering digraph in

$$\mathcal{O}\left(\,|\,I\,(P)\,|\,\omega\,(\tilde{P}) + \,|\,P\,|^{2}\,\omega\,(\tilde{P})\right).$$

We also elucidate two particular cases which demonstrate a better performance. In the first case, the covering relations of the poset are known and we obtain a construction in $\mathcal{O}\left(\,|\,I\,(P)\,|\,\omega\,(\tilde{P})\right)$. In the second case, the orders elements are given with respect to a linear extension and we obtain a construction in $\mathcal{O}\left(\,|\,I\,(P)\,| + E_{\mathrm{Cov}\,(\widetilde{I\,(P)})} + \,|\,P\,|^{2}\,\omega\,(\tilde{P})\right)$ where $E_{\mathrm{Cov}\,(\widetilde{I\,(P)})}$ is the number of edges in the digraph.

The present paper is organized as follows: first, after giving definitions of specific terms, we illustrate our meaning of an "on-line" paradigm in subsection 1.3. In section 2, we deal with the structural aspects of the problem, showing the links between two main steps of our construction. Finally, in section 3, we focus on the algorithmic aspects, in one general example and in two specific examples.

1.2. Definitions

A set P associated with a partial order relation (*i.e.* an antisymmetric, transitive and reflexive binary relation on P) is called a *partially ordered set* (*poset* for short) and is denoted by $\tilde{P} = (P, \leq_{\tilde{P}})$. When $x \leq_{\tilde{P}} y$ and $x \neq y$ we simply write $x <_{\tilde{P}} y$.

Let x, $y \in P$: we say that x and y are *comparable* in $\tilde{P}$ when either $x \leq_{\tilde{P}} y$ or $y \leq_{\tilde{P}} x$. Otherwise, x and y are called *incomparable* in $\tilde{P}$. We say that x is *covered* by y in $\tilde{P}$, denoted by $x \prec_{\tilde{P}} y$, if $x <_{\tilde{P}} y$ and $\forall z \in P$,

$(x <_{\tilde{P}} z \leq_{\tilde{P}} y) \Rightarrow (z = y)$; then x is an *immediate predecessor* of y, and y is an *immediate successor* of x in $\tilde{P}$. The directed graph (*digraph* for short) associated with this *covering relation* is called the *covering digraph* of $\tilde{P}$ and is denoted by $\text{Cov}(\tilde{P}) = (P, E_P)$.

A *chain* (resp. an *antichain*) in $\tilde{P}$ is a subset A of P such that each pair of distinct elements of A is comparable (resp. incomparable) in $\tilde{P}$. The *width* of $\tilde{P}$, denoted $\omega(\tilde{P})$, is the maximum number of elements in an antichain in $\tilde{P}$. A linear extension of $\tilde{P}$ is a chain on P, say $\tilde{L} = (P, \leq_{\tilde{L}})$, which preserves $\tilde{P}$, that is: $x \leq_{\tilde{P}} y \Rightarrow x \leq_{\tilde{L}} y$.

Let A be a subset of P: the *subposet* of $\tilde{P}$ induced by A is the poset $\tilde{A} = (A, \leq_{\tilde{A}})$ where $\forall x, y \in A, (x \leq_{\tilde{A}} y) \Leftrightarrow (x \leq_{\tilde{P}} y)$. An element $x \in A$ is *maximal* (resp. *minimal*) in $\tilde{A}$ for $\tilde{P}$ if $\forall a \in A$, $(x \leq_{\tilde{P}} a) \Rightarrow (a = x)$ (resp. $(a \leq_{\tilde{P}} x) \Rightarrow (a = x)$). We denote the *predecessor* (resp. *successor*) set of A in $\tilde{P}$ by $\downarrow_{\tilde{P}}^{]} A = \{x \in P, \exists a \in A, x \leq_{\tilde{P}} a\}$ (resp. $\uparrow_{\tilde{P}}^{]} A = \{x \in P, \exists a \in A, a \leq_{\tilde{P}} x\}$). For a singleton $\{x\}$, we denote $\downarrow_{\tilde{P}}^{]} x = \downarrow_{\tilde{P}}^{]} \{x\}$, $\uparrow_{\tilde{P}}^{]} x = \uparrow_{\tilde{P}}^{]} \{x\}$, $\downarrow_{\tilde{P}}^{[} x = \downarrow_{\tilde{P}}^{]} x - \{x\}$, $\uparrow_{\tilde{P}}^{[} x = \uparrow_{\tilde{P}}^{]} x - \{x\}$. The *immediate predecessor* (resp. *successor*) set of $\{x\}$ in $\tilde{P}$ is denoted by $\downarrow_{\tilde{P}}^{im} x = \{y \in P, y \prec_{\tilde{P}} x\}$, (resp. $\uparrow_{\tilde{P}}^{im} x = \{y \in P, x \prec_{\tilde{P}} y\}$).

A set A is an *ideal* in $\tilde{P}$ if it is "closed" by predecessors, that is: $\downarrow_{\tilde{P}}^{]} A = A$. The set of all ideals in $\tilde{P}$ is denoted as $I(P)$. This set $I(P)$ ordered by inclusion forms a distributive lattice [1] denoted as $\widetilde{I(P)}$. Observe that the set of all maximal elements in an ideal forms an antichain. This provides a one-to-one correspondence between the ideals and the antichains of a poset. In the sequel, all the considered posets are finite.

1.3. Model and Paradigm

All complexity results are calculated using a RAM model with uniform cost criterion (*see* Aho, Hopcroft and Ullman [1]). Our "on-line" paradigm that we call the "*subposet*" hypothesis, is as follows: given a poset $\tilde{P} = (P, \leq_{\tilde{P}})$, a new element $x \notin P$ and two possibly empty subsets of P, say $\mathcal{P}(x)$ and $\mathcal{S}(x)$, we consider the poset $\tilde{P}' = (P', \leq_{\tilde{P}'})$ where $P' = P \cup \{x\}$ and where $\tilde{P}$ is the subposet of $\tilde{P}'$ induced by P. We also assume that

---

[1] For any I_1, I_2 in $I(P)$, $I_1 \cup I_2$ and $I_1 \cap I_2$ belongs to $I(P)$. Moreover, $I_1 \cup I_2$ (resp. $I_1 \cap I_2$) is the smallest (resp. greatest) element of $I(P)$ including (resp. included in) both I_1 and I_2. Thus $\widetilde{I(P)}$ is a lattice.

$\downarrow \stackrel{im}{\tilde{P}'} x \subseteq \mathcal{P}(x) \subseteq \downarrow \stackrel{[}{\tilde{P}'} x$ and $\uparrow \stackrel{im}{\tilde{P}'} x \subseteq \mathcal{S}(x) \subseteq \uparrow \stackrel{[}{\tilde{P}'} x$. That is, the poset "grows", vertex by vertex. Each incoming vertex x comes with its associated "environment" (that we denote $\mathcal{P}(x)$ and $\mathcal{S}(x)$) in $\tilde{P}'$ the poset currently being built. The "subposet" hypothesis is used in order to preserve the same poset "structure" throughout the different steps. This case is identical to those we encounter in our practical problems. In order to remain as comprehensive as possible, the only assumption we make is that the immediate neighbours of x in $\tilde{P}'$ are included in the environment (which is required to be able to correctly define $\tilde{P}'$).

The data associated with the subsets $\mathcal{P}(x)$ and $\mathcal{S}(x)$ are either direct references to the elements of P they represent or a "coding" (usually a label) of those elements. Accessing to the corresponding element of P takes a constant time in the first case and $\mathcal{O}(\log(|P|))$ time in the second case, by structuring the elements of P in a height-balanced tree.

In the present paper we consider that each elements of the subsets $\mathcal{P}(x)$ and $\mathcal{S}(x)$ is a direct references to the element of P it represents. After considering our "subposet" hypothesis and in order to improve our time complexity results, we will also strengthen this hypothesis with additional assumptions based on the two given subsets $\mathcal{P}(x)$ and $\mathcal{S}(x)$. In a first approach, which we call the *"covering relations"* hypothesis, we assume that both $\downarrow \stackrel{im}{\tilde{P}'} x = \mathcal{P}(x)$ and $\uparrow \stackrel{im}{\tilde{P}'} x = \mathcal{S}(x)$. That is, at each step we deal with the covering relations of the new element in the built poset. In a second approach, which we call the *"linear extension"* hypothesis, we assume that $\mathcal{S}(x) = \varnothing$. At each step, the order of the incoming elements which build the current poset respects one of its linear extensions.

2. COMPUTING $\mathrm{Cov}(I(P \widetilde{\cup \{x\}}))$ FROM $\mathrm{Cov}(\widetilde{I(P)})$

Let $\tilde{P} = (P, \leq_{\tilde{P}})$ be a poset and let $x \notin P$. Let $\tilde{P}' = (P', \leq_{\tilde{P}'})$ be a poset such that $P' = P \cup \{x\}$ and $\tilde{P}$ be a subposet of $\tilde{P}'$. Given $\mathcal{P}(x)$ and $\mathcal{S}(x)$, two subsets of P such that $\downarrow \stackrel{im}{\tilde{P}'} x \subseteq \mathcal{P}(x) \subseteq \downarrow \stackrel{[}{\tilde{P}'} x$ and $\uparrow \stackrel{im}{\tilde{P}'} x \subseteq \mathcal{S}(x) \subseteq \uparrow \stackrel{[}{\tilde{P}'} x$, our goal is to compute $\mathrm{Cov}(\widetilde{I(P')})$ knowing $\mathrm{Cov}(\widetilde{I(P)})$. In the sequel, and unless otherwise stated, $\tilde{P}$ and $\tilde{P}'$ denote such posets.

To that purpose, we first show the relationships between $I(P)$ and $I(P')$ and then we characterize the covering relations of $\widetilde{I(P')}$ using these of $\widetilde{I(P)}$.

This characterization, given in a constructive way, gives the underlying structure of our algorithm.

2.1. Sets Relationships

Our study of the relationships between $I(P)$ and $I(P')$ is based on the following observations: given an ideal I of $\tilde{P}$, if I contains an immediate successor of x then I is replaced by $I \cup \{x\}$ in $I(P')$. If there exists an immediate predecessor of x which is not in I, then I is an ideal of $\tilde{P}'$. In the other cases, both I and $I \cup \{x\}$ are ideals of $\tilde{P}'$.

Therefore, since $\uparrow_{\tilde{P}'}^{im} x \subseteq \mathcal{S}(x)$ and $\downarrow_{\tilde{P}'}^{im} x \subseteq \mathcal{P}(x)$, $I(P)$ can be divided into the three following groups:

- $A_1 = \{I \in I(P), \mathcal{P}(x) \not\subseteq I\}$
- $A_2 = \{I \in I(P), \mathcal{P}(x) \subseteq I \text{ and } \mathcal{S}(x) \cap I = \varnothing\}$
- $A_3 = \{I \in I(P), \mathcal{S}(x) \cap I \neq \varnothing\}$

Since $\tilde{P}$ is a subposet of $\tilde{P}'$ then any ideal of P which contains an element of $\mathcal{S}(x)$ contains all the elements of $\mathcal{P}(x)$. This allows us to state the following proposition:

PROPOSITION 1: *The non empty A_i's define a partition of $I(P)$.*

Remark 1:

- *Let $I \in A_i$, $J \in A_j$ such that $I \subseteq J$, then $i \leq j$.*
- $\downarrow_{\tilde{P}}^{1} (\downarrow_{\tilde{P}'}^{im} x) = \downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ *is the infimum of A_2 in $\widetilde{I(P)}$* (²).
- $P - \uparrow_{\tilde{P}}^{1} (\uparrow_{\tilde{P}'}^{im} x) = P - \uparrow_{\tilde{P}}^{1} (\mathcal{S}(x))$ *is the supremum of A_2 in $\widetilde{I(P)}$* (³).

Following our previous observations, in order to characterize $\widetilde{I(P')}$, it remains to introduce the two sets $A_2' = \{I \cup \{x\}, I \in A_2\}$ and $A_3' = \{I \cup \{x\}, I \in A_3\}$. Thus we obtain:

PROPOSITION 2: $I(P') = A_1 \cup A_2 \cup A_2' \cup A_3'$

Proof: Since $\tilde{P}$ is a subposet of $\tilde{P}'$, it is clear that

$$A_1 \cup A_2 \cup A_2' \cup A_3' \subseteq I(P')$$

and that $\forall I \in I(P')$, if $x \notin I$ then $I \in A_1 \cup A_2$. Assume that $x \in I$, if $\exists y \in I$ such that $x \leq_{\tilde{P}'} y$ then $I - \{x\} \in A_3$, else $I - \{x\} \in A_2$. ■


(²) The greatest element among the elements of $I(P)$ smaller than all the elements of A_2
(³) The least element among the elements of $I(P)$ bigger than all the elements of A_2.

2.2. Covering relations

The definition of $I(P')$ in terms of $I(P)$ as given in section 2.1, and the fact that poset ideals grow by adding one element at a time have as a straightforward consequence the fact that the covering relations of $\widetilde{I(P')}$ may be easily deduced from those of $\widetilde{I(P)}$. This means that, the covering relations of $\widetilde{I(P')}$ on $A_1 \cup A_2$ (resp. on $A_2' \cup A_3'$) are those of $I(P)$ on $A_1 \cup A_2$ (resp. on $A_2 \cup A_3$), and that each element of A_2 is covered by the element it "generates" in A_2'. These relations are illustrated in Figure 1 and are formally expressed in Theorem 1.

First of all, for the proof of this theorem, we recall the following well known lemma, stating that the ideals grow by adding one element at a time. We give a proof for the reader's benefit.
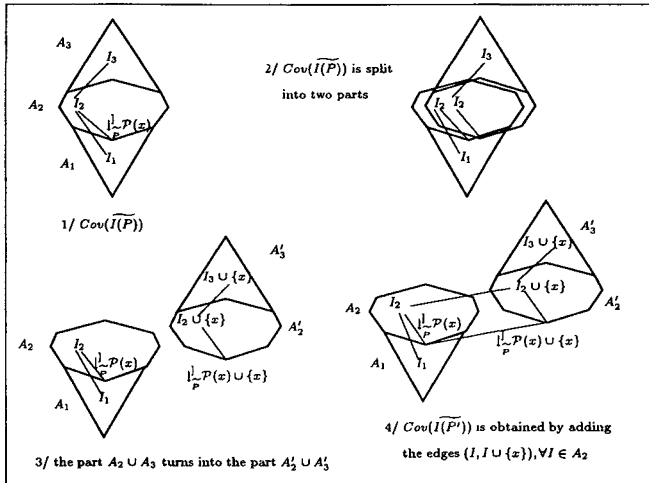
Figure 1. – $\mathrm{Cov}(\widetilde{I(P')})$ from $\mathrm{Cov}(\widetilde{I(P)})$

LEMMA 1: *Let $\tilde{Q}$ be a poset, then*

$$\forall I, J \in I(Q), \quad I \prec_{\widetilde{I(Q)}} J \Leftrightarrow I \subset J \quad \text{and} \quad |J - I| = 1.$$

Proof: Assume that $I \prec_{\widetilde{I(Q)}} J$, then $I \subset J$ and thus $|J - I| \geq 1$. If $|J - I| \geq 2$, let $x, y \in J - I$ such that $x \neq y$. W.l.o.g., assumes $y \not\leq_{\tilde{Q}} x$, then $I \prec_{\widetilde{I(Q)}} I \cup \downarrow_{\tilde{Q}}^{1} x \prec_{\widetilde{I(Q)}} J$ which contradicts $I \prec_{\widetilde{I(Q)}} J$. The converse is a consequence of the fact that $\forall I, J \in I(Q), I \prec_{\widetilde{I(Q)}} J \Leftrightarrow I \subset J$. ■

THEOREM 1: $\operatorname{Cov}(I(\widetilde{P'})) = (A_1 \cup A_2 \cup A'_2 \cup A'_3, E_{I(P')})$, *where* $E_{I(P')}$
satisfies:

- $((A_1 \cup A_2) \times (A_1 \cup A_2)) \cap E_{I(P')}$ *is isomorphic to*
  $((A_1 \cup A_2) \times (A_1 \cup A_2)) \cap E_{I(P)}$ *by the identity mapping.*
- $((A'_2 \cup A'_3) \times (A'_2 \cup A'_3)) \cap E_{I(P')}$ *is isomorphic to*
  $((A_2 \cup A_3) \times (A_2 \cup A_3)) \cap E_{I(P)}$ *by the one to one mapping*
  $\phi : A_2 \cup A_3 \to A'_2 \cup A'_3$, *such that* $\phi(I) = I \cup \{x\}$.
- $((A_1 \cup A_2) \times (A'_2 \cup A'_3)) \cap E_{I(P')} = \{(I, I \cup \{x\}), I \in A_2\}$.

Proof: $\forall I \in A_1 \cup A_2$, $\forall J \in A'_2 \cup A'_3$, $J \not\subseteq I$, so

$$((A_1 \cup A_2) \times (A_1 \cup A_2)) \cap E_{I(P')}$$

is exactly the set inclusion on $A_1 \cup A_2$. Moreover, $\forall I, J \in A'_2 \cup A'_3$, $I \subseteq J \Leftrightarrow I \cup \{x\} \subseteq J \cup \{x\}$, then $((A'_2 \cup A'_3) \times (A'_2 \cup A'_3)) \cap E_{I(P')}$ is exactly the set inclusion on $A'_2 \cup A'_3$.

For the remaining case, notice that Lemma 1 forbids any covering relations between elements of A_1 and elements of $A'_2 \cup A'_3$ (because if $I \in A'_2 \cup A'_3$ then $\{x\} \cup \mathcal{P}(x) \subseteq I$) as well as between elements of A_2 and elements of A'_3 (because if $I \in A'_3$ then $x \in I$ and $I \cap \mathcal{S}(x) \neq \varnothing$). According to Lemma 1, we obtain that $\forall I \in A_2$, $I \prec_{\widetilde{I(P')}} I \cup \{x\}$. It remains to show that I is the only element of A_2 covered by $I \cup \{x\}$, which is coherent with Lemma 1 since $\forall J \in A'_2$, $\forall I_1, I_2 \in A_2$ such that $I_1 \prec_{\widetilde{I(P')}} J$ and $I_2 \prec_{\widetilde{I(P')}} J$, we have $I_1 = J - \{x\}$ and $I_2 = J - \{x\}$. ∎

The decomposition of the covering relations of $I(\widetilde{P'})$ and the links with the covering relations of $I(\widetilde{P})$, as displayed in Theorem 1, give the underlying structure of our algorithm. Roughly speaking, this means: the determination of the set A_2, the duplication of this set and the computation of the new covering relations.

Before giving a complete description of the algorithm, we will first present it informally through the example illustrated in Figure 2. Notice that in this figure, an edge of the covering digraph of an ideal lattice is labeled by the set difference between the two ideals associated with its endpoints. Thus, the set corresponding to an ideal is the union of these labels in a path from the bottom to the vertex which represents it (for example, $I_5 = \{1, 2, 3\}$ while $I_9 = \{1, 2, 3, 4, 6\}$).

Consider the poset $\tilde{P}$ and assume that on one hand we have

$$\downarrow^{im}_{\tilde{P}'} x = \{3\} \subset \mathcal{P}(x) = \{2, 3\} \subset \downarrow^{[}_{\tilde{P}'} x = \{1, 2, 3\}$$

and on the other hand we have $\uparrow^{im}_{\tilde{P}'} x = \{5, 6\} = \mathcal{S}(x) \subseteq \uparrow^{[}_{\tilde{P}'} x = \{5, 6, 7\}$. The ideals not including $\mathcal{P}(x)$ are in $A_1 = \{I_1, I_2, I_3, I_4\}$, the ideals including $\mathcal{P}(x)$ but no element of $\mathcal{S}(x)$ are in $A_2 = \{I_5, I_6\}$, and the remaining ideals are in $A_3 = \{I_7, I_8, I_9, I_{10}, I_{11}\}$ (notice that each ideal of $A_2 \cup A_3$ is greater than or equal to $I_5 = \downarrow^{]}_{\tilde{P}} \mathcal{P}(x)$).

In order to obtain $\mathrm{Cov}(\widetilde{I(P')})$, we duplicate A_2 into $A'_2 = \{I'_5, I'_6\}$ and we transform A_3 in A'_3 where any I in $A_2 \cup A_3$ is now $I' = I \cup \{x\}$ in $A'_2 \cup A'_3$ (for example, $I'_9 = I_9 \cup \{x\} = \{1, 2, 3, 4, 6, x\}$). It remains to "delete" the covering relations between A_2 and the new A'_3 and to "add" the covering relations between A_2 and A'_2. That is, we have to remove the edges (I_5, I'_7) and (I_6, I'_8) and to add the edges (I_5, I'_5) and (I_6, I'_6).
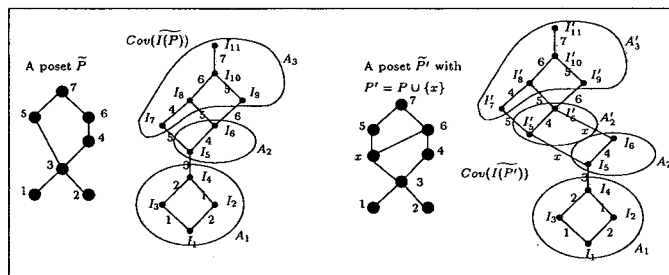

Figure 2. – The partition of the covering relations

3. COMPUTATION OF $\mathrm{Cov}(I(\widetilde{P \cup \{x\}}))$

As stated previously, in order to compute $\mathrm{Cov}(\widetilde{I(P')})$, our algorithm must be broken down into two main parts. In the first part, we find the infimum of $A_2 \cup A_3$ in $\widetilde{I(P)}$, and in the second part we duplicate A_2 and we make the corresponding updates on the covering relations. We present two different ways of obtaining the infimum of $A_2 \cup A_3$ (i.e. the ideal $\downarrow^{]}_{\tilde{P}} \mathcal{P}(x)$) and we give the whole algorithm in the general case. We then study some specific cases, which are the "covering relations" hypothesis and the "linear extension" hypothesis. The efficiency of these two given way to compute $\downarrow^{]}_{\tilde{P}} \mathcal{P}(x)$ differs on those particular cases. First of all, let us introduce our data structure assumptions.

3.1. Data Structure Assumptions

The new vertex x of P' is given with two lists, $\mathcal{P}(x)$ and $\mathcal{S}(x)$ (recall that $\downarrow \, {}^{im}_{P'} x \subseteq \mathcal{P}(x) \subseteq \downarrow \, {}^{\downharpoonleft}_{\tilde{P}'} x$ and $\uparrow \, {}^{im}_{P'} x \subseteq \mathcal{S}(x) \subseteq \uparrow \, {}^{\downharpoonleft}_{\tilde{P}'} x$) such that from any element in a list, we can access the element of P it represents in $\mathcal{O}(1)$. We also have direct access from any element $y \in P$ to its associated ideal $\downarrow \, {}^{\downharpoonleft}_{\tilde{P}} y$ in $\mathrm{Cov}\,(\widetilde{I(P)})$.

Two lists $\mathrm{succ}\,(I)$ and $\mathrm{pred}\,(I)$ are associated with each element I of $\mathrm{Cov}\,(\widetilde{I(P)})$. These are respectively the list of immediate successors and predecessors of I in $\mathrm{Cov}\,(\widetilde{I(P)})$. Moreover, with each element J in $\mathrm{succ}\,(I)$ (resp. in $\mathrm{pred}\,(I)$), an element is associated, denoted by $J - I$ (resp. by $I - J$), corresponding to the unique vertex of P which represents the set difference between the ideals I and J (*cf.* Lemma 1). The element $J - I$ (or $I - J$) is called the label of the edge IJ (or the edge JI). Using classical data structures, we access in $\mathcal{O}(1)$ from any element in $\mathrm{succ}\,(I)$ or in $\mathrm{pred}\,(I)$ to the element of $\mathrm{Cov}\,(\widetilde{I(P)})$ it represents and from any element of type $J - I$ to the element of P it represents.

3.2. The General Case

3.2.1. *Finding the Ideal* $\downarrow \, {}^{\downharpoonleft}_{\mathbf{P}} \mathcal{P}(x)$

We give two algorithms to find the ideal $\downarrow \, {}^{\downharpoonleft}_{\tilde{P}} \mathcal{P}(x)$. They are both based on the same idea: first, we go on a specific ideal in $\mathrm{Cov}\,(\widetilde{I(P)})$ and then we stay along a path in $\mathrm{Cov}\,(\widetilde{I(P)})$ until we reach $\downarrow \, {}^{\downharpoonleft}_{\tilde{P}} \mathcal{P}(x)$. These two algorithms are equivalent in the general case, but not in the specific cases. We first need the present Proposition 3, which, given a poset $\tilde{Q}$, describes the structure of maximal paths joining any two elements of $I(Q)$ to their supremum in $\widetilde{I(Q)}$. In $\mathrm{Cov}\,(\widetilde{I(Q)})$ each edge corresponding to a covering relation between two ideals I and J is labeled by the unique vertex of Q defined by $J - I$.

PROPOSITION 3: *Let* $I_1, I_2 \in I(Q)$, *then every path of* $\mathrm{Cov}\,(\widetilde{I(Q)})$ *from* I_1 *to* $I_1 \cup I_2$ *has the same length and the label set of its edges is exactly* $I_2 - I_1$. *Moreover, there exists at least one such path.*

Proof: Given any pair of ideals (I, I'), we know from Lemma 1 that if $I \subseteq I'$ then the label set of any path of $\mathrm{Cov}\, I(Q))$ from I to I' is exactly $I' - I$. ∎

The first algorithm starts the search from the ideal $\downarrow_{\tilde{P}}^{1} y$ for $y \in \mathcal{S}(x)$. Thus, we are on an ideal which contains $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ and then we "remove" $\downarrow_{\tilde{P}}^{1} y - \downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$. More precisely, the algorithm is the following:

Algorithm "FROM ABOVE"

(1) Mark all elements of $\mathcal{P}(x)$.

(2) If $\mathcal{S}(x) = \varnothing$ Then $I = $ the top of $\mathrm{Cov}(\widetilde{I(P)})$

   Else $I = \downarrow_{\tilde{P}}^{1} y$ for $y \in \mathcal{S}(x)$.

(3) While there exists an element J in $\mathrm{pred}(I)$ such that $I - J$ is not marked, $I := J$.

LEMMA 2: *The algorithm "FROM ABOVE" finds* $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ *in* $\mathrm{Cov}(\widetilde{I(P)})$ *in* $\mathcal{O}(|P| \omega(\tilde{P}))$.

Proof: The termination is obvious. After step (2), by definition I contains $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$. Let J be the ideal chosen in $\mathrm{pred}(I)$ inside the while loop.

Since $I - J$ is not marked, J contains $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ because I contains $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ and $\downarrow_{\tilde{P'}}^{im} x \subseteq \mathcal{P}(x)$. So at the end of the algorithm, we have $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x) \subseteq I$. Assume that $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x) \subset I$ and let $y \in I - \downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ such that y is maximal in $I - \downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$. Then by Lemma 1, we have $I - \{y\} \prec_{\widetilde{I(P)}} I$, which contradicts that $\forall J \in \mathrm{pred}(I)$, $I - J$ is marked.

For the time complexity, step (1) is performed in $\mathcal{O}(|P|)$. With Proposition 3, any path from the first ideal I to $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ contains exactly $I - \downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ edges, which is bounded by $|P|$. We conclude since $\forall I \in I(P)$, $|\mathrm{pred}(I)| \leq \omega(\tilde{P})$. ∎

The second algorithm to find $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ starts the search from an ideal $\downarrow_{\tilde{P}}^{1} y$ for $y \in \mathcal{P}(x)$. We are on an ideal which is contained in $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ and we have to "add" $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x) - \downarrow_{\tilde{P}}^{1} y$. This idea was first presented in [10] and [11].

Algorithm "FROM BELOW"

(1) Mark all elements of $\downarrow_{\tilde{P}}^{[} x$.

(2) If $\mathcal{P}(x) = \varnothing$ Then $\downarrow_{\tilde{P}}^{1} \mathcal{P}(x)$ is the bottom of $\mathrm{Cov}(\widetilde{I(P)})$

   Else Let $I = \downarrow_{\tilde{P}}^{1} y$ for $y \in \mathcal{P}(x)$.

(3)   While there exists an element J in $\mathrm{succ}(I)$ such that $J - I$ is marked, $I := J$.

   End If

LEMMA 3: *The algorithm "FROM BELOW" finds* $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$ *in* $\mathrm{Cov}\,(\widetilde{I(P)})$
in $\mathcal{O}\,(\,|\,P\,|\,\omega\,(\tilde{P}))$.

Proof: The termination is obvious. If $\mathcal{P}(x) = \varnothing$ then $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x) = \varnothing$.
Otherwise, by definition, $\downarrow^{\mathsf{l}}_{\tilde{P}} y \subseteq \downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$ for any $y \in \mathcal{P}(x)$. Since
$\downarrow^{\mathsf{l}}_{\tilde{P}} x = \downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$, inside the while loop we always have $I \subseteq \downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$. So at
the end of the algorithm we have $I \subseteq \downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$. Assume that $I \subset \downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$
and let $y \in \downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x) - I$ such that y is minimal in $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x) - I$. Then by
Lemma 1, we have $I \prec_{\widetilde{I(P)}} I \cup \{y\}$ which contradicts that $\forall\, J \in \mathrm{succ}\,(I)$
$J - I$ is not marked.

For the time complexity, step (1) can be performed in $\mathcal{O}\,(\,|\,P\,|\,\omega\,(\tilde{P}))$.
With proposition 3, any path from the first ideal I to $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$ contains
exactly $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x) - I$ edges, which is bounded by $|\,P\,|$. We conclude since
$\forall\, I \in I(P)$, $|\,\mathrm{succ}\,(I)\,| \leq \omega\,(\tilde{P})$. ∎ ■

3.2.2. *Duplication*

Knowing $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$ in $\mathrm{Cov}\,(\widetilde{I(P)})$, we want to compute $\mathrm{Cov}\,(\widetilde{I(P')})$. In
order to do so, and, according to Theorem 1, we have to duplicate A_2 into A_2'
so that elements of A_2 "keep" the edges between A_1 and A_2, while elements
of A_2' "keep" the edges between A_2 and A_3. In this way, A_3 "becomes" A_3',
because every path in $\mathrm{Cov}\,(\widetilde{I(P')})$ from the bottom to an element of A_3'
has an edge labeled by x. Our algorithm is based on a Breadth-First-Search
starting from $\downarrow^{\mathsf{l}}_{\tilde{P}} \mathcal{P}(x)$. It uses a list called *List*. We do not detail data
structure modifications. Just note that the instruction *Duplicate K into K'*
means the creation of a vertex K', the addition of K' in $\mathrm{succ}\,(K)$ with x as a
label for the edge $K\,K'$ and the addition of K in $\mathrm{pred}\,(K')$ with x as a label
for the edge $K'\,K$. K' is always the duplication of K. Finally, the *mark* of
steps (1) and (6) is on vertices of P while the *mark* of step (7) is on ideals.

LEMMA 4: *The algorithm "DUPLICATION" computes* $\mathrm{Cov}\,(\widetilde{I(P')})$ *in*
$\mathcal{O}\,((\,|\,I(P')\,| - |\,I(P)\,|\,)\,\omega\,(\tilde{P}) + |\,\mathcal{S}(x)\,|)$.

Proof: The termination is clear, since we make a classical Breadth-First-
Search and since elements are put in *List* only once. The correctness is a
consequence of Theorem 1.

Algorithm "DUPLICATION"

(1) Mark all elements of $\mathcal{S}(x)$.

(2) Let $I = \downarrow_{\tilde{P}}^{]} \mathcal{P}(x)$; Duplicate I into I'; Put I in *List*.

(3) While *List* $\neq \varnothing$

    (4) Take I in *List*; Delete I from *List*

    (5) Forall $J \in \text{succ}(I)$

        (6) If $J - I$ is not marked Then

            (7) If J is not marked Then

                Mark J; Duplicate J into J'; Put J in *List*.

            End If

        (8) Add J' in $\text{succ}(I')$ and I' in $\text{pred}(J')$, with label $I - J$

        Else

        (9) Add J in $\text{succ}(I')$ and I' in $\text{pred}(J)$, with label $I - J$

        (10) remove edge IJ

        End If

    End Forall

  End While

Clearly, step (1) is in $\mathcal{O}(|\mathcal{S}(x)|)$. All steps except steps (3) and (5) are performed in $\mathcal{O}(1)$. For one loop of the while of step (3) we have at most $\omega(\tilde{P})$ loops of the Forall of step (5). Indeed, $\forall I \in I(P)$, $|\text{succ}(I)| \leq \omega(\tilde{P})$. Since *List* contains only elements of A_2 and only once each, and since $|I(P')| - |I(P)| = |A_2|$, we get the announced time complexity. ∎

3.2.3. *Whole Computation*

When we perform the algorithm "FROM BELOW" (or "FROM ABOVE") and then the algorithm "DUPLICATION", we obtain $\text{Cov}(\widetilde{I(P')})$ from $\text{Cov}(\widetilde{I(P)})$. So using Lemma 3 (or Lemma 2) and Lemma 4, we deduce:

THEOREM 2: *Let $\tilde{P} = (P, \leq_{\tilde{P}})$ be a poset and $\tilde{P}' = (P', \leq_{\tilde{P}'})$ be a poset such that $P' = P \cup \{x\}$, $x \notin P$ and $\tilde{P}$ is a subposet of $\tilde{P}'$. Given $\mathcal{P}(x)$ and $\mathcal{S}(x)$ two subsets of P such that $\downarrow_{\tilde{P}'}^{im} x \subseteq \mathcal{P}(x) \subseteq \downarrow_{\tilde{P}'}^{[} x$ and $\uparrow_{\tilde{P}'}^{im} x \subseteq \mathcal{S}(x) \subseteq \uparrow_{\tilde{P}'}^{[} x$, we can compute $\text{Cov}(\widetilde{I(P')})$ knowing $\text{Cov}(\widetilde{I(P)})$ in*

$$\mathcal{O}((|I(P')| - |I(P)|)\omega(\tilde{P}) + |P|\omega(\tilde{P}))$$

Remark 2: It is clear that all the data structures can be maintained and restored without increasing the announced time complexities.

As a consequence of this theorem, we are able to achieve the computation of the covering digraph of the ideal lattice of a poset given any sequence of vertices w.r.t our assumptions. We just have to apply our construction $|P|$ times in order to obtain:

THEOREM 3: *Let $\tilde{P}$ be a poset. The covering digraph of $\widetilde{I(P)}$ can be computed "on-line" in $\mathcal{O}\left(\,|\,I(P)\,|\,\omega\,(\tilde{P}) + |\,P\,|^2\,\omega\,(\tilde{P})\right)$.*

3.3. The "Covering Relations" Hypothesis

We assume that $\mathcal{S}(x) = \uparrow_{\tilde{P}'}^{im} x$ and $\mathcal{P}(x) = \downarrow_{\tilde{P}'}^{im} x$: that is, we know the covering relations of x in $\tilde{P}'$. Under this hypothesis, we are able to improve the first part of our algorithm. Indeed, using the algorithm "FROM ABOVE", we begin the search from an ideal $\downarrow_{\tilde{P}}^{]} y$, where $y \in \mathcal{S}(x)$. But now, $\mathcal{S}(x) = \uparrow_{\tilde{P}'}^{im} x$ and, since $\tilde{P}$ is a subposet of $\tilde{P}'$, we have the following property:

PROPERTY 1: *For $y \in \uparrow_{\tilde{P}'}^{im} x$, $\forall\, I \in I(P)$ such that $\downarrow_{\tilde{P}}^{]} \mathcal{P}(x) \subseteq I \subset \downarrow_{\tilde{P}}^{]} y$, $I \in A_2$.*

This property allows us to deduce that the "FROM ABOVE" algorithm finds $\downarrow_{\tilde{P}}^{]} \mathcal{P}(x)$ in $\mathcal{O}\left((\,|\,A_2\,| + 1)\,\omega\,(\tilde{P})\right)$, that is

$$\mathcal{O}\left((|\,I(P')\,| - |\,I(P)\,| + 1)\,\omega\,(\tilde{P})\right).$$

So together with the "DUPLICATION" algorithm, we can compute $\mathrm{Cov}\,(\widetilde{I(P')})$ from $\mathrm{Cov}\,(\widetilde{I(P)})$ in $\mathcal{O}\left((|\,I(P')\,| - |\,I(P)\,| + 1)\,\omega\,(\tilde{P})\right)$ (note that under this hypothesis, $|\,\mathcal{S}(x)\,| \leq \omega\,(\tilde{P})$), thus we obtain.

THEOREM 4: *Let $\tilde{P}$ be a poset. Under the "covering relations" hypothesis, the covering digraph of $\widetilde{I(P)}$ can be computed "on-line" in $\mathcal{O}\left(\,|\,I(P)\,|\,\omega\,(\tilde{P})\right)$.*

3.4. The "Linear Extension" Hypothesis

We assume that the poset is known under a linear extension, that is, the new element x is maximal in $\tilde{P}'$. So we only have a list $\mathcal{P}(x)$, such that $\downarrow_{\tilde{P}'}^{im} x \subseteq \mathcal{P}(x) \subseteq \downarrow_{\tilde{P}'}^{[} x$. Under this hypothesis, we are able to improve the second part of our algorithm. Indeed, the set A_3 is always empty, thus if $E_{\mathrm{Cov}\,(\widetilde{I(P)})}$ denotes the number of edges of $\mathrm{Cov}\,(\widetilde{I(P)})$, the "DUPLICATION" algorithm

runs now in $\mathcal{O}\left(\,\left|I\left(P'\right)\right| + E_{\mathrm{Cov}\left(\widetilde{I\left(P'\right)}\right)} - \left|I\left(P\right)\right| - E_{\mathrm{Cov}\left(\widetilde{I\left(P\right)}\right)}\right)$. Thus we obtain:

THEOREM 5: *Let $\tilde{P}$ be a poset. Under the "linear extension" hypothesis, the covering digraph of $\widetilde{I\left(P\right)}$ can be computed "on-line" in*

$$\mathcal{O}\left(\,\left|I\left(P\right)\right| + E_{\mathrm{Cov}\left(\widetilde{I\left(P\right)}\right)} + \left|P\right|^2 \omega\left(\tilde{P}\right)\right).$$

Remark 3: On forest posets (*i.e.* a poset such that each element has no more than one immediate predecessor), under both the "linear extension" and the "covering relations" hypothesis, our algorithm is optimal; that is, it generates $\mathrm{Cov}\left(\widetilde{I\left(P\right)}\right)$ in $\mathcal{O}\left(\,\left|I\left(P\right)\right| + E_{\mathrm{Cov}\left(\widetilde{I\left(P\right)}\right)}\right)$. Ideals of such posets are useful for network partitioning: *see* Koda and Ruskey [17] for an algorithm which generates all ideals in a gray code manner ([4]) in $\mathcal{O}\left(\,\left|I\left(P\right)\right|\right)$.

4. CONCLUSION

Motivated by the practical problem of checking global predicates in distributed systems, we were interested in building the covering digraph of the ideal lattice of finite posets. Thus, a preliminary version of the algorithm under the "subposet" hypothesis and one under the "linear extension" hypothesis have already been implemented. The complete integration of these procedures into a software verification tool for analyzing traces of distributed systems is in progress, as a part of the French national project "Trace".

To call attention to the accuracy of the algorithmic principle we have discussed throughout this paper, we are going to analyse more precisely the time complexity yielded by each step of the proposed algorithms. Notice that any algorithm building the covering digraph of the ideal lattice of finite posets has a lower time complexity bound in $\Omega\left(\,\left|I\left(P\right)\right| + E_{\mathrm{Cov}\left(\widetilde{I\left(P\right)}\right)}\right)$.

Under the "linear extension" hypothesis, and in order to reach the lower bound, the time complexity drawback of our algorithm comes from the search of the infimum of A_2. It is then easy to see that using the "FROM ABOVE" algorithm, we obtain a time complexity in $\mathcal{O}\left(\,\left|I\left(P\right)\right| + E_{\mathrm{Cov}\left(\widetilde{I\left(P\right)}\right)} + \sum_{x \in P} \sum_{I \in B\left(x\right)} \delta^-\left(I\right)\right)$ where $B\left(x\right)$ is the set of

---

([4]) That is according to a Hamiltonian path in the Hasse diagram of the lattice.

elements in a path from the ideal $\downarrow_{\tilde{P}}^{[} x$ to the top of the current lattice (when x is the incoming vertex) which has immediate predecessors not including $\downarrow_{\tilde{P}}^{[} x$ and where $\delta^-(I)$ is the number of such predecessors. Although we have to notice that even if there are $\Omega(|P|)$ elements x such that for at least one I in $B(x)$ we have $\delta^-(I)) \in \Omega(w(P))$, then the sum $\sum_{x \in P} \sum_{I \in B(x)} \delta^-(I))$ belongs to $\Omega(|P|w(P))$ and can be the main factor of our time complexity ([5]). Thus, a natural question is to give a characterization of posets such that

$$\sum_{x \in P} \sum_{I \in B(x)} \delta^-(I)) \in \mathcal{O}(|I(P)| + E_{\text{Cov}(\widetilde{I(P)})}).$$

Under the "covering relations" hypothesis, the proposed algorithm gets one more additional time complexity factor. Indeed, in the duplication part we have to perform a search algorithm and the costly points appear when we are on an element of A_2 having immediate successors which contain at least an element of the set $S(x)$. Thus, we have to add to the previous final time complexity, a factor in $\mathcal{O}(\sum_{x \in P} \sum_{I \in C(x)} \delta^+(I))$ where $C(x)$ is the set of elements in the set A_2 of the current lattice (when x is the incoming vertex) which has immediate successors containing at least one element of the successors set of x and where $\delta^+(I)$ is the number of such successors. The characterization of posets such that $\sum_{x \in P} \sum_{I \in C(x)} \delta^+(I) \in \mathcal{O}(|I(P)| + E_{\text{Cov}(\widetilde{I(P)})})$ has then to be related to the previous question, in order to reach the lower bound.

After all our algorithmic principle seems to be a good candidate for obtaining an optimal "on-line" construction of the covering digraph of the ideal lattice of posets. In any case, it has already had some fruitful repercussion because it allows us to get the same time complexity than the best known "off-line" algorithm (*i.e.* the whole poset is known in advance) building this digraph.

Apart from our practical motivations we are also interested in abstractions (subposets) of this ideal lattice. More particularly, we have already followed a similar algorithmic principle in [16] for building the lattice of maximal antichains for inclusion. Indeed, as suggestion by Janicki and Koutny [15], such antichains can be associated with points on the observer time scale.

([5]) Let $\tilde{P}$ be the poset obtained by the series composition of an antichain of size k with a total order of size p. It suffices to take $k = \log_2(\log_2(p))$.

Our conviction is that the study of ideal lattice abstractions is one of the relevant challenges for parallel program debugging.

REFERENCES

1. A. V. AHO, J. E. HOPCROFT and J. D. ULLMAN, *The design and Analysis of Computer Algorithms*, Addison-Wesley Series in Computer Science and Information Processing, 1974.

2. O. BABAOGLU and M. RAYNAL, *Sequence-Based Global Predicates for Distributed Computations: Definitions and Detection Algorithms*, IRISA research report No. 729, Mai 1993.

3. G. BIRKHOFF, Rings of Sets, *Duke Math.*, 1937, J-3, pp. 311-316.

4. R. BONNET and M. POUZET, Extensions et stratifications d'ensembles dispersés, *C. R. Acad. Sci. Paris*, 1969, t. 268, Série A, pp. 1512-1515.

5. J. P. BORDAT, Calcul des idéaux d'un ordoné fini, *Recherche opérationnelle/Operations Research*, 1991, 25, n° 4, pp. 265 à 275.

6. J. P. BORDAT, *Sur l'algorithme combinatoire d'ordres finis*, Doctorat d'État es Sciences USTL Montpellier, 1992.

7. V. BOUCHITTÉ and M. HABIB, *The Calculation of Invariants of Ordered Sets*, I. Rival (ed.), Algorithms and Order, 231-279, NATO Series C, 255, Kluwer Academic Publishers, 1989.

8. G. CHATY and M. CHEIN, Ordered Matchings and Matchings without Alternating Cycles in Bipartite Graphs, *Utilitas Mathematica*, 1979, 16, pp. 183-187.

9. R. COOPER and K. MARZULLO, Consistent Detection of Global Predicates, In Proc. ACM/ONR *Workshop on Parallel and Distributed Debugging*, pp. 163-173, Santa Cruz, California, May 1991.

10. C. DIEHL, Analyse de la relation de causalité dans les exécutions réparties, Thèse, 14 Septembre 1992, Université de Rennes 1.

11. C. DIEHL, C. JARD and J. X. RAMPON, Reachability Analysis on Distributed Executions, TAPSOFT'93: Theory and Practice of Software Development, in *Lecture Notes in Computer Science*, 1993, No. 668, Springer-Verlag, pp. 629-643.

12. C. DIEHL, C. JARD and J. X. RAMPON, *Computing On-Line the Covering Graph of the Ideal Lattice of Posets*, IRISA research report No. 703, February 1993.

13. C. FIDGE, *Timestamps in Message Passing Systems that Preserve the Partial Ordering*, In Proc. 11th Australian Computer Science Conference, 55-66, February 1988.

14. M. HABIB and L. NOURINE, Linear Time Recognition Algorithm for Distributive Lattices, Order 11: pp. 197-210, 1994.

15. R. JANICKI and M. KOUNTY, Structure of Concurrency, *Theoretical Computer Science*, 1993, *112*, pp. 5-52.

16. G. V. JOURDAN, J. X. RAMPON and C. JARD, Computing On-Line the Maximal Antichain Lattice of Posets, *Order* 11: pp. 197-210, 1994.

17. Y. KODA and F. RUSKEY, A Gray Code for the Ideals of a Forest Poset, *Journal of Algorithms*, 1993, *15*, pp. 324-340.

18. L. LAMPORT, Time, Clocks and the Ordering of Events in a Distributed System, *Communications of the ACM*, July 1978, 21 (7), pp. 558-565.

19. F. MATTERN, *Virtual Time and Global States of Distributed Systems*, In Cosnard, Quinton, Raynal and Robert, editors, Proc. Int. Workshop on Parallel and Distributed Algorithms, Bonas France October 1988, North-Holland, 1989.

20. G. STEINER, An Algorithm to Generate the Ideals of a Partial Order, *Operations Research Letters*, 1986, *5*, No. 6, pp. 317-320.