Linear Time Recognition Algorithm for Distributive Lattices

Michel HABIB and Lhouari NOURINE
Département d'**I**nformatique **F**ondamentale,
LIRMM, Université Montpellier II — CNRS UMR C09928: 161 rue Ada,
34392 Montpellier Cedex 5, FRANCE.
(email: (habib,nourine)@lirmm.fr)

September 30, 1998

Abstract

This paper studies some algorithmic problems on distributive lattices. We use a partition theorem of distributive lattices to obtain linear time-complexity algorithms (Theorems 3 and 4) for recognizing and computing the transitive closure of graphs whose transitive closure is a distributive lattice. As a by-product we obtain a new algorithm for building the lattice of ideals of a given partial order. These results show in that example of distributive lattices that efficient algorithms are closely related to structural properties. This approach can be generalized to other classes of orders.

Keywords: partially ordered set (poset), distributive lattice, acyclic graphs, transitive closure or reduction, recognition algorithm.

1 Preliminaries

Recently there was a growing interest for algorithms on lattices, this is mainly due to the many applications of lattice theory in computer science. For example the use of type lattices for programming languages as in Aït-Kaci *et al* [2] and for multiple inheritance hierarchies in Caseau [6], the concept lattice theory developed by Wille for knowledge representation [17, 18], the use of the events lattice in the analysis of distributed computations Mattern [14], Charron-Bost [7]. Perhaps these distributive lattices are very important in computer science because they can be easily associated with set families.

When dealing with algorithms on orders, one has to precise the input involved, since the various directed acyclic graphs that can represent an order are not equivalent from the point of view of complexity. Indeed, one may use the transitive closure graph (ie. $G = (X, E_{tc})$ s.t. $xy \in E$ iff $x < y$) or the covering graph (ie. $D = (X, E_{tr})$ s.t. $xy \in E$ iff y covers x) or any graph H whose transitive closure is $G = (X, E_{tc})$. The algorithm that compute D or G from H is not linear in the general case, see [1].

From the original definition of distributive lattices, one can produce a naive recognition algorithm running in $O(n^3)$, when the input is a covering graph $D = (X, E_{tr})$ with $|X| = n$. Under the same assumption, recently Bordat [5] has proposed an $O(|E_{tr}|loglog\Delta)$ algorithm based upon traversals of directed graphs, where Δ is the maximal out-degree of the input graph which is a covering graph. When the input is any graph whose transitive closure is the lattice, Bordat's algorithm turns out to be in $O(n^2)$.

We present in this paper two algorithms to recognize if a graph represent a distributive lattice. The input for our first distributive lattice recognition algorithm is supposed to be a covering graph $D = (X, E_{tr})$. This algorithm runs in $O(|E_{tr}|)$. In this case Karpushov (a russian paper) has independently obtained the same result. Then we present an optimal algorithm to compute the transitive closure graph of a distributive lattice represented by any graph H. We give after a second recognition algorithm which accepts as input any directed acyclic graph whose transitive closure is the distributive lattice. This algorithm runs in $O(|E_{tc}|)$. Our main result emphasizes the algorithmic applications of a well-known representation theorem for distributive lattices [3, 4, 8]. We also use part of the knowledge on algorithmic order theory slowly gathered on basic problems such as transitive (closure and reduction) calculations on acyclic graphs [5, 12, 10]. These results show in that example of distributive lattices that efficient algorithms are closely related to structural properties. This approach can be generalized to other classes of orders.

An order P is said to be *graded* if all maximal chains between any two elements of P have the same length. It should be noticed that the covering graph of graded orders with a unique minimal element and unique maximal element, can be obtained in linear time using the rank function (see for example [10]).

We denote by $x \wedge y$ (resp. $x \vee y$) the join (resp. meet) of x and y. Since we are dealing with algorithms on lattice, the lattices we consider are supposed to be finite.

Let L be a lattice. We denote the set of join-irreducible elements of L by $J(L)$ and the set of meet-irreducible elements by $M(L)$. If the lattice L is represented by its covering graph then join-irreducible (resp. meet-irreducible) elements are exactly the vertices with in-degree (resp. out-degree) one.

Remark : If the input covering graph is already known to be a lattice. Then using a nice characterization of Avann [A 61]: *L is distributive if and only if L is graded and $|M(L)| = |J(L)|$ =the height of L*, the distributivity property can be tested in linear time-complexity. Since testing if an order is graded and computing the number of join-irreducible and meet-irreducible can be done in linear time. But the best algorithm known for recognizing graphs which represent lattice is still in $O(n^{5/2})$, which unfortunately is not linear [9].

2 Partition theorem for distributive lattices

Let us begin with a well-known representation result for distributive lattices. Let L be a finite distributive lattice. Then the map $\varphi : m \rightarrow \wedge(L\backslash \downarrow m)$ is an order isomorphism from $(M(L), \leq_L)$ into $(J(L), \leq_L)$. The reverse map is given by $\varphi^{-1} : j \rightarrow \vee(L\backslash \uparrow j)$. The Figure 1 shows a geometric interpretation of the isomorphism φ.

This result play a crucial role in the study of distributive lattices. Indeed, it has been recently used to characterize distributivity of maximal antichains lattices of an order [15]. The nice property coming out of it, is the partitioning of distributive lattices into sublattices, which yields immediately recursive algorithms. Without loss of generality, we will show that any distributive lattice can be divided into two sublattices, such that the small one is isomorphic to a sublattice of the remaining part.

Let $m \in M(L)$ and $\varphi(m) = j \in J(L)$, we denote by m^* (resp. j_*) the unique element of L that covers m (resp. is covered by j). We have $j \vee m = m^*$ and $j \wedge m = j_*$. Given two comparable elements z and t, we call the sublattice $[z, t] = \{x \in L \text{ s.t. } x \leq t \text{ and } x \geq z\}$ an *interval*.

Proposition 1 *If $j \in J(L)$ is minimal not above $m \in M(L)$. Then $\Psi_m(z) = z \vee j$ provides an isomorphism between the intervals $[j_*, m]$ and $[j, m^*]$. Note that $\Psi_m(z)$ is the unique upper cover of z in L, out of $[j_*, m]$.*

Proof: Let $z \in [j_*, m]$, therefore $j_* \leq_L z \leq_L m$ and this implies $j_* \vee j \leq_L z \vee j \leq_L m \vee j$, thus $j \leq_L \Psi_m(z) \leq_L m^*$, and $\Psi_m(z) \in [j, m^*]$.

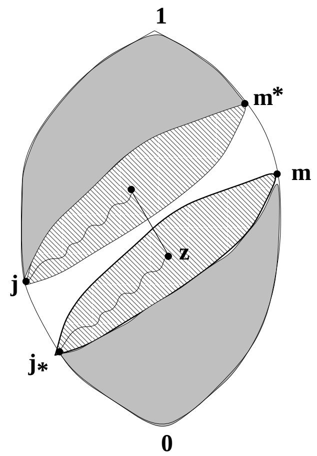

Figure 1: Isomorphism between $(M(L), \leq_L)$ and $(J(L), \leq_L)$

It is clear that Ψ_m is monotone, to show the isomorphism we just have to notice that $j \vee z = t$ is equivalent to $z = t \wedge m$.

Let us suppose that there exists $t' \in L - [j_*, m]$, $t' \neq \Psi_m(z)$ that covers z. Since $j \leq_L t'$, and $z \vee j = \Psi_m(z)$, necessarily $\Psi_m(z) \leq_L t'$. Thus since t' covers z, we have $t = \Psi_m(z)$. $\square$

Let us now state the partition theorem of a distributive lattice into sublattices that we need for our algorithms.

Theorem 1 *Let L be a distributive lattice and $\tau = \{m_1, ..., m_n\}$ a linear extension of $M(L)$. Then*

(i) $[j_1, \top] = L - [j_{1*}, m_1]$ *is a distributive lattice.*

(ii) $M([j_1, \top]) = M(L) - \{m_1\}$.

(iii) $\tau_2 = (m_2, ..., m_n)$ *is a linear extension of* $M([j_1, \top])$.

We denote by L_i the interval $[j_i, m_i]$ corresponding to i^{th} element in τ. The isomorphism Ψ_{mi} is noted Ψ_i related to the interval $[j_i*, m_i]$.*

By induction, we obtain a partition of L into disjoint sublattices $\{L_1, ..., L_n, L_{n+1}\}$ where L_{n+1} is reduced to the top element $\top$.

Proof:

(i) trivial.

(ii)(iii) The removal of the sublattice $L_1 = [j_1*, m_1]$ does not change the number of meet-irreducible elements in $[j_1, \top]$, since it changes only the in-degree. Furthermore the induced order of meet-irreducible elements is unchanged. Thus $\tau_2 = (m_2, ..., m_n)$ is a linear extension of $M([j_1, \top])$.

□

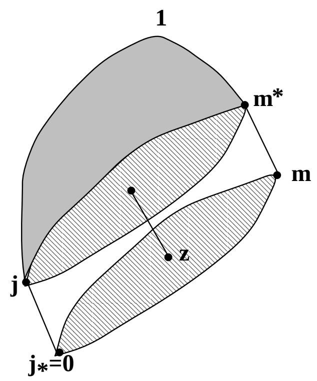

Figure 2: Partition scheme

Remark : The partition yielded by this process is not unique, since it depends on the linear extension taken on $M(L)$. One can notice that if such a partition was unique, a linear time-complexity algorithm for isomorphism testing between two distributive lattices could easily be derived. In fact the number of partitions is exactly the number of linear extensions of $M(L)$.

3 Recognition of graphs which are the covering graphs of distributive lattices

The main idea of the algorithm can be easily explained as follows:

1. Let m be a minimal element of $M(L)$;
2. Check the isomorphism between $[\perp = j_*, m]$ and $[j, m^*]$;
3. Goto 1. with $L - [\perp, m]$.

First we need some specific graph notations in order to adapt the above idea to an input graph.

Let $G = (X, E)$ be a covering graph, having a unique minimal (resp. maximal) element denoted by $\perp$ (resp. $\top$).

Let us define $M(G) = \{x \in X, d^+_G(x) = 1\}$ and $J(G) = \{x \in X, d^-_G(x) = 1\}$ where $d^+_G(x)$ (resp. $d^-_G(x)$) is the out-degree (resp. in-degree) in the covering graph. Notice that if G is a covering graph of a lattice then these sets are exactly the meet-irreducible and join-irreducible elements.

For $m \in Min(M(G))$, let m^* be the unique upper cover of m, we say that G is *m-regular* if the following two conditions are satisfied:

(i) $Min(G - [\perp, m]) = \{j\}$

(ii) *There exists an order isomorphism θ from $[\perp, m]$ onto $[j, m^*]$ such that $\forall z \in [\perp, m]$, $(z, \theta(z)) \in E$, and $\theta(z)$ is the unique upper cover of z out of $[\perp, m]$.*

Theorem 2 *Let $G = (X, E)$ and $m \in Min(M(G))$, if G is m-regular then: G is the covering graph of a distributive lattice if and only if $G' = G - [\perp, m]$ is the covering graph of a distributive lattice.*

Proof:

$\Rightarrow$ Suppose that G is the covering graph of a distributive lattice L, then we can apply the above results, in particular Proposition 1, therefore $m \in Min(M(L))$ and G' is the covering graph of the distributive lattice $[j, \top]$.

$\Leftarrow$ Conversely, suppose that G' is the covering graph of some distributive lattice L'. Since G is m-regular then $[\perp, m]$ is a distributive sublattice (ie. isomorphic to $[j, m^*]$). Therefore, if G is not the covering graph of some distributive lattice, this may be possible from some obstruction in between $[\perp, m]$ and G'. That is some lattice N_5 or M_3 (see Figure 3). Let us consider these cases, first we suppose that G contains a subgraph isomorphic to M_3.

At least $z \in [\perp, m]$, necessarily also $a \in [\perp, m]$. Therefore it is not possible that both t and u belongs to $X' = X - [\perp, m]$, since this would imply that $t \wedge u = a \in X'$. The only case that remains, $u \in [\perp, m]$, $t \in X'$. But then $z \vee u = b$ must be in $[\perp, m]$, a contradiction.

Similarly if G contains a subgraph isomorphic to N_5, with $z \in [\perp, m]$, therefore $a \in [\perp, m]$.

a) If $t, u \in X'$. Using the isomorphism θ, let us consider the set $\{\theta(a), \theta(z), u, t, b\}$. Since z and u have only one upper cover out of $[\perp, m]$, we have $\theta(a) < u$ and $\theta(z) < b$ and so this gives an N_5 in L'.

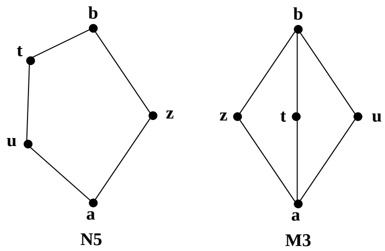

Figure 3: N_5 and M_3 forbidden sublattices in distributive lattice

b) If $t \in [\perp, m]$ and $u \in X'$. The set $\{\theta(a), \theta(z), u, \theta(t), b\}$ yields some N_5 in L'.

The last case we have to consider is $z \in X'$, and $u, a \in [\perp, m]$.

Similarly if $t \in X'$, then $\{\theta(a), z, \theta(u), t, b\}$ is a N_5 in L' and if $t \in [\perp, m]$ we consider $\{\theta(a), z, \theta(u), \theta(t), b\}$.

□

Algorithm 1: Distributive Lattice Recognition

Data : A covering graph denoted by $G = (X, E_{tr})$, represented by its adjacency lists.
Result: Is G the covering graph of a distributive lattice?
begin
  Compute $M(G)$ and a linear extension τ of $M(G)$;
  Verify(G);
  if $G = 2$ *(with* $2 = \{\{0,1\}, 0 < 1\}$*)* **then**
    G is a covering graph of a distributive lattice;
  end
  else
    choose $m \in Min(M(G))$
  end
  if G *is m-regular* **then**
    Verify($G - [\perp, m]$);
  end
  else
    G is not a covering graph of a distributive lattice;
  end
end

Theorem 3 *Algorithm 1 recognizes if a graph $G = (X, E)$ is a covering graph of a distributive lattice in $O(|E_{tr}|)$ time complexity.*

Proof:

The correctness of the algorithm comes directly from the Theorem 2.

Step(1), the computation of $M(G)$ can be done in linear time, using some traversing algorithm which computes (for example a depth-first-search) the degree sequence and a linear extension. The choice for $m \in Min(M(G))$ can be done via the linear extension τ.

The main issue for the algorithm is that m-regularity checking. This can be done as follows :

First compute the graph $[\perp, m]$, this can done by traversing G downwards from m, marking the vertices in red. Similarly using another traversing algorithm we can compute $[j, m^*]$ where m^* is the unique upper cover of m. In this second downwards traversing we mark in blue the vertices not already marked. Now it just remains to check:

(1) For every blue vertex x there exists only one edge ending in a red vertex z, let us denote this vertex by $\theta(x)$.

(2) For every edge $zt \in [\perp, m]$, check if $\theta(z)\theta(t)$ is an edge of $[j, m^*]$, and conversely.

Clearly the computation of $[\perp, m]$, $[j, m^*]$, step 1 and 2 can be done in the size of the interval $[\perp, m]$.

Since every vertex and edge of $[\perp, m]$ is visited at most twice and then can be deleted, therefore the overall complexity of the algorithm is in $O(|E_{tr}|)$.
□

4 Transitive closure and a general recognition algorithm

This section emphasizes uses of the partition theorem for computing in linear time the transitive closure graph of a distributive lattice represented by any graph. We recall that only few classes of orders admit such a good transitive closure algorithm, namely: series-parallel, interval and two dimensional orders, see [12, 10].

First we begin with an obvious property.

Proposition 2 *Given a partition $\{L_1, ..., L_n, L_{n+1}\}$ of a distributive lattice L and let $x \in L_i$ and $y \in L_k$. If $i < k$ then y is not less than x in L.*

Let L be a distributive lattice and $x \in L$. We denote by $Succ(x)$ (resp. $ImSucc(x)$) the successors set of x in the transitive closure graph (resp. covering graph) associated with L. $Pred(x)$ and $Impred(x)$ are defined dually.

Proposition 3 *Given a partition* $\{L_1, ..., L_n, L_{n+1}\}$ *and* $z \in L_i$. *Then* $Succ(z) = \{\Psi_{m_i}(z)\} \cup Succ(\Psi_{m_i}(z)) \cup \{\Psi_{m_i}^{-1}(t) \mid t \in Succ(\Psi_{m_i}(z))\}$. *In particular, if* $z \in M(L)$ *then* $Succ(z) = \{z^*\} \cup Succ(z^*)\}$.

Proof: It is clear that all the successors of $\Psi_{m_i}(z)$ are also successors of z. According to the isomorphism Ψ_{m_i}, we can have the successors of z in the interval L_i by using $\Psi_{m_i}^{-1}$ (see Figure 4). If $z \in M(L)$ then z is a maximal element in L_i. $\Box$

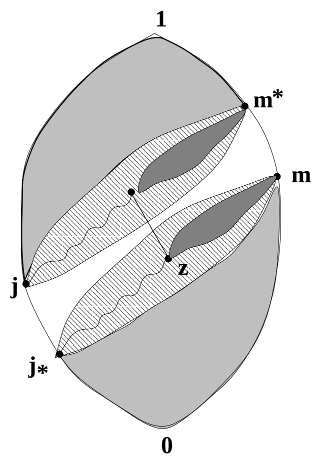

Figure 4: Transitive closure

Proposition 3 leads to a greedy linear time (optimal) algorithm for computing the transitive closure graph of any graph representing a distributive lattice L.

The main idea of the next algorithm is to compute the transitive closure graph of the interval L_i for $1 \leq i \leq n+1$ from the transitive closure graph of the interval $[j_i, \top]$ already computed. Note that the interval L_{n+1} is reduced to the element $\top$.

Algorithm 2: Transitive Closure

Data : Any graph $G = (X, E)$ representing a distributive lattice L
Result: The transitive closure graph of G denoted by $G_{tc} = (X, E_{tc})$
begin
(1) Compute the covering graph of G (using the fact that G is graded with $\perp$ and $\top$ elements);
(2) Compute a partition $\{L_1, ., L_n, L_{n+1}\}$(as in Algorithm 1)
To each element x of L_i, we associate:
(a) Its image $\Psi_{m_i}(x)$;
(b) A stack of all pairs $(i, \Psi^{-1}_{m_i}(x))$ (this simplifies the computing of $\Psi^{-1}_{m_i}$);
(3) The interval L_{n+1} is reduced to top element $\top$ then $Succ(\top) = \{\}$;
for $i = n$ *to* 1 **do**
Compute the successors sets of elements in the interval L_i from those elements in L_{i+1} using the formula given by proposition 2;
At the end of each step, Delete all the pairs used from stacks of the elements in the interval L_i;
end

Theorem 4 *Algorithm 2 computes in* $O(|X|+|E_{tc}|)$ *time the transitive closure graph of any graph* G*, whose transitive closure is a distributive lattice.*

Proof: *Step 1:* This can be done in $O(|X| + |E|)$ using the fact that an arc $e = (x, y)$ belong to the covering graph $D = (X, E_{tr})$ if $rank(y) = rank(x) + 1$ (see [10] for details).

Step 2: Computing the partition $\{L_1, ..., L_n, L_{n+1}\}$ of L can be done in $O(|E_{tr}|)$ from the covering graph obtained by step 1.

Step 3 : At each step i of the loop for, we compute the formula $Succ_{tc}(x) = \Psi_{m_i}(x) \cup Succ_{tc}(\Psi_{m_i}(x)) \cup \{\Psi^{-1}_{m_i}(z) \mid z \in Succ_{tc}(\Psi_{m_i}(x))\}$ of Proposition 4 for each element x of L.

Let us show that this can be done in $O(|Succ_{tc}(x)|)$ for each element in L_i.

The formula is divided in three composantes: $\Psi_{m_i}(x)$ is unique and known by the partition process, $Succ_{tc}(\Psi_{m_i}(x))$ is already computed by the previous step. The third one is the most difficult, the use of the stack simplifies the problem. Indeed, we take all elements in $A = \Psi_{m_i}(x) \cup Succ_{tc}(\Psi_{m_i}(x))$ which are the successors of x out of the interval L_i. For each $x \in A$, we look at the head of its stack, if it contains a pair (i, z) this mean that z is a successor of x in L_i. If x is the meet irreducible m_i then z is not added. At the end of the step i we delete all pairs (i, z) which are all visited.

The correctness of the algorithm is given by Proposition 2. □

Therefore we can now reach our main result. In fact we show that the recognition process can be done starting with any graph in input whose transitive closure graph is distributive lattice.

Theorem 5 *There exists an algorithm in* $O(|X| + |E_{tc}|)$ *time-complexity to recognize if any acyclic graph* $G = (X, E)$ *is a representation of a distributive lattice.*

Proof: If G is a representation of a distributive lattice, then G is graded with a $\perp$ and T element. For such a graph, the transitivity arcs can de defined with the rank function and therefore the transitive closure graph can be computed in linear time.

1. Check if G admits a $\perp$ and $\top$, and compute the rank function.

2. Compute G the covering graph as if G was graded.

3. Apply algorithm 1 on G to check if G is the covering graph of a distributive lattice.

4. Compute G_{tc} with Algorithm 2.

5. Check that every arc in G is also in G_{tc}.

Steps 4. and 5. are really needed, since it is possible that $G_{tr} \neq G$, some examples are available, i.e. using the rank function, we can possibly delete edges which are not transitivity arcs in G. The time-complexity is as follows:

1, 2. $O(|X| + |E|)$
3. $O(|X| + |E_{tr}|)$
4. $O(|X| + |E_{tc}|)$
5. $O(|X| + |E|)$.

Therefore from complexity point of view, the dominating step is the fourth one and yields the overall complexity of the algorithm. □

In this algorithm we have avoided the use of a general purpose algorithm for computation of the covering of G, since such an algorithm has not a linear time-complexity [1]. But the complexity of the above algorithm is not optimal, since it might be possible to produce an algorithm in $O(|X|+|E|)$. In fact using more sophisticated techniques we already have obtained an algorithm [16] that achieves $O(|X|+|E|\times|M(L)|)$ time-complexity, where L is the lattice represented by G.

Example : This example illustrate the use of the partition for transitive closure graph computations.

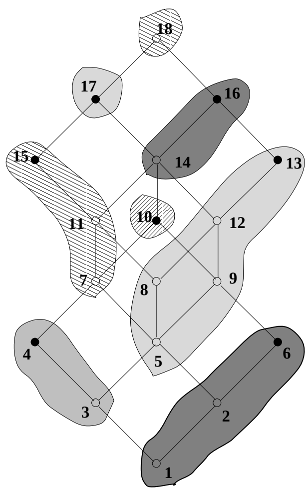

Figure 5: Partition and transitive closure graph

Consider the linear extension of the order induced by meet-irreducible elements $M(L)$, $\tau=\{6,4,13,15,10,16,17\}$. This linear extension induces a partition of L given by $L_1=\{1,2,6\}, L_2=\{3,4\}, L_3=\{5,8,9,12,13\}$, $L_4=\{7,11,15\}, L_5=\{10\}, L_6=\{14,16\}, L_7=\{17\}, L_8=\{18\}$.

The different isomorphism are the following : $\Psi_6=\{(1,3),(2,5),(6,9)\}$, $\Psi_4=\{(3,5),(4,7)\}, \Psi_{13}=\{(5,7),(8,11),(9,10),(12,14),(13,16)\}$, $\Psi_{15}=\{(7,10),(11,14),(15,17)\}, \Psi_{10}=\{(10,14)\}, \Psi_{16}=\{(14,17),(16,18)\}, \Psi_{17}=\{(17,18)\}$.

Each pair (i,j) of an isomorphism Ψ_k is interpreted as : $j=\Psi_k(i)$ and $i=\Psi_k^{-1}(j)$.

L_i	x	$\Psi_i(x)$	List of pairs $(i, \Psi_i^{-1}(x))$	Succ(x)
8	18	{}	(6,16),(7,17)	{}
7	17	18	(4,15),(6,14)	18
6	14	17	(3,12),(4,11),{5,10}	17,18,16
6	16	18	{3,13}	18
5	10	14	{3,9},{4,7}	14,17,18,16
4	7	10	{2,4},(3,5)	10,14,17,18,16,11,15
4	11	14	(3,8)	14,17,18,16,15
4	15	17	{}	17,18
3	5	7	(1,2),(2,3)	7,10,14,17,18,16,11,15,9,12,13
3	8	11	{}	11,14,17,18,16,15,12,13
3	9	10	(1,6)	10,14,17,18,16,12,13
3	12	14	{}	14,17,18,16,13
3	13	16	{}	16,18
2	3	5	(1,1)	5,7,10,14,17,18,16,11,15,9,12,13,8,4
2	4	7	{}	7,10,14,17,18,16,11,15
1	1	3	{}	3,5,7,10,14,17,18,16,11,15,9,12,13,8,4,2,6
1	2	5	{}	5,7,10,14,17,18,16,11,15,9,12,13,6
1	6	9	{}	9,10,14,17,18,16,12,13

5 Conclusions

Notice that using the dual property one can derive easily an incremental algorithm to generate distributive lattice and therefore the down-sets lattice of a partial order. An algorithm with such property can be found in [11].

Now consider a partition $\{L_1, ..., L_n, L_{n+1}\}$ of a distributive lattice L. Recall that if $x \in L_i$ then $\Psi_i(x)$ is the unique element that cover x out of L_i. The set of pairs $(x, \Psi_i(x))$ define a tree rooted in 1. This tree can be associated to L and it is a representation of L (see [11]). Moreover it is easy to generalize the algorithms presented above for extremal lattices and locally distributive lattices as defined recently by Markowsky [13].

References

[1] A.V. Aho, M.R. Garey, and J.D Ullman. The transitive reduction of a directed graph. *SIAM J. Computing*, 1(2):131–137, 1972.

[2] H. Aït-Kaci, R. Boyer, P. Lincoln, and R. Nasr. Efficient implementation of lattice operations. *ACM Transactions on Programming Langages and Systems*, 11(1):115–146, january 1989.

[3] S.P. Avann. Dual symmetry of projective sets in finite modular lattice. *Trans. AMS*, 89:541–558, 1958.

[4] G. Birkhoff. *Lattice Theory*, volume 25 of *Coll. Publ. XXV*. American Mathematical Society, Providence, 3rd edition, 1967.

[5] J.P. Bordat. Calcul des idéaux d'un ordonné fini. *Recherche Opérationnelle / Operations Research*, 25(4):265–275, 1991.

[6] Yves Caseau. Efficient handling of multiple inheritance hierarchies. In *OOPSLA'93*, pages 271–287, 1993.

[7] B. Charron-Bost. *Mesures de la Concurrence et du Parallélisme des Calculs Répartis.* PhD thesis, Université Paris VII, Paris, France, Septembre 1989.

[8] B. A. Davey and H. A. Priestley. *Introduction to lattices and orders.* Cambridge University Press, second edition, 1991.

[9] P. Goralcik, A. Goralcikova, V. Koubek, and V. Rodl. Fast recognition of rings and lattices. In *Proceedings of FCT*, pages 137–145, 1981.

[10] M. Habib, M. Morvan, and J.-X. Rampon. On the calculation of transitive reduction-closure of orders. *Discrete Mathematics*, To appear 1993.

[11] M. Habib and L. Nourine. Tree structure for distributive lattices and its applications. *Theor. Comp. Sci*, 165(2):391–405, October 1996.

[12] T.H. Ma and J. spinrad. Transitive closure for restricted classes of partial orders, 1991.

[13] G. Markowsky. Primes, irreducibles and extremal lattices. *Order*, 9:265–290, 1992.

[14] F. Mattern. Virtual time and global states of distributed systems. In M. Cosnard and al., editors, *Parallel and Distributed Algorithms*, pages 215–226. Elsevier / North-Holland, 1989.

[15] M. Morvan and L. Nourine. Sur la distributivité du treillis des antichaînes maximales d'un ensemble ordonné. *C.R. Acad. Sci.*, t. 317-Série I:129–133, 1993.

[16] L. Nourine. *Quelques propriétés algorithmique des treillis.* PhD thesis, Universite de Montpellier II, France, 1993.

[17] R. Wille. Restructuring lattice theory: An approach based on hierarchies of contexts. *in Ordered sets,I. Rival, Eds. NATO ASI No 83, Reidel, Dordecht, Holland*, pages 445–470, 1982.

[18] R. Wille. Subdirect product construction of concept lattices. *Algebra Universalis*, 17:275–287, 1983.