An Algorithm for Minimal Insertion in a Type Lattice

Petko Valtchev
INRIA Rhône-Alpes
ZIRST, 655 av. de l'Europe, 38330 Montbonnot Saint-Martin, France
tel. + 33 4 76 61 53 75, fax + 33 4 76 61 52 07
`Petko.Valtchev@inrialpes.fr`

Abstract. We consider the insertion of a new element x in a lattice of types L. As the poset $L + x$ obtained by the direct insertion of x in L, is not necessarily a lattice, a set of auxiliary elements should be added in order to restore the lattice properties. We describe an approach towards the lattice insertion based on a global definition of the set of necessary auxiliary elements and their location in L. For that reason, the set of GLB (LUB) of elements in L superior (inferior) to x is considered. Each GLB and LUB which is incomparable with x gives rise to exactly one auxiliary element. The obtained lattice, L^+, is isomorphic to the Dedekind-McNeille completion of $L + x$ and thus it is minimal in size. In addition, the insertion strategy is more general than the existing ones, since it deals with general kind lattices and makes no hypothesis on the location of x in L. An algorithm computing L^+ from L and x of time complexity $O(|L|\omega(L)(|L| + \omega^2(L)))$ is provided.

Keywords: type lattices, lattice insertion, Dedekind-McNeille completion

1 Introduction

Given a lattice $L = \langle X, \leq \rangle$, the insertion of a new element x in it amounts to building a new lattice L^+ containing both x and L. In general, the set of immediate predecessors and immediate successors of x in L are either known or computable. However, the poset $L + x$ obtained by directly inserting x into L with respect to these elements need not to be a lattice. In fact, some sub-sets of $L + x$ may have no GLB or LUB. Thus, $L + x$ is only a sub-order of L^+ or, what is equivalent, L^+ is a lattice completion of $L+x$. Consequently, in order to obtain L^+, some extra elements, called *auxiliary*, have to be inserted into $L+x$.

In the present paper we focus on the insertion in *type lattices*. A type lattice is a lattice where ground set elements are labeled by *type expression*, or simply *types*. Types are members of a type domain, partially ordered by a *sub-typing* relation, and no hypothesis on the nature of types is made. The lattice order follows the sub-typing, that is labeling represents an order-embedding of the ground set into the type domain. Type lattices are used within a wide range of object-oriented and functional languages (see [1], [2], [3]). In fact, various ordered structures like *abstract type lattices* in [4], *M-lattices* in [1], *multiple* *inheritance hierarchies* in [2] and even *concept lattices* [5] may be seen as type lattices. In most cases, these are dynamically evolving structures whence the practical interest of the insertion operation.

The completion of the poset of types $L + x$ into a type lattice L^+ may be decomposed into two disjoint tasks: first, building of the ground lattice L^+ which is an ordinary lattice completion operation, and second, the computation of auxiliary element labels (called *hidden types* in [1]).

A possible approach towards the insertion into a type lattice is described in [1]. The scope of the strategy is however limited to type lattices closed under the join operator, called *M-lattices*. A more general problem, namely the lattice completion of any poset of types is addressed in [2]. Both papers describe algorithmic procedures which can insert only terminal x, i.e. whose only immediate predecessor in L is the bottom element. Moreover, the tasks mentioned above are carried out simultaneously by both procedures. Thus, when the algorithm in [2] is used for lattice insertion, the number of auxiliary elements is larger than the minimum required for restoring the lattice structure. The reason for this unnecessary increase is that auxiliary types may violate lattice structure and thus involve further insertions.

In our own approach, the lattice completion and auxiliary type calculation are dealt with separately. This allows the first task to be seen as the computation of $DM(L + x)$, the Dedekind-McNeille completion of $L + x$. $DM(L + x)$ is the unique, up to isomorphism, smallest lattice to contain $L + x$ as a sub-order. A possible way of computing $DM(L + x)$ from $L + x$, namely as the lattice of the maximal anti-chains of the bipartite order $(L \cup \{x\}, L \cup \{x\}, \not\leq_{L+x})$ is described in [6]. The operation is computationally expensive ($O(|L|^4)$) due to structure of the bipartite order (the size of maximal antichains is in $O(|L|)$). Sub-structures of L^+ of lesser size may be used by the algorithm as reported in [7] but their detection within $L + x$ requires extra computations. Finally, for terminal x, algorithms computing $DM(L + x)$ as the *concept lattice* of the context $K = (L \cup \{x\}, L \cup \{x\}, \leq_{L+x})$ [5] may be found in [8].

In our completion strategy the lattice L^+ is still isomorphic to $DM(L + x)$. However, we design a new procedure for building $DM(L+x)$ from L and x which takes the greatest advantage of the existing lattice structure. Thus, subsets of X which have no GLB (LUB) in $L + x$ are detected together with their initial GLB (LUB) in L, called *odds*. L^+ is obtained by inserting an auxiliary element for each *odd* in $L + x$. The localization of an *odd* in L is taken into account to compute the type of the respective auxiliary.

The paper starts with definitions of some lattice substructures necessary to determine the set of auxiliary elements and their places within $L+x$ (Section 2). Then, the exact way auxiliaries are inserted into $L + x$ to transform it into L^+ is described (Section 3). In Section 4 we provide the proof that L^+ has a lattice structure and is isomorphic to $DM(L + x)$. Finally, an effective algorithm for lattice insertion is presented which uses the previous theoretical results (Section 5).

2 Definitions and Preliminaries

In the following, we assume the reader is familiar with basic concepts of the ordered structure theory like partial order, lattice, covering graph, precedence relation, anti-chain (see [9] for an introduction). Let $L = \langle\, X, \leq_L \rangle$ be a lattice where the ground set X is a sub-set of a bigger set $\mathcal{X}$ assumed potentially infinite. Let also L be given through its *covering graph* $Cov(L) = (X, \prec_L)$, i.e. for each a in X the set of its immediate predecessors, $Pred(a)$, and its immediate successors, $Succ(a)$, are associated to a.

We shall note $\downarrow_L a$ ($\uparrow_L a$) the set of all *predecessors* (*successors*) of a in L and call it the *order ideal* (*order filter*) associated with a. In the following, we shall omit the index on $\downarrow$ ($\uparrow$) when no confusion is possible. Ideals and filters may be generalized to sets as well.

Types may be considered as labels on elements of L. Thus, we consider a mapping $lab : X \to \mathcal{T}$ where $\mathcal{T}$ is a domain of type expressions. $\mathcal{T}$ is provided with a sub-typing relation $\leq_{\mathcal{T}}$ and lattice operators $\wedge_{\mathcal{T}}$ and $\vee_{\mathcal{T}}$. The following condition, called *labeling condition*, holds in L:

$$\forall a, b \in X, a \leq_L b \Leftrightarrow lab(a) \leq_{\mathcal{T}} lab(b).$$

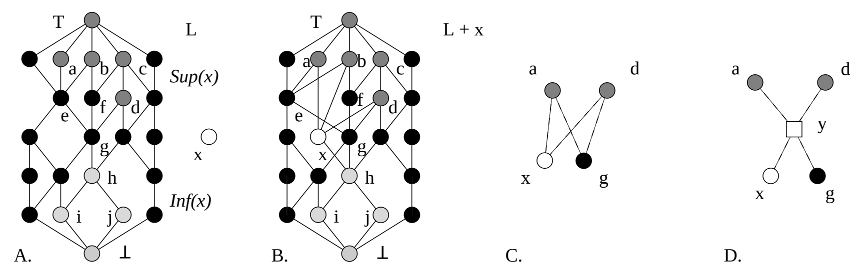

Fig. 1: Sample of direct insertion into a lattice with a possible completion.

A new element $x \in \mathcal{X}$ which is to be inserted into L, is given with its label, $lab(x)$. Thus, by testing $\leq_{\mathcal{T}}$ between $lab(x)$ and the types in L, one may automatically determine the following sub-sets of X: (i) the set of elements superior (inferior) to x, $Sup(x)$ ($Inf(x)$), (ii) the minima (maxima) of $Sup(x)$ ($Inf(x)$), noted $Sup^{im}(x)$ ($Inf^{im}(x)$). On Fig.1.A, an example of L and x is drawn (types are omitted) where the elements of $Sup(x)$ are colored in dark gray, those of $Inf(x)$ in light gray. When x is inserted in L with respect to $Sup^{im}(x)$ and $Inf^{im}(x)$, as shown on Fig.1.B., the obtained structure, $L + x$, need no more to be a lattice. In fact, there may be sub-sets of $Sup(x)$ ($Inf(x)$) with no GLB (LUB) in $L + x$. This is the case of $\{a, d\}$ (see Fig.1.C) whose GLB in L, g, is incomparable with x . Intuitively, restoring the lattice structure amounts to introducing some *auxiliary* elements to become the new GLB (LUB) of those sets in L^+. On Fig.1.D, the auxiliary y is the new GLB of a and d. Elements like g, will be referred to as *odds* and their set will be denoted by $ODD(x)$. Actually, $ODD(x)$ is the union of two disjoint sets of odds. Thus, $ODD_U(x) = \{a | a \| x, \exists b, c \in Sup(x) \ with \ a = b \wedge_L c\}$ will denote the set of GLB of elements in $Sup(x)$, called *upper odds*, whereas $ODD_L(x) = \{a | a \| x, \exists b, c \in Inf(x) \ with \ a = b \vee_L c\}$ will denote the set of LUB of elements in $Inf(x)$, called *lower odds*. The duality principle, allows us to consider only $ODD_U(x)$.

Odds give rise to auxiliary elements so they may be compared to *generators* in [8] and to *canonical representatives* in [1]. The existing strategies for lattice insertion differ as to the way odds are detected and the number of auxiliaries per odd. Thus, the algorithm in [2] checks the GLB of all couples of elements in $Sup(x)$ and inserts an auxiliary each time this GLB is incomparable to x. In this way, the same odd may provoke the generation of a set of auxiliaries. For example, the element g on Fig.1 is an odd since $g = \bigwedge_L \{a, b, d\}$ and $g \| x$. With the algorithm in [2], g will give rise to at least three auxiliary elements, one for each of the couples $\{a, b\}$, $\{a, d\}$ and $\{b, d\}$. Unlike this, the algorithms described in [8] and in [1] generate a single auxiliary per odd.

Our own completion strategy follows the same principle. For this purpose, given $a \in ODD_U(x)$, the set of all superiors to a in $Sup(x)$, $A = \uparrow a \cap Sup(x)$ should be known so that the auxiliary element become the new GLB of A in L^+. For all such a, A is an up-set in L, so for computational purposes, the whole set may be replaced by the set of its minima. Thus, we define a mapping $\mathcal{R} : X \rightarrow 2^{Sup(x)}$ such that $\mathcal{R}(a) = Min(\uparrow a \cap Sup(x))$. Please, notice that for all $a \in Sup(x)$, $\mathcal{R}(a) = \{a\}$ and for all $a \in ODD_U(x)$, $a = \bigwedge_L \mathcal{R}(a)$. From the latter fact follows that for a given $a \in ODD_U(x)$ with $\mathcal{R}(a) = A'$, a is the maximum of the set $\{b | \mathcal{R}(b) = A'\}$. In other words, odds are elements of L incomparable with x and maximal for the respective value of $\mathcal{R}$. Consequently, elements in $ODD_U(x)$ may be characterized locally, i.e. only with respect to their immediate successors.

Proposition 1 $\forall c \in X$

$$c \in ODD_U(x) \ \Leftrightarrow \ (c \not\| x \ and \ \forall e \in X (c \prec_L e \Rightarrow \mathcal{R}(e) \neq \mathcal{R}(c)))$$

Proof: $(\Rightarrow)$ Follows from the definition of $ODD_U(x)$ and $\mathcal{R}(c)$.
$(\Leftarrow)$ By absurd. Suppose there exists $d = \bigwedge_L \mathcal{R}(c)$. Clearly $c \leq_L d$, so there is $e \in X$ such that $c \prec_L e \leq_L d$. On the one hand, $(\uparrow d \cap Sup(x)) \subseteq (\uparrow e \cap Sup(x)) \subseteq (\uparrow c \cap Sup(x))$. On the other hand, by definition of d, $(\uparrow e \cap Sup(x)) = (\uparrow d \cap Sup(x))$. Consequently, $(\uparrow c \cap Sup(x)) = (\uparrow e \cap Sup(x))$. ant $\mathcal{R}(e) = \mathcal{R}(c)$ which is a contradiction. $\square$

Computationally, the checking the above condition for any a in X would be a rather expensive task. Therefore, we limit our consideration to a smaller sub-set of X, $OZ_U(x)$ (stands for *odd zone*). $OZ_U(x)$ is composed of all *potential* odds and lays between $Sup(x)$ and its GLB, $p(x) = \bigwedge_L \ Sup(x)$. Formally,

$$OZ_U(x) = \uparrow p(x) \cap (\bigcup_{a \in Sup(x)} \downarrow a - Sup(x)).$$

$OZ_U(x)$ is a *convex* subset of L, i.e. $\forall a, b \in OZ_U(x), \forall c \in X \ a \leq c \leq b \Rightarrow c \in OZ_U(x)$. The element $p(x)$ may but need not to belong to $ODD_U(x)$ since it may happen that $p(x) \leq x$. In this particular case, $ODD_L(x) = \emptyset$. In the rest of the paper, we shall consider $p(x) \notin Inf(x)$, and indicate how our strategy changes to fit the case $p(x) \in Inf(x)$.

Let $OP_U = \langle ODD_U(x), \leq_{ODD_U(x)} \rangle$ where $\leq_{ODD_U(x)}$ is the restriction of $\leq_L$ on $ODD_U(x)$. The following proposition states that if $p(x) \| x$ then OP_U is a lower semi-lattice with minimal element $p(x)$.

Proposition 2 $\forall a, b \in ODD_U(x)$, *with* $c = a \wedge_L b$, $c \in ODD_U(x) \;\Leftrightarrow\; c \not\| x$.

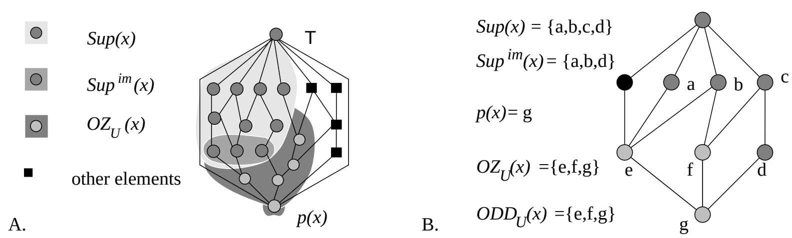

Fig. 2: Detailed structure of $\uparrow p(x)$ with an example.

The mutual position of $\uparrow p(x)$, $Sup(x)$, $Sup^{im}(x)$ and $OZ_U(x)$ in L is illustrated on Fig. 2.A. We shall also denote by $Sup^{bd}(x)$ (^{bd} stands for border), the set of all superiors of x which have an immediate predecessor in $OZ_U(x)$. Formally, $Sup^{bd}(x) = \{a | a \in Sup(x), \exists b \in OZ_U(x), \ b \prec a\}$. It is easy to see that $\mathcal{R}(a) \subseteq Sup^{bd}(x)$.

When constituting the precedence relation in L^+, for each odd a only a subset of $\mathcal{R}(a)$ will be considered. Thus we define the set $\mathcal{R}^{sing}(a)$ as the members of $\mathcal{R}(a)$ which are not superior to odds greater than a. Formally, the mapping $\mathcal{R}^{sing} : \uparrow p(x) \rightarrow 2^{Sup(x)}$ is such that $\forall a, c \in X, b \in \mathcal{R}(a)$: $b \in \mathcal{R}^{sing}(a) \Rightarrow ((a \leq_L c \text{ and } b \in \mathcal{R}(c)) \Rightarrow \mathcal{R}(c) = \{b\})$. In other words, for all predecessors of a which are inferior to such a b, the respective value of $\mathcal{R}$ is limited to $\{b\}$. On the example drawn on Fig. 2.B, $\mathcal{R}(g) = \{a, b, d\}$ and $\mathcal{R}^{sing}(g) = \{d\}$, whereas $\mathcal{R}(f) = \mathcal{R}^{sing}(f) = \{b, c\}$. The elements $s(x)$, $OZ_L(x)$, $\mathcal{S}$ and $\mathcal{S}^{sing}$ are dual to $p(x)$, $OZ_U(x)$, $\mathcal{R}$ and $\mathcal{R}^{sing}$ respectively. Please, notice that $s(x) \leq_L p(x)$.

3 Completion Principles

In the previous section, the set of odds in $L+x$ has been defined. A completion strategy which preserves the initial structure of $L+x$ is proposed in the following. More precisely, we describe a poset $L^+ = \langle X^+, \leq^+ \rangle$ where $X \cup \{x\} \subseteq X^+$ and $\leq_L \subseteq \leq^+$. First, the set of auxiliary elements to be inserted into $L+x$ is defined. Then, we describe the way $\prec^+$ is obtained from $\prec_L$.

3.1 Auxiliary Elements

Let $AUX(x) \subseteq \mathcal{X}$ be the set of auxiliary elements to insert into $L + x$ in order to obtain L^+, i.e. $X^+ = AUX(x) \cup X \cup \{x\}$. Our insertion strategy relies on a set of auxiliary which satisfies $|AUX(x)| = |ODD(x)|$. Let also $\varphi : ODD(x) \longrightarrow \mathcal{X}$ be a bijective mapping. In other words, to each odd a an auxiliary $\varphi(a)$ is assigned. Furthermore, let $AUX_U(x) = Image_\varphi(ODD_U(x))$ and $AUX_L(x) = Image_\varphi(ODD_L(x))$. The set $AUX_U(x)$ is provided with an order relation $\leq_U^A$ which satisfies $\forall a, b \in ODD_U(x)\ \varphi(a) \leq_U^A \varphi(b) \Leftrightarrow a \leq_L b$. Thus, φ defines an order isomorphism between OP_U and $\langle AUX_U(x), \leq_U^A \rangle$. In particular, if $p(x) \in ODD_U(x)$ then $\langle AUX_U(x), \leq_U^A \rangle$ is a lower semi-lattice with a minimal element $\varphi(p(x))$. Fig. 3.A illustrates the way in which the structure of $\langle AUX_U(x), \leq_U^A \rangle$ is obtained from OP_U. Dually, $\langle AUX_L(x), \leq_L^A \rangle$ is an upper semi-lattice with $Max(AUX_L(x)) = \{\varphi(s(x))\}$. In sum, the completion of $L+x$ amounts to integrating a couple of semi-lattices (a single one if $p(x) \in Inf(x)$).

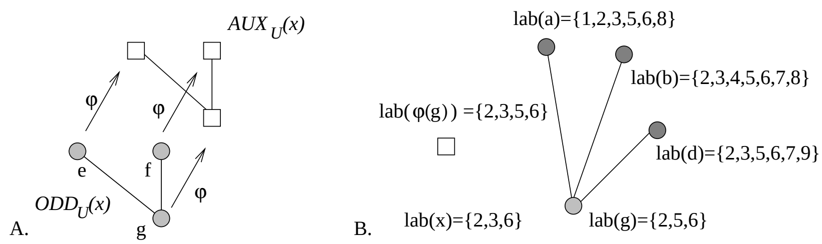

Fig. 3: Computation of $AUX_U(x)$ and an example of lab for the lattice on Fig. 1.

Finally, lab is defined on $AUX(x)$ in the following way:

- for $a \in AUX_U(x)$, $lab(a) = \bigwedge_{\mathcal{T}}\{lab(b) | b \in \mathcal{R}(\varphi^{-1}(a))\}$
- for $a \in AUX_L(x)$, $lab(a) = \bigvee_{\mathcal{T}}\{lab(b) | b \in \mathcal{R}(\varphi^{-1}(a))\}$

The type computation on auxiliary elements is exemplified on Fig. 3.B. Thus, we suppose $\mathcal{T}$ to be a domain of integer sets, where $\leq_{\mathcal{T}}$ is the set inclusion. As we noticed previously, $\mathcal{R}(g) = \{a, b, d\}$, so $lab(\varphi(g)) = lab(a) \wedge_{\mathcal{T}} lab(b) \wedge_{\mathcal{T}} lab(d) = \{2, 3, 5, 6\}$.

3.2 Linking Auxiliaries

We suppose that L is given through its covering graph $Cov(L) = (X, \prec_L)$. The way $\prec^+$ is obtained from $\prec_L$. is exemplified by Fig. 4.

First, the precedence relations $\prec_U^A$ and $\prec_L^A$ are preserved in L^+. In other words, for $a, b \in ODD_U(x)$, if $a \prec_{ODD_U(x)} b$ then $\varphi(a) \prec^+ \varphi(b)$. Next, each auxiliary element is linked to its original, that is for all $a \in AUX_U(x)$, we set $\varphi^{-1}(a) \prec^+ a$. Dually, for all $b \in AUX_L(x)$, $b \prec^+ \varphi^{-1}(b)$. For example, on figure Fig. 4.A, $\varphi(f)$ is linked to f. In addition, an auxiliary $e = \varphi(a)$ precedes

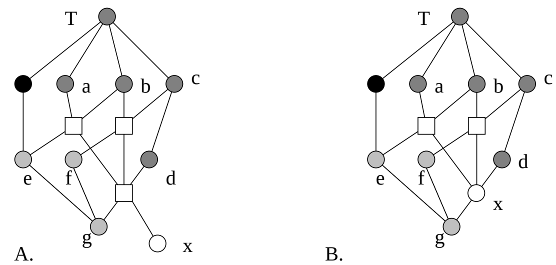

Fig. 4: Lattice completion: integration of $AUX_U(x)$.

all the elements in $\mathcal{R}^{sing}(a)$ (see definition of $\mathcal{R}^{sing}$). So, $\forall a \in ODD_U(x), \forall b \in \mathcal{R}^{sing}(a), \varphi(a) \prec^+ b$. On the same example of Fig. 4.A, $\varphi(g)$ precedes d since d is in $\mathcal{R}(g)$. With the last completion, a link between an odd element a and an element of $\mathcal{R}^{sing}(a)$ becomes redundant so it should be dropped out. In other words, $\forall a \in ODD_U(x), \forall b \in \mathcal{R}(a), a \prec_L b \Rightarrow a \not\prec^+ b$. On the figure above, c is no more preceded by f since $\varphi(f)$ is inserted between them. Finally, minimal elements in $AUX_U(x)$ are preceded by x. In case $p(x) \in ODD_U(x)$ this means that $x \prec^+ \varphi^{-1}(p(x))$ as shown on Fig. 4.A. In case $p(x) \in Inf(x)$, the place of $\varphi(p(x))$ is simply taken by x which should be inserted between $p(x)$ and $Min(AUX_U(x)) \cup \mathcal{R}^{sing}(p(x))$ (with redundant links dropped out). Therefore, we shall further simplify our notation by the assumption, somehow abusive, that in case $p(x) \in Inf(x)$, $\varphi(p(x)) = x$. This situation is illustrated on Fig. 4.B, where the same structure of $\uparrow p(x)$ is supposed with $p(x) = g \leq^+ x$.

4 Lattice Operators

In the present section we shall prove that $L^+ = \langle X^+, \leq^+ \rangle$ where $X^+ = X \cup AUX(x) \cup \{x\}$ and $\leq^+$ be the transitive closure of $\prec^+$ is a lattice. For this purpose, we shall prove that each couple of elements in X^+ has a GLB and a LUB in L^+. Actually, only those couples will be considered which contain at least one new element, i.e. either auxiliary or x, or their GLB (LUB) in L^+ are new. The rest of the elements preserve their bounds due to the way L^+ is defined.

4.1 Preliminaries

First, one needs a mapping which associates to a give a in $OZ_U(x)$ the greatest element in X which shares the same $\mathcal{R}$ value with a. Thus, we define $\nu : OZ_U(x) \to OZ_U(x) \cup Sup(x)$ with $\nu(a) = \bigwedge_L \mathcal{R}(a)$. The dual map μ is defined for all b in $OZ_L(x)$ as $\mu(a) = \bigvee_L \mathcal{S}(a)$. Next, it is easy to see that $\leq_L$ is a sub-set of $\leq^+$, i.e. $\forall a, b \in X, a \leq_L b \Leftrightarrow a \leq^+ b$. Moreover, by definition of $\prec^+$, $\forall a, b \in AUX_U(x), a \leq^+ b \Leftrightarrow \varphi^{-1}(a) \leq_L \varphi^{-1}(b)$.

In addition, both $AUX_U(x)$ and $AUX_L(x)$ are *convex* sets in L^+, that is

$\forall a, b \in AUX_U(x), c \in X^+, a \leq^+ c \leq^+ b \;\Leftrightarrow\; c \in AUX_U(x)$. Finally, for all a in $AUX_U(x)$: (i) $x \leq^+ a$ and $p(x) \leq^+ a$, (ii) $\uparrow_{L^+} a \subseteq Sup(x) \cup AUX_U(x)$ and (iii) $\forall b \in AUX_L(x), b \leq^+ a$.

The following proposition asserts that the completion procedure described in the previous section is correct with respect to $\leq_L$.

Proposition 3 $\forall a \in X, \forall b \in AUX_U(x)$

- $a \leq^+ b \;\Leftrightarrow\; a \leq_L \varphi^{-1}(b)$,
- $b \leq^+ a \;\Rightarrow\; \varphi^{-1}(b) \leq_L a$ *and if* $a \in Sup(x)$ *then* $\varphi^{-1}(b) \leq_L a \;\Rightarrow\; b \leq^+ a$.

Proof In the *first* case, the $\Leftarrow$ part is trivial. The $\Rightarrow$ part can be proved by induction on the length of the shortest path between a and b in the covering graph $Cov(L^+)$. The same kind of induction is used in the proof of both necessary and sufficient conditions in the *second* case. $\Box$

4.2 Lower Bounds in L^+

A constructive proof of the fact that for arbitrary a, b in X^+, $a \wedge^+ b$ exists is given below. First, when both a and b lay in $Sup(x)$ their GLB in L is either itself in $Sup(x)$ and in this case it remains the same in L^+ or it is an odd and the new GLB is the respective auxiliary.

Lemma 1 *For a couple of elements* $a, b \in Sup(x)$,

- $a \wedge_L b \in Sup(x) \Rightarrow a \wedge^+ b = a \wedge_L b$
- $a \wedge_L b \notin Sup(x) \Rightarrow a \wedge^+ b = \varphi(a \wedge_L b)$

Proof In the *first* case, we only examine lower bounds e of a, b with $e \in AUX_U(x)$. Thus, $e \leq^+ a \wedge_L b$ follows trivially from Proposition 3. In the *second* case, $d = a \wedge_L b$ is clearly in $ODD_U(x)$ and thus $\varphi(d) = c$ is a lower bound for a, b in L^+ with $d <^+ c$. Again, only $e \in AUX_U(x)$ with $e \leq^+ a, e \leq^+ b$ are considered. We derive from Proposition 3, $\varphi^{-1}(e) \leq_L d$ so $e \leq^+ c$. $\Box$

The GLB of couples where one element is in $Sup(x)$ and the other is a new element are also new elements.

Lemma 2 *For all* $a \in Sup(x)$:

- *if* $b \in X - Sup(x)$ *then* $a \wedge^+ b = a \wedge_L b$,
- *if* $b \in AUX_U(x)$ *then* $a \wedge^+ b = \varphi(a \wedge_L \varphi^{-1}(b))$,
- *if* $b \in AUX_L(x) \cup \{x\}$ *then* $a \wedge^+ b = b$.

Proof In the *first* case, the only auxiliary lower bounds c of a, b may lay in $AUX_L(x)$. By Proposition 3, $\varphi^{-1}(c) \wedge_L a \wedge_L b$ and thus $c \wedge^+ a \wedge_L b$. In the *second* case, observe that $a \wedge_L \varphi^{-1}(b)$ is in $ODD_U(x)$. Thus, the proof is similar to the second case of Lemma 1. The proof of the *third* case is trivial. □

The GLB of auxiliary elements are either itself in $AUX_U(x)$ or coincide with the GLB of the respective odds.

Lemma 3 *For all* $a \in AUX_U(x)$,

- *if* $b \in AUX_U(x)$ *then* $a \wedge^+ b = \varphi(\varphi^{-1}(a) \wedge_L \varphi^{-1}(b))$,
- *if* $b \in X - Sup(x)$ *then* $a \wedge^+ b = \varphi^{-1}(a) \wedge_L b$.

Proof In the *first* case, the proof follows immediately from the definition of $\leq^+$ and Proposition 3. In the *second* case, $c = \varphi^{-1}(a) \wedge_L b$ is a lower bound. Observe that any other lower bound c of a, b is in $X \cup AUX_L(x)$. In case $c \in X$, the proof follows trivially from Proposition 3. Let $c \in AUX_L(x)$, then $\varphi^{-1}(c) \leq_L p(x) \leq_L \varphi^{-1}(a)$ and, by Proposition 3, $\varphi^{-1}(c) \leq_L b$. Consequently, $c \leq^+ \varphi^{-1}(c) \leq^+ \varphi^{-1}(a) \wedge_L b$. □

Finally, let us examine the GLB of x.

Lemma 4 *For a given element* $a \in X \backslash (Sup(x) \cup Inf(x))$ *with* $e = a \wedge s(x)$*:*

- *if* $e \in OZ_L(x)$ ***and*** $\mu(e) \in OZ_L(x)$ *then* $x \wedge^+ a = \varphi(\mu(e))$
- *if* $e \in OZ_L(x)$ ***and*** $\mu(e) \in Inf(x)$ *then* $x \wedge^+ a = \mu(e)$,
- *if* $e \in Inf(x)$ *then* $x \wedge^+ a = e$,
- *if* $e \in \downarrow s(x) \backslash \{OZ_L(x) \cup Inf(x)\}$ *then* $x \wedge^+ a = -$

Proof Observe that the concerned lower bounds are in $AUX_L(x) \cup Inf(x)$. In the *first* case, $\varphi(\mu(e))$ (see Section 4.1) is an obvious lower bound of a, x ($\varphi(\mu(e)) \leq^+ \mu(e) \leq_L e \leq_L a$). For another lower bound $c \in Inf(x)$, c is a lower bound of $a, s(x)$ as well, so $c \leq_L e$. Consequently $c \leq^+ \varphi(\mu(e))$. For a $c \in AUX_L(x)$, $\varphi^{-1}(c)$ is, by Proposition 3, a lower bound of $a, s(x)$, so $\varphi^{-1}(c) \leq_L e$. Thus, by definition of μ, $\varphi^{-1}(c) \leq_L \mu(e)$, so $c \leq^+ \varphi(\mu(e))$. In the *second* case, a simplified version of the above reasoning schema may be applied. The proofs of the *third* and the *fourth* case are immediate. □

In case $p(x)$ is in $Inf(x)$, the above lemmas remain valid when $s(x)$ is replaced with $p(x)$ and $\varphi(p(x))$ – with x. Finally, they may be summarized, together with the respective dual assertions, in the following theorem.

Theorem 1 $\forall a, b \in X^+$, $a \wedge^+ b$ *exists.*

4.3 Upper Bounds in L^+

As in the previous section, only upper bounds relevant to $AUX_U(x)$ will be examined. Thus, the LUB of elements in $X - Inf(x)$ do not need being considered since they remain the same as in L (due to Proposition 3). Furthermore, the results on GLB involving elements in $AUX_U(x) \cup \{x\}$ presented in the previous section are dually valid for LUB and $AUX_L(x) \cup \{x\}$. For a couple of auxiliaries, their LUB is the maximal element of L^+ which shares the same value for $\mathcal{R}$ with the LUB of the respective odds.

Lemma 5 *For all $a, b \in AUX_U(x)$, with $\varphi^{-1}(a) \vee_L \varphi^{-1}(b) = f$:*

- $f \in OZ_U(x)$ *and* $\nu(f) \in OZ_U(x) \Rightarrow a \vee^+ b = \varphi(\nu(f))$,
- $f \in OZ_U(x)$ *and* $\nu(f) \in Sup(x) \Rightarrow a \vee^+ b = \nu(f)$,
- $f \in Sup(x) \Rightarrow a \vee^+ b = f$.

Proof First, for any $a \in AUX_U(x)$, $a \leq^+ b$ implies $b \in Sup(x) \cup AUX_U(x)$. In the *first* case, clearly, $\varphi(\nu(f))$ is an upper bound of a and b since $\nu(f)$ is an upper bound of $\varphi^{-1}(a)$ and $\varphi^{-1}(b)$. We shall prove that for an upper bound d of a and b $\varphi(\nu(f)) \leq^+ d$. Let first $d \in AUX_U(x)$. By definition of $AUX_U(x)$, $\varphi^{-1}(d)$ is an upper bound of $\varphi^{-1}(a)$ and $\varphi^{-1}(b)$, $f \leq_L \varphi^{-1}(d)$. Considering $\mathcal{R}(f)$ and $\mathcal{R}(\varphi^{-1}(d))$ leads to $\nu(f) \leq_L \varphi^{-1}(d)$. Again, by the definition of $AUX_U(x)$, $\varphi(\nu(f)) \leq^+ d$. Let now $d \in Sup(x)$. By Proposition 3, d is an upper bound for $\varphi^{-1}(a)$ and $\varphi^{-1}(b)$, so $f \leq_L d$. The fact $\varphi(\nu(f)) \leq^+ d$ follows by the same Proposition 3. In the *second* case, $\nu(f)$ is again an upper bound of a, b (see above) and no upper bounds of a, b can be in $AUX_U(x)$. In a way similar to the above, one proves for any upper bound d of a, b, $\nu(f) \leq_L d$. In the *third* case, f is an obvious upper bound of a, b. It is the least one since all other upper bounds d lay in $Sup(x)$ and satisfy d is an upper bound for $\varphi^{-1}(a)$ and $\varphi^{-1}(b)$. $\square$

LUB of auxiliary and initial elements are computed in a similar way.

Lemma 6 *For a couple of elements $a \in X$, $b \in AUX_U(x)$ with $a \vee_L \varphi^{-1}(b) = f$:*

- $f \in OZ_U(x)$ ***and*** $\nu(f) \in OZ_U(x) \Rightarrow a \vee^+ b = \varphi(\nu(f))$,
- $f \in OZ_U(x)$ ***and*** $\nu(f) \in Sup(x) \Rightarrow a \vee^+ b = \nu(f)$,
- $f \in Sup(x) \Rightarrow a \vee^+ b = f$,
- $f \in X \backslash (Sup(x) \cup OZ_U(x) \Rightarrow a \vee^+ b = \top$.

The next theorem follows directly from the above lemmas and their dual assertions concerning $AUX_L(x)$ within $OZ_L(x)$.

Theorem 2 $\forall a, b \in X^+$, *$a \vee^+ b$ exists.*

4.4 Minimalness

The Dedekind-McNeille completion of $L + x$, $DM(L + x)$, is defined as the set of all subsets of $L + x$ closed under the ^{lu} operator (see [9]). $DM(L + x)$ is the smallest lattice which contains a sub-order isomorphic to $L + x$ and each other lattice completion of $L + x$ contains a sub-lattice isomorphic to $DM(L + x)$. A possible way to prove $L^+ \cong DM(L+x)$ may be to show a concrete isomorphism, say χ, between L^+ and $DM(L+x)$. This isomorphism will be an extension on L^+ of the standard order embedding $\lambda : L+x \to DM(L+x)$ with $\lambda(e) = \uparrow e$. Thus, it will be enough to prove that each auxiliary element a is mapped by χ into a set A which is closed under ^{lu}, i.e. $A = A^{lu}$, and has at least two minimal elements, that is $\nexists e \in L+x$ with $A = \uparrow e$. However, for the sake of compactness, we give a more direct proof of the isomorphism using the fact that any lattice completion of $L+x$ contains at least as much elements as L^+.

Lemma 7 *L^+ is the smallest lattice with $(X \cup \{x\}) \subseteq X^+$ and $\leq \subseteq \leq^+$*

Proof Suppose $\exists L' = \langle X', \leq' \rangle$ with $(X \cup \{x\}) \subseteq X'$ and $\leq \subseteq \leq'$, such that $|X^+| \leq |X'|$. Let us note $AUX'_U(x) = (X' - (X \cup \{x\})) \cap \uparrow_{L'} x$. With no loss of generality we may assume $|AUX'_U(x)| < |AUX_U(x)|$. It is easy to see that in this case there exists $a \in ODD_U(x)$ such that $\forall b \in AUX'_U(x)$, $Min(\uparrow_{L'} b \cap Sup(x)) \neq \mathcal{R}(a)$. Consider $c = \bigwedge_{L'} \mathcal{R}(a)$. Since x is a lower bound of $\mathcal{R}(a)$, necessarily $c \in AUX'_U(x)$. On the one hand, $Min(\uparrow_{L'} c \cap Sup(x)) \neq \mathcal{R}(a)$ so $\exists d \in Min(\uparrow_{L'} c \cap Sup(x)) - \mathcal{R}(a)$. Clearly, $a \not\leq_L d$ and thus $a \not\leq' c$. On the other hand, a is still a lower bound of $\mathcal{R}(a)$ in L' since $\leq$ is a subset of $\leq'$. The latter contradicts $a \not\leq' c = \bigwedge_{L'} \mathcal{R}(a)$. Consequently, $|AUX_U(x)| \leq |AUX'_U(x)|$. $\square$

As a lattice completion of $L+x$, L^+ contains a sub-lattice isomorphic to $DM(L+x)$. The latter lattice, in turn, is at least as large as the former one. Combining both facts yields an isomorphism between $DM(L+x)$ and L^+.

4.5 Labeling Condition

It may easily be proved that $\forall a \in AUX_U(x), lab(\varphi^{-1}(a)) \leq_{\mathcal{T}} lab(b)$. Dually $lab(a) \leq_{\mathcal{T}} \varphi^{-1}(a)$ for all $a \in AUX_L(x)$. Furthermore, for all a in $AUX_U(x)$ ($AUX_L(x)$), $lab(x) \leq_{\mathcal{T}} lab(a)$ ($lab(a) \leq_{\mathcal{T}} lab(x)$). Next, the labeling condition is preserved on $AUX_U(x)$.

Proposition 4 *For all a, b in $AUX_U(x)$, $a \leq^+ b \Leftrightarrow lab(a) \leq_{\mathcal{T}} lab(b)$*

Proof ($\Rightarrow$) Follows from the definition of lab on $AUX_U(x)$. ($\Leftarrow$) Consider $\varphi^{-1}(a)$ and $\varphi^{-1}(b)$. $lab(\varphi^{-1}(a)) \leq_{\mathcal{T}} lab(a)$ (see above) so $lab(\varphi^{-1}(a)) \leq_{\mathcal{T}} lab(b)$. Consequently, as the labeling condition holds for L for all d in $\mathcal{R}(\varphi^{-1}(b))$, $\varphi^{-1}(a) \leq_L d$. Thus, $\varphi^{-1}(a) \leq_L \varphi^{-1}(b)$. Finally, $a \leq^+ b$. $\square$

Finally, the new labels respect the order between auxiliary and initial elements.

Proposition 5 *For all a in $AUX_U(x)$, b in X:*

- $a \leq^+ b \Leftrightarrow lab(a) \leq_{\mathcal{T}} lab(b)$,
- $b \leq^+ a \Leftrightarrow lab(b) \leq_{\mathcal{T}} lab(a)$.

Proof First case. ($\Rightarrow$) Clearly, $b \in Sup(x)$ and according to Proposition 3 $\varphi^{-1}(a) \leq_L b$. ($\Leftarrow$) $lab(\varphi^{-1}(a)) \leq_{\mathcal{T}} lab(b)$ and since labeling condition holds on L, $\varphi^{-1}(a) \leq_L b$. By Proposition 3, $a \leq^+ b$.
Second case. ($\Rightarrow$) Follows from Proposition 3. ($\Leftarrow$) For all d in $\mathcal{R}(\varphi^{-1}(a))$, $lab(b) \leq_{\mathcal{T}} lab(a) \leq_{\mathcal{T}} lab(d)$ so $b \leq_L d$. Consequently, $b \leq_L \varphi^{-1}(a)$. $\Box$

The above couple of lemmas remain valid in case $p(x)$ is in $Inf(x)$. When taken with their dual assertions, they state the correction of our label computing procedure.

Theorem 3 *$\forall a, b \in X^+$, $a \leq^+ b \Leftrightarrow lab(a) \leq_{\mathcal{T}} lab(b)$.*

5 Algorithm

In the following we describe the steps of a global lattice completion algorithm which implements the completion procedure described in Section 3. The soundness of the algorithm, i.e. the proof that the produced structure is really a type lattice is guaranteed by theorems 1, 2, and 3

5.1 Data Structures and Primitives

The lattice L is represented by its covering graph $Cov(L) = (X, \prec_L)$ Each element e in X is represented by an object o, whereby informations related to e are stored in object fields. Thus, *o.pred* and *o.succ* represent respectively the list of the immediate predecessors and the list of immediate successors of e in L. Moreover, the type expression is stored in *o.lab*. The *o.index* field carries a real number, the index of the element in a linear extension of $\leq_L$. In addition, if $e \in OZ_U(x)$ then the $o.\mathcal{R}$ and $o.\mathcal{R}^{sing}$ fields contain the values of the respective mappings, whereas $o.ODD^{max}$ contains the set of the maximal odds, inferior to e. The fields $o.\mathcal{R}$, $o.\mathcal{R}^{sing}$ and $o.ODD^{max}$ are initially set to $\emptyset$. Finally, if $e \in ODD_U(x)$ then the *o.artificial* field refers the object representing the auxiliary element $\varphi(e)$.

In the following, we suppose that $Sup(x)$, $Sup^I(x)$, $Sup^{bd}(x)$, $p(x)$, and $OZ_U(x)$ are given since their computation is trivial. In addition, the relation $\leq_L$, on X is supposed to be directly available. Thus, the fact $a \leq_L b$ may be checked in constant time. The complexity of the incremental maintenance of $\leq$ may be roughly estimated at $O(|L|^2\omega(L))$ in time and $O(|L|^2)$ in space.

The primitives *ord-union-min*, *ord-union-max* and *ord-difference*, when applied on a couple (A, B) of antichains in L^+, compute $Min(A \cup B)$, $Max(A \cup B)$ and $A - \uparrow B$ respectively. The complexity of each primitive is $O(\omega^2(L))$.

The primitives *create-link* and *drop-link* maintain the precedence relation $\prec^+$ and take constant time.

5.2 Detecting $ODD_U(x)$

The algorithm carries out a priority-guided search through $OZ_U(x)$. In a preliminary step, the members of $Sup^{bd}(x)$ propagate their references downwards to their successors in $OZ_U(x)$ thus initializing the $\mathcal{R}^{sing}$ field of each successor. Next, the elements in $OZ_U(x)$ are processed in a global loop. At each o, the fields $o.\mathcal{R}$ and $o.\mathcal{R}^{sing}$ are first updated, then the odd condition is checked. Finally, the value of $o.\mathcal{R}$ is propagated downwards.

Algorithm 1 *Detection of $ODD_U(x)$*

---

for o **in** $Sup^{bd}(x)$ **do**
    for o' **in** o.*pred* $\cap OZ_U(x)$ **do**
        o'.$\mathcal{R}^{sing}$ $\leftarrow$ o'.$\mathcal{R}^{sing}$ $\cup$ {o}

for o **in** $OZ_U(x)$ in decreasing index order **do**
    o.$\mathcal{R}^{sing}$ $\leftarrow$ ord-difference(o.$\mathcal{R}^{sing}$,o.$\mathcal{R}$)
    o.$\mathcal{R}$ $\leftarrow$ o.$\mathcal{R}$ $\cup$ o.$\mathcal{R}^{sing}$; *card* $\leftarrow$ $\|$o.$\mathcal{R}\|$

    if *card* > 1 **then**
        is-bound $\leftarrow$ true
        for o' **in** o.*succ* $\cap OZ_U(x)$ **do**
            is-bound $\leftarrow$ *is-bound* **and not** o.$\mathcal{R}$ = o'.$\mathcal{R}$
        if *is-bound* **then**
            $ODD_U(x) \leftarrow ODD_U(x) \cup$ {o}

    for o' **in** o.*pred* $\cap OZ_U(x)$ **do**
        if *card* $= 1$ **then**
            o'.$\mathcal{R}^{sing}$ $\leftarrow$ ord-union-min(o.$\mathcal{R}$,o'.$\mathcal{R}^{sing}$)
        else
            o'.$\mathcal{R}$ $\leftarrow$ ord-union-min(o.$\mathcal{R}$,o'.$\mathcal{R}$)

The above procedure is of $O(|L|\omega^3(L))$ time complexity since antichain comparisons (in $O(\omega^2(L))$) are carried out $O(\omega(L))$ times for each member of $OZ_U(x)$ (of $O(|L|)$ size). At the end of the procedure, $ODD_U(x)$ contains all odd elements, possibly $p(x)$. Furthermore, the $\mathcal{R}^{sing}$ fields of the objects in $ODD_U(x)$ indicate the the direct successors of the respective auxiliary elements. What remains is to compute the $\prec_{AUX_U(x)}$ relation. This is done directly on odd elements, that is $\prec_{ODD_U(x)}$ is computed instead, in a way quite similarly to the computation of $\mathcal{R}$. The essential difference is that the search starts by $p(x)$ and goes upwards. The procedure is of $O(|L|\omega^3(L))$ time complexity.

Algorithm 2 *Construction of $\prec_{ODD_U(x)}$*

---

p.ODD^{max} $\leftarrow$ {p}
for o **in** $OZ_U(x)$ in increasing index order **do**
    for o' **in** o.*succ* $\cap OZ_U(x)$ **do**
        if o **in** $ODD_U(x)$ **then**
            o'.ODD^{max} $\leftarrow$ ord-union-max(o'.ODD^{max},{o})
        else
            o'.ODD^{max} $\leftarrow$ ord-union-max(o'.ODD^{max},o.ODD^{max})

5.3 Integration of $AUX_U(x) \cup \{x\}$ into L

Once all the necessary information about $\prec^+$ has been gathered, the effective creation of auxiliary elements and their integration may be carried out. This is done by strictly following the rules described in Section 3.2. Thus, after being created and linked to the respective odds auxiliaries are further linked to each element in $\mathcal{R}^{sing}$ and in ODD^{max}. The redundant links are dropped and x is connected to its auxiliary successors. Finally, the corrections necessary in case of $p(x) \in Inf(x)$ are carried out. The overall complexity of the algorithm is $O(|L|\omega^2(L))$.

Algorithm 3 *Creating and Linking the Artificial Elements*

```
for o in ODD_U(x) in increasing index order do
    new(o.artificial)
    AUX_U(x) ← AUX_U(x) ∪ {o.artificial}; link-create(o,o.artificial)
    if o ≠ p then
        for o' in o.ODD^max do
            link-create(o'.artificial,o.artificial)
    for o' in o.R^sing do
        link-create(o.artificial,o')
        if o in o'.pred then
            link-drop(o, o')
if p(x) ∈ Inf(x) then
    link-create(p(x), x)
    for o in p(x).artificial.succ
        link-create(x, o); link-drop(p(x).artificial,o)
    link-drop(p(x),p(x).artificial)
else
    link-create(x,p(x).artificial)
```

5.4 Computing Indices

The last step of the completion is the index computation for auxiliary elements. Indexes, necessary for the topological search through L^+, may be seen as a mapping $\iota : X^+ \longrightarrow \mathbb{R}$. The restriction of ι on X, already available, defines a linear extension of $\leq_L$. The computation of ι on $AUX(x) \cup \{x\}$) is a two-step process. On the first step, the index of x is computed as:

$$\iota(x) = \frac{\min_{a \in Sup^I(x)}(\iota(a)) + \max_{b \in Inf^I(x)}(\iota(b))}{2}$$

Then, the members of $AUX_U(x)$ are treated in a topological top-down order whereby the index of each $e \in AUX_U(x)$ is

$$\iota(e) = \frac{\min_{a \in successors(e)}(\iota(a)) + \max(\{\iota(x), \iota(\psi^{-1}(e))\})}{2}$$

The index of a $e \in AUX_L(x)$ is obtained in a dual manner. The next lemma follows trivially from the above definitions.

Lemma 8 *ι defines a linear extension of $\leq^+$.*

6 Conclusion

We presented a strategy for insertion of a new element x in a lattice of types L. First, the set of auxiliary elements necessary to preserve the lattice structure is characterized. Then, an appropriate way to insert these elements into $L+x$ is described. Next, the proof of the lattice structure of the obtained order is given. In addition, the obtained lattice is shown to be minimal in size and thus isomorphic to the Dedekind-McNeille completion of $L+x$. We also presented a lattice completion algorithm of time complexity $O(|L|\omega(L)(|L|+\omega^2(L)))$ and space complexity $O(|L|^2)$. Our approach is more general than the existing strategies, in particular those presented in [2], [1] and [6] since it considers non-extremal elements in a general type lattice. Thus, even for terminal elements x, the generated lattice L^+ remains smaller in size than the output in [2].

7 Acknowledgments

The author would like to thank the anonymous referees who helped to improve the paper. Thanks to Jérôme Euzenat for thoughtful comments and suggestions. Thanks go as well to Michel Habib and Bernard Ganter for the fruitful discussions on the subject and the critical remarks on the paper content.

References

1. M. Missikoff and M. Scholl. An algorithm for insertion into a lattice: Application to type classification. In *Proceedings of the 3rd International FODO Conference*, number 367 in Lecture Notes in Computer Science, 1989.
2. Y. Caseau. Efficient handling of multiple inheritance hierarchies. In A. Paepcke, editor, *Proceedings of the ACM OOPSLA Conference, Washington, DC, USA*, pages 271–287. ACM Press, 1993.
3. C. Capponi. *Identification et exploitation des types dans un modèle de connaissances à objets*. Thèse d'informatique, Université Joseph Fourier, Grenoble (FR), 1995.
4. H. Ait-Kaci, R. Boyer, P. Lincoln, and R. Nasr. Efficient implementation of lattice operations. *ACM Transactions on Programming Languages and Systems*, 11(1):115–146, January 1989.
5. R. Wille. Restructuring the lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered sets*, pages 445–470, Dordrecht-Boston, 1982.
6. G.-V. Jourdan, J.-X. Rampon, and C. Jard. Computing on-line the lattice of maximal antichains of posets. *Order*, 11(2):197–210, December 1994.
7. M. Morvan and L. Nourine. Simplicial elimination schemes, extremal lattices and maximal antichain lattices. *Order*, 13(2):159–173, 1996.
8. R. Godin, R. Missaoui, and H. Alaoui. Incremental concept formation algorithms based on galois (concept) lattices. *Computational Intelligence*, 11(2):246–267, 1995.
9. B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, 1992.