Widening Operators for Abstract Interpretation

Agostino Cortesi
Dipartimento di Informatica
Università Ca' Foscari di Venezia
I-30170 Venezia (Italy)
cortesi@unive.it

Abstract

Abstract Interpretation, one of the most applied techniques for semantics based static analysis of software, is based on two main key-concepts: the correspondence between concrete and abstract semantics through Galois connections/insertions, and the feasibility of a fixed point computation of the abstract semantics, through the fast convergence of widening operators. The latter point is crucial to ensure the scalability of the analysis to large software systems.

In this paper, we investigate which properties are necessary to support a systematic design of widening operators, by discussing and comparing different definitions in the literature, and by proposing various ways to combine them. In particular, we prove that, for Galois insertions, widening is preserved by abstraction, and we show how widening operators can be combined for the cartesian and reduced product of abstract domains.

Keywords: Static Analysis, Abstract Interpretation, Abstract Domains, Widening Operators.

1 Introduction

Abstract Interpretation is a general theory of approximation of mathematical structures, in particular those involved in the semantic models of computer systems, that has been successfully applied for the static analysis of software systems. This theory is based on two main key-concepts: the correspondence between concrete and abstract semantics through Galois connections/insertions, and the feasibility of a fixed point computation of the abstract semantics, through the fast convergence of widening operators.

While Galois connections have been widely studied, yielding to a suite of general techniques to manage the combination of abstract domains, e.g. different kind of products [10, 20, 5], and more sophisticated notions like the quotient [7], the complement [6], and the powerset [18] of abstract domains, not much attention has been given to provide general results about widening operators.

Nevertheless, widening operators play a crucial role in particular when infinite abstract domains are considered to ensure the scalability of the analysis to large software systems, as it has been shown in the case of the Astrée project for analysis of absence of run-time error of avionic critical software [8].

The first infinite abstract domain (that of intervals) was introduced in [9]. This abstract domain was later used to prove that, thanks to widening operators, infinite abstract domains can lead to effective static analyses for a given programming language that are strictly more precise and equally efficient than any other one using a finite abstract domain or an abstract domain satisfying chain conditions [12].

Specific widening operators have been also designed for type graphs [21], in domains for reordering CLP(RLin) programs [27], and in the analysis of programs containing digital filters [17], just to name a few. More recently, widenings have been used also to infer loop invariants inside an STM solver [22], and in trace partitioning abstract domain [28].

The main challenge for widening operators is when considering numerical domains. For the domain of convex polyhedra, the original widening operator proposed by Cousot and Halbwachs [13] has been improved by recent works by Bagnara et al [1], and refined for the domain of pentagons in [23]. In [2] the authors define three generic widening methodologies for a finite powerset abstract domain. The widening operators are obtained by lifting any widening operator defined on the base-level abstract domain. The proposed techniques are instantiated on powersets of convex polyhedra, a domain for which no non-trivial widening operator was previously known.

We observed that, with the noticeable exception of [12, 2], there is still a lack of general techniques that support the systematic construction of widening operators. This is mainly due to the fact that the definition of widening provides extremely weak algebraic properties, while it is extremely demanding with respect to convergence and termination.

The focus of the paper is to give a comprehensive presentation of the basic theory on widening operators. We discuss and compare different definitions introduced in the literature, namely the notion of set-widening and the most known notion of pair-widening, and we investigate which properties are necessary to support a systematic design of widening operators. In particular, we prove that, for Galois Insertions, widening is preserved by abstraction, and we show how widening operators can be combined in the cartesian and reduced product of abstract domains.

The rest of this paper is organized as follows. The next section reports some preliminary notions. In Section 3, we analyze different notions of widening, and we show their weakness points and their mutual relations. In Section 4, we show how widening operators behave with respect to the combination of domains through Galois insertions. Finally, Section 5 concludes.

2 Basic Definitions

Let us briefly recall some basic definitions on orders and lattices [3, 14].

Definition 1 (poset) *If P is a non-empty set, then by a* partial order *on P we mean a binary relation $\leq$ on P which is reflexive, anti-symmetric, and transitive. By a* poset $(P, \leq)$ *we shall mean a set P on which there is defined a partial order $\leq$.*

Definition 2 (upper and lower bounds) *Let P be a poset, and let S be a subset of P. An element $x \in P$ is an* upper bound *of S if $s \leq x$ for all $s \in S$. If the set of the upper bounds of S has a least element z, then z is called the* least upper bound *(*lub*) of S, and will be denoted by $z = \sqcup S$.*
By duality, an element $x \in P$ is a lower bound *of S if $x \leq s$ for all $s \in S$. If the set of the lower bounds of S has a maximum element z, then z is called the* greatest lower bound *(*glb*) of S, and will be denoted by $z = \sqcap S$.*

Looking ahead, we shall often adopt the neater notation $x \sqcup y$ in place of $\sqcup\{x, y\}$, and $x \sqcap y$ in place of $\sqcap\{x, y\}$.

Definition 3 (directed set, cpo) *Let S be a subset of a poset $(P, \leq)$. Then S is said to be* directed *if for each pair of elements $x, y \in S$, there exists $z \in S$ such that $x \leq z$ and $y \leq z$.*

We say that a poset $(P, \leq)$ is a cpo *(complete partially ordered set) if P has a bottom element $\perp$, and $\sqcup D$ exists for each directed subset D of P.*

Definition 4 (ACC) *A poset $(P, \leq)$ is said to satisfy the ascending chain condition (ACC) if every ascending chain $x_1 \leq x_2 \leq \ldots$ of elements of P is eventually stationary, that is, there is some positive integer n such that $x_m = x_n$ for all $m > n$.*

Definition 5 (lattice) *Let P be a non empty poset. If $x \sqcup y$ and $x \sqcap y$ exist for all $x, y \in P$, then P is a* lattice. *Moreover, if $\sqcup S$ and $\sqcap S$ exist for every $S \subseteq P$, then P is a* complete lattice.

In what follows a function's domain and range are indicated by subscripts: ε_{XY} is a function from X to Y. The ordering and the least upper bound operator defined in X are denoted by $\sqsubseteq_X$ and $\sqcup_X$, respectively.

Definition 6 (Galois connection and insertion) *Let C and D be complete lattices, and consider two functions: $\gamma_{DC} : D \rightarrow C$ and $\alpha_{CD} : C \rightarrow D$. The tuple $G_{CD} = (\gamma_{DC}, C, D, \alpha_{CD})$ is a* Galois connection *if*

$$\forall c \in C \text{ and } \forall d \in D : \alpha_{CD}(c) \leq_D d \Leftrightarrow c \leq_C \gamma_{DC}(d).$$

G_{CD} is a Galois insertion *when γ_{DC} is injective or, equivalently, when α_{CD} is onto.*

In a Galois connection or insertion G_{CD}, the functions γ_{DC} and α_{CD} are called the concretization and the abstraction function, respectively. The following are well-known properties of these functions, see [11].

Lemma 1 *Let C and D be complete lattices, and consider two monotone functions $\gamma_{DC} : D \rightarrow C$ and $\alpha_{CD} : C \rightarrow D$. Then, G_{CD} is a Galois connection if and only if*

- *$\gamma_{DC} \circ \alpha_{CD}$ is extensive: $\forall c \in C,\ c \leq_C \gamma_{DC}(\alpha_{CD}(c))$;*
- *$\alpha_{CD} \circ \gamma_{DC}$ is reductive: $\forall d \in D,\ \alpha_{CD}(\gamma_{DC}(d)) \leq_D d$.*

Moreover, G_{CD} is a Galois insertion if it is a Galois connection and $\alpha_{CD} \circ \gamma_{DC}$ is the identity function.

Lemma 2 *Let G_{CD} be a Galois connection/insertion,*

- *if α_{CD} and γ_{DC} form a Galois connection, then one of the two functions determines the other one. More precisely, for $d \in D$, $\gamma_{DC}(d) = \sqcup_C\{c \in C \mid \alpha_{CD}(c) \sqsubseteq_D d\}$, and similarly, for $c \in C, \alpha_{CD}(c) = \sqcap_D\{d \in D \mid c \sqsubseteq_C \gamma_{DC}(d)\}$. Each function is called the* adjoint *of the other one.*
- *$\alpha_{CD} \circ \gamma_{DC} \circ \alpha_{CD} = \alpha_{CD}$, and $\gamma_{DC} \circ \alpha_{CD}\gamma_{DC} = \gamma_{DC}$.*

3 Widening Operators

In Abstract Interpretation, the collecting semantics of a program is expressed as a least fix-point of a set of equations. The equations are solved over some abstract domain that captures the property of interest to be analyzed. Typically, the equations are solved iteratively; that is, successive approximations of the solution is computed until a fix-point is reached. However, for many useful abstract domains, such chains can be either infinite or too long to let the analysis be efficient. To make use of these domains, abstract interpretation theory provides very powerful tools, the widening operators, that attempt to predict the fix-point based on the sequence of approximations computed on earlier iterations of the analysis on a cpo or on a (complete) lattice. The degradation of precision of the solution obtained by widening can be partly restored by further applying a narrowing operator [12].

3.1 Set- and Pair-Widening Operators

In the Abstract Interpretation literature, two different general definitions of widening operator have been introduced. The first one defines a widening operator as a partial function on the powerset of a poset P, while the second one defines it as a binary (total) function on a poset P. In both cases, two main requirements are given: first, the widening has to be an extrapolation operator, second, it has to guarantee termination when applied to increasing sequences.

Definition 7 (set-widening [10, 11]) *Let $(P, \leq)$ be a poset. A set-widening operator is a partial function $\nabla_\star : \wp(P) \nrightarrow P$ such that*

(i) Covering: Let S be an element of $\wp(P)$. If $\nabla_\star(S)$ is defined, then $\forall x \in S,\ x \leq \nabla_\star(S)$.

(ii) Termination: For every ascending chain $\{x_i\}_{i\geq 0}$, the chain defined as

$$y_0 = x_0,\ y_i = \nabla_\star(\{x_j \mid 0 \leq j \leq i\})$$

is ascending too, and it stabilizes after a finite number of terms.

The definition above has been used recently in [15, 16], for fix-point computations over sets represented as automata, in a model checking approach.

Example 1 *Consider the lattice of intervals $L = \{\bot\} \cup \{[\ell, u] \mid \ell \in \mathbb{Z} \cup \{-\infty\},\ u \in \mathbb{Z} \cup \{+\infty\},\ \ell \leq u\}$, ordered by: $\forall x \in L, \bot \leq x$ and $[\ell_0, u_0] \leq [\ell_1, u_1]$ if $\ell_1 \leq \ell_0$ and $u_0 \leq u_1$. Let k be a fixed positive integer constant, and I be any set of indices. Consider the threshold widening operator defined on L by:*

$$\nabla^k_\star(\{\bot\}) = \bot$$
$$\nabla^k_\star(\{\bot\} \cup S) = \nabla^k_\star(S)$$
$$\nabla^k_\star(\{[\ell_i, u_i] : i \in I\}) = [h_1, h_2]$$

where

$$h_1 = min\{\ell_i : i \in I\}\ if\ min\{\ell_i : i \in I\} > -k,\ else\ -\infty$$
$$h_2 = max\{u_i : i \in I\}\ if\ max\{u_i : i \in I\}) < k,\ else\ +\infty.$$

Observe that for all k, $\nabla^k_\star$ is associative, and monotone. However, it is not reflexive. For instance, we get $\nabla^7_\star(\{[-8, 4]\}) = [-\infty, 4]$.

Definition 8 (pair-widening [12], [26]) *Let $(P, \leq)$ be a poset. A pair-widening operator is a binary operator $\nabla : P \times P \to P$ such that*

(i) Covering: $\forall x, y \in P : x \leq x\nabla y$, and $y \leq x\nabla y$.

(ii) Termination: For every ascending chain $\{x_i\}_{i\geq 0}$, the ascending chain defined as

$$y_0 = x_0,\ y_{i+1} = y_i \nabla x_{i+1}$$

stabilizes after a finite number of terms.

Definition 9 (extrapolator) *Let $(P, \leq)$ be a poset. A binary operator $\bullet : P \times P \to P$ is called* extrapolator *if it satisfies the covering property, i.e. $\forall x, y \in P : x \leq x \bullet y$, and $y \leq x \bullet y$.*

Observe that pair-widening operators are not necessarily neither commutative neither monotone, nor associative, while these properties are crucial for chaotic iteration fix-point algorithms [26].

Example 2 *Consider the binary operator introduced in [9] on the same lattice of Intervals of Example 1:*

$$\begin{array}{rcl} \bot\nabla x &=& x \\ x\nabla\bot &=& x \\ [\ell_0,u_0]\nabla[\ell_1,u_1] &=& [\textit{if } \ell_1<\ell_0 \textit{ then } -\infty \textit{ else } \ell_0, \\ && \textit{if } u_0<u_1 \textit{ then } +\infty \textit{ else } u_0]. \end{array}$$

∇ is a pair-widening operator, as it satisfies both covering and termination requirements of Def.8.
Observe that the operator is not commutative, as for instance

$$\begin{array}{rcl} [2,3]\nabla[1,4] &=& [-\infty,+\infty] \\ [1,4]\nabla[2,3] &=& [1,4] \end{array}$$

Moreover, in order to see that it is not monotone, consider $[0,1]\leq[0,3]$. *We have:*

$$\begin{array}{rcl} [0,1]\nabla[0,2] &=& [0+\infty] \\ [0,3]\nabla[0,2] &=& [0,3]. \end{array}$$

and of course $[0,+\infty]$ *is not smaller or equal to* $[0,3]$. *Finally, observe that associativity does not hold either:*

$$\begin{array}{rcl} [0,2]\nabla([0,1]\nabla[0,2]) &=& [0+\infty] \\ ([0,2]\nabla[0,1])\nabla[0,2] &=& [0,2]. \end{array}$$

Let us come back to the two definitions of widening operators introduced before. As a first contribution, we see how to build a set-widening out of a pair-widening operator.

Theorem 1 *Let $(P,\leq)$ be a poset, and let $\nabla : P\times P\to P$ be a pair-widening operator on P. Define $\nabla_\star : \wp(P)\nrightarrow P$ such that:*

- *$dom(\nabla_\star)=R_1\cup R_2$, where*
  $R_1=\{\{x,y\}\mid x,y\in P\}$, and
  $R_2=\{S\subseteq P\mid S$ is a finite ascending chain$\}$.

- $\forall\{x,y\}\in R_1$,
  $\nabla_\star(\{x,y\}) =_{def}$
  $\begin{cases} x\nabla y & \textit{if } x\leq y \\ z\in\{x\nabla y, y\nabla x\} & \textit{randomly, otherwise.}\end{cases}$

- $\forall S=\{x_i\mid x_0\leq x_1\leq\cdots\leq x_j\}\in R_2$,
  $\nabla_\star(S) =_{def} (((x_0\nabla x_1)\nabla x_2\ldots)\nabla x_j)$.

Then $\nabla_\star$ is a set-widening operator.

Proof: We have to show that both covering and termination requirements hold for $\nabla_\star$.

- *Covering*. Let $S\subseteq P$ such that $\nabla_\star(S)$ is defined. We have to show that $\forall s\in S : s\leq\nabla_\star(S)$.
  Case $S\in R_1$: it follows from the definition of ∇.
  Case $S\in R_2$: it follows by induction on the length of the ascending chain, and by the transitivity of the partial order.

- *Termination*. Consider the ascending chain $\{x_i\}_{i\geq 0}$. Consider the corresponding ascending chain $\{\hat{y}_i\}_{i\geq 0}$ obtained by ∇ (see Def. 8), and the ascending chain $\{y_i\}_{i\geq 0}$ obtained using $\nabla_\star$ (see Def. 7). We can prove by induction that for each index i, $y_i=\hat{y}_i$.
  The basis is true, as $y_0=x_0=\hat{y}_0$.
  Consider the inductive step:

$$\begin{array}{rcl} y_{i+1} &=& \nabla_\star(\{x_j\mid 0\leq j\leq i+1\}) \\ && \quad\text{by (ii) of Def. 7} \\ &=& (((x_0\nabla x_1)\nabla x_2\ldots)\nabla x_{i+1}) \\ && \quad\text{by definition of } \nabla_\star \\ &=& \nabla_\star(\{x_j\mid 0\leq j\leq i\})\nabla x_{i+1} \\ && \quad\text{again by definition of } \nabla_\star \\ &=& \hat{y}_i\nabla x_{i+1} \\ && \quad\text{by inductive hypotesis} \\ &=& \hat{y}_{i+1} \\ && \quad\text{by (ii) of Def. 8} \end{array}$$

  As the sequence $\{\hat{y}_i\}_{i\geq 0}$ stabilizes after a finite number of terms, so does $\{y_i\}_{i\geq 0}$.

□

The notion of set-widening is weaker than the notion of pair-widening. This is why, in general, there is no way to prove the dual of Theorem 1, which can be stated only under restricted conditions.

Theorem 2 *Let $(P,\leq)$ be a poset, and let $\nabla_\star : \wp(P)\nrightarrow P$ be a set-widening operator on P such that*

- *$dom(\nabla_\star)\supseteq\{\{x,y\}\mid x,y\in P\}$, and*

- *$\forall S\subseteq P$, $\forall x\in P$, if $S\cup\{x\}\subseteq dom(\nabla_\star)$ then also $S\subseteq dom(\nabla_\star)$*

- *$\forall S\subseteq P$, $\forall x\in P$, $\nabla_\star(S\cup\{x\})=\nabla_\star(\{\nabla_\star(S),x\})$.*

Then, the binary operator $\nabla : P\times P\to P$ defined by $x\nabla y=\nabla_\star(\{x,y\})$ is a pair-widening operator.

Proof: First, observe that ∇ is well defined. The covering requirement follows immediately from the definition of ∇ and the covering property of $\nabla_\star$. Now, consider an ascending chain $\{x_i\}_{i\geq 0}$ in P, and the ascending chain $y_0=x_0, y_{i+1}=y_i\nabla x_i$. As $\nabla_\star$ is a set-widening, we know that the sequence $y'_0 = x_0, y'_i = \nabla_\star(\{x_j \mid 0 \leq j \leq i\}$ stabilizes finitely. We show by induction that for each i, $y_i = y'_i$. The basis is true, as $y_0 = x_0 = y'_0$. On the induction step,

$$\begin{array}{rcl} y'_{i+1} & = & \nabla_\star(\{x_j \mid 0 \leq j \leq i+1\} \\ & & \quad \text{by point (ii) of Def. 7} \\ & = & \nabla_\star(\{\nabla_\star(\{x_j \mid 0 \leq j \leq i\}), x_{i+1}\}) \\ & & \quad \text{by hypothesis on } \nabla_\star \\ & = & \nabla_\star(\{y'_i, x_{i+1}\}), \text{ by point (ii) of Def. 7} \\ & = & \nabla_\star(\{y_i, x_{i+1}\}), \text{by inductive hypothesis} \\ & = & y_i \nabla x_{i+1}, \text{ by definition of } \nabla \\ & = & y_{i+1}, \text{by point (ii) of Def. 8.} \end{array}$$

As the sequence $\{y'_i\}_{i \geq 0}$ stabilizes after a finite number of terms, so does $\{y_i\}_{i \geq 0}$. □

Observe that the set-widening operator $\nabla^k_\star$ of Example 1 satisfies the conditions of Theorem 2 above, yielding to a corresponding pair-widening operator.

3.2 Pair Widening and Cartesian Product

The next theorem shows that pair-widening operators can be combined when considering the cartesian product of posets.

Theorem 3 *Let ∇_A and ∇_D be pair-widening operators defined on the posets A and D, respectively. The binary operator $\nabla : (A \times D) \times (A \times D) \to (A \times D)$ defined by $\forall \langle a, d \rangle, \langle a', d' \rangle \in A \times D : \langle a, d \rangle \nabla \langle a', d' \rangle = \langle a \nabla_A a', d \nabla_D d' \rangle$ is a pair-widening operator.*

Proof:

- *Covering*

$$\begin{array}{rl} & a \leq a \nabla_A a' \text{ and } d \leq d \nabla_D d' \\ & \quad \text{by covering of } \nabla_A, \nabla_D \\ \Rightarrow & \langle a, d \rangle \leq \langle a \nabla_A a', d \nabla_D d' \rangle \\ & \quad \text{by definition of } \leq \text{ on } A \times D \\ \Rightarrow & \langle a, d \rangle \leq \langle a, d \rangle \nabla \langle a', d' \rangle \\ & \quad \text{by definition of } \nabla. \end{array}$$

- *Termination* Let $\{\langle a_i, d_i \rangle\}_{i \geq 0}$ be an ascending chain in the cartesian product $A \times D$. We have to show that the sequence $\langle u_0, v_0 \rangle = \langle a_0, d_0 \rangle$, $\langle u_{i+1}, v_{i+1} \rangle = \langle u_i, v_i \rangle \nabla \langle a_i, d_i \rangle$ stabilizes after a finite number of terms.

  By the termination property of ∇_A and ∇_D, both the sequence $\hat{a}_0 = a_0$, $\hat{a}_{i+1} = \hat{a}_i \nabla_A a_i$, and the sequence $\hat{d}_0 = d_0$, $\hat{d}_{i+1} = \hat{d}_i \nabla_D d_i$ stabilize finitely.

  It can be easily proved by induction that for each i, $\langle u_i, v_i \rangle = \langle \hat{a}_i, \hat{d}_i \rangle$. Therefore, the sequence $\{\langle u_j, v_j \rangle\}_{j \geq 0}$ stabilizes finitely too. □

3.3 Combination of pair-widening operators on the same poset

What happens when more than one widening operator is defined on a poset P? Is it possible to get a more precise and/or a more efficient widening operator by combining them in a suitable way? Unfortunately, in general the answer is negative. And the reason relies on the fact that the possibly non monotonic behavior of the widening operators becomes an issue when trying to prove termination of their combination on an ascending chain. However, as soon as stronger termination conditions are guaranteed on the poset P, some positive results can be easily derived.

Theorem 4 *Let $(P, \leq)$ be a lattice satisfying the ascending chain property. Let ∇_1, ∇_2 be two pair-widening operators on P. Then, the binary operators $\nabla_\sqcap, \nabla_\sqcup$ defined by*

$$\begin{array}{rcl} x \nabla_\sqcap y & = & (x \nabla_1 y) \sqcap (x \nabla_2 y) \\ x \nabla_\sqcup y & = & (x \nabla_1 y) \sqcup (x \nabla_2 y) \end{array}$$

are pair-widening operators.

Proof: It follows by properties of $\sqcup$ and $\sqcap$. □

This result may apply for instance to widening operators defined on the (infinite) domain of congruences [19], where prime factorization is an issue, in order to tune performance vs. accuracy of the analysis. In fact, $\nabla_\sqcup$ may gain in efficiency with respect to both ∇_1 and ∇_2, while $\nabla_\sqcap$ may better keep accuracy, thus returning a more accurate result.

3.4 Strong Pair-Widening Operators

For numerical domains like polyhedra, where the abstract elements computed at each iteration of the analysis are not necessarily ordered, a stronger notion of widening is used for forcing termination of the analysis. This is the case, for instance, of the trace partitioning abstract domain of Astrée, an abstract interpretation-based analyzer aiming at proving automatically the absence of run time errors in programs written in the C programming language, which has been applied with success to large safety critical real-time software for avionics [4, 8].

Definition 10 (strong pair-widening [28]) *Let $(P, \leq)$ be a poset. A strong pair-widening operator is a binary operator $\nabla : P \times P \to P$ such that* *(i) Covering:* $\forall x, y \in P : x \leq x \nabla y$*, and* $y \leq x \nabla y$*.*

(ii) Termination: For every sequence $\{x_i\}_{i \geq 0}$*, the ascending chain defined as* $y_0 = x_0,\ y_{i+1} = y_i \nabla x_{i+1}$ *stabilizes after a finite number of terms.*

Observe that this definition is strictly stronger than Definition 8, as termination is required starting from every (not necessarily increasing) sequence.

Example 3 *The octagon domain [24, 25] is based on invariants of the form* $\pm x \pm y \leq c$*, where* x *and* y *are numerical variables and* c *is a numeric constant. Sets described by such invariants are special kind of polyhedra called octagons because they feature at most eight edges in dimension 2. These constraints are expressed through Different Bound Matrices, which are adjacency matrices of weighted graphs. The widening operator defined on this domain consists on removing unstable constraints. In this case, termination has to be guaranteed for the chain of widened elements starting from a sequence of elements possibly incomparable. This is why the strongest notion of pair widening has to be used.*

The two notions of Pair-widening and Strong pair-widening are equivalent for a lattice P, under associativity conditions, as shown in Theorem 5. In order to prove it, we introduce the following auxiliary Lemma.

Lemma 3 *Let* ∇ *be a pair-widening operator on a lattice* $(P, \leq)$*, such that for every finite set* $\{x_i\}_{0 \leq i \leq n}$ *and for every* $y \in P$*,* $(((x_0 \nabla x_1) \nabla \dots) \nabla x_n)\ \nabla\ (x_0 \sqcup x_1 \sqcup \dots \sqcup x_n \sqcup y)$ $=$ $(((x_0 \nabla x_1) \nabla \dots) \nabla x_n) \nabla y$*, then* ∇ *is a strong pair-widening operator.*

Proof: We need to focus only on the termination property. Consider the sequence $\{x_i\}_{0 \leq i \leq n}$, and the increasing sequence

$$z_0 = x_0,\ z_{i+1} = x_0 \sqcup \dots \sqcup x_{i+1}$$

We show by induction that the two increasing sequences $y_0 = x_0,\ y_{i+1} = y_i \nabla x_{i+1}$ and $h_0 = z_0,\ h_{i+1} = h_i \nabla z_{i+1}$ are such that $\forall i : y_i = h_i$.
The basis is trivial, as $y_0 = x_0 = z_0 = h_0$.

The induction step:

$$\begin{aligned}
h_{i+1} &= h_i \nabla z_{i+1} \\
&\qquad \text{by def. of } \{h_j\}_{j \geq 0} \\
&= y_i \nabla z_{i+1} \\
&\qquad \text{by inductive hypothesis} \\
&= (((x_0 \nabla x_1) \nabla \dots) \nabla x_i) \nabla z_{i+1} \\
&\qquad \text{by def. of } \{y_j\}_{j \geq 0} \\
&= (((x_0 \nabla x_1) \nabla \dots) \nabla x_i) \nabla (x_0 \sqcup \dots \sqcup x_{i+1}) \\
&\qquad \text{by def. of } \{z_j\}_{j \geq 0} \\
&= (((x_0 \nabla x_1) \nabla \dots) \nabla x_i) \nabla x_{i+1} \\
&\qquad \text{by hypothesis on } \nabla \\
&= y_{i+1} \\
&\qquad \text{by def. of } \{y_j\}_{j \geq 0}
\end{aligned}$$

As the increasing sequence $\{h_j\}_{j \geq 0}$ stabilizes after a finite number of terms, so does $\{y_j\}_{j \geq 0}$. □

Theorem 5 *Let* ∇ *be an associative pair-widening operator on a lattice* $(P, \leq)$*, such that for* $\forall x, y \in P : x \nabla y = x \nabla (x \sqcup y)$*, then* ∇ *is a strong pair-widening operator.*

Proof: By Lemma 3, it is sufficient to prove by induction that for every finite set $\{x_i\}_{0 \leq i \leq n}$ and for every $y \in P$, $(((x_0 \nabla x_1) \nabla \dots) \nabla x_n)\ \nabla\ (x_0 \sqcup x_1 \sqcup \dots \sqcup x_n \sqcup y) = (((x_0 \nabla x_1) \nabla \dots) \nabla x_n) \nabla y$.
The basis ($n = 1$) follows immediately from the hypothesis.
Induction step:

$$\begin{aligned}
&(((x_0 \nabla x_1) \nabla \dots) \nabla x_n)\ \nabla\ (x_0 \sqcup \dots \sqcup x_n \sqcup y)\ = \\
&\qquad \text{by inductive hypothesis} \\
&(((x_0 \nabla x_1) \nabla \dots) \nabla (x_0 \sqcup \dots \sqcup x_n))\ \nabla\ (x_0 \sqcup \dots \sqcup x_n \sqcup y)\ = \\
&\qquad \text{by associativity of } \nabla \text{ and of } \sqcup \\
&((x_0 \nabla x_1) \nabla \dots) \nabla ((x_0 \sqcup \dots \sqcup x_n) \nabla ((x_0 \sqcup \dots \sqcup x_n) \sqcup y))\ = \\
&\qquad \text{by applying the hypothesis} \\
&((x_0 \nabla x_1) \nabla \dots) \nabla ((x_0 \sqcup \dots \sqcup x_n) \nabla y)\ = \\
&\qquad \text{by associativity of } \nabla \\
&(((x_0 \nabla x_1) \nabla \dots) \nabla x_n) \nabla y.
\end{aligned}$$

□

Example 4 *Observe that the pair-widening operator on intervals obtained from the set-widening of Example 1 following the construction of Theorem 2, satisfies the condition of Theorem 5, and it is in fact a strong pair widening operator. However, not every pair-widening operator is also a strong one. On the same lattice of intervals, consider for instance the pair-widening* ∇ *defined by:*

$$\bot \nabla x = x \qquad \textit{and} \qquad x \nabla \bot = x$$

$$[\ell_0, u_0]\nabla[\ell_1, u_1] =$$

$$= \begin{cases} [-\infty, +\infty] \\ \quad \textit{if } [\ell_0, u_0] \le [\ell_1, u_1] \textit{ or } [\ell_1, u_1] \le [\ell_0, u_0] \\ \\ [min(\ell_0, \ell_1), max(u_0, u_1)] \\ \quad \textit{otherwise} \end{cases}$$

On increasing sequences, the widened sequence terminates immediately, whereas if we consider for instance the sequence $\{[i, i+1]\}_{i\ge0}$, ∇ yields the ascending sequence $\{[0, i]\}_{i\ge1}$, which does not terminate.

4 Widening Operators and Galois Insertions

Widening operators have already been used in order to derive abstract domains [29]. The next results show how to derive Galois insertions by introducing an abstraction function built on top of a widening operator. In order to do that, additional requirements have to be assumed on the widening operator, like idempotence and order-preservation on pairs/singletons.

Theorem 6 *Let ∇ be a pair-widening operator on a complete lattice $(L, \le)$ such that $\forall x, y \in L : x \le y \Rightarrow x\nabla x \le y\nabla y$. Let A be the set $\{x\nabla x \mid x \in L\}$. Then $\alpha_{LA}(x) = x\nabla x$ is the lower adjoint of a Galois insertion between L and A, with the upper adjoint being the identity function.*

Proof: According to Def. 6, we have to show that $(\gamma_{AL}, L, A, \alpha_{LA})$ is a Galois insertion, with γ_{AL} being the identity function. By Lemma 1, it is sufficient to prove that $\forall x \in L : x \le \gamma_{AL}(\alpha_{LA}(x))$, and that $\forall a \in A : a = \alpha_{LA}(\gamma_{AL}(a))$.

$$\begin{array}{ll} \forall x \in L : & x \le x\nabla x, \text{ by (i) of Def. 8} \\ \quad\Rightarrow & x \le \alpha_{LA}(x), \text{ by definition of } \alpha_{LA} \\ \quad\Rightarrow & x \le \gamma_{AL}(\alpha_{LA}(x)), \text{ as } \gamma_{AL} \text{ is the identity} \end{array}$$

$$\begin{array}{ll} \forall a \in A : & a = a\nabla a, \text{ by definition of } A \\ \quad\Rightarrow & a = (\gamma_{AL}(a))\nabla(\gamma_{AL}(a)), \text{ as } \gamma_{AL} \text{ is the identity} \\ \quad\Rightarrow & a = \alpha_{LA}(\gamma_{AL}(a)), \text{ by definition of } \alpha_{LA} \end{array}$$

□

A corresponding result can be obtained also for set-widening operators.

Theorem 7 *Let $\nabla_\star$ be a set-widening operator on a complete lattice $(L, \le)$ such that $\nabla_\star(\{x\})$ is defined for each x in L, and such that $\forall x, y \in L : x \le y \Rightarrow \nabla_\star(\{x\}) \le \nabla_\star(\{y\})$. Let A be the set $\{\nabla_\star(\{x\}) \mid x \in L\}$. Consider the function $\alpha_{LA} : L \to A$ defined by $\alpha_{LA}(x) = \nabla_\star(\{x\})$. Then, α_{LA} is the lower adjoint of a Galois insertion between L and A, with the upper adjoint being the identity function.*

Proof: The proof is similar to the proof of Theorem 6. □

4.1 Pair-widening and abstraction

The following theorem shows that pair widening is preserved through abstraction.

Theorem 8 *Let C and D be two complete lattices, s.t. $G_{CD} = (\gamma_{DC}, C, D, \alpha_{CD})$ is a Galois insertion. Let ∇_C be a pair-widening on C. The binary operator ∇_D defined by $\forall d_1, d_2 \in D$, $d_1\nabla_D d_2 = \alpha_{CD}(\gamma_{DC}(d_1)\nabla_C\gamma_{DC}(d_2))$ is a pair-widening operator on D.*

Proof:

- *Covering*. Let us show that $\forall d_1, d_2 \in D : d_1 \le d_1\nabla_D d_2$.

$$\begin{array}{rcl} \gamma_{DC}(d_1) & \le & \gamma_{DC}(d_1)\nabla_C\gamma_{DC}(d_2) \\ & & \text{by (ii) of Def. 8} \\ \alpha_{CD}(\gamma_{DC}(d_1)) & \le & \alpha_{CD}(\gamma_{DC}(d_1)\nabla_C\gamma_{DC}(d_2)) \\ & & \text{by monotonicity of } \alpha_{CD} \\ \alpha_{CD}(\gamma_{DC}(d_1)) & \le & d_1\nabla_D d_2 \\ & & \text{by definition of } \nabla_D \\ d_1 & \le & d_1\nabla_D d_2 \\ & & \text{as } G_{CD} \text{ is a Galois insertion.} \end{array}$$

  The same way, we can also prove that $\forall d_1, d_2 \in D : d_2 \le d_1\nabla_D d_2$.

- *Termination*. Consider the ascending chain $\{d_i\}_{i\ge0}$ in D. Consider the corresponding ascending chain $\gamma_{DC}(d_0) \le \gamma_{DC}(d_1) \le \ldots$ in C. And consider the sequence $y_0 = \gamma_{DC}(d_0)$, $y_{i+1} = y_i\nabla_C\gamma_{DC}(d_{i+1})$. As ∇_C is a pair-widening operator, this ascending sequence stabilizes after a finite number of terms. We have to show that also the sequence $\hat{y}_0 = d_0$, $\hat{y}_{i+1} = \hat{y}_i\nabla_D d_{i+1}$ stabilizes after a finite number of terms. By induction, we prove that for each i, $\hat{y}_i = \alpha_{CD}(y_i)$.

  The basis is trivial, as $\hat{y}_0 = d_0 = \alpha_{CD}(\gamma_{DC}(d_0)) = \alpha_{CD}(y_0)$.

Looking at the inductive step,

$$\begin{array}{rcl}
\hat{y}_{i+1} & = & \hat{y}_i \nabla_{D} d_{i+1} \\
& & \quad \text{by definition of the sequence } \{\hat{y}_j\}_{j\geq 0}. \\
& = & \alpha_{CD}(y_i)\nabla_{D} d_{i+1} \\
& & \quad \text{by inductive hypotesis} \\
& = & \alpha_{CD}(y_i)\nabla_{D}\alpha_{CD}(\gamma_{DC}(d_{i+1})) \\
& & \quad \text{as } G_{CD} \text{ is a Galois insertion} \\
& = & \alpha_{CD}(y_i\nabla_{C}\gamma_{DC}(d_{i+1})) \\
& & \quad \text{by definition of } \nabla_{D} \\
& = & \alpha_{CD}(y_{i+1}) \\
& & \quad \text{by definition of the sequence } \{y_j\}_{j\geq 0}.
\end{array}$$

□

As a corollary of Theorem 8, we can prove that pair-widening operators are preserved also when projecting a cartesian product of lattices on one of its components.

Corollary 1 *Let A and D be complete lattices, and let ∇ be a pair-widening operator over the cartesian product $A \times D$. Let π_1 be the projection on the first argument. The binary operator $\nabla_{A} : A \times A \to A$ defined by*

$$a\nabla_{A}a' = \pi_1(\langle a, \top\rangle\nabla\langle a', \top\rangle)$$

is a pair-widening operator.

Proof: It is sufficient to observe that the monotone functions $\alpha : A \times D \to A$ and $\gamma : A \to A \times D$ defined by

$$\begin{array}{l}
\forall(a, d) \in A \times D : \ \alpha(\langle a, d\rangle) = a \\
\forall a \in A : \ \gamma(a) = \langle a, \top\rangle
\end{array}$$

form a Galois insertion between A and D. Therefore, by applying Theorem 8, the binary operator $\nabla' = \alpha(\gamma(a)\nabla\gamma(a'))$ is a pair widening operator on A. To conclude, it is sufficient to observe that $\nabla_{A} = \nabla'$. □

4.2 Pair-widening and Reduced Product

A very important operator to combine abstract domains in Abstract Interpretation, is the *reduced product* [10]. We have already seen in Theorem 3 that the pair-widening operators can be combined when considering the cartesian product of two posets. Unfortunately, this result cannot be fully extended to the reduced product, due to the fact that pair-widening operators in general are not required to be monotone. However, getting results relating widening operators in case of reduced product may have great impact on abstract domains used for the analysis of critical software. For instance, the octagon domain [25] can be seen as the reduced product of $2n^2$ abstract domains, each one of them focusing on an invariant of the form $\pm x \pm y \leq c$.

Definition 11 *Let C, A, D be complete lattices, and let $G_{CD} = (\gamma_{DC}, C, D, \alpha_{CD})$ and $G_{CA} = (\gamma_{AC}, C, A, \alpha_{CA})$ be Galois insertions.*
Consider the function reduce*:$A \times D \to A \times D$ defined by* reduce$(\langle a, d\rangle)$ = $\sqcap\{\langle a', d'\rangle \mid \gamma_{AC}(a)\sqcap\gamma_{DC}(d) = \gamma_{AC}(a')\sqcap \gamma_{DC}(d')\}$
The reduced product $A \sqcap D$ is defined as follows:

$$A \sqcap D = \{\text{reduce}(\langle a, d\rangle) \mid a \in A, d \in D\}.$$

Moreover, the function $\gamma : A \sqcap D \to C$ defined by $\gamma(\langle a, d\rangle) = \gamma_{AC}(a) \sqcap \gamma_{DC}(d)$ is the upper adjoint of a Galois insertion between $A \sqcap D$ and the domain C.

We can prove (Lemma 5) that by combining two pair-widening operators in the reduced product at least covering is preserved, i.e. we can obtain an extrapolation operator (which not necessarily terminates on ascending sequences, see for instance the domain of octagons [25]). The following auxiliary Lemma says that *reduce* well behaves with respect to the ordering in the reduced product $A \sqcap D$.

Lemma 4 *Let C, A, D be complete lattices, and let $G_{CD} = (\gamma_{DC}, C, D, \alpha_{CD})$ and $G_{CA} = (\gamma_{AC}, C, A, \alpha_{CA})$ be Galois insertions. For $\hat{a} \in A, \hat{d} \in D$, $\langle a, d\rangle \in A \sqcap D$, if $a \leq \hat{a}$ and $d \leq \hat{d}$, then $\langle a, d\rangle \leq$* reduce$(\langle\hat{a}, \hat{d}\rangle)$.

Proof: By $\sqcap$ properties and monotonicity of γ functions, $\gamma_{AC}(a) \sqcap \gamma_{DC}(d) \leq \gamma_{AC}(\hat{a}) \sqcap \gamma_{DC}(\hat{d})$. Therefore, *reduce*$(\langle\hat{a}, \hat{d}\rangle)$ is such that

$$\gamma(\langle a, d\rangle) \leq \gamma(\textit{reduce}(\langle\hat{a}, \hat{d}\rangle)$$

where γ is the upper adjoint of the Galois insertion $(\gamma, C, A \sqcap D, \alpha)$ as in Def. 11.
By applying α to both expressions, by monotonicity of α we get

$$\alpha(\gamma(\langle a, d\rangle)) \leq \alpha(\gamma(\textit{reduce}(\langle\hat{a}, \hat{d}\rangle)))$$

and by Galois insertion properties, as $\alpha \circ \gamma$ is the identity function, we get

$$\langle a, d\rangle \leq \textit{reduce}(\langle\hat{a}, \hat{d}\rangle)$$

□

Lemma 5 *Let C, A, D be complete lattices, and let $G_{CD} = (\gamma_{DC}, C, D, \alpha_{CD})$ and $G_{CA} = (\gamma_{AC}, C, A, \alpha_{CA})$ be Galois insertions.*
Let ∇_A and ∇_D be pair-widening operators defined on the lattice A and D, respectively.
The binary operator $\bullet : (A \sqcap D) \times (A \sqcap D) \to (A \sqcap D)$ defined by $\forall \langle a, d\rangle, \langle a', d'\rangle \in A \sqcap D : \langle a, d\rangle \bullet \langle a', d'\rangle =$ reduce$(\langle a \nabla_A a', d \nabla_D d'\rangle)$ *is an extrapolator operator.*

Proof: Let $\langle a, d\rangle, \langle a', d'\rangle \in A \sqcap D$. We have to prove that $\langle a, d\rangle \leq \langle a, d\rangle \bullet \langle a', d'\rangle$.

$$\begin{array}{rrcl} & \langle a, d\rangle & \leq & \langle a \nabla_A a', d \nabla_D d'\rangle \\ & & & \quad \text{by covering of } \nabla_A, \nabla_D \\ \Rightarrow & \langle a, d\rangle & \leq & \textit{reduce}(\langle a \nabla_A a', d \nabla_D d'\rangle) \\ & & & \quad \text{by Lemma 4} \\ \Rightarrow & \langle a, d\rangle & \leq & \langle a, d\rangle \bullet \langle a', d'\rangle \\ & & & \quad \text{by definition of } \bullet. \end{array}$$

In the same way, we can also prove that $\langle a', d'\rangle \leq \langle a, d\rangle \bullet \langle a', d'\rangle$. □

The last Theorem shows that in a reduced product, when the pair-widening operators on the two domains are not affected by *reduce*, the extrapolator of Lemma 5 enjoys also the termination property, thus resulting into a pair-widening operator too.

Theorem 9 *Let C, A, D be complete lattices, and let $G_{CD} = (\gamma_{DC}, C, D, \alpha_{CD})$ and $G_{CA} = (\gamma_{AC}, C, A, \alpha_{CA})$ be Galois insertions.*
Let ∇_A and ∇_D be pair-widening operators defined on the lattice A and D, respectively, such that $\forall \langle a, d\rangle \in A \sqcap D$, $\forall a' \in A, \forall d' \in D : \langle a \nabla_A a', d \nabla_D d'\rangle \in A \sqcap D$.
Then the binary operator $\nabla : (A \sqcap D) \times (A \sqcap D) \to (A \sqcap D)$ defined by $\forall \langle a, d\rangle, \langle a', d'\rangle \in A \sqcap D : \langle a, d\rangle \nabla \langle a', d'\rangle =$ reduce$(\langle a \nabla_A a', d \nabla_D d'\rangle)$ *is a pair-widening operator.*

Proof: By Lemma 5, we need to focus only on the termination property.
Consider the increasing sequence $\langle a_0, d_0\rangle \leq \langle a_1, d_1\rangle \dots$ in $A \sqcap D$. As the ordering $\leq$ in $A \sqcap D$ is the same as in the cartesian product $A \times D$, we may consider the increasing sequence $a_0 \leq a_1 \leq \dots$ in A, and the increasing sequence $d_0 \leq d_1 \leq \dots$ in D. By the termination property of ∇_A and ∇_D, we know that the corresponding sequences $\hat{a}_0 = a_0,\ \hat{a}_{i+1} = \hat{a}_i \nabla_A a_{i+1}$, and $\hat{d}_0 = d_0,\ \hat{d}_{i+1} = \hat{d}_i \nabla_D d_{i+1}$ stabilize after a finite number of terms.

We show by induction that the increasing sequence $\langle a'_0, d'_0\rangle = \langle a_0, d_0\rangle,\ \langle a'_{i+1}, d'_{i+1}\rangle = \langle a'_i, d'_i\rangle \nabla \langle a_{i+1}, d_{i+1}\rangle$ is such that $\forall i :\ \langle a'_i, d'_i\rangle = \langle \hat{a}_i, \hat{d}_i\rangle$.

The basis is trivial, as $\langle a'_0, d'_0\rangle = \langle a_0, d_0\rangle = \langle \hat{a}_0, \hat{d}_0\rangle$.
Induction step:

$$\begin{array}{rcl} \langle a'_{i+1}, d'_{i+1}\rangle & = & \langle a'_i, d'_i\rangle \nabla \langle a_{i+1}, d_{i+1}\rangle \\ & & \quad \text{by definition of } \{\langle a'_j, d'_j\rangle\}_{j \geq 0} \\ & = & \text{reduce}(a'_i \nabla_A a_{i+1}, d'_i \nabla_D d_{i+1}) \\ & & \quad \text{by def. of } \nabla \\ & = & \langle a'_i \nabla_A a_{i+1}, d'_i \nabla_D d_{i+1}\rangle \\ & & \quad \text{by the hypothesis} \\ & = & \langle \hat{a}_{i+1}, \hat{d}_{i+1}\rangle \\ & & \quad \text{by def. of } \{\hat{a}_j\}_{j \geq 0} \text{ and } \{\hat{d}_j\}_{j \geq 0} \end{array}$$

It follows that $\{\langle a'_j, d'_j\rangle\}_{j \geq 0}$ converges in a finite number of steps, namely the maximum between the termination indexes of $\{\hat{a}_j\}_{j \geq 0}$ and $\{\hat{d}_j\}_{j \geq 0}$. □

5 Conclusions and Future Work

We investigated which properties are necessary to support a systematic design of widening operators. As far as we know, this is the first attempt to provide a general comparison of the different notions of widening used in the literature and a first comprehensive discussion of their main features. More work deserves to be done in order to support a broader range of widening operators defined on abstract domains where only the concretization function is available or where the least upper bound operator is not always defined. We are currently investigating how to enhance domains and widening operators with suitable metrics that allow to get a quantitative comparison of their precision and/or of their speed to reach a fixed-point.

Aknowledgments

This paper was conceived while the author was visiting the École Normale Supérieure, invited by Patrick and Radhia Cousot. Many thanks to Xavier Rival, Pietro Ferrara, and Flemming Nielson for their comments on the a first draft of this work.
Work partially supported by PRIN 2007 MUR Project 200793N42R "SOFT - Tecniche formali orientate alla sicurezza".

References

[1] R. Bagnara, P. M. Hill, E. Ricci, and E. Zaffanella. Precise widening operators for convex polyhedra. *Science of Computer Programming*, 58(1-2):28–56, 2005.

[2] R. Bagnara, P. M. Hill, and E. Zaffanella. Widening operators for powerset domains. *Software Tools for Technology Transfer*, 8(4/5):449–466, 2006.

[3] Gareth Birkhoff. *Lattice Theory*. American Mathematical Society Colloquium Publications, Rhode Island, 1973.

[4] Bruno Blanchet, Patrick Cousot, Radhia Cousot, Jérome Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. A static analyzer for large safety-critical software. In *PLDI '03: Proc. of the ACM SIGPLAN 2003 conference on Programming language design and implementation*, pages 196–207, 2003.

[5] Agostino Cortesi, Baudouin Le Charlier, and Pascal Van Hentenryck. Combinations of abstract domains for logic programming: open product and generic pattern construction. *Science of Computer Programming*, 38(1–3):27–71, 2000.

[6] Agostino Cortesi, Gilberto Filé, Francesco Ranzato, Roberto Giacobazzi, and Catuscia Palamidessi. Complementation in abstract interpretation. *ACM Trans. Program. Lang. Syst.*, 19(1):7–47, 1997.

[7] Agostino Cortesi, Gilberto Filé, and William Winsborough. The quotient of an abstract interpretation. *Theoretical Computer Science*, 202(1-2):163 – 192, 1998.

[8] P. Cousot. Proving the absence of run-time errors in safety-critical avionics code. In C. Kirsch and R. Wilhelm, editors, *Proc. 7th ACM & IEEE International Conference on Embedded Sofware, Embedded Systems, (EMSOFT 2007)*, pages 7–9, Salzburg, Austria, 2007. ACM press.

[9] P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In *Proceedings of the Second International Symposium on Programming*, pages 106–130. Dunod, Paris, France, 1976.

[10] P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Conference Record of the Sixth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 269–282, San Antonio, Texas, 1979. ACM Press, New York, NY.

[11] P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, August 1992.

[12] P. Cousot and R. Cousot. Comparing the Galois connection and widening/narrowing approaches to abstract interpretation. In *Proc. Int. Workshop on Programming Language Implementation and Logic Programming*, volume 631 of *LNCS*, pages 269–295. Springer-Verlag, 1992.

[13] P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *5th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 84–97. ACM Press, 1978.

[14] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, 1990.

[15] Vijay D'Silva. Widening for automata. In *PhD thesis, Institut fur Informatik, Universitaat Zurich*, 2006.

[16] Vijay D'Silva, Mitra Purandare, and Daniel Kroening. Approximation refinement for interpolation-based model checking. In *VMCAI*, 2008.

[17] Jérôme Feret. Static analysis of digital filters. In *European Symposium on Programming (ESOP'04)*, number 2986 in LNCS. Springer-Verlag, 2004.

[18] Roberto Giacobazzi and Francesco Ranzato. The reduced relative power operation on abstract domains. *Theoretical Computer Science*, 216(1-2):159 – 211, 1999.

[19] P. Granger. Static analysis of linear congruence equalities among variables of a program. In *Int. Joint Conference on Theory and Practice of Software Development (TAPSOFT'91)*, volume 464 of *LNCS*, pages 169–192. Springer-Verlag, April 1991.

[20] P. Granger. Improving the results of static analyses programs by local decreasing iteration. In *Proceedings of FSTTCS*, volume 652 of *Lectures Notes in Computer Science*, pages 68–79. Springer-Verlag, 1992.

[21] Pascal Van Hentenryck, Agostino Cortesi, and Baudouin Le Charlier. Type analysis of prolog using type graphs. In *SIGPLAN Conference on Programming Language Design and Implementation*, pages 337–348, 1994.

[22] K. Rustan M. Leino and Francesco Logozzo. Using widenings to infer loop invariants inside an smt solver, or: A theorem prover as abstract domain. In *Workshop on Invariant Generation (WING 2007), Hagenberg, Austria, June 25-26*, 2007.

[23] Francesco Logozzo and Manuel Fahndrich. A weakly relational domain for the efficient validation of array accesses. In *23th ACM Symposium on Applied Computing (SAC 2008), Fortaleza, Brazil*, 2008.

[24] A. Miné. The octagon abstract domain. In *AST 2001 in WCRE 2001*, IEEE, pages 310–319. IEEE CS Press, October 2001.

[25] A. Miné. The octagon abstract domain. *Higher-Order and Symbolic Computation*, 19(1):31–100, 2006.

[26] F. Nielson, Riis H. Nielson, and C. L. Hankin. *Principles of Program Analysis*. Springer, second printing, 2005 edition, 1999.

[27] Viswanath Ramachandran, Pascal Van Hentenryck, and Agostino Cortesi. Abstract domains for reordering clp(rlin) programs. *J. Log. Program.*, 42(3):217–256, 2000.

[28] Xavier Rival and Laurent Mauborgne. The trace partitioning abstract domain. *ACM Trans. Program. Lang. Syst.*, 29(5):7–47, 2007.

[29] Arnaud Venet. Abstract cofibered domains: Application to the alias analysis of untyped programs. In *Proc. of the 3rd Int. Symposium on Static Analysis (SAS 96)*, pages 366–382. Springer-Verlag, 1996.