Comparing the Galois Connection and Widening/Narrowing Approaches to Abstract Interpretation*

Patrick Cousot[1] and Radhia Cousot[2]

[1] LIENS, DMI, École Normale Supérieure, 45, rue d'Ulm, 75230 Paris cedex 05 (France)
cousot@dmi.ens.fr
[2] LIX, École Polytechnique, 91128 Palaiseau cedex (France)
radhia@polytechnique.fr

Abstract. The use of infinite abstract domains with widening and -narrowing for accelerating the convergence of abstract interpretations is shown to be more powerful than the Galois connection approach restricted to finite lattices (or lattices satisfying the chain condition).

1 Introduction

A widely-held opinion is that finite lattices (or lattices satisfying the chain condition, i.e., such that all strictly increasing chains are finite) can be used instead of widenings and narrowings to ensure the termination of abstract interpretations of programs on infinite lattices. We show that, in general, this can only be to the detriment of precision and prove that the use of infinite abstract domains with widenings and narrowings is more powerful than the Galois connection approach for finite lattices (or lattices satisfying the chain condition). By way of example, various widenings are suggested for solving non-convergence problems left open in the literature.

2 Upper Approximation of the Collecting Semantics

Following [CC76,CC77a,CC79b], the abstract interpretation of a program can be formalized as the effective computation of an upper approximation A of the *collecting semantics* of the program.

This collecting semantics can often be specified as the least fixed point $\mathrm{lfp}_{\perp}(F)$ of a continuous[1] operator $F \in L \xrightarrow{\text{con}} L$ on a cpo $L(\sqsubseteq, \sqcup)$ greater than a *basis* $\perp$ satisfying $\perp \sqsubseteq F(\perp)$[2]. By Kleene fixpoint theorem (Prop. 23 in the appendix), $\mathrm{lfp}_{\perp}(F)$ is the least upper bound $\bigsqcup_{n\in\mathbb{N}} F^n(\perp)$ of the *iterates* $F^n(\perp)$ defined by $F^0(x) \stackrel{\text{def}}{=} x$ and $F^{n+1}(x) \stackrel{\text{def}}{=} F(F^n(x))$ for all $x \in L$.

* This work was supported in part by Esprit BRA action 3124 "Sémantique".

[1] Monotony is sufficient by considering transfinite iterations [CC79a].

[2] The basis $\perp$ is often the infimum $\perp$ of the cpo, in which case $\mathrm{lfp}_{\perp} F$ is written $\mathrm{lfp}\, F$.

This approximation A must be *sound* in the sense that $\mathrm{lfp}(F) \sqsubseteq A$[3].

Example 1 ((Imperative programs)). Assume that the collecting semantics of the following PASCAL program:

```
program P;
   var I : integer ;
begin
   I := 1;
   while I <= 100 do
      begin
         { I ∈ [1, 100] }
         I := I + 1;
      end;
   { I = 101 }
end.
```

is the set of possible values of integer variable `I` when starting execution of the loop body. It is the least fixed point $\mathrm{lfp}(F) = \mathrm{lfp}_{\emptyset}(F) = \{i \in \mathbb{Z} \mid 1 \leq i \leq 100\}$ of the continuous (and even additive) operator:

$$F \in L \xrightarrow{\text{con}} L \;=\; \lambda\, X \cdot (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leq 100\} \qquad (1)$$

on the complete lattice $L = \wp(\mathbb{Z})(\subseteq, \emptyset, \mathbb{Z}, \cap, \cup)$ where $\mathbb{Z}$ is the set of integers and $\wp(S)$ is the powerset of the set S.

A sound upper approximation is the loop invariant $A = \{i \in \mathbb{Z} \mid i \geq 0\}$ specifying that `I` is non-negative. □

Example 2 ((Logic programs)). Let P be a logic program (containing at least one constant), B_P be its Herbrand universe over a family $\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{F}^n$ of n-ary functors $f \in \mathcal{F}^n$ and $\mathrm{ground}(P)$ be the set of all ground instances of clauses in P. The *immediate consequence operator* is $T_P \in \wp(B_P) \xrightarrow{\text{con}} \wp(B_P)$ such that:

$$T_P = \lambda\, X \cdot \{A \mid A \leftarrow B_1, \ldots, B_n \in \mathrm{ground}(P) \wedge \forall i = 1, \ldots, n : B_i \in X\} \;.$$

A *model* of P is a set $I \subseteq B_P$, such that $T_P(I) \subseteq I$. The characterization theorem of van Emden and Kowalski [vEK76] shows that P has a least model M_P in the complete lattice $\wp(B_P)(\subseteq, \emptyset, \cup)$ such that $M_P = \mathrm{lfp}_{\emptyset}\, T_P = \bigcup_{n \in \mathbb{N}} {T_P}^n(\emptyset)$. □

Example 3 ((Functional programs)). Following [CC92c], the *relational semantics* of the functional factorial program:

`f(n)` $\equiv$ **if** `n` $=$ `0` **then** `1` **else** `n` $*$ `f(n` $-$ `1)`;

is $f \in \wp(\mathbb{N}_\perp \times \mathbb{N}_\perp)$, where $\mathbb{N}_\perp \stackrel{\text{def}}{=} \mathbb{N} \cup \{\perp\}$ and $\perp$ represents non-termination, such that: $f = \{\langle \perp, \perp \rangle\} \cup \{\langle n, \perp \rangle \mid n < 0\} \cup \{\langle n, n! \rangle \mid n \geq 0\}$. It is the least fixpoint $\mathrm{lfp}_{\perp}\, F$ of $F \in \wp(\mathbb{N}_\perp \times \mathbb{N}_\perp) \xrightarrow{\text{con}} \wp(\mathbb{N}_\perp \times \mathbb{N}_\perp)$ such that:

$$F(f) = \{\langle \perp, \perp \rangle\} \cup \{\langle 0, 1 \rangle\} \cup \{\langle n, n * \rho \rangle \mid \langle n-1, \rho \rangle \in f\}$$

[3] Although commonly satisfied, these hypotheses on the definition of the collecting semantics and the specification of the approximation are stronger than strictly necessary, see a discussion of various weaker hypotheses in [CC92b].

where $\bot - \rho = \rho - \bot = \bot$ and $\bot * \rho = \rho * \bot = \bot$. The semantic domain $\wp(\mathbb{N}_\bot \times \mathbb{N}_\bot)(\sqsubseteq, \pm, \mp, \sqcup)$ is a complete lattice, where:

$$\begin{array}{rcl} \pm & \stackrel{\text{def}}{=} & \mathbb{N}_\bot \times \{\bot\} \\ \mp & \stackrel{\text{def}}{=} & \mathbb{N}_\bot \times \mathbb{N} \\ f \sqsubseteq f' & \stackrel{\text{def}}{=} & (f \cap \mp) \subseteq (f' \cap \mp) \ \wedge \ (f \cap \pm) \supseteq (f' \cap \pm) \\ \sqcup_{i\in\Delta} f_i & \stackrel{\text{def}}{=} & \cup_{i\in\Delta}(f_i \cap \mp) \ \cup \ \cap_{i\in\Delta}(f_i \cap \pm) \ . \end{array}$$

Observe that $f \sqsubseteq f'$ if and only if f' produces more output results in $\mathbb{N}$ that f for a given terminating or non-terminating argument in $\mathbb{N}_\bot$ and f' terminates more frequently than f. □

3 The Galois Connection Approach to Abstract Interpretation

Principle of the Approach. The Galois connection approach to abstract interpretation [CC76,CC77a] formalizes the idea that the equation $X = F(X)$ can be first simplified into $\overline{X} = \overline{F}(\overline{X})$, where $\overline{F} \in \overline{L} \stackrel{\text{mon}}{\longmapsto} \overline{L}$ and $\overline{L}(\overline{\sqsubseteq}, \overline{\sqcup})$ is a poset, and then solved iteratively starting from the basis $\overline{\pm}$. The technique consists in understanding $\overline{L}$ as a discrete approximation of L and in extending this notion of approximation, in various ways, to semantic domains such as products $L \times L$, powersets $\wp(L)$ and function spaces $L \longmapsto L$ [CC77b,CC79b].

Galois Connection. The correspondence between the semantic domain L and its abstract version $\overline{L}$ can be formalized by a Galois connection (also called *pair of adjoined functions*).

Definition 4. * If L $(\sqsubseteq)$ and $\overline{L}$ $(\overline{\sqsubseteq})$ are posets, then $\langle \alpha, \gamma \rangle$ is a *Galois connection*, written $L \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \overline{L}$, if and only if $\alpha \in L \longmapsto \overline{L}$ and $\gamma \in \overline{L} \longmapsto L$ are functions such that:

$$\forall x \in L, \overline{y} \in \overline{L} : \left(\alpha(x) \ \overline{\sqsubseteq} \ \overline{y}\right) \iff (x \sqsubseteq \gamma(\overline{y})) \ . \tag{2}$$

$\alpha(x)$ is the *abstraction* of x, i.e., the most precise approximation of $x \in L$ in $\overline{L}$. $\gamma(\overline{y})$ is the *concretization* of $\overline{y}$, i.e., the most imprecise element of L which can be soundly approximated by $\overline{y} \in \overline{L}$.

Example 5 ((Intervals)). In [CC76], $\wp(\mathbb{Z})$ ordered by $\subseteq$ is approximated using the abstract lattice of intervals $\overline{L} = \{\bot\} \cup \{[\ell, u] \mid \ell \in \mathbb{Z} \cup \{-\infty\} \wedge u \in \mathbb{Z} \cup \{+\infty\} \wedge \ell \leq u\}$ ordered by $\overline{\sqsubseteq}$, such that:

$$\begin{array}{rcl} \bot & \overline{\sqsubseteq} & [\ell, u] \ \stackrel{\text{def}}{=} \ \text{true} \\ [\ell_0, u_0] & \overline{\sqsubseteq} & [\ell_1, u_1] \ \stackrel{\text{def}}{=} \ \ell_1 \leq \ell_0 \leq u_0 \leq u_1 \ . \end{array} \tag{3}$$

This approximation is formalized by the Galois connection defined by:

$$\begin{array}{ll} \gamma(\bot) = \emptyset & \alpha(\emptyset) = \bot \\ \gamma([\ell, u]) = \{x \in \mathbb{Z} \mid \ell \leq x \leq u\} & \alpha(X) = [\min X, \max X] \ . \end{array}$$

For example the set $\{1, 2, 5\} \in \wp(\mathbb{Z})$ is soundly approximated by $[1, 5] \in \overline{L}$. □

Soundness and Precision. Here, the concrete and abstract notions of soundness and precision are formalized in the same way, by the respective partial orders $\sqsubseteq$ on L and $\overline{\sqsubseteq}$ on $\overline{L}$. $x \sqsubseteq y$ is interpreted as "y is a sound approximation of x", "x is a more precise concrete assertion than y" or "x logically implies y". The same way $\overline{x} \mathrel{\overline{\sqsubseteq}} \overline{y} \mathrel{\overline{\sqsubseteq}} \overline{z}$ means that $\overline{y}$ and $\overline{z}$ are sound approximations of $\overline{x}$ but $\overline{y}$ is more precise than $\overline{z}$. We may have $\overline{x} \mathrel{\overline{\sqsubseteq}} \overline{y}$ and $\overline{x} \mathrel{\overline{\sqsubseteq}} \overline{z}$ but neither $\overline{y} \mathrel{\overline{\sqsubseteq}} \overline{z}$ nor $\overline{z} \mathrel{\overline{\sqsubseteq}} \overline{y}$ in which case $\overline{y}$ and $\overline{z}$ are non-comparable sound approximations of $\overline{x}$. Equation (2) states that the concrete and abstract notions of soundness and precision coincide, up to an approximation, which consists in representing several concrete assertions $\{x \mid \alpha(x) = \overline{x}\}$ by the same abstract assertion $\overline{x}$.

Example 6 ((Intervals, continued)). For intervals considered in Ex. 5, the concrete approximation relation $\sqsubseteq$ is subset inclusion $\subseteq$ whereas the abstract approximation relation $\overline{\sqsubseteq}$ is defined by (3). For example, $\{1\,,\,2\,,\,5\} \subseteq \{i \in \mathbb{Z} \mid i \geq 1\}$ and $\{1\,,\,2\,,\,5\} \subseteq \{i \in \mathbb{Z} \mid i \leq 5\}$ since the assertion that the value of a variable can only be 1, 2 or 5 during execution is more precise than saying that it is strictly positive. These assertions are respectively abstracted by $[1,\,5] \mathrel{\overline{\sqsubseteq}} [1,\,+\infty]$ and $[1,\,5] \mathrel{\overline{\sqsubseteq}} [-\infty,\,5]$ but these approximations are not comparable since $[1,\,+\infty] \mathrel{\overline{\not\sqsubseteq}} [-\infty,\,5]$ and $[-\infty,\,5] \mathrel{\overline{\not\sqsubseteq}} [1,\,+\infty]$. $[1,\,5]$ is the best possible abstract approximation of the concrete assertion $\{1\,,\,2\,,\,5\}$. □

Extension to Function Spaces. The concrete approximation relation $\sqsubseteq \in \wp(L \times L)$ can be extended to the function space $L \longmapsto L$ pointwise, i.e., $F \sqsubseteq F' \stackrel{\text{def}}{=} \forall x \in L : F(x) \sqsubseteq F'(x)$. The intuition is that F is more precise than F' if and only if F always yields more precise results than F'.

Then, the approximation of L by $\overline{L}$ can be extended to the approximation of the function space $L \longmapsto L$ by $\overline{L} \longmapsto \overline{L}$ using the functional abstraction $\vec{\alpha}$ and concretization $\vec{\gamma}$ defined, as in [CC77b], by:

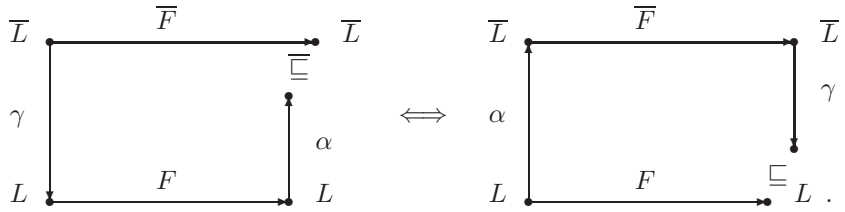
$$\begin{array}{ll} \vec{\alpha} \in (L \longmapsto L) \longmapsto (\overline{L} \longmapsto \overline{L}) & \vec{\gamma} \in (\overline{L} \longmapsto \overline{L}) \longmapsto (L \longmapsto L) \\ \vec{\alpha}(\varphi) \stackrel{\text{def}}{=} \alpha \circ \varphi \circ \gamma & \vec{\gamma}(\overline{\varphi}) \stackrel{\text{def}}{=} \gamma \circ \overline{\varphi} \circ \alpha \end{array} \tag{4}$$

such that, by Prop. 25 in the appendix:

$$(L \stackrel{\text{mon}}{\longmapsto} L) \underset{\vec{\alpha}}{\overset{\vec{\gamma}}{\leftrightarrows}} (\overline{L} \stackrel{\text{mon}}{\longmapsto} \overline{L}) \ . \tag{5}$$

Intuitively, $\vec{\alpha}(F)$ is the abstract image of F up to the Galois connection $L \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \overline{L}$. It follows, by Prop. 30 in the appendix, that if $L(\sqsubseteq, \sqcup)$ is a poset, $F \in L \stackrel{\text{con}}{\longmapsto} L$, and $\overline{\bot}$ is $\alpha(\bot)$, then $\mathrm{lfp}_{\bot}(F) \sqsubseteq \gamma\left(\mathrm{lfp}_{\overline{\bot}}(\vec{\alpha}(F))\right)$. Otherwise stated, the fixpoint operator lfp preserves the soundness of the approximation [CC77b].

Functional Abstraction. In practice $\vec{\alpha}(F)$ may not be easy to program. In this case we can use an upper approximation $\overline{F}$. More precisely, $\overline{F} \in (\overline{L} \longmapsto \overline{L})$ is an *abstraction* of $F \in (L \stackrel{\text{con}}{\longmapsto} L)$ if and only if $\vec{\alpha}(F) \mathrel{\overline{\sqsubseteq}} \overline{F}$ or, equivalently, $F \sqsubseteq \vec{\gamma}(\overline{F})$. Diagrammatically:



Intuitively, $\overline{F}(\overline{x})$ is an approximation of $F(x)$ when applied to an approximation $\overline{x}$ of x.

Definition 7. * $\langle \overline{L}, \overline{\bot}, \overline{F} \rangle$ is an *abstract interpretation* of $\langle L, \bot, F \rangle$, written $\langle L, \bot, F \rangle \xleftarrow[\alpha]{\gamma} \langle \overline{L}, \overline{\bot}, \overline{F} \rangle$, if and only if $L \xleftarrow[\alpha]{\gamma} \overline{L}$, $\alpha(\bot) \mathrel{\overline{\sqsubseteq}} \overline{\bot}$ and $\vec{\alpha}(F) \mathrel{\overline{\sqsubseteq}} \overline{F}$[4].

If $\langle L, \bot, F \rangle \xleftarrow[\alpha]{\gamma} \langle \overline{L}, \overline{\bot}, \overline{F} \rangle$ and $\overline{A}$ is an upper bound of the abstract iterates $\overline{F}^n(\overline{\bot})$, $n \in \mathbb{N}$, then $\mathrm{lfp}_\bot(F) \sqsubseteq \gamma(\overline{A})$, as shown by Prop. 31 in the appendix[5]. Otherwise stated any upper bound of the abstract iterates is a sound approximation of the collecting semantics.

Example 8 ((Intervals, continued)). Given the interval abstraction of Ex. 5, the approximate equation $\overline{X} = \overline{F}(\overline{X})$ corresponding to (1) for program P is defined by:

$$\overline{F} \in \overline{L} \overset{\text{mon}}{\longmapsto} \overline{L} = \lambda\, X \cdot ([1,\, 1] \sqcup (X \oplus [1,\, 1])) \sqcap [-\infty,\, 100]$$

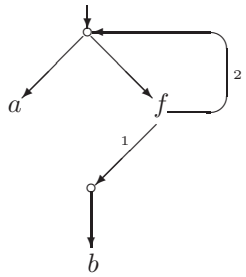
where $\bot \sqcup X = X \sqcup \bot = X$, $[\ell_0,\, u_0] \sqcup [\ell_1,\, u_1] = [\min(\ell_0,\, \ell_1),\, \max(u_0,\, u_1)]$, $\bot \sqcap X = X \sqcap \bot = \bot$, $[\ell_0,\, u_0] \sqcap [\ell_1,\, u_1] =$ if $\max(\ell_0,\, \ell_1) > \min(u_0,\, u_1)$ then $\bot$ else $[\max(\ell_0,\, \ell_1),\, \min(u_0,\, u_1)]$, $\bot \oplus X = X \oplus \bot = \bot$ and $[\ell_0,\, u_0] \oplus [\ell_1,\, u_1] = [\ell_0 + \ell_1,\, u_0 + u_1]$. It can be solved iteratively starting from the infimum $\bot$. The successive iterates are $\bot$, $[1,\, 1]$, $[1,\, 2]$, $\ldots$, $[1,\, 100]$. This sequence might be infinite and strictly increasing (e.g. for nonterminating programs). □

In practice, finite convergence of the abstract iterates $\overline{F}^n(\overline{\bot})$, $n \in \mathbb{N}$ must be ensured. This leads to hypotheses on $\overline{L}$ and $\overline{F}$ such as, e.g., $\overline{L}$ is finite or $\overline{F}^n(\overline{\bot})$, $n \in \mathbb{N}$ is an increasing chain and no strictly increasing chain in $\overline{L}$ can be infinite (i.e. $\overline{L}$ satisfies the so-called *ascending chain condition*). Observe that various hypotheses ensure that $\overline{F}^n(\overline{\bot})$, $n \in \mathbb{N}$ is an increasing chain. For example, $\overline{F}$ might be *extensive* (i.e., $\forall \overline{x} \in \overline{L} : \overline{x} \mathrel{\overline{\sqsubseteq}} \overline{F}(\overline{x})$) or $\overline{\bot}$ may be a *prefixpoint* of $\overline{F}$ (i.e., $\overline{\bot} \mathrel{\overline{\sqsubseteq}} \overline{F}(\overline{\bot})$) and $\overline{F} \in \overline{L} \overset{\text{mon}}{\longmapsto} \overline{L}$ may be monotone. For more details or equivalent approaches, see [CC79b], [Cou78, chapter 4] and [CC92a].

Example 9 ((Descriptive types)). In Prolog type analysis of Bruynooghe et al. [BJCD87,JB92], a set of ground terms is approximated by a *type graph* such as the following one (where a and b are constants of arity 0 and f is a binary functor):

[4] $\alpha(\bot) \mathrel{\overline{\sqsubseteq}} \overline{\bot}$ is equivalent to $\bot \sqsubseteq \gamma(\overline{\bot})$ and $\vec{\alpha}(F) \mathrel{\overline{\sqsubseteq}} \overline{F}$ is equivalent to $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$ or to $\alpha \circ F \mathrel{\overline{\sqsubseteq}} \overline{F} \circ \alpha$ (see Prop. 26 in the appendix), so that we can dispense with either α or γ, [CC92b].

[5] which is the case for $\overline{A} = \mathrm{lfp}^{\overline{\sqsubseteq}}_{\overline{\bot}} \overline{F}$ whenever this least fixpoint exists.



A type graph $G \in \mathcal{G}$ is a finite bipartite graph, consisting of:

1. A finite set N_t of type nodes (marked $\circ$ in diagrams),
2. A finite set N_f of functor nodes m, labeled with n-ary functors $f(m) \in \mathcal{F}^n$, and such that $N_t \cap N_f = \emptyset$,
3. A root $r \in N_t$ such that there is a path from r to any node of G,
4. A set $A \in \wp(N_t \times N_f) \cup \wp(N_f \times \mathbb{N} \times N_t)$ of arcs, such that:
   (a) All type nodes $k \in N_t$ have at least one outgoing arc and all outgoing arcs $\langle k,\, m\rangle$ go to functors nodes $m \in N_f$ with distinct labels $f(m)$,
   (b) All functors nodes $m \in N_f$ labeled with a functor $f(m) \in \mathcal{F}^n$ of arity $n \in \mathbb{N}$ have n outgoing arcs $\langle m,\, i,\, k^i\rangle$, $1 \le i \le n$ (so that there is no outgoing arc when $n = 0$).

We write $k : g(k^1, \ldots, k^n)$ for $\exists m \in N_f$: $\langle k,\, m\rangle \in A \wedge f(m) = g \in \mathcal{F}^n \wedge \forall i \in [1,\, n]$: $\langle m,\, i,\, k^i\rangle \in A$ and say that type nodes $k^1, \ldots, k^n$ are the *sons* of node k. A ground term $t \in B_P$ is said to *fold on type node k of type graph G*, if and only if:

1. $t = c \in \mathcal{F}^0$ and $k : c$,
2. $t = g(t_1, \ldots, t_n)$, $k : g(k^1, \ldots, k^n)$ and each ground term t_i, $1 \le i \le n$ folds on type node k^i of graph G.

The concretization function is defined by:

$$\gamma(G) = \{t \in B_P \mid t \text{ folds on the root of } G\}$$

For the type graph G above, we have $\gamma(G) = \{a,\, f(b,\, a),\, f(b,\, f(b,\, a)),\, f(b,\, f(b,\, f(b,\, a))),\, \ldots\}$.

Define the equivalence relation $G \equiv G'$ by $\gamma(G) = \gamma(G')$. The partial order relation $\sqsubseteq$ on $\overline{L} = \mathcal{G}/_{\equiv}$ is defined by $G \sqsubseteq G'$ if and only if $\gamma(G) \subseteq \gamma(G')$. We have $G \sqsubseteq G'$ if and only if all paths in G exist in G', which can be checked by path-finding algorithms. □

Example 10 ((Strictness analysis)). In Mycroft's strictness analysis [Myc80], a relation $f \in \wp(\mathbb{N}_\perp \times \mathbb{N}_\perp)$ is approximated by a function $f^\sharp \in \{0, 1\} \xrightarrow{\text{mon}} \{0, 1\}$ such that $0 \sqsubseteq 1$ and $f^\sharp(0) = 0$ only if f is *strict*, that is: $\forall \rho \in \mathbb{N}_\perp$: $\langle \perp,\, \rho\rangle \in f \Longrightarrow \rho = \perp$. This approximation is formalized by the Galois connection defined by:

$$\begin{array}{ll|ll}
\gamma(\lambda\, x{\cdot}\, 0) = \mathbb{N}_\perp \times \{\perp\} & & \alpha(f) = \lambda\, x{\cdot}\, 0 & \text{if } f = \mathbb{N}_\perp \times \{\perp\} \\
\gamma(\lambda\, x{\cdot}\, x) = \{\langle \perp,\, \perp\rangle\} \cup \mathbb{N} \times \mathbb{N}_\perp & & \alpha(f) = \lambda\, x{\cdot}\, x & \text{if } \langle \perp,\, \rho\rangle \in f \Longrightarrow \rho = \perp \\
\gamma(\lambda\, x{\cdot}\, 1) = \mathbb{N}_\perp \times \mathbb{N}_\perp & & \alpha(f) = \lambda\, x{\cdot}\, 1 & \text{otherwise .}
\end{array}$$

This abstract interpretation can be lifted to higher-order functions using (4). □

4 The Widening/Narrowing Approach to Abstract Interpretation

Another method [CC76,CC77a] for enforcing termination of the abstract interpretation consists in using a *widening* $\triangledown \in L \times L \longmapsto L$ such that:

$$\forall x, y \in L : x \sqsubseteq x \triangledown y \tag{6}$$

$$\forall x, y \in L : y \sqsubseteq x \triangledown y \tag{7}$$

for all increasing chains $x^0 \sqsubseteq x^1 \sqsubseteq \ldots$, the increasing chain defined by $y^0 = x^0, \ldots, y^{i+1} = y^i \triangledown x^{i+1}, \ldots$ is not strictly increasing . (8)

It follows, as shown by Prop. 33 in the appendix, that the *upward iteration sequence with widening*:

$$\begin{array}{rll} \hat{X}^0 &= \bot & \\ \hat{X}^{i+1} &= \hat{X}^i & \text{if } F(\hat{X}^i) \sqsubseteq \hat{X}^i \\ &= \hat{X}^i \triangledown F(\hat{X}^i) & \text{otherwise} \end{array} \tag{9}$$

is ultimately stationary and its limit $\hat{A}$ is a sound upper approximation of $\mathrm{lfp}_{\bot}(F)$[6]. Observe that if L is a join-semi-lattice (the least upper bound $x \sqcup y$ exists for all x, $y \in L$) satisfying the ascending chain condition, then $\sqcup$ is a widening.

This approximation can then be improved using a *narrowing* operator $\triangle \in L \times L \longmapsto L$ such that:

$$\forall x, y \in L : (y \sqsubseteq x) \implies (y \sqsubseteq (x \triangle y) \sqsubseteq x) \tag{10}$$

for all decreasing chains $x^0 \sqsupseteq x^1 \sqsupseteq \ldots$, the decreasing chain defined by $y^0 = x^0, \ldots, y^{i+1} = y^i \triangle x^{i+1}, \ldots$ is not strictly decreasing . (11)

It follows, as shown by Prop. 34 in the appendix, that the *downward abstract iteration sequence with narrowing*:

$$\begin{array}{l} \check{X}^0 = \hat{A} \\ \check{X}^{i+1} = \check{X}^i \triangle F(\check{X}^i) \end{array} \tag{12}$$

is ultimately stationary and its limit $\check{A}$ as well as each term $\check{X}^i$ of this decreasing chain is a sound upper approximation of $\mathrm{lfp}_{\bot}(F)$. Observe that if $F(\check{X}^i) = \check{X}^i$

[6] Numerous variants are possible. For example, we might assume $x \sqsubseteq y$ in (6) and (7), and use $\hat{X}^{i+1} = \hat{X}^i \triangledown (\hat{X}^i \sqcup F\left(\hat{X}^i\right))$ in (9), or use a different widening for each iterate (as in [Cou81]) or even have a widening which depends upon all previous iterates.

then $\check{X}^{i+1} = \check{X}^i$ so that if the approximation $\hat{A}$ of $\mathrm{lfp}^{\sharp}_{\perp} F$ is a fixpoint of F then it cannot be improved by (12). Observe also that if L is a meet-semi-lattice (the greatest lower bound $x \sqcap y$ exists for all $x, y \in L$) satisfying the *descending chain condition* (no strictly decreasing chain in L can be infinite), then $\sqcap$ is a narrowing.

Example 11 ((Widening and narrowing for intervals)). The widening and narrowing introduced in [CC76] for the lattice of intervals $\overline{L} = \{\bot\} \cup \{[\ell, u] \mid \ell \in \mathbb{Z} \cup \{-\infty\} \wedge u \in \mathbb{Z} \cup \{+\infty\} \wedge \ell \leq u\}$ are defined as follows:

$$\begin{aligned} \bot \triangledown X &= X \\ X \triangledown \bot &= X \\ [\ell_0, u_0] \triangledown [\ell_1, u_1] &= [\text{if } \ell_1 < \ell_0 \text{ then } -\infty \text{ else } \ell_0, \\ &\qquad \text{if } u_1 > u_0 \text{ then } +\infty \text{ else } u_0] \ . \end{aligned} \tag{13}$$

The widening (13) extrapolates unstable bounds to infinity. Observe that the widening (13) is not monotone. For example $[0, 1] \sqsubseteq [0, 2]$ but $[0, 1] \triangledown [0, 2] = [0, +\infty] \not\sqsubseteq [0, 2] = [0, 2] \triangledown [0, 2]$.

The narrowing introduced in [CC76] for the lattice of intervals $\overline{L} = \{\bot\} \cup \{[\ell, u] \mid \ell \in \mathbb{Z} \cup \{-\infty\} \wedge u \in \mathbb{Z} \cup \{+\infty\} \wedge \ell \leq u\}$ is defined by:

$$\begin{aligned} \bot \triangle X &= \bot \\ X \triangle \bot &= \bot \\ [\ell_0, u_0] \triangle [\ell_1, u_1] &= [\text{if } \ell_0 = -\infty \text{ then } \ell_1 \text{ else } \ell_0, \\ &\qquad \text{if } u_0 = +\infty \text{ then } u_1 \text{ else } u_0] \ . \end{aligned} \tag{14}$$

The narrowing (14) improves infinite bounds only.

Resolution of the equation:

$$X = \overline{F}(X) = ([1, 1] \sqcup (X \oplus [1, 1])) \sqcap [-\infty, 100]$$

considered in Ex. 8 starts with the following increasing iterates:

$$\begin{aligned} \hat{X}^0 &= \bot \\ \hat{X}^1 &= \hat{X}^0 \triangledown \Big(([1, 1] \sqcup (\hat{X}^0 \oplus [1, 1])) \sqcap [-\infty, 100]\Big) \\ &= \bot \triangledown \Big(([1, 1] \sqcup (\bot \oplus [1, 1])) \sqcap [-\infty, 100]\Big) \\ &= ([1, 1] \sqcup \bot) \sqcap [-\infty, 100] \\ &= [1, 1] \\ \hat{X}^2 &= \hat{X}^1 \triangledown \Big(([1, 1] \sqcup (\hat{X}^1 \oplus [1, 1])) \sqcap [-\infty, 100]\Big) \\ &= [1, 1] \triangledown \Big(([1, 1] \sqcup ([1, 1] \oplus [1, 1])) \sqcap [-\infty, 100]\Big) \\ &= [1, 1] \triangledown (([1, 1] \sqcup [2, 2]) \sqcap [-\infty, 100]) \\ &= [1, 1] \triangledown ([1, 2] \sqcap [-\infty, 100]) \end{aligned}$$

$$
\begin{aligned}
&= [1,\,1] \mathbin{\triangledown} [1,\,2] \\
&= [1,\,+\infty] \\
\hat{X}^3 &= \hat{X}^2 \mathbin{\triangledown} \Big(([1,\,1] \sqcup (\hat{X}^2 \oplus [1,\,1])) \sqcap [-\infty,\,100] \Big) \\
&= [1,\,+\infty] \mathbin{\triangledown} \Big(([1,\,1] \sqcup ([1,\,+\infty] \oplus [1,\,1])) \sqcap [-\infty,\,100] \Big) \\
&= [1,\,+\infty] \mathbin{\triangledown} \big(([1,\,1] \sqcup [2,\,+\infty]) \sqcap [-\infty,\,100] \big) \\
&= [1,\,+\infty] \mathbin{\triangledown} ([1,\,+\infty] \sqcap [-\infty,\,100]) \\
&= [1,\,+\infty] \mathbin{\triangledown} [1,\,100] \\
&= [1,\,+\infty] \\
&\overline{\sqsubseteq}\ \hat{X}^2 \ .
\end{aligned}
$$

Then the decreasing iterates are as follows:

$$
\begin{aligned}
\check{X}^0 &= \hat{X}^2 \\
\check{X}^1 &= \check{X}^0 \mathbin{\triangle} \Big(([1,\,1] \sqcup (\check{X}^0 \oplus [1,\,1])) \sqcap [-\infty,\,100] \Big) \\
&= [1,\,+\infty] \mathbin{\triangle} \Big(([1,\,1] \sqcup ([1,\,+\infty] \oplus [1,\,1])) \sqcap [-\infty,\,100] \Big) \\
&= [1,\,+\infty] \mathbin{\triangle} \big(([1,\,1] \sqcup [2,\,+\infty]) \sqcap [-\infty,\,100] \big) \\
&= [1,\,+\infty] \mathbin{\triangle} ([1,\,+\infty] \sqcap [-\infty,\,100]) \\
&= [1,\,+\infty] \mathbin{\triangle} [1,\,100] \\
&= [1,\,100] \\
\check{X}^2 &= \check{X}^1 \mathbin{\triangle} \Big(([1,\,1] \sqcup (\check{X}^1 \oplus [1,\,1])) \sqcap [-\infty,\,100] \Big) \\
&= [1,\,100] \mathbin{\triangle} \Big(([1,\,1] \sqcup ([1,\,100] \oplus [1,\,1])) \sqcap [-\infty,\,100] \Big) \\
&= [1,\,100] \\
&= \check{X}^1 \ .
\end{aligned}
$$

In what follows, we will consider the fact that given two integer constants $n_1 \leq n_2$, the abstract interpreter SYNTOX [Bou90] will analyze the program:

```
program Pn1n2;
   var I : integer ;
begin
   I := n1;
   while I <= n2 do
      begin
         { I ∈ [n1, n2] }
         I := I + 1;
      end;
   { I = n2 + 1 }
end.
```

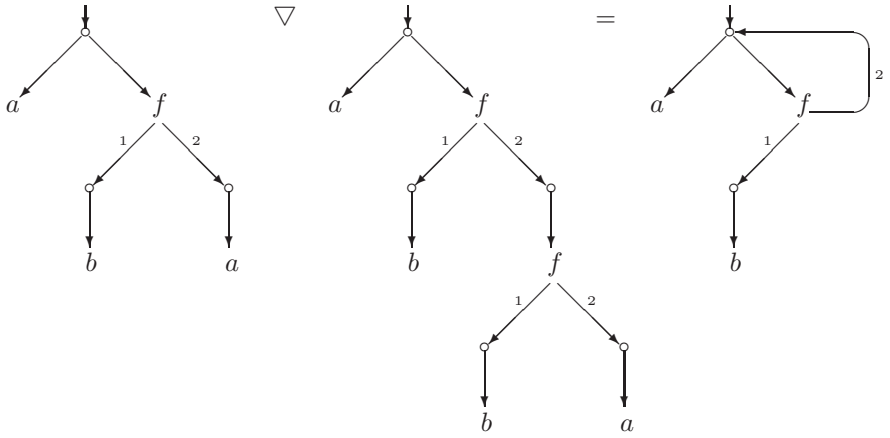
by solving a system of fixpoint equations equivalent to:

$$X = F(X) = ([n_1,\,n_1] \sqcup (X \oplus [1,\,1])) \sqcap [-\infty,\,n_2]$$

and automatically discover the loop invariant:

$$\{\ \mathtt{I} \in [n_1,\ n_2]\ \}\ .$$ □

Example 12 ((Type graphs widening)). [BJCD87] have defined the *restriction* of type graphs. It is a widening. For example:



More precisely, the widening $G = G_1 \triangledown G_2$ of two type graphs G_1 and G_2 is obtained by:

1. Initializing G with a copy of G_1 and G_2 where roots are merged (merging consists in joining type nodes without removing any arc),
2. and then, in repeatedly applying the following transformations to G:
   (a) type nodes k of G with distinct sons $k : g(k_1^1, \ldots, k_1^n)$ and $k : g(k_2^1, \ldots, k_2^n)$ with the same functor g have their sons k_1^i and k_2^i pairwise merged,
   (b) distinct type nodes $k_1 : g(k_1^1, \ldots, k_1^n)$ and $k_2 : g(k_2^1, \ldots, k_2^n)$ with the same functor g on an acyclic path from the root are merged[7].

All sons of a type node must have different functors, so that the breadth of a type graph is finite. No acyclic path starting from the root can contain the same functor twice, so that the depth of a widened type graph is finite. It follows that a strictly increasing chain of type graphs is finite. □

5 Combining the Galois Connection and Widening/Narrowing Approaches to Abstract Interpretation

In practice both Galois connection and widening/narrowing approaches are used simultaneously [CC76,CC77a]. First a Galois connection is used to obtain approximate equations $\overline{X} = \overline{F}(\overline{X})$ on an abstract domain $\overline{L}$. The goal is to obtain computer representable properties of programs. These fixpoint equations are then solved iteratively. Widenings and narrowings are used when the domain $\overline{L}$ has infinite or very long strictly ascending chains or even when it is finite but

[7] As noticed by [BJCD87], several solutions are possible.

very large. The goal is then to enforce or accelerate the convergence. For more details see [CC76,CC77a], consult [Cou81] to minimize the number of widenings within loops and chapter 4 of [Cou78] for dual problems.

The use of Galois connections corresponds to an ideal situation where concrete assertions have a unique best abstract interpretation [CC79b]. In practice this property is not always satisfied for reasons of efficient computer representation of abstract properties. Moreover the abstract domain need not be partially ordered since many equivalent abstract values can be used to represent the same abstract assertion or least upper bounds may not exist or may not be efficiently computable. In this case, widenings and narrowings can be used to palliate the non-existence of least upper bounds or greatest lower bounds in the abstract domain $\overline{L}(\overline{\sqsubseteq})$ [CC92b]. Proposition 35 in the appendix can be applied in this case. Examples of such a situation are given in [Bou92,BJCD87,CH78,Deu92,MS88,Str88].

6 Unappreciated Conjectures about the Two Approaches

The widening/narrowing approach to abstract interpretation is not so well understood as the Galois connection approach, as exemplified by [AH87] where no paper refers to the convergence acceleration method.

An often used argument for 'proving' the uselessness of the widening/narrowing approach is that given an infinite abstract domain together with specific widening and narrowing operators, it is possible to find a finite lattice which will give the same results. For example [KN87] claim that "One may wonder whether or not it is necessary to choose a finite domain for abstract interpretation, since apparently more information can be obtained from an interpretation over an infinite domain. The answer is that if uniform termination of the abstract interpretation is required, no more information can be obtained by choosing an infinite domain". In [HH90], a fixpoint approximation method is considered which consists in an upwards iteration using a safe approximation $\vec{\alpha}(F)$ of the function $F \in L \overset{\text{con}}{\longmapsto} L$ in a finite small lattice $\overline{L}$ such that $L \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \overline{L}$ and in which the problem of finding fixpoints is tractable, followed by a downwards iteration from $\gamma\big(\text{lfp}_{\alpha(\pm)}\,(\vec{\alpha}(F))\big)$ in L (or in a sequence of intermediate lattices larger than $\overline{L}$). [HH90] claim that "We have now shown the *equivalence* of step 1 of that process and the Cousot's notion of widening." For step 2, which consists in working in a larger lattice, [HH90] claim that "the refinement of the upper bound in intermediate lattice corresponds to narrowing".

7 Comparing the Two Approaches

To correct these overstatements, we show that, on the contrary and in general, no finite abstract domain (or domain satisfying the ascending chain condition) can be used instead of widening/narrowing operators on infinite domains to obtain the same results (or equivalent ones, up to the computer representation).

7.1 Finite Abstract Domains (or Domains Satisfying the Ascending Chain Condition) Cannot Do for Widenings and Narrowings

More precisely, we prove in this section that there exist infinite domains and widening/narrowing operators such that:

1. For each program there exists a finite lattice which can be used for this program to obtain results equivalent to those obtained using widening/narrowing operators;
2. No such a finite lattice will do for all programs;
3. For all programs, infinitely many abstract values are necessary;
4. For a particular program it is not possible to infer the set of needed abstract values by a simple inspection of the text of the program.

Let $\mathrm{lfp}^{\bot}_{\bot}(F)$ where $F \in L \stackrel{\text{mon}}{\longmapsto} L$ be the collecting semantics of a given program P. Assume that $\langle L, \bot, F\rangle \stackrel{\gamma}{\underset{\alpha}{\leftarrow\!\rightarrow}} \langle \overline{L}, \overline{\bot}, \overline{F}\rangle$ is an abstract interpretation such that $\mathrm{lfp}^{\bot}_{\bot}(\overline{F})$ is not computable iteratively in finitely many steps and $\overline{A}$ is an upper approximation of $\mathrm{lfp}^{\bot}_{\bot}(\overline{F})$ effectively computed using the widening-/narrowing approach. We have $\mathrm{lfp}^{\bot}_{\bot}(F) \sqsubseteq \gamma\left(\mathrm{lfp}^{\overline{\bot}}_{\overline{\bot}}(\overline{F})\right)$ and $\mathrm{lfp}^{\overline{\bot}}_{\overline{\bot}}(\overline{F}) \mathrel{\overline{\sqsubseteq}} \overline{A}$, so that, by monotony and transitivity, $\mathrm{lfp}^{\bot}_{\bot}(F) \sqsubseteq \gamma(\overline{A})$. We want to find a finite equivalent abstract interpretation $\langle L, \bot, F\rangle \stackrel{\gamma'}{\underset{\alpha'}{\leftarrow\!\rightarrow}} \langle \overline{\overline{L}}, \overline{\overline{\bot}}, \overline{\overline{F}}\rangle$ such that $\overline{\overline{L}}$ is finite and $\mathrm{lfp}^{\overline{\overline{\bot}}}_{\overline{\overline{\bot}}}(\overline{\overline{F}})$ gives results equivalent to $\overline{A}$, i.e., $\gamma'\left(\mathrm{lfp}^{\overline{\overline{\bot}}}_{\overline{\overline{\bot}}}(\overline{\overline{F}})\right) = \gamma(\overline{A})$.

We choose the finite lattice $\overline{\overline{L}}$ consisting of the elements $\overline{\overline{\bot}} \mathrel{\overline{\overline{\sqsubseteq}}} \overline{A} \mathrel{\overline{\overline{\sqsubseteq}}} \overline{\overline{\top}}$ and the operator $\overline{\overline{F}} \in \overline{\overline{L}} \stackrel{\text{mon}}{\longmapsto} \overline{\overline{L}}$ such that $\overline{\overline{F}}(\overline{\overline{\bot}}) = \overline{\overline{F}}(\overline{A}) = \overline{A}$ and $\overline{\overline{F}}(\overline{\overline{\top}}) = \overline{\overline{\top}}$. Define the Galois connection $\overline{L} \stackrel{\overline{\gamma}}{\underset{\overline{\alpha}}{\leftarrow\!\rightarrow}} \overline{\overline{L}}$ such that $\overline{\alpha}(X) =$ if $X = \bot$ then $\overline{\overline{\bot}}$ elsif $X \mathrel{\overline{\sqsubseteq}} \overline{A}$ then $\overline{A}$ else $\overline{\overline{\top}}$ and $\overline{\gamma}(\overline{\overline{\bot}}) = \overline{\bot}$, $\overline{\gamma}(\overline{A}) = \overline{A}$, $\overline{\gamma}(\overline{\overline{\top}}) = \overline{\top}$, where $\overline{\top}$ is the supremum of L (which is added to $\overline{L}$ if no one exists). We have $L \stackrel{\gamma\circ\overline{\gamma}}{\underset{\overline{\alpha}\circ\alpha}{\leftarrow\!\rightarrow}} \overline{\overline{L}}$, $\overline{\alpha}\circ\alpha\circ F\circ\gamma\circ\overline{\gamma} \mathrel{\overline{\overline{\sqsubseteq}}} \overline{\overline{F}}$ and $\mathrm{lfp}^{\overline{\overline{\bot}}}_{\overline{\overline{\bot}}}(\overline{\overline{F}}) = \overline{A}$. It follows that the effective computation of any upper approximation $\overline{A}$ of $\mathrm{lfp}^{\bot}_{\bot}(F)$ (obtained by widening/narrowing) can also be done by iteration of a fixpoint operator $\overline{\overline{F}}$ on a finite lattice $\overline{\overline{L}}$.

If equivalent results are required for the two approaches, we observe that $\overline{\overline{L}}$ must contain an element $\mathrm{lfp}^{\overline{\overline{\bot}}}_{\overline{\overline{\bot}}}(\overline{\overline{F}})$ such that $\gamma'\left(\mathrm{lfp}^{\overline{\overline{\bot}}}_{\overline{\overline{\bot}}}(\overline{\overline{F}})\right) = \gamma(\overline{A})$ for each program P. For the family of programs $\mathtt{P}n_1n_2$ defined in Ex. 11, this lattice $\overline{\overline{L}}$ would have to contain infinitely many different elements equivalent to $\gamma(\overline{A})$ where $\overline{A} = [n_1, n_2]$. It follows that in general, $\overline{\overline{L}}$ cannot be finite and must contain infinite strictly increasing chains.

Since the above proof is rather contrived, it could be argued that the finite subset of $\overline{L}$ which is needed for analyzing a given program can be directly derived from a simple inspection of its text. This is not possible in general since, as shown by the series of examples below, the invariant $\overline{A}$ which is found by the analysis does not necessarily appear in the program and, more generally, is not a simple function of the program text.

Example 13 ((Interval analysis)). Given an integer constant n, the abstract interpreter SYNTOX [Bou90] will analyze the program `Function91ofMcCarthy` below (known for $n = 100$) and automatically discover the invariant given as comment:

```
program Function91ofMcCarthy;
  var X, Y : integer;

  function F(X : integer) : integer;
  begin
    if X > n then
      F := X - 10
    else
      F := F(F(X + 11));
  end;

begin
  readln(X);
  Y := F(X);
  { Y ∈ [n - 9, maxint - 10] }
end.
```

Observe that the integer constants $(n-9)$ and $(\text{maxint}-10)$ which are found as bounds for `Y` by the automatic interval analysis do not appear in the program. Even more convincing is the following example:

```
program Function91ofMcCarthy;
  var X, Y : integer;

  function F(X : integer) : integer;
  begin
     if X > 100 then
        F := X - 10
        { F ∈ [91, maxint - 10] }
     else
        F := F(F(F(X + 33)));
        { F ∈ [91, 93] }
     { F ∈ [91, maxint - 10] }
 end;

begin
  readln(X);
  Y := F(X);
  { Y ∈ [91, maxint - 10] }
end.
```

□

Example 14 ((Rational congruence analysis)). [Gra91a] considers the discovery of arithmetical congruences of the form $x \equiv p[q]$ where $p, q \in \mathbb{Q}$ are rational numbers automatically determined by the analysis and x denotes the value of a program variable. The non-extremal elements of the corresponding lattice $\overline{L} = \{\bot, \top\} \cup (\mathbb{Q} \times \mathbb{Q})$ are denoted $p + q\mathbb{Z}$ since $\gamma(p + q\mathbb{Z}) = \{x \in \mathbb{Q} \mid \exists k \in \mathbb{Z} : x = p + q.k\}$. This lattice does not satisfy the ascending chain condition since:

$$\frac{1}{2^0}\mathbb{Z} \sqsubset \frac{1}{2^1}\mathbb{Z} \sqsubset \frac{1}{2^2}\mathbb{Z} \sqsubset \ldots \sqsubset \frac{1}{2^n}\mathbb{Z} \sqsubset \ldots .$$

Using the widening/narrowing approach to abstract interpretation, the following loop invariant is derived in [Gra91b]:

```
program PC;
```

```
   var X : real;
begin
   X := 2.8542;
   while ... do begin
      { X ≡ 1/5000 [1/500] }
      X := X + 1/500;
   end;
end.
```

Observe that the constant 5000 which is derived from $2.8542 = \frac{14271}{5000}$ does not appear in the program text. □

Example 15 ((Linear inequality analysis)). The abstract interpretation introduced in [CH78] has been designed, using the widening/narrowing approach, to discover linear invariants such as:

```
program PL;
   var I, J : integer;
begin
   I := 2; J := 0;
   while ... do begin
      { 2J + 2 ≤ I  ∧  0 ≤ J }
      if ... then begin
         I := I + 4;
         { 2J + 6 ≤ I  ∧  0 ≤ J }
      end else begin
         I := I + 2; J := J + 1;
         { 2J + 2 ≤ I  ∧  1 ≤ J }
      end;
      { 2J + 2 ≤ I  ∧  6 ≤ I + 2J  ∧  0 ≤ J }
   end;
end.
```

Observe that the analysis discovers relations between variables that never appear within the same command. Incidentally, this fact can be used to prove automatically the termination of loops [Hal79]: a new counter is added to the program for each loop which is initialized to zero and incremented by one within the loop body. The analysis will relate its value to that of the other variables of the program. If the value of the counter is bounded on loop exit, then termination is automatically proved. □

7.2 Widenings and Narrowings Can Do for Finite Abstract Domains (or Domains Satisfying the Ascending Chain Condition)

To prove that the widening/narrowing approach is more general than the Galois connection approach, it remains to show that given an infinite domain L, it is always possible to find widening/narrowing operators giving results similar (in precision and speed of convergence) to the ones that could be obtained by approximations of the domain L based upon Galois connections $L \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \overline{L}$.

Assume that $L(\sqsubseteq, \sqcup)$ is a poset, $F \in L \overset{\text{con}}{\longmapsto} L$ is continuous, $\bot \in L$ is such that $\bot \sqsubseteq F(\bot)$, and $\mathrm{lfp}_\bot(F) = \bigsqcup_{n\in\mathbb{N}} F^n(\bot)$ exists (see Prop. 23 in the appendix). Assume as well that $\overline{L}(\overline{\sqsubseteq}, \overline{\sqcup})$ is a poset satisfying the ascending

chain condition and $\langle L,\ \bot,\ F\rangle \underset{\alpha'}{\overset{\gamma}{\leftrightarrows}} \langle \overline{L},\ \overline{\bot},\ \overline{F}\rangle$. We can assume that $\gamma(\overline{\bot}) = \bot$ since otherwise more precision could be obtained by considering $\overline{L} \cup \{\overline{\bot}\}$ where $\overline{\bot} \notin \overline{L}$ is a new abstract element such that $\gamma(\overline{\bot}) = \bot$ and $\overline{\bot}\ \overline{\sqsubseteq}\ \overline{\bot}$. For simplicity, we can also assume that α is surjective since otherwise by choosing $\overline{L} = \{\alpha(x) \mid x \in L\}$ we could eliminate useless abstract values (these abstract value $\overline{x}$ are useless since they can be replaced by $\alpha \circ \gamma(\overline{x})$ without any loss of information). Consequently, by Prop. 29 in the appendix, $\forall \overline{x} \in \overline{L}$: $\alpha \circ \gamma(\overline{x}) = \overline{x}$. Finally we assume that $\overline{F} = \alpha \circ F \circ \gamma$ which, by Prop. 31 in the appendix, is the most precise $\overline{F}$ among those satisfying $\alpha \circ F \circ \gamma\ \overline{\sqsubseteq}\ \overline{F}$. By (24) and (25), $\overline{F}$ is monotone, hence it is continuous since $\overline{L}$ satisfies the ascending chain condition. Together with Prop. 30 in the appendix, this implies that the increasing chain $\overline{X}^0 = \overline{\bot}, \ldots, \overline{X}^{i+1} = \overline{F}(\overline{X}^i), \ldots$ converges in n steps to the limit $\overline{X}^n = \mathrm{lfp}^{\overline{\bot}}_{\overline{\bot}}(\overline{F})$, which is the result obtained by the Galois connection approach. The result of the analysis is sound since $\mathrm{lfp}_{\bot}(F) \sqsubseteq \gamma(\overline{X}^n)$. Define the partial widening:

$$\begin{array}{c} \triangledown \in L \times L \mapsto L \\ x \triangledown y = \gamma\big(\alpha(x)\,\overline{\sqcup}\,\alpha(y)\big)\ . \end{array} \tag{15}$$

According to (9), the widening approach consists in computing the iteration sequence $X^0 = \bot, \ldots, X^{i+1} =$ if $F(X^i) \sqsubseteq X^i$ then X^i else $X^i \triangledown F(X^i), \ldots$ This sequence is well-defined and converges in m steps to $X^m = A$ which is the result obtained by the widening approach. We have $m = n$ and $A = \gamma\big(\mathrm{lfp}^{\overline{\bot}}_{\overline{\bot}}(\overline{F})\big)$ so that both approaches have the same cost and precision (up to $\langle \alpha,\ \gamma\rangle$ as far as the representation of abstract values is concerned). The proof is as follows:

Proof. First we must show that (15) defines a widening. Observe that if $x, y \in L$ then $\alpha(x)\ \overline{\sqsubseteq}\ \alpha(x)\,\overline{\sqcup}\,\alpha(y)$ by definition of upper bounds so that, by (22) and (25), $x \sqsubseteq \gamma \circ \alpha(x) \sqsubseteq \gamma\big(\alpha(x)\,\overline{\sqcup}\,\alpha(y)\big) = x \triangledown y$ proving (6). The same way, (7) holds. Let $x^0 \sqsubseteq x^1 \sqsubseteq \ldots$ be an increasing chain such that $y^0 = x^0, \ldots, y^{i+1} = y^i \triangledown x^{i+1}, \ldots$ is well-defined. By (6) and (24), $y^i,\ i \in \mathbb{N}$ and $x^i,\ i \in \mathbb{N}$ hence $\alpha(y^i),\ i \in \mathbb{N}$ and $\alpha(x^i),\ i \in \mathbb{N}$ are increasing chains. Since $\overline{L}$ satisfies the ascending chain condition, there exists $\ell' \in \mathbb{N}$ such that $\alpha(y^{\ell'}) = \alpha(y^{\ell'+k})$ for $k \geq \ell'$ and $\ell'' \in \mathbb{N}$ such that $\alpha(x^{\ell''}) = \alpha(x^{\ell''+k})$ for $k \geq \ell''$. So let ℓ be the maximum of ℓ' and ℓ''. For all $k \geq \ell$, we have $\alpha(y^k) = \alpha(y^\ell)$ and $\alpha(x^k) = \alpha(x^\ell)$ so that, by (15), $y^{k+1} = y^k \triangledown x^k = \gamma\big(\alpha(y^k) \sqcup \alpha(x^k)\big) = \gamma\big(\alpha(y^\ell) \sqcup \alpha(x^\ell)\big) = y^\ell \triangledown x^\ell = y^{\ell+1}$, proving that $\forall k > \ell$: $y^k = y^{\ell+1}$, so that $y^i,\ i \in \mathbb{N}$ is eventually stable, as required by (8).

Since $\overline{L}$ is a poset, the least upper bound $\overline{\sqcup}$ may not exist in (15). Therefore, we must show that the iteration sequence $X^i,\ i \in \mathbb{N}$ is well-defined. More precisely, we prove that $\forall n \in \mathbb{N}$: X^n is well-defined such that $\alpha(X^n)\ \overline{\sqsubseteq}\ \alpha\big(F(X^n)\big)$. For the basis, we have $X^0 \sqsubseteq F(X^0)$ since $X^0 = \bot \sqsubseteq F(\bot)$ whence $\alpha(X^0)\ \overline{\sqsubseteq}\ \alpha\big(F(X^0)\big)$ by (24). If, by induction hypothesis, X^n is well-defined and such that $\alpha(X^n)\ \overline{\sqsubseteq}\ \alpha\big(F(X^n)\big)$, then $\alpha(X^n)\,\overline{\sqcup}\,\alpha\big(F(X^n)\big) = \alpha\big(F(X^n)\big)$ exists, whence, by (15), $X^n \triangledown F(X^n) = \gamma\big(\alpha(X^n)\,\overline{\sqcup}\,\alpha(F(X^n))\big) = \gamma \circ \alpha\big(F(X^n)\big)$ is well-defined. If $F(X^n) \sqsubseteq X^n$ then $X^{n+1} = X^n$ whence, by induction hypothesis, X^{n+1} is well-defined such that $\alpha(X^{n+1})\ \overline{\sqsubseteq}\ \alpha\big(F(X^{n+1})\big)$. Otherwise, we have shown that

$X^{n+1} = X^n \triangledown F(X^n) = \gamma \circ \alpha\big(F(X^n)\big)$ is well-defined. Moreover, by (22), induction hypothesis and (25), we have $X^n \sqsubseteq \gamma \circ \alpha(X^n) \sqsubseteq \gamma \circ \alpha\big(F(X^n)\big) = X^{n+1}$. From $X^n \sqsubseteq X^{n+1}$, we derive by (26) and continuity, hence monotony of F, (24), (25) that $\alpha(X^{n+1}) = \alpha \circ \gamma \circ \alpha\big(F(X^n)\big) = \alpha\big(F(X^n)\big) \mathrel{\overline{\sqsubseteq}} \alpha\big(F(X^{n+1})\big)$. By recurrence, we conclude that $\forall n \in \mathbb{N}$: X^n is well-defined such that $\alpha(X^n) \mathrel{\overline{\sqsubseteq}} \alpha\big(F(X^n)\big)$.

We now prove, by recurrence, that $\forall k \in \mathbb{N}$: $\gamma(\overline{X}^k) = X^k$. For the basis, we have $\gamma(\overline{X}^0) = X^0$ since $\gamma(\overline{\bot}) = \bot$. For the induction step, assume $\gamma(\overline{X}^k) = X^k$ so that $\overline{X}^k = \alpha \circ \gamma(\overline{X}^k) = \alpha(X^k)$. If $F(X^k) \not\sqsubseteq X^k$ then $\gamma(\overline{X}^{k+1}) = \gamma(\overline{X}^k \mathbin{\overline{\sqcup}} \overline{X}^{k+1})$ [since $\overline{X}^k \mathrel{\overline{\sqsubseteq}} \overline{X}^{k+1}$] $= \gamma\big(\overline{X}^k \mathbin{\overline{\sqcup}} \overline{F}(\overline{X}^k)\big)$ [since $\overline{X}^{k+1} = \overline{F}(\overline{X}^k)$] $= \gamma\big(\overline{X}^k \mathbin{\overline{\sqcup}} \alpha \circ F \circ \gamma(\overline{X}^k)\big)$ [by definition of $\overline{F} = \alpha \circ F \circ \gamma$] $= \gamma\big(\alpha(X^k) \mathbin{\overline{\sqcup}} \alpha \circ F(X^k)\big)$ [by induction hypothesis] $= X^k \triangledown F(X^k)$ [by (15)] $= X^{k+1}$ [by definition of X^{k+1} when $F(X^k) \not\sqsubseteq X^k$]. Otherwise $F(X^k) \sqsubseteq X^k$ in which case $\gamma(\overline{X}^{k+1}) = \gamma\big(\overline{F}(\overline{X}^k)\big)$ [by definition of $\overline{X}^{k+1}$] $= \gamma \circ \alpha \circ F \circ \gamma(\overline{X}^k)$ [by definition of $\overline{F}$] $= \gamma \circ \alpha \circ F(X^k)$ [by induction hypothesis] $\sqsubseteq \gamma \circ \alpha(X^k)$ [by $F(X^k) \sqsubseteq X^k$, (24) and (25)] $= \gamma(\overline{X}^k)$ [since $\alpha(X^k) = \overline{X}^k$, by induction hypothesis] $= X^k$ [by induction hypothesis] $= X^{k+1}$ [by (9) when $F(X^k) \sqsubseteq X^k$]. Moreover $\overline{X}^k$, $k \in \mathbb{N}$ is an increasing chain so that $\overline{X}^k \sqsubseteq \overline{X}^{k+1}$ whence $X^k = \gamma(\overline{X}^k) \sqsubseteq \gamma(\overline{X}^{k+1})$ by (25). By antisymmetry, we have $\gamma(\overline{X}^{k+1}) = X^{k+1}$.

Observe that the chain X^k, $k \in \mathbb{N}$ is increasing but not strictly by (8), so that there exists $\ell \in \mathbb{N}$ such that $X^{\ell+1} = X^\ell$. By (9), we have $F(X^\ell) \sqsubseteq X^\ell$ or $X^{\ell+1} = X^\ell \triangledown F(X^\ell) = X^\ell$, whence $F(X^\ell) \sqsubseteq X^\ell$ by (7). So let m be the smallest ℓ such that $F(X^\ell) \sqsubseteq X^\ell$. The chain $\overline{X}^k$, $k \in \mathbb{N}$ is increasing but not strictly since $\overline{L}$ satisfies the ascending chain condition so let n be the smallest natural such that $\overline{X}^{n+1} = \overline{F}(\overline{X}^n) = \overline{X}^n$.

We have $F\big(\gamma(\overline{X}^m)\big) \sqsubseteq \gamma(\overline{X}^m)$ which implies $\alpha\big(F\big(\gamma(\overline{X}^m)\big)\big) \mathrel{\overline{\sqsubseteq}} \overline{X}^m$ that is $\overline{F}(\overline{X}^m) \mathrel{\overline{\sqsubseteq}} \overline{X}^m$. Since the sequence $\langle \overline{X}^i,\ i \geq 0 \rangle$ is increasing, we have $\overline{X}^m \mathrel{\overline{\sqsubseteq}} \overline{X}^{m+1} = \overline{F}(\overline{X}^m)$ so that $\overline{F}(\overline{X}^m) = \overline{X}^m$, by antisymmetry. It follows that $n \leq m$. Reciprocally, $\overline{F}(\overline{X}^n) = \overline{X}^n$ so that $\alpha \circ F \circ \gamma(\overline{X}^n) = \overline{X}^n$ whence $F \circ \gamma(\overline{X}^n) \sqsubseteq \gamma(\overline{X}^n)$ by (2) that is $F(X^n) \sqsubseteq X^n$ and therefore $m \leq n$. We conclude $m = n$ and $A = X^m = \gamma(\overline{X}^m) = \gamma(\overline{X}^n) = \gamma\big(\mathrm{lfp}^{\overline{\sqsubseteq}}_{\overline{\bot}}(\overline{F})\big)$. □

8 Remarks on the Design of Widenings and Narrowings

The design of abstract domains using Galois connections is rather familiar since a great number of examples is available and because it can be presented using a number of equivalent and well understood mathematical objects such as upper closure operators, Moore families, topologies, complete join congruence relations, families of principal ideals (see [CC79b]). On the contrary, the design of widenings and narrowings is often thought off to be more difficult since it appears as an heuristic to cope with induction. The following remarks can help in the design of widenings and narrowings.

■ The rapprochement between the two approaches can be made by observing that whenever a Galois connection $L \underset{\alpha}{\overset{\gamma}{\leftarrows}} \overline{L}$ is available and $\overline{L}$ ($\overline{\sqsubseteq}$, $\overline{\sqcup}$) is a join-semi-lattice satisfying the ascending chain condition then a widening $\overline{\nabla} \in \overline{\overline{L}} \times \overline{\overline{L}} \longmapsto \overline{\overline{L}}$ can be defined on any infinite abstract domain $\overline{\overline{L}}$ ($\overline{\overline{\sqsubseteq}}$) such that $L \underset{\overline{\alpha}}{\overset{\overline{\gamma}}{\leftarrows}} \overline{\overline{L}}$ by projection into $\overline{\overline{L}}$ of the least upper bound $\overline{\sqcup}$ defined on $\overline{L}$, as follows:

$$x \mathbin{\overline{\nabla}} y \;=\; \overline{\alpha}\Big(\overline{\gamma}\big(\alpha \circ \overline{\gamma}(x) \mathbin{\overline{\sqcup}} \alpha \circ \overline{\gamma}(y)\big)\Big) \;. \tag{16}$$

By Prop. 37 in the appendix, if $\overline{\alpha}$ is surjective then (16) defines a widening. In particular, when L is $\overline{\overline{L}}$ so that $\overline{\alpha}$ and $\overline{\gamma}$ are identity functions, we obtain the widening defined in (15). Similarly, if L ($\sqsubseteq$, $\sqcap$) is a meet-semi-lattice and $\overline{L}$ ($\overline{\sqsubseteq}$) is a poset satisfying the descending chain condition, then $\triangle \in L \times L \longmapsto L$ can be defined on L for speeding up the convergence by projection in $\overline{L}$, as follows:

$$x \mathbin{\triangle} y \;=\; x \sqcap \gamma \circ \alpha(y) \;. \tag{17}$$

Proposition 38 in the appendix shows that $\triangle$ is a narrowing.

Example 16 ((Rule of signs based widening and narrowing for interval analysis)). Assume that L is the lattice of intervals and $\overline{L}$ is the lattice of signs $\{\overline{\bot},\ 0,\ -,\ +,\ \overline{\top}\}$ [CC79b], such that $\overline{\bot} \mathbin{\overline{\sqsubseteq}} 0 \mathbin{\overline{\sqsubseteq}} - \mathbin{\overline{\sqsubseteq}} \overline{\top}$, $0 \mathbin{\overline{\sqsubseteq}} + \mathbin{\overline{\sqsubseteq}} \overline{\top}$, and $\overline{\gamma}(\overline{\bot}) = \bot$, $\overline{\gamma}(0) = [0,\ 0]$, $\overline{\gamma}(-) = [-\infty,\ 0]$, $\overline{\gamma}(+) = [0,\ +\infty]$, $\overline{\gamma}(\overline{\top}) = [-\infty,\ +\infty]$. Then (15) becomes:

$$\begin{aligned}
\bot \mathbin{\overline{\nabla}} X &= X \\
X \mathbin{\overline{\nabla}} \bot &= X \\
[\ell_0,\ u_0] \mathbin{\overline{\nabla}} [\ell_1,\ u_1] &= \text{if } \ell_0 = \mathrm{u}_0 = \ell_1 = \mathrm{u}_1 = 0 \text{ then } [0,\ 0] \\
&\quad \text{elsif } (\mathrm{u}_0 \leq 0) \wedge (\mathrm{u}_1 \leq 0) \text{ then } [-\infty,\ 0] \\
&\quad \text{elsif } (0 \leq \ell_0) \wedge (0 \leq \ell_1) \text{ then } [0,\ +\infty] \\
&\quad \text{else } [-\infty,\ +\infty] \;.
\end{aligned}$$

Similarly, (17) becomes:

$$\begin{aligned}
\bot \mathbin{\triangle} X &= \bot \\
X \mathbin{\triangle} \bot &= \bot \\
[\ell_0,\ u_0] \mathbin{\triangle} [\ell_1,\ u_1] &= [\text{if } \ell_0 \leq 0 \leq \ell_1 \text{ then } 0 \text{ else } \ell_0, \\
&\qquad \text{if } \mathrm{u}_1 \leq 0 \leq \mathrm{u}_0 \text{ then } 0 \text{ else } \mathrm{u}_0] \;.
\end{aligned}$$ □

Another example of application of (16) and (17) for boolean-based abstract interpretations of higher-order functional languages such as strictness analysis is given by [HH90]. However the restriction to a finite lattice $\overline{L}$ is unfortunate since expressiveness can be severely restricted without necessary speed up since only the length of strictly increasing and decreasing chains has to be taken into account. The use of (15) and (17) with finite lattices $\overline{L}$ should be understood as a last resort since the power of the widening/narrowing approach relies on the

ability to extrapolate to infinitely many distinct abstract values for all programs but to a finite number only for any given program.

■ The results obtained using an infinite domain with a widening can be worse than those obtained using a finite domain corresponding to a coarser Galois connection. This is the case for example when using intervals with widening (13) which can give worse results than those obtained by application of the rule of signs [CC79b], as shown by the following:

Example 17 (On loose widenings).

```
program S;
   var X : integer;
begin
   X := 1;
   while ... do begin
      { X1 }
      if ... then
         X := X + 1
      else
         X := 0;
    { X2 }
   end;
end.
```

The following system of approximate equations on intervals for program S:

$$\begin{aligned} X_1 &= F_1(X_2) = [1,\ 1] \sqcup X_2 \\ X_2 &= F_2(X_1) = (X_1 \oplus [1,\ 1]) \sqcup [0,\ 0] \end{aligned}$$

can be solved iteratively using widening (13), as follows:

$$\begin{aligned} &\hat{X}_1^0 = \bot \\ &\hat{X}_2^0 = \bot \\ &\hat{X}_1^1 = \hat{X}_1^0 \triangledown F_1(\hat{X}_2^0) = [1,\ 1] \\ &\hat{X}_2^1 = F_2(\hat{X}_1^1) = [0,\ 2] \\ &\hat{X}_1^2 = \hat{X}_1^1 \triangledown F_1(\hat{X}_2^1) = [1,\ 1] \triangledown [0,\ 2] = [-\infty,\ +\infty] \\ &\hat{X}_2^2 = F_2(\hat{X}_1^2) = [-\infty,\ +\infty]\ . \end{aligned}$$

The system of approximate equations on signs for program S:

$$\begin{aligned} X_1 &= F_1(X_2) = + \sqcup X_2 \\ X_2 &= F_2(X_1) = (X_1 \oplus +) \sqcup 0 \end{aligned}$$

can be solved iteratively as follows:

$$\begin{aligned} &\hat{X}_1^0 = \bot \\ &\hat{X}_2^0 = \bot \\ &\hat{X}_1^1 = F_1(\hat{X}_2^0) = + \\ &\hat{X}_2^1 = F_2(\hat{X}_1^1) = + \\ &\hat{X}_1^2 = F_1(\hat{X}_2^1) = + \\ &\hat{X}_2^2 = F_2(\hat{X}_1^2) = + \end{aligned}$$

and this yields better results, that is $\mathtt{x} \in [0, +\infty]$. □

The remedy is very simple and consists in using a widening that does not lose more information than the Galois connection.

Example 18 (On reducing the loss of information by widening). The widening (13) and narrowing (14) can be improved to give results always better than the rule of signs analysis, as follows:

$$\begin{aligned}
\bot \triangledown X &= X \qquad (18)\\
X \triangledown \bot &= X\\
[\ell_0, u_0] \triangledown [\ell_1, u_1] &= [\text{if } 0 \le \ell_1 < \ell_0 \text{ then } 0 \text{ elsif } \ell_1 < \ell_0 \text{ then } -\infty \text{ else } \ell_0,\\
&\qquad \text{if } \mathrm{u}_0 < \mathrm{u}_1 \le 0 \text{ then } 0 \text{ elsif } \mathrm{u}_0 < \mathrm{u}_1 \text{ then } +\infty \text{ else } \mathrm{u}_0]\\
\bot \triangle X &= \bot \qquad (19)\\
X \triangle \bot &= \bot\\
[\ell_0, u_0] \triangle [\ell_1, u_1] &= [\text{if } (\ell_0 \le 0 \le \ell_1) \vee (\ell_0 = -\infty) \text{ then } \ell_1 \text{ else } \ell_0,\\
&\qquad \text{if } (\mathrm{u}_1 \le 0 \le \mathrm{u}_0) \vee (\mathrm{u}_0 = +\infty) \text{ then } \mathrm{u}_1 \text{ else } \mathrm{u}_0] \ .
\end{aligned}$$

The widening (18) extrapolates unstable bounds to zero or infinity whereas the narrowing (19) improves these bounds. Other bounds such as -1 and $+1$ or even declared bounds might also be taken into account in these definitions. The iterates are now:

$$\begin{aligned}
\hat{X}_1^0 &= \bot\\
\hat{X}_2^0 &= \bot\\
\hat{X}_1^1 &= \hat{X}_1^0 \triangledown F_1(\hat{X}_2^0) = [1,\ 1]\\
\hat{X}_2^1 &= F_2(\hat{X}_1^1) = [0,\ 2]\\
\hat{X}_1^2 &= \hat{X}_1^1 \triangledown F_1(\hat{X}_2^1) = [1,\ 1] \triangledown [0,\ 2] = [0,\ +\infty]\\
\hat{X}_2^2 &= F_2(\hat{X}_1^2) = [0,\ +\infty] \ .
\end{aligned}$$

Another solution is to alternate the collection of bounds on one iteration and the extrapolation of the unstable ones by widening on the next iteration. □

■ A suggestion for designing widenings and narrowings consists in using least upper bounds/greatest lower bounds as long as the iterates follow finite chains in the lattice $\overline{L}$ and in extrapolating as soon as some iterate belongs to an infinite chain.

Example 19 ((Widening for congruence analysis)). Let us come back to the lattice $\overline{L} = \{\pm,\ \top\} \cup (\mathbb{Q} \times \mathbb{Q})$ considered in Ex. 14 for discovering arithmetical congruences of the form $x \equiv p[q]$. The widening proposed by [Gra91b] is:

$$\begin{aligned}
p_0 + q_0\mathbb{Z} \triangledown p_1 + q_1\mathbb{Z} = {} & \text{if } 0 \ne |\mathrm{q}_0| \ne |\mathrm{q}_1| \ne 0 \qquad (20)\\
& \text{then } \overline{\top}\\
& \text{else } \mathrm{p}_0 + \mathrm{q}_0\mathbb{Z} \sqcup \mathrm{p}_1 + \mathrm{q}_1\mathbb{Z}
\end{aligned}$$

where the least upper bound is defined by:

$$p_0 + q_0\mathbb{Z} \sqcup p_1 + q_1\mathbb{Z} = p_0 + \gcd(q_0,\ q_1,\ p_0 - p_1)\mathbb{Z} \ .$$

The idea is that extrapolation is necessary only when the modulus of the congruence class is not constant through consecutive iterates. □

■ The general idea of widenings is to eliminate unstable components through consecutive iterates (or through all previous iterates, which is equivalent, up to the choice of a different abstract domain which would allow for the accumulation of successive iterates in a single abstract value). Hence a very brute force widening would be:

$$x \mathbin{\overline{\nabla}} y = \text{if y } \overline{\sqsubseteq} \text{ x then x else } \overline{\top} \ .$$

Similarly, a naïve narrowing consists in immediately stopping the decreasing iteration sequence:

$$x \mathbin{\overline{\triangle}} y = x \ .$$

The above definitions prove that widenings and narrowings always exist, but this is not a convincing argument. Therefore the idea can always be softened by introducing a *extrapolation threshold* under which the least upper bound $\overline{\sqcup}$ or greatest lower bound $\overline{\sqcap}$ is used and above which extrapolation is enforced. A simple way to do this is to limit the number of exact iterations to some given positive integer n, as follows (abstract values are extended to pairs so as to memorize the number of iterations):

$$\begin{aligned}
\langle x,\, i\rangle \mathbin{\overline{\nabla}} \langle y,\, i+1\rangle = {} & \text{if y } \overline{\sqsubseteq} \text{ x then } \langle \text{x},\, \text{i}+1\rangle \\
& \text{elsif i } \leq \text{ n then } \langle \text{x} \sqcup \text{y},\, \text{i}+1\rangle \\
& \text{else } \langle \text{x} \mathbin{\nabla} \text{y},\, \text{i}+1\rangle \\
\langle x,\, i\rangle \mathbin{\overline{\triangle}} \langle y,\, i+1\rangle = {} & \text{if i } \leq \text{ n then } \langle \text{x} \sqcap \text{y},\, \text{i}+1\rangle \\
& \text{else } \langle \text{x} \mathbin{\triangle} \text{y},\, \text{i}+1\rangle \ .
\end{aligned}$$

Example 20 ((Widening for congruence analysis, continued)). Following this idea of extrapolation threshold, [Gra91b] proposes to improve the widening (20) as follows:

$$\begin{aligned}
p_0 + q_0\mathbb{Z} \mathbin{\nabla} p_1 + q_1\mathbb{Z} = {} & \text{if } (0 \neq |\text{q}_0| \neq |\text{q}_1| \neq 0) \wedge (|\text{q}_0| < \text{r}) \\
& \text{then } \overline{\top} \\
& \text{else } \text{p}_0 + \text{q}_0\mathbb{Z} \sqcup \text{p}_1 + \text{q}_1\mathbb{Z} \ .
\end{aligned}$$

The idea is that extrapolation is necessary only when the modulus is not constant and less than some fixed rational number $r > 0$, which, for example, can be chosen equal to 1 or to some modulus encountered during the first iterates. □

9 Using Widenings to Solve Convergence Problems Left Open in the Literature

Numerous program analysis methods can be found in the literature which can be easily generalized by expressing them as abstract interpretations. Non-convergence problems which are dodged by resorting to restricted classes of programs or to human interaction can be solved using widenings. We consider two examples.

9.1 Simple Sections

Balasundaram and Kennedy [BK89] use simple sections to provide a compact representation of commonly encountered array access shapes in Fortran programs. A *simple section* for program variables $x_1, \ldots, x_n$ is either $\emptyset$ (such that $\gamma(\emptyset) = \text{false}$) or a pair $\langle \ell, u \rangle$ representing the predicate:

$$\begin{aligned}\gamma(\langle \ell, u \rangle) = \quad & \bigwedge_{i=1}^{n} \ell_i \leq x_i \leq u_i \\ \wedge & \bigwedge_{i=1}^{n} \bigwedge_{j=1,\, j\neq i}^{n} \ell_{ij}^{+} \leq x_i + x_j \leq u_{ij}^{+} \\ \wedge & \bigwedge_{i=1}^{n} \bigwedge_{j=1,\, j\neq i}^{n} \ell_{ij}^{-} \leq x_i - x_j \leq u_{ij}^{-}\end{aligned}$$

where the ℓ_i, u_i, ℓ_{ij}^{+}, u_{ij}^{+}, ℓ_{ij}^{-}, u_{ij}^{-} belong to $\mathbb{Z}^{\infty} = \mathbb{Z} \cup \{-\infty, +\infty\}$.

Since Balasundaram and Kennedy consider only relationships between loop indices in Fortran programs that consist of a sequence of perfectly-nested `DO`-loops in which all subroutines calls are expanded inline, they can infer the loop invariants directly from the program text ([BK89], page 47). This solves the convergence problem but for a very particular class of programs only.

The simple sections analysis can be generalized to arbitrary programs using the framework of abstract interpretation. One obtains a slight extension of interval analysis [CC76] and a very restricted form of linear invariants [CH78]. To do this it is formally sufficient to specify the corresponding Galois connection (which is uniquely determined by the function γ above) as well as the widening operator:

$$\begin{aligned}\emptyset \mathbin{\triangledown} \langle \ell, u \rangle &= \langle \ell, u \rangle \\ \langle \ell, u \rangle \mathbin{\triangledown} \emptyset &= \langle \ell, u \rangle \\ \langle \ell, u \rangle \mathbin{\triangledown} \langle \ell', u' \rangle &= \langle \ell'', u'' \rangle\end{aligned}$$

where, x standing for one of the ℓ_i, ℓ_{ij}^{+}, ℓ_{ij}^{-}, x' for ℓ_i', $\ell_{ij}^{+\prime}$, $\ell_{ij}^{-\prime}$ and x'' for the corresponding ℓ_i'', $\ell_{ij}^{+\prime\prime}$, $\ell_{ij}^{-\prime\prime}$, and similarly y, y', y'' standing for bounds in u, u' and u'', we have:

$$\begin{aligned}x'' &= \text{if x}' < \text{x then } -\infty \text{ else x} \\ y'' &= \text{if y}' > \text{y then } +\infty \text{ else y}\end{aligned}$$

and the narrowing operator:

$$\emptyset \mathbin{\triangle} \langle \ell,\, u \rangle = \emptyset$$
$$\langle \ell,\, u \rangle \mathbin{\triangle} \emptyset = \emptyset$$
$$\langle \ell,\, u \rangle \mathbin{\triangle} \langle \ell',\, u' \rangle = \langle \ell'',\, u'' \rangle$$

where:

$$x'' = \text{if x} = -\infty \text{ then x}' \text{ else x}$$
$$y'' = \text{if y} = +\infty \text{ then y}' \text{ else y} \ .$$

Again more precision can be obtained by widening or narrowing to -1, 0, 1 and bounds given by declarations of scalar variables of subrange type or arrays.

The step size of each loop index variable is ignored when computing simple sections. As noticed by [BK89], "this is an inadequacy in the simple section representation". A simple way to cope with this problem is to combine simple sections with arithmetical congruences [Gra89,Gra91b].

9.2 Deriving Constraints on the Sizes of Data Structures

van Gelder [vG90] proposes a method for deriving constraints among argument sizes in logic programs: the set of possible n-tuples of arguments of a logic procedure is approximated by the set of tuples of the sizes of these data structures which in turn is approximated by the polyhedron which is the convex hull of these points in $\mathbb{R}^n$ so as to obtain an invariant in form of a conjunction of inequalities proven to hold among the argument sizes. He shows that this invariant can be defined as a fixpoint of an operator associated with the logic program and observes that the iteration process to compute this fixpoint may not converge in finitely many steps. Therefore he proposes "an heuristic which often works" else resorts to human interaction, verifies experimentally that fixpoints are difficult to guess, therefore indicates in his "directions for further work" that "we need more ways to generate candidates for the fixpoint" and concludes that "there is still much work to be done in the automatic analysis of argument term size constraints".

Thinking in terms of abstract interpretation, a major step towards this goal was taken by [CC77a] who observed that post-fixpoints are upper approximations of the least fixpoint (as shown by Prop. 32 in the appendix) and that post-fixpoints are much easier to compute than fixpoints. Another major step towards this goal was taken by [CC76,CC77a] who used a widening/narrowing approach to enforce convergence of the iterates. As far as linear inequalities are concerned, one can choose the widening proposed in [CH78] and further improved in [Hal79] as follows:

If P_1 and P_2 are two polyhedra in $\mathbb{R}^n$, respectively defined by two sets of linear inequalities $S_1 = \{\beta_1, \beta_2, \ldots, \beta_n\}$ and $S_2 = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ then

$$P_1 \mathbin{\triangledown} P_2 = {S_1}' \cup S_2' \tag{21}$$

where:

- $S_1{}'$ is the subset of S_1 consisting of all inequalities β_i which are satisfied by all points of P_2;
- $S_2{}'$ is the subset of S_2 such that: $\gamma_i \in S_2{}'$ if and only if there exists $\beta_j \in S_1$ such that $(S_1 - \{\beta_j\}) \cup \{\gamma_i\}$ defines the same polyhedron than P_1.

Observe that this widening also mitigates the non-existence of least upper bounds (the circle is the limit of inscribed polygons), see [CC92b].

Example 21 ((Widening for linear inequality analysis)). If $P_1 = \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid 0 \leq x \leq 1 \wedge y = 0\}$ and $P_2 = \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid x \leq 2 \wedge 0 \leq y \wedge y \leq x\}$ then $S_1 = \{0 \leq x\,,\ x \leq 1\,,\ y \leq 0\,,\ 0 \leq y\}$ and $S_2 = \{x \leq 2\,,\ 0 \leq y\,,\ y \leq x\}$. The extremal points of P_2 are $\langle 0,\, 0\rangle$, $\langle 2,\, 0\rangle$ and $\langle 2,\, 2\rangle$. They only satisfy the constraints $0 \leq x$ and $0 \leq y$ in S_1 so that $S_1' = \{0 \leq x\,,\ 0 \leq y\}$. The constraint $0 \leq x$ of S_1 can be replaced by $y \leq x$ without changing P_1. The constraint $0 \leq y$ appears in S_1 and S_2. The constraint $x \leq 2$ can replace no constraint in S_1 without changing P_1. It follows that $S_2' = \{0 \leq y\,,\ y \leq x\}$. We have $S_1' \cup S_2' = \{0 \leq x\,,\ 0 \leq y\,,\ y \leq x\}$ where the constraint $0 \leq x$ is redundant. Consequently, $P_1 \bigtriangledown P_2 = \{\langle x,\, y\rangle \in \mathcal{R}^2 \mid 0 \leq y \leq x\}$. □

A simple narrowing is obtained by limiting the length of the decreasing iteration sequence to some $k \geq 1$ (experience shows that $k > 1$ often brings no significant improvement).

Example 22 ((Argument size analysis)). The logic procedure below [vG90] might test for precedence in some partial order, thinking of s as successor:

$$\begin{array}{c} p(X,\, X) \\ p\,(X,\, s(Y)) \leftarrow p(X, Y)\ . \end{array}$$

Knowing that $\mathrm{size}(c) = 0$ if c is a constant and that $\mathrm{size}\,(s(Y)) = 1 + \mathrm{size}(Y)$, the constraints among argument sizes of predicate p are upper approximations to the least solution of the fixpoint equation:

$$p = F(p) = \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid x \geq 0 \wedge y \geq 0 \wedge ((x = y) \vee \langle x,\, y - 1\rangle \in p)\}$$

which can be effectively computed by the following iteration sequence with widening (21):

$$\begin{array}{l} p^0 = \emptyset \\ p^1 = p^0 \bigtriangledown F(p^0) = \emptyset \bigtriangledown F(p^0) = F(p^0) = \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid 0 \leq x = y\} \\ p^2 = p^1 \bigtriangledown F(p^1) \\ \quad = \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid 0 \leq x = y\} \bigtriangledown \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid 0 \leq x \leq y \leq x + 1\} \\ \quad = \{\langle x,\, y\rangle \in \mathbb{R}^2 \mid 0 \leq x \leq y\} \end{array}$$

which is such that $F(p^2) = p^2$. □

Let us conclude with [vG90] that "the method may be applicable to other languages in which the sizes of data structures can be determined syntactically" and refer to chapter 7.3 of [Hal79] for examples illustrating this point of view.

10 Conclusion

The widening/narrowing approach [CC77a] to abstract interpretation, which is more powerful than the popular variations on the Galois connection approach [CC77a], deserves to be better understood since it can significantly improve the precision of the analyses as well as the speed of convergence including in the case of finite lattices which too large for the fixpoint finding problem to be tractable. Our practical experience is that the combination of the two approaches using infinite abstract domains is worthwhile.

Acknowledgments. We thank P. Granger and C. Hankin for their comments on a first version of this paper.

References

AH87. S. Abramsky and C. Hankin, editors. *Abstract Interpretation of Declarative Languages*. Computers and their Applications. Ellis Horwood, Chichester, U.K., 1987. 279

BJCD87. M. Bruynooghe, G. Janssens, A. Callebaut, and B. Demoen. Abstract interpretation: towards the global optimization of Prolog programs. In *Proceedings of the 1987 International Symposium on Logic Programming*, San Francisco, California, pages 192–204. IEEE Computer Society Press, Los Alamitos, California, August 31–September 4, 1987. 273, 278, 279

BK89. V. Balasundaram and K. Kennedy. A technique for summarizing data access and its use in parallelism enhancing transformations. In *SIGPLAN'89 Conference on Programming Language Design and Implementation*, pages 41–53, Portland, Oregon, June 21–23, 1989. 289, 290

Bou90. F. Bourdoncle. Interprocedural abstract interpretation of block structured languages with nested procedures, aliasing and recursivity. In P. Deransart and Małuszyński, editors, *Proceedings of the International Workshop PLILP'90, Programming Language Implementation and Logic Programming*, Linköping, Sweden, Lecture Notes in Computer Science 456, pages 307–323. Springer-Verlag, Berlin, Germany, August 20–22, 1990. 277, 281

Bou92. F. Bourdoncle. Abstract interpretation by dynamic partitioning. *Journal of Functional Programming* 2(4):407–435, 1992. 279

CC76. P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In *Proceedings of the 2nd International Symposium on Programming*, pages 106–130. Dunod, Paris, France, 1976. 269, 271, 275, 276, 278, 279, 289, 290

CC77a. P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the 44th ACM Symposium on Principles of Programming Languages*, pages 238–252, Los Angeles, California, 1977. 269, 271, 275, 278, 279, 290, 292

CC77b. P. Cousot and R. Cousot. Static determination of dynamic properties of recursive procedures. In E.J. Neuhold, editor, *IFIP Conference on Formal Description of Programming Concepts*, St-Andrews, N.B., Canada, pages 237–277. North-Holland Pub. Co., Amsterdam, the Netherlands, 1977. 271, 272

CC79a. P. Cousot and R. Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific Journal of Mathematics*, 82(1):43–57, 1979. 269

CC79b. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Conference Record of the 6$^{\text{th}}$ ACM Symposium on Principles of Programming Languages*, pages 269–282, San Antonio, Texas, 1979. 269, 271, 273, 279, 284, 285, 286

CC92a. P. Cousot and R. Cousot. P. Cousot and R. Cousot. Abstract interpretation and application to logic programs. *Journal of Logic Programming*, 13(2–3):103–179, 1992. 273

CC92b. P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation* 2(4):511–547, 1992. 270, 273, 279, 291, 295

CC92c. P. Cousot and R. Cousot. Inductive definitions, semantics and abstract interpretation. In *Conference Record of the 19$^{\text{th}}$ ACM Symposium on Principles of Programming Languages*, pages 83–94, Albuquerque, New Mexico, 1992. 270

CH78. P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the 5$^{\text{th}}$ ACM Symposium on Principles of Programming Languages*, pages 84–97, Tucson, Arizona, 1978. 279, 282, 289, 290

Cou78. P. Cousot. *Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes*. Thèse d'état ès sciences mathématiques, Université scientifique et médicale de Grenoble, Grenoble, France, 21 March 1978. 273, 279

Cou81. P. Cousot. Semantic foundations of program analysis. In S. S. Muchnick and N. D. Jones, editors, *Program Flow Analysis: Theory and Applications*, chapter 10, pages 303–342. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981. 275, 279

Deu92. A. Deutsch. A storeless model of aliasing and its abstraction using finite representations of right-regular equivalence relations. In *Proceedings of the 1992 International Conference on Computer Languages*, Oakland, California, pages 2–13. IEEE Computer Society Press, Los Alamitos, California, April 20–23, 1992. 279

Gra89. P. Granger. Static analysis of arithmetical congruences. *International Journal of Computer Mathematics*, 30:165–190, 1989. 290

Gra91a. P. Granger. *Analyses sémantiques de congruence*. Thèse de l'École Polytechnique en informatique, LIX, École Polytechnique, Palaiseau, France, 12 July 1991. 281

Gra91b. P. Granger. Static analysis of linear congruence equalities among variables of a program. In S. Abramsky and T.S.E. Maibaum, editors, *TAPSOFT'91, Proceedings of the International Joint Conference on Theory and Practice of Software Development*, Brighton, U.K., Volume 1 (CAAP'91), Lecture Notes in Computer Science 493, pages 169–192. Springer-Verlag, Berlin, Germany, 1991. 281, 287, 288, 290

Hal79. N. Halbwachs. *Détermination automatique de relations linéaires vérifiées par les variables d'un programme*. Thèse de 3$^{\text{ème}}$ cycle d'informatique, Université scientifique et médicale de Grenoble, Grenoble, France, 12 March 1979. 282, 290, 291

HH90. C. Hankin and S. Hunt. Approximate fixed points in abstract interpretation. In B. Krieg-Brückner, editor, *Proceedings of the 4$^{\text{th}}$ European Symposium on Programming, ESOP '92*, pages 219–232. Springer-Verlag, Berlin, Germany, Rennes, France, February 26–28 1990. 279, 285

JB92. G. Janssens and M. Bruynooghe. On abstracting the procedural behaviour of logic programs. In A. Voronkov, editor, *Proceedings of the First Russian Conference on Logic Programming, Irkutsk, Russia, September 14–18, 1990 and of the Second Russian Conference on Logic Programming, St. Petersburg, Russia, September 11-16, 1991*, pages 240–262. Springer-Verlag, Berlin, Germany, 1992. 273

KN87. R. B. Kieburtz and M. Napierala. Abstract semantics. In S. Abramsky and C. Hankin, editors, *Abstract Interpretation of Declarative Languages*, chapter 7, pages 143–180. Ellis Horwood, Chichester, U.K., 1987. 279

MS88. K. Marriott and H. Søndergaard. On describing success patterns of logic programs. Technical Report 88/12, Department of Computer Science, University of Melbourne, Melbourne, Australia, 1988. 279

Myc80. A. Mycroft. The theory and practice of transforming call-by-need into call-by-value. In B. Robinet, editor, *Proceedings of the Fourth International Symposium on Programming*, Paris, France, 22-24 April 1980, Lecture Notes in Computer Science 83, pages 270–281. Springer-Verlag, Berlin, Germany, 1980. 274

Str88. J. Stransky. *Analyse sémantique de structures de données dynamiques avec application au cas particulier de langages LISPiens.* Thèse de doctorat en science, Université de Paris-sud, Orsay, 28 June 1988. 279

vEK76. M. H. van Emden and R. A. Kowalski. The semantics of predicate logic as a programming language. *Journal of the Association for Computing Machinary*, 23(4):733–742, October 1976. 270

vG90. A. van Gelder. Deriving constraints among argument sizes in logic programs. In *Proceedings of the 9th ACM Symposium on Principles of Database Systems*, pages 47–60, Nashville, Tennesse, 1990. 290, 291

Appendix

A *preorder* is a preordered set $L(\sqsubseteq)$ where $\sqsubseteq \in \wp(L \times L)$ is reflexive ($\forall x \in L : x \sqsubseteq x$) and transitive ($\forall x, y, z \in L$: $(x \sqsubseteq y \wedge y \sqsubseteq z) \Longrightarrow x \sqsubseteq z$). A *poset* is a preorder $L(\sqsubseteq)$ where $\sqsubseteq$ is antisymmetric ($\forall x, y \in L$: $(x \sqsubseteq y \wedge y \sqsubseteq x) \Longrightarrow x = y$). An *upper bound* u of $X \subseteq L$ is such that $\forall x \in X : x \sqsubseteq u$. The *least upper bound*, written $\sqcup X$ is an upper bound such that for all upper bounds u, $\sqcup X \sqsubseteq u$. When it exists, the least upper bound $\sqcup X$ is unique, by antisymmetry. A *strict* poset has an *infimum* $\bot$ such that $\forall x \in L : \bot \sqsubseteq x$. A *cpo* is a *complete* poset i.e., such that any $\mathbb{N}$-termed sequence $c_i \in L$, $i \in \mathbb{N}$, which is an *increasing chain* (i.e., $\forall i \in \mathbb{N} : c_i \sqsubseteq c_{i+1}$) has a least upper bound $\bigsqcup_{i \in \mathbb{N}} c_i$. A *complete lattice* is a poset such that every subset $X \subseteq L$ has a least upper bound $\sqcup X$ and a greatest lower bound $\sqcap X$. A map $F \in L \stackrel{\text{mon}}{\longmapsto} L$ is *monotone* i.e. $x \sqsubseteq y$ implies $F(x) \sqsubseteq F(y)$. It is *continuous* (written $F \in L \stackrel{\text{con}}{\longmapsto} L$) if and only if $F(\bigsqcup_{i \in \mathbb{N}} c_i) = \bigsqcup_{i \in \mathbb{N}} F(c_i)$ for all increasing chains $c_i \in L$, $i \in \mathbb{N}$ such that the least upper bound $\bigsqcup_{i \in \mathbb{N}} c_i$ exists. Continuity implies monotony.

Proposition 23 ((Kleene fixpoint theorem)). *If $L(\sqsubseteq, \sqcup)$ is a poset, $F \in L \stackrel{\text{con}}{\longmapsto} L$ is continuous, and $\bot \in L$ is such that $\bot \sqsubseteq F(\bot)$[8], then $F^n(\bot)$,*

[8] If this is not true and L is a lattice, we can iterate with $\lambda X \cdot X \sqcup F(X)$ instead.

$n \in \mathbb{N}$ *is an increasing chain. If* $\bigsqcup_{n\in\mathbb{N}} F^n(\bot)$ *exists*[9]*, then it is the least fixpoint* $\mathrm{lfp}_\bot(F)$ *of* F *greater than or equal to* $\bot$.

Proposition 24 ((Characteristic property of Galois connections)). *If* $L(\sqsubseteq)$ *and* $\overline{L}(\overline{\sqsubseteq})$ *are posets, then (2) is equivalent to:*

$$\forall x \in L : x \sqsubseteq \gamma \circ \alpha(x) \quad \textit{and} \tag{22}$$
$$\forall \overline{x} \in \overline{L} : \alpha \circ \gamma(\overline{x}) \mathrel{\overline{\sqsubseteq}} \overline{x} \quad \textit{and} \tag{23}$$
$$\alpha \in L \overset{\text{mon}}{\longmapsto} \overline{L} \quad \textit{and} \tag{24}$$
$$\gamma \in \overline{L} \overset{\text{mon}}{\longmapsto} L \ . \tag{25}$$

Proposition 25 ((Functional Galois connection)). *If* $L(\sqsubseteq)$ *and* $\overline{L}(\overline{\sqsubseteq})$ *are posets, and* $\alpha \in L \longmapsto \overline{L}$ *and* $\gamma \in \overline{L} \longmapsto L$ *satisfy (2), then (4) implies (5).*

Proposition 26 ((Function approximation)). *If* $L(\sqsubseteq)$ *and* $\overline{L}(\overline{\sqsubseteq})$ *are posets,* $F \in L \overset{\text{mon}}{\longmapsto} L$ *and* $\overline{F} \in \overline{L} \overset{\text{mon}}{\longmapsto} \overline{L}$ *are monotone, then (2) and (4) imply that* $\vec{\alpha}(F) \mathrel{\overline{\sqsubseteq}} \overline{F}$ *is equivalent to* $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$*, or to* $\alpha \circ F \mathrel{\overline{\sqsubseteq}} \overline{F} \circ \alpha$.

Proposition 27 ((Least upper bounds inducing)). *If* $L(\sqsubseteq, \sqcup)$ *and* $\overline{L}(\overline{\sqsubseteq}, \overline{\sqcup})$ *are posets such that* $L \underset{\alpha'}{\overset{\gamma}{\leftrightarrows}} \overline{L}$ *is a Galois connection,* $X \subseteq L$*, and* $\sqcup X$ *exists, then* $\overline{\bigsqcup}_{x\in X} \alpha(x)$ *exists and is equal to* $\alpha(\sqcup X)$.

Proposition 28 ((Connection property)). *If* $L(\sqsubseteq)$ *and* $\overline{L}(\overline{\sqsubseteq})$ *are posets such that* $L \underset{\alpha'}{\overset{\gamma}{\leftrightarrows}} \overline{L}$ *is a Galois connection, then:*

$$\alpha \circ \gamma \circ \alpha = \alpha \tag{26}$$
$$\gamma \circ \alpha \circ \gamma = \gamma \ . \tag{27}$$

Proposition 29 ((Galois surjection)). *If* $L(\sqsubseteq)$ *and* $\overline{L}(\overline{\sqsubseteq})$ *are posets such that* $L \underset{\alpha'}{\overset{\gamma}{\leftrightarrows}} \overline{L}$ *is a Galois connection, then* α *is surjective if and only if* $\forall \overline{x} \in \overline{L}$*:* $\alpha \circ \gamma(\overline{x}) = \overline{x}$.

Proposition 30 ((Fixpoint abstraction)). *If* $L(\sqsubseteq, \sqcup)$ *is a cpo,* $\overline{L}(\overline{\sqsubseteq}, \overline{\sqcup})$ *is a poset*[10]*,* $L \underset{\alpha'}{\overset{\gamma}{\leftrightarrows}} \overline{L}$ *is a Galois connection,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,* $\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$*, and* $\vec{\alpha}$ *is defined by (4), then* $\mathrm{lfp}_\bot(F) \mathrel{\overline{\sqsubseteq}} \gamma(\overline{A})$ *where the least upper bound* $\overline{A} = \overline{\bigsqcup}_{n\in\mathbb{N}} \vec{\alpha}(F)^n\big(\alpha(\bot)\big)$ *of the increasing chain* $\vec{\alpha}(F)^n\big(\alpha(\bot)\big)$*,* $n \in \mathbb{N}$ *exists and is such than* $\overline{A} \mathrel{\overline{\sqsubseteq}} \vec{\alpha}(F)(\overline{A}) \mathrel{\overline{\sqsubseteq}} \overline{x}$ *whenever* $\overline{\bot} \mathrel{\overline{\sqsubseteq}} \overline{x} = \vec{\alpha}(F)(\overline{x})$*. In particular, if* $\vec{\alpha}(F) \in \overline{L} \overset{\text{con}}{\longmapsto} \overline{L}$*, then* $\overline{A} = \mathrm{lfp}_{\alpha(\bot)}\big(\vec{\alpha}(F)\big)$*, but this equality does not hold in general*[11].

Proposition 31 ((Fixpoint abstract approximation)). *If* $L(\sqsubseteq, \sqcup)$ *is a cpo,* $\overline{L}(\overline{\sqsubseteq}, \overline{\sqcup})$ *is a poset,* $L \underset{\alpha'}{\overset{\gamma}{\leftrightarrows}} \overline{L}$ *is a Galois connection,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,*

[9] This is the case when $L(\sqsubseteq, \sqcup)$ is a cpo.
[10] Not necessarily a cpo, see [CC92b] for even weaker hypotheses.
[11] But it does by considering transfinite iterates.

$\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$, $\vec{\alpha}$ *is defined by (4),* $\overline{F} \in \overline{L} \longmapsto \overline{L}$ *is such that* $\vec{\alpha}(F)$ $\overline{\sqsubseteq}$ $\overline{F}$ *for the pointwise ordering* $\overline{\sqsubseteq}$, $\overline{\bot} \in \overline{L}$ *is such that* $\alpha(\bot)$ $\overline{\sqsubseteq}$ $\overline{\bot}$, *and* $\overline{A} \in \overline{L}$ *is such that* $\forall n \in \mathbb{N}$: $\overline{F}^n(\overline{\bot})$ $\overline{\sqsubseteq}$ $\overline{A}$, *then,* $\mathrm{lfp}_{\bot}(F)$ $\overline{\sqsubseteq}$ $\gamma\left(\overline{\bigsqcup}_{n\in\mathbb{N}} \vec{\alpha}(F)^n\left(\alpha(\bot)\right)\right)$ $\overline{\sqsubseteq}$ $\gamma(\overline{A})$.

Proposition 32 ((Approximation by postfixpoints)). *If* $L(\sqsubseteq, \sqcup)$ *is a poset,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,* $\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$, $\bigsqcup_{n\in\mathbb{N}} F^n(\bot)$ *exists, and* $A \in L$ *is such that* $\bot \sqsubseteq A$ *and* $F(A) \sqsubseteq A$, *then* $\mathrm{lfp}_{\bot}(F) \sqsubseteq A$.

Proposition 33 ((Upward iteration sequence with widening)). *If* $L(\sqsubseteq, \sqcup)$ *is a cpo,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,* $\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$, $\triangledown \in L \times L \longmapsto L$ *satisfies (6), (7) and (8), then the upward iteration sequence with widening* $\hat{X}^n$, $n \in \mathbb{N}$ *defined by (9) is ultimately stationary and its limit* $\hat{A}$ *is such that* $\mathrm{lfp}_{\bot}\, F \sqsubseteq \hat{A}$ *and* $F(\hat{A}) \sqsubseteq \hat{A}$.

Proposition 34 ((Downward iteration sequence with narrowing)). *If* $L(\sqsubseteq, \sqcup)$ *is a cpo,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,* $\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$, $\triangle \in L \times L \longmapsto L$ *satisfies (10) and (11), then the downward iteration sequence with narrowing* $\check{X}^n$, $n \in \mathbb{N}$ *defined by (12), where* $\mathrm{lfp}_{\bot}\, F \sqsubseteq \hat{A}$ *and* $F(\hat{A}) \sqsubseteq \hat{A}$, *is ultimately stationary and all terms* $\check{X}^n$, $n \in \mathbb{N}$ *are such that* $\mathrm{lfp}_{\bot}\, F \sqsubseteq F(\check{X}^n) \sqsubseteq \check{X}^n$.

Proposition 35 ((Preordered upward iteration with widening)). *If* $L(\sqsubseteq, \sqcup)$ *is a poset,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,* $\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$, $\bigsqcup_{n\in\mathbb{N}} F^n(\bot)$ *exists,* $\overline{L}$ *is a set,* $\gamma \in \overline{L} \longmapsto L$, $\overline{\sqsubseteq}$ *is the preorder defined by* $x \,\overline{\sqsubseteq}\, y \overset{\text{def}}{=} \gamma(x) \sqsubseteq \gamma(y)$, $\overline{\bot} \in \overline{L}$ *is such that* $\bot \sqsubseteq \gamma(\overline{\bot})$, $\overline{F} \in \overline{L} \overset{\text{mon}}{\longmapsto} \overline{L}$ *is such that* $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$ *and* $\triangledown \in \overline{L} \times \overline{L} \longmapsto \overline{L}$ *satisfies (6), (7) and (8) (where* $\sqsubseteq$, $\bot$ *and* F *are respectively* $\overline{\sqsubseteq}$, $\overline{\bot}$ *and* $\overline{F}$*) then the upward iteration sequence with widening (9) is ultimately stationary with limit* $\hat{A}$ *such that* $\mathrm{lfp}_{\bot}(F) \sqsubseteq \gamma(\hat{A})$ *and* $\overline{F}(\hat{A})$ $\overline{\sqsubseteq}$ $\hat{A}$.

Proposition 36 ((Preordered downward iteration with narrowing)). *If* $L(\sqsubseteq, \sqcup)$ *is a poset,* $F \in L \overset{\text{con}}{\longmapsto} L$ *is continuous,* $\bot \in L$ *is such that* $\bot \sqsubseteq F(\bot)$, $\bigsqcup_{n\in\mathbb{N}} F^n(\bot)$ *exists,* $\overline{L}$ *is a set,* $\gamma \in \overline{L} \longmapsto L$, $\overline{\sqsubseteq}$ *is the preorder defined by* $x \,\overline{\sqsubseteq}\, y \overset{\text{def}}{=} \gamma(x) \sqsubseteq \gamma(y)$, $\overline{\bot} \in \overline{L}$ *is such that* $\bot \sqsubseteq \gamma(\overline{\bot})$, $\overline{F} \in \overline{L} \overset{\text{mon}}{\longmapsto} \overline{L}$ *is such that* $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$ *and* $\triangle \in \overline{L} \times \overline{L} \longmapsto \overline{L}$ *satisfies (10) and (11) where* $\sqsubseteq$ *is* $\overline{\sqsubseteq}$, *then the downward iteration sequence with narrowing* $\check{X}^n$, $n \in \mathbb{N}$ *defined by (12) where* F *is* $\overline{F}$, $\mathrm{lfp}_{\bot}(F) \sqsubseteq \gamma(\hat{A})$ *and* $\overline{F}(\hat{A})$ $\overline{\sqsubseteq}$ $\hat{A}$, *is ultimately stationary and all terms* $\check{X}^n$, $n \in \mathbb{N}$ *are such that* $\mathrm{lfp}_{\bot}\, F \sqsubseteq \gamma(\check{X}^n)$ *and* $\overline{F}(\check{X}^n)$ $\overline{\sqsubseteq}$ $\check{X}^n$.

Proposition 37 ((Widening inducing)). *Let* $L(\sqsubseteq)$ *and* $\overline{\overline{L}}(\overline{\overline{\sqsubseteq}})$ *be posets and* $\overline{L}(\overline{\sqsubseteq}, \overline{\sqcup})$ *be a join-semi-lattice satisfying the ascending chain condition, such that* $L \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \overline{L}$, $L \underset{\overline{\alpha}}{\overset{\overline{\gamma}}{\leftrightarrows}} \overline{\overline{L}}$ *and* $\overline{\alpha}$ *is surjective. Then* $\overline{\triangledown} \in \overline{\overline{L}} \times \overline{\overline{L}} \longmapsto \overline{\overline{L}}$ *defined by (16) is a widening on* $\overline{\overline{L}}$. *(6) and (7) may not hold when* $\overline{\alpha}$ *is not surjective.*

Proposition 38 ((Narrowing inducing)). *if* L $(\sqsubseteq, \sqcap)$ *is a meet-semi-lattice and* $\overline{L}$ $(\overline{\sqsubseteq})$ *is a poset satisfying the descending chain condition then* $\triangle \in L \times L \longmapsto L$ *defined by (17) is a narrowing satisfying (10) and (11).*

This article was processed using the LaTeX macro package with LLNCS style

This article is an invited paper reprinted from the proceedings of the fourth international symposium PLILP'92 :

"Programming Language Implementation and Logic Programming"
Leuven, Belgium, August 13–17, 1992
Lecture Notes in Computer Science 631
Maurice Bruynooghe and Martin Wirsing (Eds.)