AAA616: Program Analysis

Lecture 4 — Abstract Interpretation Framework

Hakjoo Oh
2016 Fall

Abstract Interpretation Framework

A powerful framework for designing correct static analysis

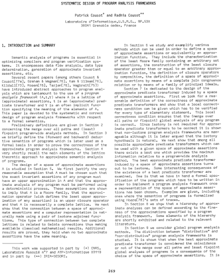
- "framework": correct static analysis comes out, reusable
- "powerful": all static analyses are understood in this framework
- "simple": prescription is simple
- "eye-opening": any static analysis is an abstract interpretation


ABSTRACT INTERPRETATION : A UNIFIED LATTICE MODEL FOR STATIC ANALYSIS OF PROGRAMS BY CONSTRUCTION OR APPROXIMATION OF FIXPOINTS

Patrick Cousot and Radhia Cousot

Laboratoire d'Informatique, U.S.M.G., BP. 53 38041 Grenoble cedex, France

CC77


SYSTEMATIC DESIGN OF PROGRAM ANALYSIS FRAMEWORKS

Patrick Cousot and Radhia Cousot

Laboratoire d'Informatique,U.S.M.G., BP.53X 38041 Grenoble cedex, France

CC79

Step 1: Define Concrete Semantics

The concrete semantics describes the real executions of the program.
Described by semantic domain and function.

- A *semantic domain* D, which is a CPO:
  - D is a partially ordered set with a least element $\bot$.
  - Any increasing chain $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$ in D has a least upper bound $\bigsqcup_{n \geq 0} d_n$ in D.
- A *semantic function* $F : D \to D$, which is continuous: for all chains $d_0 \sqsubseteq d_1 \sqsubseteq \ldots$,

$$F(\bigsqcup_{n \geq 0} d_i) = \bigsqcup_{n \geq 0} F(d_n).$$

Then, the concrete semantics (or collecting semantics) is defined as the least fixed point of *semantic function* $F : D \to D$:

$$\mathit{fix} F = \bigsqcup_{i \in N} F^i(\bot).$$

Example: Concrete Semantics

- Program representation:
  - P is represented by control flow graph $(\mathbb{C}, \rightarrow, c_0)$
  - Each program point c is associated with a command $\mathbf{cmd}(c)$

$$
\begin{array}{rcl}
cmd & \rightarrow & skip \mid x := e \\
e & \rightarrow & n \mid x \mid e + e \mid e - e.
\end{array}
$$

- Semantics of commands:
- Concrete memory states: $\mathbb{M} = \textbf{Var} \rightarrow \mathbb{Z}$
- Concrete semantics:

$$
\llbracket c \rrbracket \quad : \quad \mathbb{M} \rightarrow \mathbb{M}
$$

$$
\begin{array}{rcl}
\llbracket skip \rrbracket(m) & = & m \\
\llbracket x := e \rrbracket(m) & = & m[x \mapsto \llbracket e \rrbracket(s)]
\end{array}
$$

$$
\llbracket e \rrbracket \quad : \quad \mathbb{M} \rightarrow \mathbb{Z}
$$

$$
\begin{array}{rcl}
\llbracket n \rrbracket(m) & = & n \\
\llbracket x \rrbracket(m) & = & m(x) \\
\llbracket e_1 + e_2 \rrbracket(m) & = & \llbracket e_1 \rrbracket(m) + \llbracket e_2 \rrbracket(m) \\
\llbracket e_1 - e_2 \rrbracket(m) & = & \llbracket e_1 \rrbracket(m) + \llbracket e_2 \rrbracket(m)
\end{array}
$$

Example: Concrete Semantics

- Program states: $\textbf{State} = \mathbb{C} \times \mathbb{M}$
- A trace $\sigma \in \textbf{State}^+$ is a (partial) execution sequence of the program:

$$\sigma_0 \in I \ \wedge \ \forall k. \sigma_k \rightsquigarrow \sigma_{k+1}$$

where $I \subseteq \textbf{State}$ is the initial program states

$$I = \{(c_0, m_0) \mid m_0 \in \mathbb{M}\}$$

and $(\rightsquigarrow) \subseteq \textbf{State} \times \textbf{State}$ is the relation for the one-step execution:

$$(c_i, s_i) \rightsquigarrow (c_j, s_j) \iff c_i \to c_j \ \wedge \ s_j = [\![\textbf{cmd}(c_j)]\!](s_i)$$

Example: Concrete Semantics

The collecting semantics of program P is defined as the set of all finite traces of the program:

$$[\![P]\!] = \{\sigma \in \mathbf{State}^{+} \mid \sigma_0 \in I \;\wedge\; \forall k.\sigma_k \rightsquigarrow \sigma_{k+1}\}$$

The semantic domain:

$$D = \wp(\mathbf{State}^{+})$$

The semantic function:

$$\begin{array}{rcl} F & : & \wp(\mathbf{State}^{+}) \to \wp(\mathbf{State}^{+}) \\ F(\Sigma) & = & I \cup \{\sigma \cdot (c, m) \mid \sigma \in \Sigma \;\wedge\; \sigma_{\dashv} \rightsquigarrow (c, m)\} \end{array}$$

Lemma

$[\![P]\!] = \mathit{fix} F$.

Step 2: Define Abstract Semantics

Define the abstract semantics of the input program.

- Define an *abstract semantic domain* CPO $\hat{D}$.
  - Intuition: $\hat{D}$ is an abstraction of D
- Define an *abstract semantic function* $\hat{F} : \hat{D} \to \hat{D}$.
  - Intuition: $\hat{F}$ is an abstraction of F.
  - $\hat{F}$ must be monotone:

$$\forall \hat{x}, \hat{y} \in \hat{D}.\ \hat{x} \sqsubseteq \hat{y} \implies \hat{F}(\hat{x}) \sqsubseteq \hat{F}(\hat{y})$$

(or extensive: $\forall x \in \hat{D}.\ x \sqsubseteq \hat{F}(x)$)

Then, static analysis is to compute an upper bound of:

$$\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\bot)$$

How can we ensure that the result soundly approximate the concrete semantics?

Requirement 1: Galois Connection

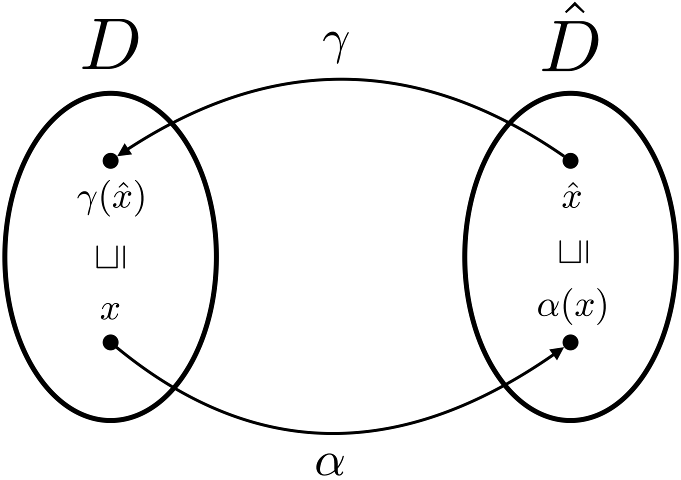
D and $\hat{D}$ must be related with Galois-connection:

$$D \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{D}$$

That is, we have

- *abstraction function*: $\alpha \in D \to \hat{D}$
  - represents elements in D as elements of $\hat{D}$
- *concretization function*: $\gamma \in \hat{D} \to D$
  - gives the meaning of elements of $\hat{D}$ in terms of D
- $\forall x \in D, \hat{x} \in \hat{D}.\ \alpha(x) \sqsubseteq \hat{x} \iff x \sqsubseteq \gamma(\hat{x})$
  - α and γ respect the orderings of D and $\hat{D}$

Galois-Connection



Example: Sign Abstraction

Sign abstraction:

$$\wp(\mathbb{Z}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \{\bot, +, 0, -\top\}$$

where

$$\begin{array}{rcl}
\alpha(Z) & = & \begin{cases} \bot & Z = \emptyset \\ + & \forall z \in Z.\ z > 0 \\ 0 & Z = \{0\} \\ - & \forall z \in Z.\ z < 0 \\ \top & \text{otherwise} \end{cases} \\
\gamma(\bot) & = & \emptyset \\
\gamma(\top) & = & \mathbb{Z} \\
\gamma(+) & = & \{z \in \mathbb{Z} \mid z > 0\} \\
\gamma(0) & = & \{0\} \\
\gamma(-) & = & \{z \in \mathbb{Z} \mid z < 0\}
\end{array}$$

Example: Interval Abstraction

$$\wp(\mathbb{Z}) \xleftrightarrows[\alpha]{\gamma} \{\perp\} \cup \{[a,b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}\}$$

$$\begin{array}{rcl}
\gamma(\perp) & = & \emptyset \\
\gamma([a,b]) & = & \{z \in \mathbb{Z} \mid a \le z \le b\} \\
\gamma([a,+\infty]) & = & \{z \in \mathbb{Z} \mid z \ge a\} \\
\gamma([-\infty,b]) & = & \{z \in \mathbb{Z} \mid z \le b\} \\
\gamma([-\infty,+\infty]) & = & \mathbb{Z}
\end{array}$$

Requirement 2: $\hat{F}$ and F

- $\hat{F}$ and F must satisfy

$$\alpha \circ F \sqsubseteq \hat{F} \circ \alpha \quad (\text{i.e., } F \circ \gamma \sqsubseteq \gamma \circ \hat{F})$$

- or, alternatively,

$$\alpha(x) \sqsubseteq \hat{x} \implies \alpha(F(x)) \sqsubseteq \hat{F}(\hat{x})$$

Soundness Guarantee

Theorem (Fixpoint Transfer)

Let D *and* $\hat{D}$ *be related by Galois-connection* $D \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{D}$. *Let* $F : D \to D$ *be a continuous function and* $\hat{F} : \hat{D} \to \hat{D}$ *be a monotone function such that* $\alpha \circ F \sqsubseteq \hat{F} \circ \alpha$. *Then,*

$$\alpha(\mathit{fix} F) \sqsubseteq \bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot}).$$

Theorem (Fixpoint Transfer2)

Let D *and* $\hat{D}$ *be related by Galois-connection* $D \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{D}$. *Let* $F : D \to D$ *be a continuous function and* $\hat{F} : \hat{D} \to \hat{D}$ *be a monotone function such that* $\alpha(x) \sqsubseteq \hat{x} \implies \alpha(F(x)) \sqsubseteq \hat{F}(\hat{x})$. *Then,*

$$\alpha(\mathit{fix} F) \sqsubseteq \bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot}).$$

A Property of Galois-Connection

The functional composition of two Galois-connections is also Galois-connection:

Lemma

If $D_1 \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} D_2$ *and* $D_2 \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} D_3$, *then*

$$D_1 \underset{\alpha_2 \circ \alpha_1}{\overset{\gamma_1 \circ \gamma_2}{\leftrightarrows}} D_3.$$

Proof.

Exercise □

Example: Partitioning Abstraction

Galois-connection: $\wp(\textbf{State}^+) \underset{\alpha_1}{\overset{\gamma_1}{\leftrightarrows}} \mathbb{C} \to \wp(\mathbb{M})$

$$\alpha_1(\Sigma) \;=\; \lambda c.\{m \in \mathbb{M} \mid \exists \sigma \in \Sigma \;\wedge\; \exists i.\sigma_i = (c, m)\}$$

Semantic function:

$$\hat{F}_1 : (\mathbb{C} \to \wp(\mathbb{M})) \to (\mathbb{C} \to \wp(\mathbb{M}))$$

$$\hat{F}_1(X) = \alpha_1(I) \sqcup \lambda c \in \mathbb{C}.\; f_c(\bigcup_{c' \to c} X(c'))$$

where $f_c : \wp(\mathbb{M}) \to \wp(\mathbb{M})$ is a *transfer function* at program point c:

$$f_c(M) \;=\; \{m' \mid m \in M \;\wedge\; m' = [\![\textbf{cmd}(c)]\!](m)\}$$

Lemma (Soundness of Partitioning Abstraction)

$\alpha_1(\mathit{fix}\,F) \sqsubseteq \bigsqcup_{i \in \mathbb{N}} \hat{F}_1^i(\bot)$.

Example: Memory State Abstraction

Galois-connection:

$$\mathbb{C} \to \wp(\mathbb{M}) \underset{\alpha_2}{\overset{\gamma_2}{\leftrightarrows}} \mathbb{C} \to \hat{\mathbb{M}}$$

$$\begin{array}{rcl} \alpha_2(f) & = & \lambda c.\ \alpha_m(f(c)) \\ \gamma_1(\hat{f}) & = & \lambda c.\ \gamma_m(\hat{f}(c)) \end{array}$$

where we assume

$$\wp(\mathbb{M}) \underset{\alpha_m}{\overset{\gamma_m}{\leftrightarrows}} \hat{\mathbb{M}}$$

Semantic function $\hat{F} : (\mathbb{C} \to \hat{\mathbb{M}}) \to (\mathbb{C} \to \hat{\mathbb{M}})$:

$$\hat{F}(X) = (\alpha_2 \circ \alpha_1)(I) \sqcup \lambda c \in \mathbb{C}.\ \hat{f}_c(\bigsqcup_{c' \to c} X(c'))$$

where *abstract transfer function* $\hat{f}_c : \hat{\mathbb{M}} \to \hat{\mathbb{M}}$ is given such that

$$\alpha_m \circ f_c \sqsubseteq \hat{f}_c \circ \alpha_m \qquad (1)$$

Theorem (Soundness)

$\alpha(\mathit{fix} F) \sqsubseteq \bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\bot)$ *where* $\alpha = \alpha_2 \circ \alpha_1$.

Example: Sign Analysis

Memory state abstraction:

$$\wp(\mathbb{M}) \xleftarrow[\alpha_m]{\gamma_m} \hat{\mathbb{M}}$$

$$\alpha_m(M) \quad = \quad \lambda x \in \textbf{Var}.\ \alpha_s(\{m(x) \mid m \in M\})$$

where α_s is the sign abstraction:

$$\wp(\mathbb{Z}) \xleftarrow[\alpha_s]{\gamma_s} \hat{\mathbb{Z}}$$

The transfer function $\hat{f}_c : \hat{\mathbb{M}} \to \hat{\mathbb{M}}$:

$$\begin{array}{rcll} \hat{f}_c(\hat{m}) & = & \hat{m} & c = skip \\ \hat{f}_c(\hat{m}) & = & \hat{m}[x \mapsto \hat{\mathcal{V}}(e)(\hat{m})] & c = x := e \end{array}$$

$$\begin{array}{rcl} \hat{\mathcal{V}}(n)(\hat{m}) & = & \alpha_s(\{n\}) \\ \hat{\mathcal{V}}(x)(\hat{m}) & = & \hat{m}(x) \\ \hat{\mathcal{V}}(e_1 + e_2) & = & \hat{\mathcal{V}}(e_1)(\hat{m}) \mathbin{\hat{+}} \hat{\mathcal{V}}(e_2)(\hat{m}) \\ \hat{\mathcal{V}}(e_1 - e_2) & = & \hat{\mathcal{V}}(e_1)(\hat{m}) \mathbin{\hat{-}} \hat{\mathcal{V}}(e_2)(\hat{m}) \end{array}$$

Lemma

$\alpha_m \circ f_c \sqsubseteq \hat{f}_c \circ \alpha_m$

Example: Interval Analysis

Memory state abstraction:

$$\alpha_m(M) \quad = \quad \lambda x \in \textbf{Var}.\ \alpha_n(\{m(x) \mid m \in M\})$$

where α_n is the interval abstraction:

$$\wp(\mathbb{Z}) \underset{\alpha_n}{\overset{\gamma_n}{\leftrightarrows}} \hat{\mathbb{Z}}$$

$$\hat{\mathbb{Z}} = \{\bot\} \cup \{[l, u] \mid l, u \in \mathbb{Z} \cup \{-\infty, +\infty\} \wedge l \leq u\}$$

The transfer function $\hat{f}_c : \hat{\mathbb{M}} \to \hat{\mathbb{M}}$:

$$\begin{array}{rcll} \hat{f}_c(\hat{m}) & = & \hat{m} & c = skip \\ \hat{f}_c(\hat{m}) & = & \hat{m}[x \mapsto \hat{\mathcal{V}}(e)(\hat{m})] & c = x := e \end{array}$$

$$\begin{array}{rcl} \hat{\mathcal{V}}(n)(\hat{m}) & = & \alpha_s(\{n\}) \\ \hat{\mathcal{V}}(x)(\hat{m}) & = & \hat{m}(x) \\ \hat{\mathcal{V}}(e_1 + e_2) & = & \hat{\mathcal{V}}(e_1)(\hat{m}) \mathbin{\hat{+}} \hat{\mathcal{V}}(e_2)(\hat{m}) \\ \hat{\mathcal{V}}(e_1 - e_2) & = & \hat{\mathcal{V}}(e_1)(\hat{m}) \mathbin{\hat{-}} \hat{\mathcal{V}}(e_2)(\hat{m}) \end{array}$$

Lemma

$\alpha_m \circ f_c \sqsubseteq \hat{f}_c \circ \alpha_m$

Computing an upper bound of $\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot})$

- If the abstract domain $\hat{D}$ has finite height (i.e., all chains are finite), we can directly calculate

$$\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot}).$$

- If the domain $\hat{D}$ has infinite height, the computation may not terminate. In this case, we find a finite chain $\hat{X}_0 \sqsubseteq \hat{X}_1 \sqsubseteq \hat{X}_2 \sqsubseteq \ldots$ such that

$$\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot}) \sqsubseteq \lim_{i \in \mathbb{N}} \hat{X}_i$$

Finite Chain $\hat{X}_i$

Define finite chain $\hat{X}_i$ by an widening operator $\triangledown : \hat{D} \times \hat{D} \to \hat{D}$:

$$\begin{array}{rcll} \hat{X}_0 & = & \bot & \\ \hat{X}_i & = & \hat{X}_{i-1} & \text{if } \hat{F}(\hat{X}_{i-1}) \sqsubseteq \hat{X}_{i-1} \\ & = & \hat{X}_{i-1} \triangledown \hat{F}(\hat{X}_{i-1}) & \text{otherwise} \end{array} \tag{2}$$

Conditions on $\triangledown$:

- $\forall a, b \in \hat{D}.\ (a \sqsubseteq a \triangledown b)\ \wedge\ (b \sqsubseteq a \triangledown b)$
- For all increasing chains $(x_i)_i$, the increasing chain $(y_i)_i$ defined as

$$y_i = \begin{cases} x_0 & \text{if } i = 0 \\ y_{i-1} \triangledown x_i & \text{if } i > 0 \end{cases}$$

  eventually stabilizes (i.e., the chain is finite).

Then, the limit of the chain is safe analysis result.

Theorem (Widening's Safety)

Let $\hat{D}$ be a CPO, $\hat{F} : \hat{D} \to \hat{D}$ a monotone function, $\triangledown : \hat{D} \times \hat{D} \to \hat{D}$ a widening operator. Then, chain $(\hat{X}_i)_i$ defined as (2) eventually stabilizes and

$$\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot}) \sqsubseteq \lim_{i \in \mathbb{N}} \hat{X}_i.$$

Narrowing

- We can refine the widening result $\mathbf{lim}_{i\in\mathbb{N}}\, \hat{X}_i$ by a narrowing operator $\triangle : \hat{D} \times \hat{D} \to \hat{D}$.
- Compute chain $(\hat{Y}_i)_i$

$$\hat{Y}_i = \begin{cases} \mathbf{lim}_{i\in\mathbb{N}}\, \hat{X}_i & \text{if } i = 0 \\ \hat{Y}_{i-1} \triangle \hat{F}(\hat{Y}_{i-1}) & \text{if } i > 0 \end{cases} \tag{3}$$

- Conditions on $\triangle$
  - $\forall a, b \in \hat{D}.\ a \sqsubseteq b \implies a \sqsubseteq a \triangle b \sqsubseteq b$
  - For all decreasing chain $(x_i)_i$, the decreasing chain $(y_i)_i$ defined as

$$y_i = \begin{cases} x_i & \text{if i} = 0 \\ y_{i-1} \triangle x_i & \text{if } i > 0 \end{cases}$$

    eventually stabilizes.

Theorem (Narrowing's Safety)

Let $\hat{D}$ be a CPO, $\hat{F} : \hat{D} \to \hat{D}$ a monotone function, $\triangle : \hat{D} \times \hat{D} \to \hat{D}$ a narrowing operator. Then, chain $(\hat{Y}_i)_i$ defined as (3) eventually stabilizes and

$$\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot}) \sqsubseteq \lim_{i \in \mathbb{N}} \hat{Y}_i.$$

Widening/Narrowing Example

```
i = 0;
while (i<10)
  i++;
```

- Abstract equation:

$$\begin{array}{rcl} X_1 & = & [0,0] \\ X_2 & = & (X_1 \sqcup X_3] \sqcap [-\infty, 9] \\ X_3 & = & X_2 \hat{+} [1,1] \\ X_4 & = & (X_1 \sqcup X_3) \sqcap [10, +\infty] \end{array}$$

- Abstract domain $\hat{D} =$ Interval $\times$ Interval $\times$ Interval $\times$ Interval
- Semantic function $\hat{F} : \hat{D} \to \hat{D}$ such that

$$(X_1, X_2, X_3, X_4) = \hat{F}(X_1, X_2, X_3, X_4)$$

Widening/Narrowing Example

$$\begin{array}{rcl} X_1 & = & [0,0] \\ X_2 & = & (X_1 \sqcup X_3] \sqcap [-\infty, 9] \\ X_3 & = & X_2 \hat{+} [1,1] \\ X_4 & = & (X_1 \sqcup X_3) \sqcap [10, +\infty] \end{array}$$

$\bigsqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\bot})$:

	0	1	2	3	4	5	6	...	
X_1	$\hat{\bot}$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$		$[0,0]$
X_2	$\hat{\bot}$	$\hat{\bot}$	$[0,0]$	$[0,0]$	$[0,1]$	$[0,1]$	$[0,2]$		$[0,9]$
X_3	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$[1,1]$	$[1,1]$	$[1,2]$	$[1,2]$		$[1,10]$
X_4	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$		$[10,10]$

Widening/Narrowing Example

A simple widening operator for the Interval domain:

$$\begin{array}{rcll} [a,b] & \bigtriangledown & \bot & = [a,b] \\ \bot & \bigtriangledown & [c,d] & = [c,d] \\ [a,b] & \bigtriangledown & [c,d] & = [(c < a ? -\infty : a), (b < d ? +\infty : b)] \end{array}$$

A simple narrowing operator:

$$\begin{array}{rcll} [a,b] & \bigtriangleup & \bot & = \bot \\ \bot & \bigtriangleup & [c,d] & = \bot \\ [a,b] & \bigtriangleup & [c,d] & = [(a = -\infty ? c : a), (b = +\infty ? d : b)] \end{array}$$

Widening/Narrowing Example

Widening iteration:

	0	1	2	3	4	5	6	7
X_1	$\hat{\bot}$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$
X_2	$\hat{\bot}$	$\hat{\bot}$	$[0,0]$	$[0,0]$	$[0,+\infty]$	$[0,+\infty]$	$[0,+\infty]$	$[0,+\infty]$
X_3	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$[1,1]$	$[1,1]$	$[1,+\infty]$	$[1,+\infty]$	$[1,+\infty]$
X_4	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$\hat{\bot}$	$[10,+\infty]$	$[10,+\infty]$

Narrowing iteration:

	0	1	2	3	4
X_1	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$	$[0,0]$
X_2	$[0,+\infty]$	$[0,9]$	$[0,9]$	$[0,9]$	$[0,9]$
X_3	$[1,+\infty]$	$[1,+\infty]$	$[1,10]$	$[1,10]$	$[1,10]$
X_4	$[10,+\infty]$	$[10,+\infty]$	$[10,+\infty]$	$[10,10]$	$[10,10]$

Worklist Algorithm

$$
\begin{array}{l}
W \in \mathit{Worklist} = \wp(\mathbb{C}) \\
T \in \mathbb{C} \to \hat{\mathbb{S}} \\
\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}} \\
\\
W := \mathbb{C} \\
T := \lambda c.\bot \\
\textbf{repeat} \\
\quad c := \text{choose}(W) \\
\quad W := W - \{c\} \\
\quad \hat{s}_{in} := \bigsqcup_{c' \to c} \hat{f}_{c'}(T(c')) \\
\quad \textbf{if } \hat{s}_{in} \not\sqsubseteq \hat{X}(c) \\
\qquad \textbf{if } c \text{ is a head of a flow cycle} \\
\qquad\quad \hat{s}_{in} := T(c) \bigtriangledown \hat{s}_{in} \\
\qquad \hat{X}(c) := \hat{s}_{in} \\
\qquad W := W \cup \{c' \mid c \to c'\} \\
\textbf{until } W = \emptyset
\end{array}
$$

Example