

AAA616: Program Analysis

Lecture 6 — Abstract Interpretation Example (1)

Hakjoo Oh
2018 Spring

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 $\mathbf{cmd}(c)$

$$\begin{aligned} \mathbf{cmd} &\rightarrow \mathit{skip} \mid x := e \\ e &\rightarrow n \mid x \mid e + e \mid e - e. \end{aligned}$$

- Concrete memory states: $\mathbb{M} = \mathit{Var} \rightarrow \mathbb{Z}$
- Concrete semantics:

$$\begin{aligned} \llbracket c \rrbracket &: \mathbb{M} \rightarrow \mathbb{M} \\ \llbracket \mathit{skip} \rrbracket(m) &= m \\ \llbracket x := e \rrbracket(m) &= m[x \mapsto \llbracket e \rrbracket(m)] \\ \llbracket e \rrbracket &: \mathbb{M} \rightarrow \mathbb{Z} \\ \llbracket n \rrbracket(m) &= n \\ \llbracket x \rrbracket(m) &= m(x) \\ \llbracket e_1 + e_2 \rrbracket(m) &= \llbracket e_1 \rrbracket(m) + \llbracket e_2 \rrbracket(m) \\ \llbracket e_1 - e_2 \rrbracket(m) &= \llbracket e_1 \rrbracket(m) - \llbracket e_2 \rrbracket(m) \end{aligned}$$

Concrete Semantics

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

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

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

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

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

$$(c_i, s_i) \rightsquigarrow (c_j, s_j) \iff c_i \rightarrow c_j \wedge s_j = \llbracket \mathbf{cmd}(c_j) \rrbracket (s_i)$$

Concrete Semantics

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

$$\llbracket P \rrbracket = \{\sigma \in \mathbf{State}^+ \mid \sigma_0 \in I \wedge \forall k. \sigma_k \rightsquigarrow \sigma_{k+1}\}$$

The semantic domain:

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

The semantic function:

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

Lemma

$$\llbracket P \rrbracket = \mathit{fix} F.$$

Partitioning Abstraction

Galois-connection: $\wp(\mathbf{State}^+) \xleftarrow[\alpha_1]{\gamma_1} \mathbb{C} \rightarrow \wp(\mathbb{M})$

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

Semantic function:

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

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

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

$$f_c(M) = \{m' \mid m \in M \wedge m' = \llbracket \mathbf{cmd}(c) \rrbracket(m)\}$$

Lemma (Soundness of Partitioning Abstraction)

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

Memory State Abstraction

Galois-connection:

$$\begin{aligned} \mathbb{C} \rightarrow \wp(\mathbb{M}) &\xleftrightarrow[\alpha_2]{\gamma_2} \mathbb{C} \rightarrow \hat{\mathbb{M}} \\ \alpha_2(f) &= \lambda c. \alpha_m(f(c)) \\ \gamma_1(\hat{f}) &= \lambda c. \gamma_m(\hat{f}(c)) \end{aligned}$$

where we assume

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

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

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

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

$$\alpha_m \circ f_c \sqsubseteq \hat{f}_c \circ \alpha_m \tag{1}$$

Theorem (Soundness)

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

Sign Analysis

Memory state abstraction:

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

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

where α_s is the sign abstraction:

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

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

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

$$\hat{V}(n)(\hat{m}) = \alpha_s(\{n\})$$

$$\hat{V}(x)(\hat{m}) = \hat{m}(x)$$

$$\hat{V}(e_1 + e_2) = \hat{V}(e_1)(\hat{m}) \hat{+} \hat{V}(e_2)(\hat{m})$$

$$\hat{V}(e_1 - e_2) = \hat{V}(e_1)(\hat{m}) \hat{-} \hat{V}(e_2)(\hat{m})$$

Lemma

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

Interval Analysis

Memory state abstraction:

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

where α_n is the interval abstraction:

$$\wp(\mathbb{Z}) \xrightleftharpoons[\alpha_n]{\gamma_n} \hat{\mathbb{Z}}$$

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

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

$$\hat{f}_c(\hat{m}) = \hat{m} \quad c = \text{skip}$$

$$\hat{f}_c(\hat{m}) = \hat{m}[x \mapsto \hat{V}(e)(\hat{m})] \quad c = x := e$$

$$\hat{V}(n)(\hat{m}) = \alpha_s(\{n\})$$

$$\hat{V}(x)(\hat{m}) = \hat{m}(x)$$

$$\hat{V}(e_1 + e_2) = \hat{V}(e_1)(\hat{m}) \hat{+} \hat{V}(e_2)(\hat{m})$$

$$\hat{V}(e_1 - e_2) = \hat{V}(e_1)(\hat{m}) \hat{-} \hat{V}(e_2)(\hat{m})$$

Lemma

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

Widening/Narrowing Example

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

- Abstract equation:

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

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

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

Widening/Narrowing Example

$$X_1 = [0, 0]$$

$$X_2 = (X_1 \sqcup X_3) \sqcap [-\infty, 9]$$

$$X_3 = X_2 \hat{+} [1, 1]$$

$$X_4 = (X_1 \sqcup X_3) \sqcap [10, +\infty]$$

$\sqcup_{i \in \mathbb{N}} \hat{F}^i(\hat{\perp})$:

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

Widening/Narrowing Example

A simple widening operator for the Interval domain:

$$\begin{aligned} [a, b] \nabla \perp &= [a, b] \\ \perp \nabla [c, d] &= [c, d] \\ [a, b] \nabla [c, d] &= [(c < a? -\infty : a), (b < d? +\infty : b)] \end{aligned}$$

A simple narrowing operator:

$$\begin{aligned} [a, b] \Delta \perp &= \perp \\ \perp \Delta [c, d] &= \perp \\ [a, b] \Delta [c, d] &= [(a = -\infty?c : a), (b = +\infty?d : b)] \end{aligned}$$

Widening/Narrowing Example

Widening iteration:

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

Mid-term Exam

- April 24 (in class)
- Lecture 1 – 6
- Closed book