AAA616: Program Analysis

Lecture 6 — A Static Analyzer for C-like Languages

Hakjoo Oh
2016 Fall

A C-like Language

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

$$c \rightarrow lv := e \mid lv := alloc_l(a) \mid x < n \mid f_x(e) \mid return_f$$

$$\begin{array}{llll} \text{expression} & e & \rightarrow & n \mid e_1 + e_2 \mid lv \mid \& lv \\ \text{l-value} & lv & \rightarrow & x \mid *e \mid e_1[e_2] \mid e.x \\ \text{allocation} & a & \rightarrow & [e] \mid \{x\} \end{array}$$

Abstract Semantics

- Abstract Domain

$$\begin{array}{rcl}\mathbb{D} & = & \mathbb{C} \to \mathbb{S} \\ \mathbb{S} & = & \mathbb{L} \to \mathbb{V} \\ \mathbb{L} & = & \textbf{Var} + \textbf{AllocSite} + \textbf{AllocSite} \times \textbf{FieldName} \\ \mathbb{V} & = & \mathbb{I} \times \wp(\mathbb{L}) \times \wp(\textbf{AllocSite} \times \mathbb{I} \times \mathbb{I}) \times \wp(\textbf{AllocSite} \times \wp(\textbf{FieldName}))\end{array}$$

- Abstract Semantic Function:

$$F(X) = \lambda c. \bigsqcup_{c' \to c} f_{c'}(X(c'))$$

$$f_c(s) =$$

$$\begin{cases} s[\hat{\mathcal{L}}(lv)(s) \overset{w}{\mapsto} \hat{\mathcal{V}}(e)(s)] & c = lv := e \\ s[\hat{\mathcal{L}}(lv)(s) \overset{w}{\mapsto} \langle \bot, \bot, \{\langle l, [0, 0], \hat{\mathcal{V}}(e)(s).1 \rangle\}, \bot \rangle] & c = lv := alloc_l([e]) \\ s[\hat{\mathcal{L}}(lv)(s) \overset{w}{\mapsto} \langle \bot, \bot, \bot, \{\langle l, \{x\} \rangle\} \rangle] & c = lv := alloc_l(\{x\}) \\ s[x \mapsto \langle s(x).1 \sqcap [-\infty, n-1], s(x).2, s(x).3, s(x).4 \rangle] & c = x < n \\ s[x \mapsto \hat{\mathcal{V}}(e)(s)] & c = f_x(e) \\ s & c = return_f \end{cases}$$

Abstract Semantics

$$
\begin{array}{rcl}
\hat{\mathcal{V}}(e) & \in & \mathbb{S} \to \mathbb{V} \\
\hat{\mathcal{V}}(n)(s) & = & \langle \alpha_{\hat{\mathbb{Z}}}(n), \bot, \bot, \bot \rangle \\
\hat{\mathcal{V}}(e_1 + e_2)(s) & = & \hat{\mathcal{V}}(e_1)(s) \hat{+} \hat{\mathcal{V}}(e_2)(s) \\
\hat{\mathcal{V}}(lv)(s) & = & \bigsqcup \{ s(l) \mid l \in \hat{\mathcal{L}}(lv)(s) \} \\
\hat{\mathcal{V}}(\& lv)(s) & = & \langle \bot, \hat{\mathcal{L}}(lv)(s), \bot, \bot \rangle
\end{array}
$$

$$
\begin{array}{rcl}
\hat{\mathcal{L}}(lv) & \in & \mathbb{S} \to \wp(\mathbb{L}) \\
\hat{\mathcal{L}}(x)(s) & = & \{x\} \\
\hat{\mathcal{L}}(*e)(s) & = & \hat{\mathcal{V}}(e)(s).2 \cup \{ l \mid \langle l, o, s \rangle \in \hat{\mathcal{V}}(e)(s).3 \} \\
& & \cup \{ \langle l, x \rangle \mid \langle l, X \rangle \in \hat{\mathcal{V}}(e)(s).4 \wedge x \in X \} \\
\hat{\mathcal{L}}(e_1[e_2])(s) & = & \{ l \mid \langle l, o, s \rangle \in \hat{\mathcal{V}}(e_1)(s).3 \} \\
\hat{\mathcal{L}}(e.x)(s) & = & \{ \langle l, x \rangle \mid \langle l, X \rangle \in \hat{\mathcal{V}}(e)(s).4 \wedge x \in X \}
\end{array}
$$

Fixed Point Algorithm

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