AAA616: Program Analysis

Lecture 7 — Abstract Interpretation Example (2)

Hakjoo Oh
2022 Fall

Programs

Language:

$$\begin{array}{rcl} lv & \rightarrow & x \mid *x \\ e & \rightarrow & n \mid lv \mid \&lv \mid e_1 + e_2 \mid e_1 \star e_2 \mid e_1 - e_2 \\ b & \rightarrow & \texttt{true} \mid \texttt{false} \mid e_1 = e_2 \mid e_1 \leq e_2 \mid \neg b \mid b_1 \wedge b_2 \\ c & \rightarrow & lv := e \mid lv := \texttt{alloc} \mid \texttt{skip} \mid c_1; c_2 \mid \texttt{if}\ b\ c_1\ c_2 \mid \texttt{while}\ b\ c \end{array}$$

Assume programs are represented by control flow graphs. Let $(N, \rightarrow)$ be a control-flow graph and $\mathbf{cmd}(n)$ be the command associated with node n:

$$lv := e \mid lv := \texttt{alloc} \mid assume(b)$$

Examples

- ```
  x = 1;
  p = &x;
  *p = *p + 1
  ```
- ```
  p = alloc;
  q = &p;
  **q = 1
  ```
- ```
  x = 1;
  while (x < 10) {
      p = alloc;
      *p = *p + x;
      x = x + 1
  }
  ```
- ```
  if (...) { p = alloc; *p = 1; }
  else     { p = alloc; *p = 2; }
  *p = 3
  ```

Concrete Semantics

$$\begin{array}{rcl} Mem & = & Loc \to Val \\ Loc & = & Var + HeapAddr \\ Val & = & Int + Loc \end{array}$$

- $[\![lv]\!] : Mem \to Loc$:

$$\begin{array}{rcl} [\![x]\!](m) & = & x \\ [\![*x]\!](m) & = & m(x) \end{array}$$

- $[\![e]\!] : Mem \to Val$:

$$\begin{array}{rcl} [\![n]\!](m) & = & n \\ [\![lv]\!](m) & = & m([\![lv]\!](m)) \\ [\![\&lv]\!](m) & = & [\![lv]\!](m) \\ [\![e_1 + e_2]\!](m) & = & [\![e_1]\!](m) +_{Int} [\![e_2]\!](m) \end{array}$$

- $[\![b]\!] : \mathit{Mem} \to \mathit{Bool}$:

$$\begin{array}{rcl} [\![\texttt{true}]\!](m) & = & \mathit{true} \\ [\![\texttt{false}]\!](m) & = & \mathit{false} \\ [\![e_1 = e_2]\!](m) & = & [\![e_1]\!](m) =_{\mathit{Int}} [\![e_2]\!](m) \\ [\![e_1 \leq e_2]\!](m) & = & [\![e_1]\!](m) \leq_{\mathit{Int}} [\![e_2]\!](m) \\ [\![\neg b]\!](m) & = & \neg [\![b]\!](m) \\ [\![b_1 \wedge b_2]\!](m) & = & [\![b_1]\!](m) \wedge [\![b_2]\!](m) \end{array}$$

- $f_n : \wp(\mathit{Mem}) \to \wp(\mathit{Mem})$:

$$\begin{array}{lll} f_n(M) = \{m[[\![lv]\!](m) \mapsto [\![e]\!](m)] \mid m \in M\} & \cdots & \textbf{cmd}(n) = lv := e \\ f_n(M) = \{m[[\![lv]\!](m) \mapsto l, l \mapsto 0] \mid m \in M\} & \cdots & \textbf{cmd}(n) = lv := \texttt{alloc}, \\ & & l \text{ is new} \\ f_n(M) = \{m \in M \mid [\![b]\!](m) = \mathit{true}\} & \cdots & \textbf{cmd}(n) = \mathit{assume}(b) \end{array}$$

- $F : (N \to \wp(\mathit{Mem})) \to (N \to \wp(\mathit{Mem}))$:

$$F(X) = \lambda n.\ f_n(\bigcup_{n' \to n} X(n'))$$

- Collecting semantics:

$$\mathit{fix} F \in N \to \wp(\mathit{Mem})$$

Abstract Domain

$$\begin{array}{rcl} Mem & = & Loc \to Val \\ Loc & = & Var + HeapAddr \\ Val & = & Int + Loc \end{array} \qquad \begin{array}{rcl} \widehat{Mem} & = & \widehat{Loc} \to \widehat{Val} \\ \widehat{Loc} & = & Var + AllocSite \\ \widehat{Val} & = & Interval \times \wp(\widehat{Loc}) \end{array}$$

- $\wp(HeapAddr) \xleftarrow[\alpha_{HeapAddr}]{\gamma_{HeapAddr}} \wp(AllocSite)$

$$\alpha_{HeapAddr}(H) = \{\textsf{allocsite}(h) \mid h \in H\}$$

- $\wp(Loc) \xleftarrow[\alpha_{Loc}]{\gamma_{Loc}} \wp(\widehat{Loc})$

$$\alpha_{Loc}(L) = \{x \mid x \in L\} \uplus \alpha_{HeapAddr}(\{h \mid h \in L\})$$

- $\wp(Int) \xleftarrow[\alpha_{Int}]{\gamma_{Int}} Interval$
- $\wp(Val) \xleftarrow[\alpha_{Val}]{\gamma_{Val}} \widehat{Val}$

$$\alpha_{Val}(V) = \langle \alpha_{Int}(\{z \mid z \in V\}), \alpha_{Loc}(\{l \mid l \in V\})\rangle$$

- $\wp(Mem) \xleftarrow[\alpha_{Mem}]{\gamma_{Mem}} \widehat{Mem}$

$$\alpha_{Mem}(M) = \lambda l. \begin{cases} \bigsqcup\{m(l) \mid m \in M\} & \cdots\ l \in Var \\ \bigsqcup\{m(a) \mid m \in M, a \in \gamma_{HeapAddr}(l)\} & \cdots\ l \in AllocSite \end{cases}$$

- $N \to \wp(Mem) \xleftarrow[\alpha]{\gamma} N \to \widehat{Mem}$: $\alpha(X) = \lambda n.\alpha_{Mem}(X(n))$

Abstract Semantics

- $[\![lv]\!] : \widehat{Mem} \to \wp(Loc)$

$$\begin{array}{rcl} [\![x]\!](m) & = & \{x\} \\ [\![*x]\!](m) & = & m(x).2 \end{array}$$

- $[\![e]\!] : \widehat{Mem} \to Val$

$$\begin{array}{rcl} [\![n]\!](m) & = & \langle [n,n], \emptyset \rangle \\ [\![lv]\!](m) & = & \bigsqcup_{l \in [\![lv]\!](m)} m(l) \\ [\![\&lv]\!](m) & = & \langle \bot, [\![lv]\!](m) \rangle \\ [\![e_1 + e_2]\!](m) & = & [\![e_1]\!](m) \mathbin{\hat{+}} [\![e_2]\!](m) \end{array}$$

- $[\![b]\!] : \widehat{Mem} \to Bool$

$$\begin{array}{rcl} [\![\mathtt{true}]\!](m) & = & true \\ [\![\mathtt{false}]\!](m) & = & false \\ [\![e_1 = e_2]\!](m) & = & [\![e_1]\!](m) \mathbin{\hat{=}} [\![e_2]\!](m) \\ [\![e_1 \le e_2]\!](m) & = & [\![e_1]\!](m) \mathbin{\hat{\le}} [\![e_2]\!](m) \\ [\![\neg b]\!](m) & = & \neg [\![b]\!](m) \\ [\![b_1 \land b_2]\!](m) & = & [\![b_1]\!](m) \land [\![b_2]\!](m) \end{array}$$

- $\hat{f}_n : \widehat{Mem} \to \widehat{Mem}$:

$$
\begin{array}{ll}
\hat{f}_n(m) = m[x \mapsto [\![e]\!](m)] & \cdots\ \mathbf{cmd}(n) = x := e \\
\hat{f}_n(m) = m[x \mapsto [\![e]\!](m)] & \cdots\ \mathbf{cmd}(n) = lv := e, \\
 & \quad [\![lv]\!](m) = \{x\} \\
\hat{f}_n(m) = \bigsqcup_{l \in [\![lv]\!](m)} m[l \mapsto m(l) \sqcup [\![e]\!](m)] & \cdots\ \mathbf{cmd}(n) = lv := e \\
\hat{f}_n(m) = m[x \mapsto (\bot, \{n\}), n \mapsto ([0,0], \emptyset)] & \cdots\ \mathbf{cmd}(n) = x := \mathtt{alloc} \\
\hat{f}_n(m) = m[x \mapsto (\bot, \{n\}), n \mapsto ([0,0], \emptyset)] & \cdots\ \mathbf{cmd}(n) = lv := \mathtt{alloc}, \\
 & \quad [\![lv]\!](m) = \{x\} \\
\hat{f}_n(m) = \bigsqcup_{l \in [\![lv]\!](m)} m'[l \mapsto m(l) \sqcup (\bot, \{n\})] & \cdots\ \mathbf{cmd}(n) = lv := \mathtt{alloc} \\
 & \quad m' = m[n \mapsto ([0,0], \emptyset)] \\
\hat{f}_n(m) = \bigsqcup \{m' \sqsubseteq m \mid true \sqsubseteq [\![b]\!](m')\} & \cdots\ \mathbf{cmd}(n) = assume(b)
\end{array}
$$

- $\hat{F} : (N \to \widehat{Mem}) \to (N \to \widehat{Mem})$:

$$\hat{F}(X) = \lambda n.\ \hat{f}_n\Big(\bigsqcup_{n' \to n} X(n')\Big)$$

- Abstract semantics:

$$\bigsqcup_{i \geq 0} \hat{F}^i(\bot) \in N \to \widehat{Mem}$$