1 Syntax

$$
\begin{array}{ll}
\textit{program} & ::= \overline{cls}\ \overline{s} \\
\textit{cls} & ::= \texttt{class}\ C\ \{\overline{field}\ \overline{method}\} \\
\textit{field} & ::= T\ f; \\
\textit{method} & ::= T\ m(T\ x)\ contract\ \{\overline{s}\} \\
\textit{contract} & ::= \texttt{requires}\ \phi;\ \texttt{ensures}\ \phi; \\
T & ::= \texttt{int} \mid C \\
s & ::= x.f := y; \mid x := e; \mid x := \texttt{new}\ C; \mid x := y.m(z); \\
 & \mid \texttt{return}\ x; \mid \texttt{assert}\ \phi; \mid \texttt{release}\ \phi; \mid T\ x; \\
\phi & ::= \texttt{true} \mid e = e \mid e \neq e \mid \texttt{acc}(x.f) \mid x : T \mid \phi * \phi \\
e & ::= v \mid x \mid e.f \\
x & ::= \texttt{this} \mid \texttt{result} \mid \langle other \rangle
\end{array}
$$

$$
\begin{array}{ll}
H & ::= (o \rightharpoonup (C, \overline{(f \rightharpoonup v)})) \\
\rho & ::= (x \rightharpoonup v) \\
A_s & ::= \overline{(x, f)} \\
A_d & ::= \overline{(o, f)} \\
S & ::= (\rho, A_d, \overline{s}) \cdot S \mid nil
\end{array}
$$

2 Static semantics

2.1 Expressions ($A_s \vdash_{\texttt{sfrm}} e$)

$$\frac{}{A \vdash_{\texttt{sfrm}} x}\ \textsc{WFVar}$$

$$\frac{}{A \vdash_{\texttt{sfrm}} v}\ \textsc{WFValue}$$

$$\frac{(x, f) \in A}{A \vdash_{\texttt{sfrm}} x.f}\ \textsc{WFField}$$

2.2 Formulas ($A_s \vdash_{\texttt{sfrm}} \phi$)

$$\frac{}{A \vdash_{\texttt{sfrm}} \texttt{true}}\ \textsc{WFTrue}$$

$$\frac{A \vdash_{\texttt{sfrm}} e_1 \qquad A \vdash_{\texttt{sfrm}} e_2}{A \vdash_{\texttt{sfrm}} e_1 = e_2}\ \textsc{WFEqual}$$

$$\frac{A \vdash_{\texttt{sfrm}} e_1 \qquad A \vdash_{\texttt{sfrm}} e_2}{A \vdash_{\texttt{sfrm}} e_1 \neq e_2} \text{ WFNEqual}$$

$$\frac{}{A \vdash_{\texttt{sfrm}} \texttt{acc}(x, f)} \text{ WFAcc}$$

$$\frac{}{A \vdash_{\texttt{sfrm}} x : T} \text{ WFType}$$

$$\frac{A_s \vdash_{\texttt{sfrm}} \phi_1 \qquad A_s \cup \texttt{static-footprint}(\phi_1) \vdash_{\texttt{sfrm}} \phi_2}{A_s \vdash_{\texttt{sfrm}} \phi_1 * \phi_2} \text{ WFSepOp}$$

2.2.1 Implication ($\phi_1 \overset{\cdot}{\Longrightarrow} \phi_2$)

Conservative approx. of $\phi_1 \implies \phi_2$.

2.3 Footprint ($\texttt{static-footprint}(\phi) = A_s$)

$$\begin{aligned}
&\texttt{static-footprint}(\texttt{true}) &&= \emptyset \\
&\texttt{static-footprint}(e_1 = e_2) &&= \emptyset \\
&\texttt{static-footprint}(e_1 \neq e_2) &&= \emptyset \\
&\texttt{static-footprint}(\texttt{acc}(x.f)) &&= \{(x, f)\} \\
&\texttt{static-footprint}(\phi_1 * \phi_2) &&= \texttt{static-footprint}(\phi_1) \cup \texttt{static-footprint}(\phi_2)
\end{aligned}$$

2.4 Hoare ($\vdash \{\phi\}\overline{s}\{\phi\}$)

$$\frac{\vdash \{\phi_p\}s_1\{\phi_{q1}\} \qquad \phi_{q1} \implies \phi_{q2} \qquad \vdash \{\phi_{q2}\}s_2\{\phi_r\}}{\vdash \{\phi_p\}s_1; s_2\{\phi_r\}} \text{ HSec}$$

$$\frac{\phi(x) = C \qquad \texttt{fields}(C) = fs}{\vdash \{\phi\}x := \texttt{ new } C\{(\texttt{acc}(x, \overline{fs_i}) * (x \neq \texttt{null} * \phi))\}} \text{ HNewObj}$$

$$\frac{\phi(x) = C \qquad \texttt{fieldType}(C, f) = T \qquad \phi(y) = T \qquad \texttt{acc}(x, f) \in \phi \qquad x \neq \texttt{null} \in \phi}{\vdash \{\phi\}x.f := y\{\phi * x.f = y\}} \text{ HFieldAssign}$$

$$\frac{\begin{array}{c}\phi_1(x) = T \qquad \texttt{staticType}_{\phi_1}(e) = T \\ \phi_1 = \phi_2[e/x] \qquad \emptyset \vdash_{\texttt{sfrm}} \phi_1 \qquad \texttt{static-footprint}(\phi_1) \vdash_{\texttt{sfrm}} e\end{array}}{\vdash \{\phi_1\}x := e\{\phi_2\}} \text{ HVarAssign}$$

$$\frac{\phi(x) = T \qquad \phi(\texttt{result}) = T}{\vdash \{\phi\}\texttt{return}\ x\{\phi * \texttt{result} = x\}}\ \textsc{HReturn}$$

$$\frac{\begin{array}{c}\phi(y) = C \qquad \phi(x) = T_r \qquad \phi(z') = T_p \qquad y \neq \texttt{null} \in \phi \qquad \phi \implies (\phi_p * \phi_r) \\ \texttt{mpre}(C, m) = \phi_{pre} \qquad \texttt{mpost}(C, m) = \phi_{post} \qquad \texttt{mparam}(C, m) = (T_p, z) \\ mrettypeCm = T_r \qquad \phi_p = \phi_{pre}[y, z'/\texttt{this}, z] \qquad \phi_q = \phi_{post}[y, z', x/\texttt{this}, z, \texttt{result}]\end{array}}{\vdash \{\phi\}x := y.m(z')\{(\phi_q * \phi_r)\}}\ \textsc{HApp}$$

$$\frac{\phi_2 \in \phi_1}{\vdash \{\phi_1\}\texttt{assert}\ \phi_2\{\phi_1\}}\ \textsc{HAssert}$$

$$\frac{\phi_1 \implies (\phi_2 * \phi_r) \qquad \emptyset \vdash_{\texttt{sfrm}} \phi_r}{\vdash \{\phi_1\}\texttt{release}\ \phi_2\{\phi_r\}}\ \textsc{HRelease}$$

$$\frac{\phi_1(x) = \bot \qquad \phi_2 = \phi_1 * x : T}{\vdash \{\phi_1\}T\ x\{\phi_2\}}\ \textsc{HDeclare}$$

3 Dynamic semantics

3.1 Expressions ($H, \rho \vdash e \Downarrow v$)

$$\frac{}{H, \rho \vdash x \Downarrow \rho(x)}\ \textsc{EEVar}$$

$$\frac{}{H, \rho \vdash v \Downarrow v}\ \textsc{EEValue}$$

$$\frac{H, \rho \vdash x \Downarrow o}{H, \rho \vdash x.f \Downarrow H(o)(f)}\ \textsc{EEAcc}$$

3.2 Formulas ($H, \rho, A \vDash \phi$)

$$\frac{}{H, \rho, A \vDash \texttt{true}}\ \textsc{EATrue}$$

$$\frac{H, \rho \vdash e_1 \Downarrow v_1 \qquad H, \rho \vdash e_2 \Downarrow v_2 \qquad v_1 = v_2}{H, \rho, A \vDash e_1 = e_2}\ \textsc{EAEqual}$$

$$\frac{H, \rho \vdash e_1 \Downarrow v_1 \qquad H, \rho \vdash e_2 \Downarrow v_2 \qquad v_1 \neq v_2}{H, \rho, A \vDash e_1 \neq e_2} \text{ EANEqual}$$

$$\frac{\rho(x) = o \qquad (o, f) \in A}{H, \rho, A \vDash \mathtt{acc}(x, f)} \text{ EAAcc}$$

$$\frac{\rho(x) = T}{H, \rho, A \vDash x : T} \text{ EAType}$$

$$\frac{A_1 = A \backslash A_2 \qquad H, \rho, A_1 \vDash \phi_1 \qquad H, \rho, A_2 \vDash \phi_2}{H, \rho, A \vDash \phi_1 * \phi_2} \text{ EASepOp}$$

3.2.1 Implication ($\phi_1 \implies \phi_2$)

$$\phi_1 \implies \phi_2 \qquad \iff \qquad \forall H, \rho, A : H, \rho, A \vDash \phi_1 \implies H, \rho, A \vDash \phi_2$$

Drawn from def. of entailment in “A Formal Semantics for Isorecursive and Equirecursive State Abstractions”.

3.3 Footprint ($\mathtt{footprint}_{H,\rho}(\phi) = A_d$)

$$\begin{aligned}
&\mathtt{footprint}_{H,\rho}(\mathtt{true}) && = \emptyset \\
&\mathtt{footprint}_{H,\rho}(e_1 = e_2) && = \emptyset \\
&\mathtt{footprint}_{H,\rho}(e_1 \neq e_2) && = \emptyset \\
&\mathtt{footprint}_{H,\rho}(\mathtt{acc}(e.f)) && = \{(o, f)\} \text{ where } H, \rho \vdash e \Downarrow o \\
&\mathtt{footprint}_{H,\rho}(\phi_1 * \phi_2) && = \mathtt{footprint}_{H,\rho}(\phi_1) \cup \mathtt{footprint}_{H,\rho}(\phi_2)
\end{aligned}$$

3.4 Small-step ($(H, S) \rightarrow (H, S)$)

$$\frac{H, \rho \vdash x \Downarrow o \qquad H, \rho \vdash y \Downarrow v_y \qquad (o, f) \in A \qquad H' = H[o \mapsto [f \mapsto v_y]]}{(H, (\rho, A, x.f := y; \overline{s}) \cdot S) \rightarrow (H', (\rho, A, \overline{s}) \cdot S)} \text{ ESFieldAssign}$$

$$\frac{H, \rho \vdash e \Downarrow v \qquad \rho' = \rho[x \mapsto v]}{(H, (\rho, A, x := e; \overline{s}) \cdot S) \rightarrow (H, (\rho', A, \overline{s}) \cdot S)} \text{ ESVarAssign}$$

$$\frac{\begin{array}{c} H(o) = \bot \\ \mathtt{fields}(C) = \overline{f} \qquad \rho' = \rho[x \mapsto o] \qquad A' = A * \overline{(o, f_i)} \qquad H' = H[o \mapsto \mathtt{new}\ C] \end{array}}{(H, (\rho, A, x := \mathtt{new}\ C; \overline{s}) \cdot S) \rightarrow (H', (\rho', A', \overline{s}) \cdot S)} \text{ ESNewObj}$$

$$\frac{H, \rho \vdash x \Downarrow v_x \qquad \rho' = \rho[\texttt{result} \mapsto v_x]}{(H, (\rho, A, \texttt{return}\ x; \overline{s}) \cdot S) \rightarrow (H, (\rho', A, \overline{s}) \cdot S)} \text{ ESReturn}$$

$$\frac{\begin{array}{c} H, \rho \vdash y \Downarrow o \\ H, \rho \vdash z \Downarrow v \qquad H(o) = (C, c) \qquad \texttt{mbody}(C, m) = \overline{r} \qquad \texttt{mparam}(C, m) = (T, w) \\ \texttt{mpre}(C, m) = \phi \qquad \rho' = [\texttt{this} \mapsto o, w \mapsto v] \qquad H, \rho', A \vDash \phi \qquad A' = \texttt{footprint}_{H,\rho'}(\phi) \end{array}}{(H, (\rho, A, x := y.m(z); \overline{s}) \cdot S) \rightarrow (H, (\rho', A', \overline{r}) * (\rho, A \setminus A', x := y.m(z); \overline{s}) \cdot S)} \text{ ESApp}$$

$$\frac{\texttt{mpost}(C, m) = \phi \qquad H, \rho', A' \vDash \phi \qquad A'' = \texttt{footprint}_{H,\rho'}(\phi) \qquad H, \rho' \vdash \texttt{result} \Downarrow v_r}{(H, (\rho', A', \emptyset) * (\rho, A, x := y.m(z); \overline{s}) \cdot S) \rightarrow (H, (\rho[x \mapsto v_r], A * A'', \overline{s}) \cdot S)} \text{ ESAppFinish}$$

$$\frac{H, \rho, A \vDash \phi}{(H, (\rho, A, \texttt{assert}\ \phi; \overline{s}) \cdot S) \rightarrow (H, (\rho, A, \overline{s}) \cdot S)} \text{ ESAssert}$$

$$\frac{H, \rho, A \vDash \phi \qquad A' = A \setminus \texttt{footprint}_{H,\rho}(\phi)}{(H, (\rho, A, \texttt{release}\ \phi; \overline{s}) \cdot S) \rightarrow (H, (\rho, A', \overline{s}) \cdot S)} \text{ ESRelease}$$

$$\frac{\rho' = \rho[x \mapsto \texttt{defaultValue}(T)]}{(H, (\rho, A, T\ x; \overline{s}) \cdot S) \rightarrow (H, (\rho', A, \overline{s}) \cdot S)} \text{ ESDeclare}$$

4 Theorems

Hoare preserves self-framing?

$$\begin{aligned} \forall\ \Gamma, \phi_1, \phi_2, s :& \vdash \{\phi_1\} s \{\phi_2\} \\ & \implies \emptyset \vdash_{\texttt{sfrm}} \phi_1 \\ & \implies \emptyset \vdash_{\texttt{sfrm}} \phi_2 \end{aligned}$$

Hoare progress

$$\begin{aligned} \forall\ \ldots :& \vdash \{\phi_1\} s' \{\phi_2\} \\ & \implies H_1, \rho_1, A_1 \vDash \phi_1 \\ & \implies \exists H_2, \rho_2, A_2 : (H_1, (\rho_1, A_1, s'; \overline{s}) \cdot S) \rightarrow^* (H_2, (\rho_2, A_2, \overline{s}) \cdot S) \end{aligned}$$

Hoare preservation

$$\begin{aligned} \forall\ \ldots :& \vdash \{\phi_1\} s' \{\phi_2\} \\ & \implies H_1, \rho_1, A_1 \vDash \phi_1 \\ & \implies (H_1, (\rho_1, A_1, s'; \overline{s}) \cdot S) \rightarrow^* (H_2, (\rho_2, A_2, \overline{s}) \cdot S) \\ & \implies H_2, \rho_2, A_2 \vDash \phi_2 \end{aligned}$$