1 Syntax

program	$::= \overline{cls}\ \overline{s}$
cls	$::= \texttt{class}\ C\ \{\overline{field}\ \overline{method}\}$
field	$::= T\ f;$
method	$::= T\ m(T\ x)\ contract\ \{\overline{s}\}$
contract	$::= \texttt{requires}\ \phi;\ \texttt{ensures}\ \phi;$
T	$::= \texttt{int} \mid C$
s	$::= x.f := y; \mid x := e; \mid x := \texttt{new}\ C; \mid x := y.m(z);$
	$\mid \texttt{return}\ x; \mid \texttt{assert}\ \phi; \mid \texttt{release}\ \phi; \mid T\ x;$
ϕ	$::= \texttt{true} \mid e = e \mid e \neq e \mid \texttt{acc}(x.f) \mid x : T \mid \phi * \phi$
e	$::= v \mid x \mid e.f$
x	$::= \texttt{this} \mid \texttt{result} \mid \langle other \rangle$
v	$::= o \mid n \mid \texttt{null}$
n	$\in \mathbb{Z}$
H	$\in (o \rightharpoonup (C, \overline{(f \rightharpoonup v)}))$
ρ	$\in (x \rightharpoonup v)$
A_s	$::= \overline{(x, f)}$
A_d	$::= \overline{(o, f)}$
S	$::= (\rho, A_d, \overline{s}) \cdot S \mid nil$

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

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

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

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

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

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

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

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 Type ($\texttt{staticType}_\phi(e) = T$)

$$\begin{aligned}
&\texttt{staticType}_\phi(v_T) &&= T \\
&\texttt{staticType}_\phi(x) &&= T \quad \text{where } \phi \implies (x : T) \\
&\texttt{staticType}_\phi(e.f) &&= \texttt{fieldType}(C, f) \quad \text{where } \texttt{staticType}_\phi(e) = C
\end{aligned}$$

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

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

$$\frac{\texttt{staticType}_\phi(x) = C \qquad \texttt{fields}(C) = \overline{f}}{\vdash \{\phi\}x := \texttt{ new } C\{(\texttt{acc}(x, \overline{f_i}) * (x \neq \texttt{null} * \phi))\}}\ \textsc{HNewObj}$$

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

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

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

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

$$\frac{\phi_2 \implies \phi_1}{\vdash \{\phi_1\} \texttt{assert } \phi_2 \{\phi_1\}} \text{ HAssert}$$

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

$$\frac{\texttt{staticType}_{\phi_1}(x) \textit{ undefined} \qquad \phi_2 = \phi_1 * x : T}{\vdash \{\phi_1\} T\ x \{\phi_2\}} \text{ HDeclare}$$

3 Dynamic semantics

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

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

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

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

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

$$\frac{}{H, \rho, A \vDash \texttt{true}} \text{ EAT}\textsc{rue}$$

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

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

$$\frac{\rho(x) = o \qquad (o, f) \in A}{H, \rho, A \vDash \texttt{acc}(x, f)} \text{ EAA}\textsc{cc}$$

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

$$\frac{A_1 = A \backslash A_2 \qquad H, \rho, A_1 \vDash \phi_1 \qquad H, \rho, A_2 \vDash \phi_2}{H, \rho, A \vDash \phi_1 * \phi_2} \text{ EAS}\textsc{ep}\text{O}\textsc{p}$$

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 ($\texttt{footprint}_{H,\rho}(\phi) = A_d$)

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

3.4 Type ($\texttt{dynamicType}_{H,\rho}(e) = T$)

$$\texttt{dynamicType}_{H,\rho}(e) = T \quad \text{where } H, \rho \vdash e \Downarrow v_T$$

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

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

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

$$\frac{\begin{array}{c} H(o) \textit{ undefined} \qquad \texttt{fields}(C) = \overline{T}\ \overline{f} \\ \rho' = \rho[x \mapsto o] \qquad A' = A * \overline{(o, f_i)} \qquad H' = H[o \mapsto [\overline{f \mapsto \texttt{defaultValue}(T)}]] \end{array}}{(H, (\rho, A, x := \texttt{ new } C; \overline{s}) \cdot S) \rightarrow (H', (\rho', A', \overline{s}) \cdot S)} \text{ ESNewObj}$$

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

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

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

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

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

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

4 Theorems

4.1 Invariant $invariant(H, \rho, A_d, \phi)$

4.1.1 Heap consistent

$$
\begin{aligned}
\forall x, o, C : \rho(x) = o_C \implies \\
&\exists f_C, m : \texttt{fields}(C) = f_C \\
&\quad \wedge\ H(o_C) = (C, m) \\
&\quad \wedge\ (\forall (T, f) \in f_C : \texttt{dynamicType}_{H,\rho}(res(f)) = T)
\end{aligned}
$$

4.1.2 Phi holds

$$H, \rho, A_d \vDash \phi$$

4.1.3 Types preserved

$$
\begin{aligned}
&\forall e, T : \texttt{staticType}_{\phi}(e) = T \\
&\quad \implies \texttt{dynamicType}_{H,\rho}(e) = T
\end{aligned}
$$

4.2 Soundness

4.2.1 Progress

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

4.2.2 Preservation

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