1 Syntax

1.1 Syntax

$$
\begin{array}{llll}
\text{Interfaces} & \text{IL} & ::= & \texttt{interface I extends } \overline{\text{I}}\ \{\overline{\text{M}}\} \\
\text{Methods} & \text{M} & ::= & \text{I } \text{m}(\overline{\text{I}}\ \text{x}) \texttt{ override J } \{\texttt{return } e;\} \\
\text{Expressions} & e & ::= & \text{x} \mid e.\text{m}(\overline{e}) \mid \texttt{new I}() \mid e.\text{I} :: \text{m}(\overline{e}) \mid \texttt{super.I} :: \text{m}(e) \\
\text{Context} & \Gamma & ::= & \text{x}_1 : \text{I}_1 ... \text{x}_n : \text{I}_n
\end{array}
$$

1.2 Subtyping

$$\text{I} <: \text{I}$$

$$\frac{\text{I} <: \text{J} \qquad \text{J} <: \text{K}}{\text{I} <: \text{K}}$$

$$\frac{\texttt{interface I extends } \overline{\text{I}}\ \{\overline{\text{M}}\}}{\forall \text{I}_i \in \overline{\text{I}}, \text{I} <: \text{I}_i}$$

1.3 Typing Rules

$$(\text{T-Var})\ \Gamma \vdash \text{x} : \Gamma(\text{x})$$

$$(\text{T-Invk})\ \frac{\Gamma \vdash e_0 : \text{I}_0 \qquad \texttt{mtype}(\text{m}, \text{I}_0) = \overline{\text{J}} \to \text{I} \qquad \Gamma \vdash \overline{e} : \overline{\text{I}} \qquad \overline{\text{I}} <: \overline{\text{J}}}{\Gamma \vdash e_0.\text{m}(\overline{e}) : \text{I}}$$

$$(\text{T-PathInvk})\ \frac{\Gamma \vdash e_0 : \text{I}_0 \qquad \text{I}_0 <: \text{J}_0 \qquad \texttt{mtype}(\text{m}, \text{J}_0) = \overline{\text{J}} \to \text{I} \qquad \Gamma \vdash \overline{e} : \overline{\text{I}} \qquad \overline{\text{I}} <: \overline{\text{J}}}{\Gamma \vdash e_0.\text{J}_0 :: \text{m}(\overline{e}) : \text{I}}$$

$$(\text{T-SuperInvk})\ \frac{\Gamma \vdash \texttt{this} : \text{I}_0 \qquad \texttt{ext}(\text{I}_0, \text{J}_0) \qquad \texttt{mtype}(\text{m}, \text{J}_0) = \overline{\text{J}} \to \text{I} \qquad \Gamma \vdash \overline{e} : \overline{\text{I}} \qquad \overline{\text{I}} <: \overline{\text{J}}}{\Gamma \vdash {}_{\text{I}_0}\texttt{super.J}_0 :: \text{m}(\overline{e}) : \text{I}}$$

$$(\text{T-New})\ \Gamma \vdash \texttt{new I}() : \text{I}$$

$$(\text{T-Method})\ \frac{\text{I} <: \text{J} \qquad \texttt{mtype}(\text{m}, \text{J}) = \overline{\text{I}} \to \text{I}_0}{\text{I}_0\ \text{m}(\overline{\text{I}}\ \text{x}) \texttt{ override J } \{\texttt{return } e_0;\} \text{ OK IN I}}$$

$$(\text{T-Intf})\ \frac{\overline{\text{I}}\ \text{OK} \qquad \forall \text{m} \in \texttt{collectMethods}(\text{I}), \texttt{mbody}(\text{m}, \text{I}) \neq \texttt{Undefined}}{\texttt{interface I extends } \overline{\text{I}}\ \{\overline{\text{M}}\}\ \text{OK}}$$

1.4 Small-step Semantics

$$(\text{S-Invk})\ \frac{\texttt{mbody}(\text{m}, \text{I}, \text{J}) = (\overline{\text{X}}\ \overline{\text{x}}, \text{E}'\ e_0)}{< \text{J} > \texttt{new I}().\text{m}(< \overline{\text{E}} > \overline{e}) \to [< \overline{\text{X}} > \overline{e}/\overline{\text{x}}, < \text{J} > \texttt{new I}()/\texttt{this}]e_0}$$

$$(\text{S-PathInvk})\ \frac{\texttt{mbody}(\text{m}, \text{I}, \text{K}) = (\overline{\text{X}}\ \overline{\text{x}}, \text{E}'\ e_0)}{< \text{J} > \texttt{new I}().\text{K} :: \text{m}(< \overline{\text{E}} > \overline{e}) \to [< \overline{\text{X}} > \overline{e}/\overline{\text{x}}, < \text{J} > \texttt{new I}()/\texttt{this}]e_0}$$

$$(\text{S-SuperInvk})\ \frac{\texttt{mbody}(\text{m}, \text{K}, \text{K}) = (\overline{\text{X}}\ \overline{\text{x}}, \text{E}'\ e_0)}{\texttt{super.K} :: \text{m}(< \overline{\text{E}} > \overline{e}) \to [< \overline{\text{X}} > \overline{e}/\overline{\text{x}}, < \text{J} > \texttt{new I}()/\texttt{this}]e_0}$$

1.5 Congruence

$$(\text{C-Receiver})\ \frac{e \to e'}{e.\text{m}(\overline{e}) \to e'.\text{m}(\overline{e})}$$

$$(\text{C-Args})\ \frac{e_i \to e_i'}{e.\text{m}(..., e_i, ...) \to e.\text{m}(..., e_i', ...)}$$

$$(\text{C-StaticType})\ \texttt{new I}() \to < \text{I} > \texttt{new I}()$$

1.6 Auxilary Definitions

1.6.1 mbody

$$\frac{\mathrm{C}\{\mathrm{m}()\ \texttt{override}\ \mathrm{A}...\}}{\texttt{mbody}(\mathrm{m},\mathrm{C},\mathrm{A}) = (\overline{\mathrm{X}}\ \overline{\mathrm{x}},\mathrm{E}\ e_0)\ \texttt{IN}\ \mathrm{C}} \qquad \frac{\texttt{mbody}(\mathrm{m},\mathrm{C}) = \{\mathrm{A}.\mathrm{m}(),\mathrm{B}.\mathrm{m}(),...\} \qquad \nexists\ \mathrm{C}.\mathrm{m}()}{\texttt{mbody}(\mathrm{m},\mathrm{C},\mathrm{A}) = (\overline{\mathrm{X}}\ \overline{\mathrm{x}},\mathrm{E}\ e_0)\ \texttt{IN}\ \mathrm{A}}$$

$$\texttt{interface}\ \mathrm{I}\ \texttt{extends}\ \overline{\mathrm{I}}\ \{\overline{\mathrm{M}}\}$$

$\texttt{mbody}(\mathrm{m},\mathrm{I})$ algorithm:

- If m is defined in I directly, then return I.m()
- Else, let $\overline{\mathrm{I}'} = \texttt{mdefined}(\texttt{fathers}(\mathrm{I}))$, all ancestors of I that has directly defined $\mathrm{m}()$.
- $\overline{\mathrm{I}''} = \texttt{needed}(\overline{\mathrm{I}'})$, keep only interfaces that are needed, which are not super-interface of others.
- If $\overline{\mathrm{I}''}$ is unique, then return this unique one. Else if any two I1,I2 in $\overline{\mathrm{I}''}$ share a parent in $\overline{\mathrm{I}'}$, then diamond conflict is detected, report error. Else return multiple $\mathrm{m}()$s.

1.6.2 mtype

$\texttt{mtype}(\mathrm{m},\mathrm{C})$ algorithm:

- If the result of $\texttt{mbody}(\mathrm{m},\mathrm{C},\mathrm{A})$ is a unique method, $\mathrm{I}_0\ \mathrm{m}(\overline{\mathrm{I}}\ \mathrm{x})\ \texttt{override}\ \mathrm{J}\ \{\texttt{return}\ e_0;\}$, then $\texttt{mtype}(\mathrm{m},\mathrm{C}) = \overline{\mathrm{I}} \to \mathrm{I}_0$
- Else (`Undefined` or multiple methods returned), $\texttt{mtype}(\mathrm{m},\mathrm{C}) = \texttt{Error}$

1.6.3 ext

$\texttt{ext}(\mathrm{I},\mathrm{J})$ means interface I (directly) extends J.

$$\frac{\texttt{interface}\ \mathrm{I}\ \texttt{extends}\ \overline{\mathrm{I}}\ \{\overline{\mathrm{M}}\} \qquad \mathrm{J} \in \overline{\mathrm{I}}}{\texttt{ext}(\mathrm{I},\mathrm{J}) = \texttt{true}}$$

$$\texttt{ext}(\mathrm{I},\mathrm{J}) = \texttt{false}$$

1.6.4 collectMethods

$$\texttt{collectMethods}(\mathrm{I}) = \left(\bigcup_{\mathrm{I_i} \in \overline{\mathrm{I}}} \texttt{methods}(\mathrm{I_i})\right) \bigcup \texttt{methods}(\mathrm{I})$$

$$\texttt{methods}(\mathrm{I}) = \overline{\mathrm{M}}, \text{where } \mathrm{IT}(\mathrm{I}) = \texttt{interface}\ \mathrm{I}\ \texttt{extends}\ \overline{\mathrm{I}}\ \{\overline{\mathrm{M}}\}$$

1.6.5 needed