$$
\begin{array}{rcll}
\texttt{programs} & P & ::= & \texttt{def}\ A.m :: \tau_m \ = \ e \ \mid \ P;P \\
\texttt{values} & v & ::= & \texttt{nil} \ \mid \ [A] \ \mid \ \tau_b \\
\texttt{expressions} & e & ::= & v \ \mid \ x \ \mid \ \texttt{self} \ \mid \ x = e \ \mid \ e;e \\
 & & \mid & A.\texttt{new} \ \mid \ \texttt{if}\ e\ \texttt{then}\ e\ \texttt{else}\ e \\
 & & \mid & e.m(e) \ \mid \ f \ \mid \ f = e \\
\texttt{base types} & \tau_b & ::= & A \ \mid \ \texttt{tnil} \\
\texttt{types} & \tau & ::= & \tau_b \ \mid \ \Pi x.e \\
\texttt{mth types} & \tau_m & ::= & \tau \to \tau
\end{array}
$$

$x \in$ var ids, $f \in$ field ids, $m \in$ meth ids, $A \in$ class ids, and $\texttt{Type} \in A$

$$
\begin{array}{rcll}
\texttt{typ env} & \Gamma & ::= & \text{ids} \to \text{types} \\
\texttt{heap} & \mathcal{H} & ::= & \text{locs} \to \text{obj. records} \\
\texttt{effects} & \varphi & ::= & \texttt{PURE} \ \mid \ \texttt{IMPURE}
\end{array}
$$

MK:♣ Should types be their own kind of value? Or are they also just instances of a class A?♣

MK:♣ Do we want more kinds of types than just nominal types A? The interesting computations involve other types.♣

MK:♣ Is a self binding necessary within type computations? Both for evaluating and type checking it?♣

MK:♣ Can the TIf rule use unions of types if types are just defined as A/nil?♣

MK:♣ TAppBase rule: is this correct? Specifically type/effect checking of method body. Related: should effects be part of method annotation?♣

Type checking rules $\boxed{\Gamma \vdash e : \tau \;\&\; \varphi}$

$$\frac{}{\Gamma \vdash \texttt{nil} : \texttt{tnil} \;\&\; \texttt{PURE}}\ \text{TNil} \qquad \frac{}{\Gamma \vdash [A] : A \;\&\; \texttt{PURE}}\ \text{TObject} \qquad \frac{}{\Gamma \vdash \tau_b : \texttt{Type} \;\&\; \texttt{PURE}}\ \text{TType}$$

$$\frac{}{\Gamma \vdash x : \Gamma(x) \;\&\; \texttt{PURE}}\ \text{TVar} \qquad \frac{}{\Gamma \vdash \texttt{self} : \Gamma(\texttt{self}) \;\&\; \texttt{PURE}}\ \text{TSelf} \qquad \frac{\Gamma(x) = \tau_{b1} \quad \Gamma \vdash e : \tau_{b2} \;\&\; \varphi \quad \tau_{b2} \leq \tau_{b1}}{\Gamma \vdash x = e : \tau_{b2} \;\&\; \varphi}\ \text{TAssn}$$

$$\frac{\Gamma \vdash e_1 : \tau_{b1} \;\&\; \varphi_1 \quad \Gamma_1 \vdash e_2 : \tau_{b2} \;\&\; \varphi_2}{\Gamma \vdash e_1; e_2 : \tau_{b2} \;\&\; (\varphi_1 \sqcup \varphi_2)}\ \text{TSeq} \qquad \frac{}{\Gamma \vdash A.\texttt{new} : A \;\&\; \texttt{PURE}}\ \text{TNew}$$

$$\frac{\Gamma \vdash e_1 : \tau_{b1} \;\&\; \varphi_1 \quad \Gamma \vdash e_2 : \tau_{b2} \;\&\; \varphi_2 \quad \Gamma \vdash e_3 : \tau_{b3} \;\&\; \varphi_3}{\Gamma \vdash \texttt{if } e_1 \texttt{ then } e_2 \texttt{ else } e_3 : (\tau_{b2} \sqcup \tau_{b3}) \;\&\; (\varphi_1 \sqcup \varphi_2 \sqcup \varphi_3)}\ \text{TIf}$$

$$\frac{\Gamma(self) = A \quad \Gamma(A.f) = \tau_b \quad \Gamma \vdash e : \tau_{b1} \;\&\; \varphi \quad \tau_{b1} \leq \tau_b}{\Gamma \vdash f = e : \tau_{b1} \;\&\; \texttt{IMPURE}}\ \text{TFAssn} \qquad \frac{\Gamma(self) = A}{\Gamma \vdash f : \Gamma(A.f) \;\&\; \textit{I think impure?}}\ \text{TFVar}$$

$$\frac{\Gamma \vdash e_0 : A \;\&\; \varphi_1 \quad \Gamma \vdash e_1 : \tau_b \;\&\; \varphi_2 \quad \Gamma(A.m) = \tau_{b1} \to \tau_{b2} \quad \tau_b \leq \tau_{b1} \quad defOf(A.m) = e_m \quad \Gamma \vdash e_m : \tau \;\&\; \varphi_3}{\Gamma \vdash e_0.m(e_1) : \tau_{b2} \;\&\; \varphi_1 \sqcup \varphi_2 \sqcup \varphi_3}\ \text{TAppBase}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_0 : A \;\&\; \varphi_1 \quad \Gamma \vdash e_1 : \tau_b \;\&\; \varphi_2 \\ \Gamma(A.m) = \Pi x.e_{t1} \to \Pi x.e_{t2} \\ \Gamma \vdash e_{t1}[x \mapsto \tau_b] : \texttt{Type} \;\&\; \texttt{PURE} \quad \Gamma \vdash e_{t2}[x \mapsto \tau_b] : \texttt{Type} \;\&\; \texttt{PURE} \\ e_{t1}[x \mapsto \tau_b] \rightsquigarrow \tau_{b1} \quad e_{t2}[x \mapsto \tau_b] \rightsquigarrow \tau_{b2} \\ \tau_b \leq \tau_{b1} \quad defOf(A.m) = e_m \quad \Gamma \vdash e_m : \tau \;\&\; \varphi_3 \\ \text{type checking above terminates}\end{array}}{\Gamma \vdash e_0.m(e_1) : \tau_{b2} \;\&\; \varphi_1 \sqcup \varphi_2 \sqcup \varphi_3}\ \text{TAppComp}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_0 : A \;\&\; \varphi_1 \quad \Gamma \vdash e_1 : \tau_b \;\&\; \varphi_2 \\ \Gamma(A.m) = \Pi x.e_{t1} \to \tau_{b2} \quad \Gamma \vdash e_{t1}[x \mapsto \tau_b] : \texttt{Type} \;\&\; \texttt{PURE} \\ e_{t1}[x \mapsto \tau_b] \rightsquigarrow \tau_{b1} \quad \tau_b \leq \tau_{b1} \\ defOf(A.m) = e_m \quad \Gamma \vdash e_m : \tau \;\&\; \varphi_3 \\ \text{type checking above terminates}\end{array}}{\Gamma \vdash e_0.m(e_1) : \tau_{b2} \;\&\; \varphi_1 \sqcup \varphi_2 \sqcup \varphi_3}\ \text{TAppArg}$$

$$\frac{\begin{array}{c}\Gamma \vdash e_0 : A \;\&\; \varphi_1 \quad \Gamma \vdash e_1 : \tau_b \;\&\; \varphi_2 \\ \Gamma(A.m) = \tau_{b1} \to \Pi x.e_{t2} \quad \tau_b \leq \tau_{b1} \\ \Gamma \vdash e_{t2}[x \mapsto \tau_b] : \texttt{Type} \;\&\; \texttt{PURE} \quad e_{t2}[x \mapsto \tau_b] \rightsquigarrow \tau_{b2} \\ defOf(A.m) = e_m \quad \Gamma \vdash e_m : \tau \;\&\; \varphi_3 \\ \text{type checking above terminates}\end{array}}{\Gamma \vdash e_0.m(e_1) : \tau_{b2} \;\&\; \varphi_1 \sqcup \varphi_2 \sqcup \varphi_3}\ \text{TAppRet}$$

Dynamic semantics $\boxed{e \rightsquigarrow e'}$