$$
\begin{array}{rcl}
\Pi & ::= & \overline{\mathsf{defmodtype}\ X\ \mathsf{do}\ \overline{P}\ \overline{D}\ \mathsf{end}}\ \mathsf{defmodule}\ \mathit{Main}\ \mathsf{do}\ \overline{B}\ \mathsf{end} \\
N & ::= & x \\
 & | & X \\
S & ::= & \$\mathsf{behaviour}\ X \\
P & ::= & \$\mathsf{param}\ x \\
B & ::= & \mathsf{defmodule}\ X\ \mathsf{do}\ \overline{P}\ \overline{S}\ \overline{B}\ \mathsf{end} \\
 & | & x = v \\
 & | & \$\mathsf{type}\ x = t \\
 & | & \$\mathsf{opaque}\ x = t \\
E & ::= & v \\
 & | & x \\
 & | & E(\overline{E}, E) \\
 & | & \%\{\overline{\ell = E}\} \\
 & | & E.\ell \\
 & | & (E \in t)?E : E \\
 & | & \overline{X\ [\overline{x = t}]}.x \\
 & | & \overline{X\ [\overline{x = t}]}.X\ [\overline{x = t}] \\
v & ::= & c \\
 & | & \%\{\overline{\ell = v}\} \\
 & | & \$ \wedge \overline{t \to t}\ \mathsf{fn}\ \overline{x} \to E \\
 & | & \$ \cap \overline{(\overline{N : T}) \to T}\ \mathsf{fn}\ \overline{N} \to E \\
T & ::= & t \\
 & | & (\overline{N : T}) \to T \\
 & | & \star \\
 & | & \{\overline{S}; \overline{D}\} \\
t & ::= & \mathsf{int} \\
 & | & t \to t \\
 & | & \%\{\overline{f}\} \\
 & | & t \vee t \\
 & | & t \wedge t \\
 & | & \neg t \\
 & | & \alpha \\
 & | & \mathbb{O} \\
 & | & \overline{X\ [\overline{x = t}]}.x \\
D & ::= & \$\mathsf{module}\ X : T \\
 & | & \$\mathsf{callback}\ x : \bigcap \overline{T} \\
 & | & \$\mathsf{opaque}\ x \\
 & | & \$\mathsf{type}\ x = t
\end{array}
$$

Figure 1: Syntax of the surface language

$$
\begin{array}{rcl}
\tau & ::= & t \\
 & | & \star \\
 & | & (\overline{N : \tau}) \to \tau \\
 & | & \mathsf{like}\left(\overline{X\ [\overline{x = t}]}.X\ [\overline{x = t}]\right) \\
 & | & \cap \overline{\tau} \\
 & | & X\ [\overline{x = t}]
\end{array}
$$

Figure 2: Syntax of surface module types

$$
\begin{array}{ccccc}
\$\mathsf{module}\ X : T & \cap & \$\mathsf{module}\ X : T' & = & \$\mathsf{module}\ X : T \cap T' \\
\$\mathsf{callback}\ X : \bigcap \overline{T} & \cap & \$\mathsf{callback}\ X : \bigcap \overline{T'} & = & \$\mathsf{callback}\ X : \bigcap \overline{T}\,\overline{T'} \\
\$\mathsf{type}\ x = t & \cap & \$\mathsf{type}\ x = t' & = & \$\mathsf{type}\ x = t \wedge t' \\
\$\mathsf{opaque}\ x & \cap & \$\mathsf{opaque}\ x & = & \$\mathsf{opaque}\ x \\
\$\mathsf{opaque}\ x & \cap & \$\mathsf{type}\ x = t & = & \$\mathsf{type}\ x = t \\
\$\mathsf{type}\ x = t & \cap & \$\mathsf{opaque}\ x & = & \$\mathsf{type}\ x = t \\
D & \cap & D' & = & \epsilon
\end{array}
$$

Figure 3: Component-wise intersection

ElEnv-Empty

$$\frac{}{\Sigma, \Gamma \vdash \epsilon}$$

ElEnv-Expr

$$\frac{\Sigma, \Gamma \vdash t : \star}{\Sigma, \Gamma \vdash x : t, \Gamma}$$

ElEnv-Type

$$\frac{\Sigma, \Gamma \vdash t : \star}{\Sigma, \Gamma \vdash x = t, \Gamma}$$

ModEnv-ModuleType

$$\frac{\Sigma, \Gamma \vdash \left\{\overline{\$\mathsf{type}\ x = \alpha}; \overline{D}\right\}}{\Sigma, \Gamma \vdash X = \overline{x} \mapsto \overline{D}, \Sigma}$$

ModEnv-Empty

$$\frac{}{\Sigma, \Gamma \vdash \epsilon}$$

ModEnv-Module

$$\frac{\Sigma, \Gamma \vdash \left\{\overline{\$\mathsf{type}\ x = \alpha}; \overline{B}\right\}}{\Sigma, \Gamma \vdash X = \overline{x} \mapsto \overline{B}, \Sigma}$$

ModEnv-Dom

$$\Sigma = X : \overline{X} \mapsto$$

Figure 4: Formation rules for environments

EqPath-Empty

$$\frac{}{\Sigma, \Gamma \vdash \epsilon \cong \epsilon}$$

EqPath-Add

$$\frac{\Sigma, \Gamma \vdash P_1 \cong P_2 \qquad \forall i. \Sigma, \Gamma \vdash t_i \cong t'_i}{\Sigma, \Gamma \vdash P_1.X\left[\overline{x_i = t_i}\right] \cong P_2.X\left[\overline{x_i = t'_i}\right]}$$

Sub-Opaque

$$\frac{\Sigma, \Gamma \vdash P_1 \cong P_2 \qquad \mathsf{struct}(P_1) = \{\$\mathsf{opaque}\ t; \ldots\}}{\Sigma, \Gamma \vdash P_1.t \preccurlyeq P_2.t}$$

Figure 5: Subtyping rules with path

Figure 6: Typing rules for declarations

$$\textsc{Bind-DefModule}\ \frac{\Sigma;\Gamma \vdash \overline{B} : \overline{D} \qquad \Sigma;\Gamma, X : \left(\overline{P : \star}\right) \to \left\{\overline{S}; \overline{D}\right\} \vdash \overline{B_0} : \overline{D_0}}{\Sigma;\Gamma \vdash \left(\mathsf{defmodule}\, X\, \mathsf{do}\, \overline{PSB}\, \mathsf{end}\right) \overline{B_0} : \left(\$\mathsf{module}\, X : \left(\overline{P : \star}\right) \to \left\{\overline{S}; \overline{D}\right\}\right) \overline{D_0}}$$

$$\textsc{Bind-Type}\ \frac{\Sigma;\Gamma \vdash t : \star \qquad \Sigma;\Gamma, x = t \vdash \overline{B} : \overline{D}}{\Sigma;\Gamma \vdash (\$\mathsf{type}\, x = t)\, \overline{B} : (\$\mathsf{type}\, x = t)\, \overline{D}}$$

$$\textsc{Bind-Opaque}\ \frac{\Sigma;\Gamma \vdash t : \star \qquad \Sigma;\Gamma, x = t \vdash \overline{B} : \overline{D}}{\Sigma;\Gamma \vdash (\$\mathsf{opaque}\, x = t)\, \overline{B} : (\$\mathsf{opaque}\, x)\, \overline{D}}$$

$$\textsc{Bind-Empty}\ \frac{}{\Sigma;\Gamma \vdash \epsilon : \epsilon} \qquad \textsc{Bind-Value}\ \frac{\Sigma;\Gamma \vdash v : \cap \overline{T} \qquad \Sigma;\Gamma, x : \cap \overline{T} \vdash \overline{B} : \overline{D}}{\Sigma;\Gamma \vdash (x = v)\, \overline{B} : \left(\$\mathsf{callback}\, x : \cap \overline{T}\right) \overline{D}}$$

Figure 7: Typing rules for bindings

Figure 8: Typing rules for the surface language

$$\begin{array}{ccccc}
\$\mathsf{module}\, X : T & \cup & \$\mathsf{module}\, X : T' & = & \$\mathsf{module}\, X : T \cup T' \\
\$\mathsf{callback}\, X : \bigcap \overline{T} & \cup & \$\mathsf{callback}\, X : \bigcap \overline{T'} & = & ? \\
\$\mathsf{type}\, x = t & \cup & \$\mathsf{type}\, x = t' & = & \$\mathsf{type}\, x = t \cup t' \\
\$\mathsf{opaque}\, x & \cup & \$\mathsf{opaque}\, x & = & \$\mathsf{opaque}\, x \\
\$\mathsf{opaque}\, x & \cup & \$\mathsf{type}\, x = t & = & \$\mathsf{opaque}\, x \\
\$\mathsf{type}\, x = t & \cup & \$\mathsf{opaque}\, x & = & \$\mathsf{opaque}\, x \\
D & \cup & D' & = & \epsilon
\end{array}$$

Figure 9: Component-wise union