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

Figure 1: Syntax of the surface language

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

Figure 2: Syntax of surface module types

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

Figure 3: Component-wise intersection

$$
\textsc{ElEnv-Empty}\quad \frac{}{\Sigma, \Gamma \vdash \epsilon}
\qquad
\textsc{ElEnv-Expr}\quad \frac{\Sigma, \Gamma \vdash t : \star}{\Sigma, \Gamma \vdash x : t, \Gamma}
\qquad
\textsc{ElEnv-Type}\quad \frac{\Sigma, \Gamma \vdash t : \star}{\Sigma, \Gamma \vdash x = t, \Gamma}
$$

$$
\textsc{ModEnv-ModuleType}\quad \frac{\Sigma, \Gamma \vdash \left\{\overline{\$\text{type}\, x = \alpha}; \overline{D}\right\}}{\Sigma, \Gamma \vdash X = \overline{x} \mapsto \overline{D}, \Sigma}
\qquad
\textsc{ModEnv-Empty}\quad \frac{}{\Sigma, \Gamma \vdash \epsilon}
\qquad
\textsc{ModEnv-Module}\quad \frac{\Sigma, \Gamma \vdash \left\{\overline{\$\text{type}\, x = \alpha}; \overline{B}\right\}}{\Sigma, \Gamma \vdash X = \overline{x} \mapsto \overline{B}, \Sigma}
$$

Figure 4: Formation rules for environments

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

$$
\textsc{Sub-Opaque}\quad \frac{\Sigma, \Gamma \vdash P_1 \cong P_2 \qquad \text{struct}(P_1) = \{\$\text{opaque}\, t; \ldots\}}{\Sigma, \Gamma \vdash P_1.t \preccurlyeq P_2.t}
$$

Figure 5: Subtyping rules with path

Figure 6: Typing rules for declarations

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

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

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

$$\frac{}{\Sigma;\Gamma\vdash\epsilon:\epsilon}\;\textsc{Bind-Empty}\qquad\qquad \frac{\Sigma;\Gamma\vdash v:t\qquad \Sigma;\Gamma,x:t\vdash\overline{B}:\overline{D}}{\Sigma;\Gamma\vdash x=v,\overline{B}:(x:t,\overline{D})}\;\textsc{Bind-Value}$$

Figure 7: Typing rules for bindings

Figure 8: Typing rules for the surface language

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

Figure 9: Component-wise union