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

Figure 1: Syntax of the surface language

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

Figure 2: Syntax of surface module types

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

Figure 3: Component-wise intersection

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

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

Figure 4: Formation rules for environments

Struct-Com

$$\frac{}{\Sigma, \Gamma \vdash \mathsf{struct}(M \cap M') = \mathsf{struct}(M' \cap M)}$$

Struct-Assoc

$$\frac{}{\Sigma, \Gamma \vdash \mathsf{struct}(M \cap (M' \cap M'')) = \mathsf{struct}((M \cap M') \cap M'')}$$

Struct-Declaration

$$\frac{\Sigma, \Gamma \vdash \overline{D}}{\Sigma, \Gamma \vdash \mathsf{struct}\left(\left\{\overline{D}\right\}\right) = \left\{\overline{D}\right\}}$$

Struct-ModuleType

$$\frac{X = \left(\overline{x_i} \mapsto \overline{D}\right) \in \Sigma \qquad \forall i. \Sigma, (x_1 = t_1, \ldots, x_i = t_i, \Gamma) \vdash t_{i+1} : \star}{\Sigma, \Gamma \vdash \mathsf{struct}\left(X\left[\overline{x_i = t_i}\right]\right) = \left\{\overline{\$\mathsf{type}\, x_i = t_i}; \overline{D}\right\}}$$

Struct-ModuleTypesIntersection

$$\frac{\forall i \neq i', j, j'. x_j^i \neq x_{j'}^{i'} \qquad \forall i. \Sigma, \Gamma \vdash \mathsf{struct}\left(X^i[\overline{x_j^i = t_j^i}]\right) = \overline{D}^i}{\Sigma, \Gamma \vdash \mathsf{struct}\left(\bigcap \overline{X^i[\overline{x_j^i = t_j^i}]}\right) = \left\{\overline{\overline{D}^i}\right\}}$$

Struct-DeclarationIntersection

$$\frac{}{\Sigma, \Gamma \vdash \mathsf{struct}\left(\left\{\overline{D}\right\} \cap M\right) = \left\{\overline{D}\right\}}$$

Figure 5: Erasure of name subtyping

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 6: Subtyping rules with path

Figure 7: Typing rules for declarations

Figure 8: Typing rules for the surface language

\$module $X : T$	$\cup$	\$module $X : T'$	=	\$module $X : T \cup T'$
\$callback $X : \bigcap \overline{T}$	$\cup$	\$callback $X : \bigcap \overline{T'}$	=	?
\$type $x = t$	$\cup$	\$type $x = t'$	=	\$type $x = t \cup t'$
\$opaque x	$\cup$	\$opaque x	=	\$opaque x
\$opaque x	$\cup$	\$type $x = t$	=	\$opaque x
\$type $x = t$	$\cup$	\$opaque x	=	\$opaque x
D	$\cup$	D'	=	ϵ

Figure 9: Component-wise union