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

Figure 1: Syntax of the surface language

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

Figure 2: 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

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

ModEnv-Empty

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

ModEnv-Module

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

Figure 3: 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

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

Struct-ModuleTypesIntersection

$$\frac{\forall i \neq i', j, j'. x^i_j \neq x^{i'}_{j'} \qquad \forall i. \Sigma, \Gamma \vdash \mathsf{struct}\left(\overline{X^i[\overline{x^i_j = t^i_j}]}\right) = \overline{D}^i}{\Sigma, \Gamma \vdash \mathsf{struct}\left(\overline{X^i[\overline{x^i_j = t^i_j}]}\right) = \left\{\overline{\overline{D}^i}\right\}}$$

Struct-DeclarationIntersection

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

Figure 4: Erasure of name subtyping

EqPath-Empty

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

EqPath-Add

$$\frac{\Sigma, \Gamma \vdash P_1 \cong P_2 \qquad \forall i. \Sigma, \Gamma \vdash t_i \cong t_i'}{\Sigma, \Gamma \vdash P_1.X\left[\overline{x_i = t_i}\right] \cong P_2.X\left[\overline{x_i = t_i'}\right]}$$

Figure 5: Subtyping rules with path

Figure 6: Typing rules for declarations

Figure 7: 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 8: Component-wise union