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

Figure 1: Syntax of the surface language

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

Figure 2: Component-wise intersection

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

Figure 3: Component-wise union

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

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

Figure 4: Formation rules for environments

$$\textsc{Struct-Com}\quad \frac{}{\Sigma, \Gamma \vdash \mathsf{struct}(M \cap M') = \mathsf{struct}(M' \cap M)}$$

$$\textsc{Struct-Assoc}\quad \frac{}{\Sigma, \Gamma \vdash \mathsf{struct}(M \cap (M' \cap M'')) = \mathsf{struct}((M \cap M') \cap M'')}$$

$$\textsc{Struct-Declaration}\quad \frac{\Sigma, \Gamma \vdash \overline{D}}{\Sigma, \Gamma \vdash \mathsf{struct}\left(\left\{\overline{D}\right\}\right) = \left\{\overline{D}\right\}}$$

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

Figure 5: Erasure of name subtyping

Figure 6: Typing rules for declarations

Figure 7: Typing rules for the surface language