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

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

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

Figure 3: Component-wise intersection

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

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

ModEnv-Empty

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

ModEnv-Dom

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

ModEnv-Module

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

Figure 4: Formation rules for environments

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: Rules for path equivalence

Figure 6: Typing rules for declarations

BIND-DEFMODULE

$$\frac{\begin{array}{c}\Sigma;\Gamma,\overline{P:*}\vdash\overline{B}:\overline{D}\\ \Sigma;\Gamma,X:\left(\overline{P:\star}\right)\to\left\{\overline{S_i};\overline{D}\right\}\vdash\overline{B_0}:\overline{D_0}\qquad\forall i.\Sigma(S_i)=\overline{x_i:\star}\to\overline{D_i}\\ \forall i\neq j.(\overline{x_i},\mathsf{dom}(D_i)\#\overline{x_j},\mathsf{dom}(D_j))\qquad\left\{\overline{D}\right\}\preccurlyeq\left\{\overline{\overline{x_i:[=\_]D_i}}\right\}\end{array}}{\Sigma;\Gamma\vdash\left(\mathsf{defmodule}\,X\,\mathsf{do}\,\overline{PS_iB}\,\mathsf{end}\right)\overline{B_0}:\left(X:\left(\overline{P:\star}\right)\to\left\{\overline{S_i};\overline{D}\right\}\right)\overline{D_0}}$$

BIND-TYPE

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

BIND-OPAQUE

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

BIND-EMPTY

$$\frac{}{\Sigma;\Gamma\vdash\epsilon:\epsilon}$$

BIND-VALUE

$$\frac{\Sigma;\Gamma\vdash v:\cap\overline{T}\qquad\Sigma;\Gamma,x:\cap\overline{T}\vdash\overline{B}:\overline{D}}{\Sigma;\Gamma\vdash(x=v)\,\overline{B}:\left(x:\cap\overline{T}\right)\overline{D}}$$

Figure 7: Typing rules for bindings

PATH-EMPTY

$$\frac{}{\Sigma;\Gamma\vdash\epsilon:\{\epsilon;\Gamma\}}$$

PATH-SUBMODULE

$$\frac{}{}$$

Figure 8: Well-formdness rules for paths

$$\textsc{Type-Variable}\ \frac{\Sigma;\Gamma \vdash P : \left\{\overline{S};\overline{D}(x :\cap T)\overline{D'}\right\}}{\Sigma;\Gamma \vdash P.x :\cap T}$$

$$\textsc{Type-Subsumption}\ \frac{\Sigma;\Gamma \vdash T \qquad \Sigma\Gamma \vdash E :\cap\overline{T'} \qquad \Sigma;\Gamma \vdash\cap\overline{T'} \preccurlyeq T}{\Sigma;\Gamma \vdash E : T}$$

$$\textsc{Type-BigFunction}\ \frac{\Sigma;\Gamma \vdash\cap\overline{\overline{N : T} \to T'} \Sigma;\Gamma,\overline{N : T} \vdash E : T'}{\Sigma;\Gamma \vdash \$\cap\overline{\forall\overline{\alpha}\left(\overline{N : T}\right) \to T'}\ \mathsf{fn}\,\overline{N} \to E :\cap\overline{\overline{N : T} \to T'}}$$

$$\textsc{Type-Module}\ \frac{\Sigma;\Gamma \vdash P : \left\{\overline{S_0};\overline{D_0}(X : \overline{y : \star} \to \left\{\overline{S};\overline{D}\right\})\overline{D_1}\right\} \qquad \Sigma;\Gamma \vdash \overline{t} \qquad \overline{x} \simeq \overline{y}}{\Sigma;\Gamma \vdash P.X\left[\overline{x = t}\right] : \left\{\overline{S};\overline{D}\right\}}$$

Figure 9: Typing rules for the surface language

$$\textsc{Sub-StarRefl}\ \frac{}{\star \preccurlyeq \star} \qquad \textsc{Sub-Elixir}\ \frac{t \preccurlyeq t'}{\Sigma;\Gamma \vdash t \preccurlyeq t'} \qquad \textsc{Sub-Intersection}\ \frac{}{\Sigma;\Gamma \vdash\cap\overline{T_i} \preccurlyeq T_i}$$

$$\textsc{Sub-ModuleLeft}\ \frac{\Sigma;\Gamma \vdash P : \left\{\overline{S};\overline{D}(x : [= t])\overline{D'}\right\} \qquad \Sigma;\Gamma \vdash t \preccurlyeq T}{\Sigma;\Gamma \vdash P.x \preccurlyeq T}$$

$$\textsc{Sub-ModuleRight}\ \frac{\Sigma;\Gamma \vdash P : \left\{\overline{S};\overline{D}(x : [= t])\overline{D'}\right\} \qquad \Sigma;\Gamma \vdash\cap\overline{T} \preccurlyeq t}{\Sigma;\Gamma \vdash\cap\overline{T} \preccurlyeq P.x}$$

$$\textsc{Sub-Opaque}\ \frac{\Sigma;\Gamma \vdash P \cong P' \qquad \Sigma;\Gamma \vdash P : \left\{\overline{S};\overline{D}(x : \star)\overline{D'}\right\}}{\Sigma;\Gamma \vdash P.x \preccurlyeq P'.x}$$

Figure 10: Subtyping rules

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

Figure 11: Component-wise union