

Path		Module types	
$P ::= X_i \mid P.X$		$\mathcal{M}^\circ ::= \mathcal{X}_i \mid \mathcal{P}.\mathcal{X}$	(Variables)
$\mathcal{P} ::= X_i \mid \mathcal{P}.X \mid \mathcal{P}_1(\mathcal{P}_2) \mid (\mathcal{P} :> \mathcal{M})$		$\mid (= \mathcal{P})$	(Alias)
Module Expressions		$\mid \mathcal{M} \text{ with } \mathcal{C}$	(Substitution)
$M ::= P$	(Variables)	$\mid (X_i : \mathcal{M}_1) \rightarrow \mathcal{M}_2$	(Functor)
$\mid (M : \mathcal{M})$	(Opaque Ascription)	$\mid \text{sig } \mathcal{S} \text{ end}$	(Signature)
$\mid (M :> \mathcal{M})$	(Transparent Ascription)	$\mathcal{M} ::= \mathcal{M} /_i \mathcal{P}$	(Strengthening)
$\mid M_1(M_2)$	(Functor application)	$\mid \mathcal{M}_{[\mathcal{C}]}$	(Substution)
$\mid (X_i : \mathcal{M}) \rightarrow M$	(Functor)	$\mid (\mathcal{M} :> \mathcal{M}')$	(Ascription)
$\mid \text{struct } \mathcal{S} \text{ end}$	(Structure)	$\mid \mathcal{M}^\circ$	
Structures		Substitutions	
$S ::= \varepsilon \mid D; S$		$\mathcal{C} ::= P.t = \tau$	
$D ::= \text{let } x_i = e$	(Values)	$\mid P = P'$	
$\mid \text{type } t = \tau$	(Types)	Signatures	
$\mid \text{module } X_i = M$	(Modules)	$\mathcal{S} ::= \varepsilon \mid \mathcal{D}; \mathcal{S}$	
$\mid \text{module type } \mathcal{X}_i = \mathcal{M}$	(Module types)	$\mathcal{D} ::= \text{val } x_i : \tau$	(Values)
Core language		$\mid \text{type } t = \tau$	(Types)
$e ::= P.x$	(Qualified variable)	$\mid \text{type } t$	(Abstract types)
$\mid \dots$	(Other expressions)	$\mid \text{module } X_i : \mathcal{M}$	(Modules)
$\tau ::= \mathcal{P}.t$	(Qualified type)	$\mid \text{module type } \mathcal{X}_i = \mathcal{M}$	(Module types)
$\mid \dots$	(Other types)	Environments	
		$\Gamma ::= \mathcal{S}$	

Figure 1: Module language

Judgements:

$\Gamma \blacktriangleright M : \mathcal{M}$	The module M is of type \mathcal{M} in Γ . See fig. 5.
$\Gamma \blacktriangleright \mathcal{M} <: \mathcal{M}'$	The module type \mathcal{M} is a subtype of \mathcal{M}' in Γ . See fig. 6.
$\Gamma \models \mathcal{M}$	The module type \mathcal{M} is well-formed in Γ . See ??.

Operations:

$\mathcal{M} /_i \mathcal{P}$	Strengthen \mathcal{M} by \mathcal{P} while ignoring the list of module paths l .
$\mathcal{M}_{[\mathcal{C}]}$	Applies the substitution \mathcal{C} in \mathcal{M} .
$(\mathcal{M} :> \mathcal{M}')$	Computes the transparent ascription of \mathcal{M} by \mathcal{M}' .
$\text{force}_\Gamma(\mathcal{M})$	Forces all the operations in \mathcal{M} until obtaining a simple module type \mathcal{M}° .

Environment accesses:

$\Gamma(P)$	Lookup the module or module type P in the environment (i.e., the signature) Γ .
$\text{resolve}_\Gamma(\mathcal{M}^\circ)$	Resolve \mathcal{M}° until it's not a path (i.e., either an arrow or a signature).
$\text{normalize}_\Gamma(\mathcal{P})$	Normalizes the path \mathcal{P} in Γ

$$\begin{aligned}
& \mathcal{P}' /_i \mathcal{P} \rightarrow \text{resolver}_\Gamma(\mathcal{P}') /_i \mathcal{P} \\
& \text{sig } \mathcal{S} \text{ end} /_i \mathcal{P} \rightarrow \text{sig } \mathcal{S} /_i \mathcal{P} \text{ end} \\
& (\mathcal{M} \text{ with } \mathcal{C}) /_i \mathcal{P} \rightarrow (\mathcal{M} /_{l, \text{id}(\mathcal{C})} \mathcal{P}) \text{ with } \mathcal{C} \\
& ((X : \mathcal{M}) \rightarrow \mathcal{M}') /_i \mathcal{P} \rightarrow (X : \mathcal{M}) \rightarrow \mathcal{M}' /_i \mathcal{P}(X) \\
& (= \mathcal{P}') /_i \mathcal{P} \rightarrow (= \mathcal{P}) \\
& (\mathcal{M} :> \mathcal{M}') /_i \mathcal{P} \rightarrow (\mathcal{M} /_i \mathcal{P} :> \mathcal{M}') \\
\\
& \text{type } t = \tau; \mathcal{S} /_i \mathcal{P} \rightarrow \text{type } t = \tau; (\mathcal{S} /_i \mathcal{P}) & \text{when } t \notin l \\
& \text{type } t; \mathcal{S} /_i \mathcal{P} \rightarrow \text{type } t; (\mathcal{S} /_i \mathcal{P}) & \text{when } t \in l \\
& \text{type } t = \tau; \mathcal{S} /_i \mathcal{P} \rightarrow \text{type } t = \mathcal{P}.t; (\mathcal{S} /_i \mathcal{P}) \\
& \text{type } t; \mathcal{S} /_i \mathcal{P} \rightarrow \text{type } t = \mathcal{P}.t; (\mathcal{S} /_i \mathcal{P}) \\
& \text{module } X : \mathcal{M}; \mathcal{S} /_i \mathcal{P} \rightarrow \text{module } X : \mathcal{M} /_{\text{chop}(l, X)} \mathcal{P}.X; (\mathcal{S} /_i \mathcal{P}) & \text{when } X \notin l \\
& \text{module } X : \mathcal{M}; \mathcal{S} /_i \mathcal{P} \rightarrow \text{module } X : \mathcal{M}; (\mathcal{S} /_i \mathcal{P}) & \text{when } X \in l \\
& \text{module type } \mathcal{X} = \mathcal{M}; \mathcal{S} /_i \mathcal{P} \rightarrow \text{module type } \mathcal{X} = \mathcal{M}; (\mathcal{S} /_i \mathcal{P}) \\
\\
& \mathcal{M} /_i \mathcal{P} /_{i'} \mathcal{P}' \rightarrow \mathcal{M} /_i \mathcal{P} \\
& \mathcal{M}_{[c]} /_i \mathcal{P} \rightarrow \mathcal{M}_{[c]} /_i \mathcal{P}
\end{aligned}$$

Figure 2: Module strengthening operation – $\mathcal{M} /_i \mathcal{P}$

$$\begin{aligned}
& \mathcal{P}_{[c]} \rightarrow \text{resolver}_\Gamma(\mathcal{P})_{[c]} \\
& \text{sig } \mathcal{S} \text{ end}_{[c]} \rightarrow \text{sig } \mathcal{S}_{[c]} \text{ end} \\
& (\mathcal{M} \text{ with } \mathcal{C}')_{[c]} \rightarrow \mathcal{M}_{[c']}_{[c]} \\
& ((X : \mathcal{M}) \rightarrow \mathcal{M}')_{[c]} \rightarrow \text{fail} \\
& (= \mathcal{P})_{[c]} \rightarrow (= \mathcal{P}) & \text{when } \text{resolver}_\Gamma(\mathcal{P})_{[c]} \rightarrow _ \\
& (\mathcal{M} :> \mathcal{M}')_{[c]} \rightarrow (\mathcal{M}_{[c]} :> \mathcal{M}') \\
\\
& \text{type } t = \tau; \mathcal{S}_{[t=\tau']} \rightarrow \text{type } t = \tau'; \mathcal{S} & \text{when } \Gamma \triangleright \tau' <: \tau \\
& \text{module } X : \mathcal{M}; \mathcal{S}_{[X=\mathcal{P}]} \rightarrow \text{module } X : (= \mathcal{P}); \mathcal{S} & \text{when } \Gamma \blacktriangleright (= \mathcal{P}) <: \mathcal{M} \\
& \text{module } X : \mathcal{M}; \mathcal{S}_{[X.P.t=\tau]} \rightarrow \text{module } X : \mathcal{M}_{[P.t=\tau]}; \mathcal{S} \\
& \text{module } X : \mathcal{M}; \mathcal{S}_{[X.P=\mathcal{P}']} \rightarrow \text{module } X : \mathcal{M}_{[P=\mathcal{P}']} ; \mathcal{S} \\
& (\mathcal{D}; \mathcal{S})_{[c]} \rightarrow \mathcal{D}; \mathcal{S}_{[c]} & \text{when } \text{id}(\mathcal{D}) \text{ is not a prefix of } \text{id}(\mathcal{C}) \\
\\
& (\mathcal{M} /_i \mathcal{P})_{[c]} \rightarrow (\mathcal{M} /_i \mathcal{P})_{[c]} \\
& \mathcal{M}_{[c']}_{[c]} \rightarrow \mathcal{M}_{[c']}_{[c]}
\end{aligned}$$

Figure 3: Substitution operation – $\mathcal{M}_{[c]}$

$$\begin{aligned}
& (\mathcal{M} :> \mathcal{M}') \rightarrow (\text{resolver}_{\Gamma}(\mathcal{M}) :> \text{resolver}_{\Gamma}(\mathcal{M}')) \\
& (\text{sig } \mathcal{D}_1; \dots; \mathcal{D}_n \text{ end} :> \text{sig } \mathcal{D}'_1; \dots; \mathcal{D}'_m \text{ end}) \rightarrow \text{sig } (\mathcal{D}_{\pi(i)} :> \mathcal{D}_i) \text{ end} && \text{with } \pi : [1; m] \rightarrow [1; n] \\
& ((\text{val } x_i : \tau) :> (\text{val } x_i : \tau')) \rightarrow (\text{val } x_i : \tau) && \text{when } \Gamma \triangleright \tau <: \tau' \\
& ((\text{val } x_i : \tau) :> \varepsilon) \rightarrow \varepsilon \\
& ((\text{module } X_i : \mathcal{M}) :> (\text{module } X_i : \mathcal{M}')) \rightarrow (\text{module } X_i : (\mathcal{M} :> \mathcal{M}')) \\
& ((\text{module } X_i : \mathcal{M}) :> \varepsilon) \rightarrow (\text{module } X_i : (\mathcal{M} :> \varepsilon)) \\
& ((\text{type } t_i = \tau) :> (\text{type } t_i)) \rightarrow (\text{type } t_i = \tau) \\
& ((\text{type } t_i = \tau) :> (\text{type } t_i = \tau')) \rightarrow (\text{type } t_i = \tau) && \text{when } \Gamma \triangleright \tau <: \tau' \\
& ((\text{type } t_i = \tau) :> \varepsilon) \rightarrow (\text{type } t_i = \tau) \\
& ((\text{type } t_i) :> (\text{type } t_i)) \rightarrow (\text{type } t_i) \\
& ((\text{type } t_i) :> (\text{type } t_i = \tau')) \rightarrow (\text{type } t_i = \tau') && \text{when } \Gamma \triangleright t <: \tau' \\
& ((\text{type } t_i) :> \varepsilon) \rightarrow (\text{type } t_i)
\end{aligned}$$

Figure 4: Module transparent ascription – $(\mathcal{M} :> \mathcal{M}')$

$$\begin{array}{c}
\text{MODVAR} \\
\frac{\Gamma(P) = _}{\Gamma \blacktriangleright P : (= P)} \quad \frac{\Gamma \blacktriangleright M : \mathcal{M}' \quad \Gamma \blacktriangleright \mathcal{M}' <: \mathcal{M}}{\Gamma \blacktriangleright M : \mathcal{M}} \quad \frac{\Gamma \models \mathcal{M} \quad \Gamma \blacktriangleright M : \mathcal{M}}{\Gamma \blacktriangleright (M : \mathcal{M}) : \mathcal{M}} \quad \frac{\Gamma \models \mathcal{M} \quad \Gamma \blacktriangleright M : \mathcal{M}_i}{\Gamma \blacktriangleright (M :> \mathcal{M}) : (\mathcal{M}_i :> \mathcal{M})} \\
\\
\frac{\Gamma \blacktriangleright M_f : (X_i : \mathcal{M}_a) \rightarrow \mathcal{M}_r \quad \Gamma \blacktriangleright M_a : \mathcal{M}_a}{\Gamma \blacktriangleright M_f(M_a) : \mathcal{M}_{r[X=(M_a :> \mathcal{M}_a)]}} \quad \frac{\Gamma \models \mathcal{M} \quad X_i \notin \text{BV}(\Gamma) \quad \Gamma; \text{module } X_i : \mathcal{M} \blacktriangleright M : \mathcal{M}'}{\Gamma \blacktriangleright (X_i : \mathcal{M}) \rightarrow M : (X_i : \mathcal{M}) \rightarrow \mathcal{M}'} \\
\\
\frac{\Gamma \blacktriangleright S : \mathcal{S}}{\Gamma \blacktriangleright \text{struct } S \text{ end : sig } \mathcal{S} \text{ end}} \quad \frac{}{\Gamma \blacktriangleright \varepsilon : \varepsilon} \\
\\
\frac{\Gamma \triangleright e : \tau \quad x_i \notin \text{BV}(\Gamma) \quad \Gamma; \text{val } x_i : \tau \blacktriangleright S : \mathcal{S}}{\Gamma \blacktriangleright (\text{let } x_i = e; S) : (\text{val } x_i : \tau; \mathcal{S})} \quad \frac{\Gamma \blacktriangleright M : \mathcal{M} \quad X_i \notin \text{BV}(\Gamma) \quad \Gamma; \text{module } X_i : \mathcal{M} \blacktriangleright S : \mathcal{S}}{\Gamma \blacktriangleright (\text{module } X_i = M; S) : (\text{module } X_i : \mathcal{M}; \mathcal{S})} \\
\\
\frac{\Gamma \models \tau \quad t_i \notin \text{BV}(\Gamma) \quad \Gamma; \text{type } t_i = \tau \blacktriangleright S : \mathcal{S}}{\Gamma \blacktriangleright (\text{type } t = \tau; S) : (\text{type } t = \tau; \mathcal{S})} \quad \frac{\Gamma \models \mathcal{M} \quad \mathcal{X}_i \notin \text{BV}(\Gamma) \quad \Gamma; \text{module type } \mathcal{X}_i = \mathcal{M} \blacktriangleright S : \mathcal{S}}{\Gamma \blacktriangleright (\text{module type } \mathcal{X}_i = \mathcal{M}; S) : (\text{module type } \mathcal{X}_i = \mathcal{M}; \mathcal{S})}
\end{array}$$

Figure 5: Module typing rules – $\Gamma \blacktriangleright M : \mathcal{M}$

$$\begin{array}{c}
\frac{\Gamma \blacktriangleright \text{force}_\Gamma(\mathcal{M}) <: \text{force}_\Gamma(\mathcal{M}')}{\Gamma \blacktriangleright \mathcal{M} <: \mathcal{M}'} \quad \frac{\text{normalizer}_\Gamma(\mathcal{P}) = \text{normalizer}_\Gamma(\mathcal{P}')}{\Gamma \blacktriangleright (= \mathcal{P}) <: (= \mathcal{P}')} \quad \frac{\Gamma \blacktriangleright \text{resolve}_\Gamma(\mathcal{M}) <: \text{resolve}_\Gamma(\mathcal{M}')}{\Gamma \blacktriangleright \mathcal{M} <: \mathcal{M}'} \\
\\
\frac{\Gamma \blacktriangleright \mathcal{M}'_a <: \mathcal{M}_a \quad \Gamma, \text{module } X : \mathcal{M}'_a \blacktriangleright \mathcal{M}_r <: \mathcal{M}'_r}{\Gamma \blacktriangleright (X : \mathcal{M}_a) \rightarrow \mathcal{M}_r <: (X : \mathcal{M}'_a) \rightarrow \mathcal{M}'_r} \\
\\
\frac{\pi : [1; m] \rightarrow [1; n] \quad \forall i \in [1; m], \Gamma; \mathcal{D}_1; \dots; \mathcal{D}_n \blacktriangleright \mathcal{D}_{\pi(i)} <: \mathcal{D}'_i}{\Gamma \blacktriangleright \text{sig } \mathcal{D}_1; \dots; \mathcal{D}_n \text{ end} <: \text{sig } \mathcal{D}'_1; \dots; \mathcal{D}'_m \text{ end}} \quad \frac{\Gamma \triangleright \tau_1 <: \tau_2}{\Gamma \blacktriangleright (\text{val } x_i : \tau_1) <: (\text{val } x_i : \tau_2)} \\
\\
\frac{\Gamma \blacktriangleright \mathcal{M}_1 <: \mathcal{M}_2}{\Gamma \blacktriangleright (\text{module } X_i : \mathcal{M}_1) <: (\text{module } X_i : \mathcal{M}_2)} \quad \frac{\Gamma \triangleright \tau_1 <: \tau_2}{\Gamma \blacktriangleright (\text{type } t_i = \tau_1) <: (\text{type } t_i = \tau_2)} \\
\\
\frac{}{\Gamma \blacktriangleright (\text{type } t_i) <: (\text{type } t_i)} \quad \frac{\Gamma \triangleright t <: \tau}{\Gamma \blacktriangleright (\text{type } t_i) <: (\text{type } t_i = \tau)} \quad \frac{}{\Gamma \blacktriangleright (\text{type } t_i = \tau_1) <: (\text{type } t_i)}
\end{array}$$

Figure 6: Module subtyping rules – $\Gamma \blacktriangleright \mathcal{M} <: \mathcal{M}'$

$$\begin{array}{c}
\frac{\text{module } X_i : \mathcal{M} \in \Gamma}{\Gamma(X_i) = \mathcal{M}} \qquad \frac{\text{module type } \mathcal{X}_i = \mathcal{M} \in \Gamma}{\Gamma(\mathcal{X}_i) = \mathcal{M}} \qquad \frac{\Gamma(P) = \text{sig } \mathcal{S}_1; \text{module } X : \mathcal{M}; \mathcal{S}_2 \text{ end}}{\Gamma(P.X) = \mathcal{M}[n_i \mapsto P.n \mid n_i \in \text{BV}(\mathcal{S}_1)]} \\
\\
\frac{\Gamma(\mathcal{P}) = \text{sig } \mathcal{S}_1; \text{module type } \mathcal{X} = \mathcal{M}; \mathcal{S}_2 \text{ end}}{\Gamma(\mathcal{P}.\mathcal{X}) = \mathcal{M}[n_i \mapsto \mathcal{P}.n \mid n_i \in \text{BV}(\mathcal{S}_1)]} \qquad \frac{\Gamma(\mathcal{P}_f) = (X_i : \mathcal{M}_a) \rightarrow \mathcal{M}_r \quad \Gamma(\mathcal{P}_a) = \mathcal{M}_a}{\Gamma(\mathcal{P}_f(\mathcal{P}_a)) = \mathcal{M}_{r[X=(\mathcal{P}_a \succ \mathcal{M}_a)]}} \\
\\
\frac{\Gamma(\mathcal{P}) = \mathcal{M}}{\Gamma((\mathcal{P} :> \mathcal{M}')) = (\mathcal{M} :> \mathcal{M}')}
\end{array}$$

Figure 7: Lookup rules – $\Gamma(P) = \mathcal{M}$

$$\begin{array}{ll}
\text{force}_\Gamma(\mathcal{M}/_l \mathcal{P}) = \text{force}_\Gamma(\mathcal{M}/_{\textcolor{red}{l}} \mathcal{P}) & \text{resolve}_\Gamma(\mathcal{P}) = \text{resolve}_\Gamma(\Gamma(\mathcal{P}))/\mathcal{P} \\
\text{force}_\Gamma(\mathcal{M}_{[\textcolor{blue}{c}]}) = \text{force}_\Gamma(\mathcal{M}_{[\textcolor{red}{c}]}) & \text{resolve}_\Gamma((= \mathcal{P})) = \text{resolve}_\Gamma(\Gamma(\mathcal{P}))/\mathcal{P} \\
\text{force}_\Gamma((\mathcal{M} :> \mathcal{M}')) = \text{force}_\Gamma((\mathcal{M} : \textcolor{red}{>} \mathcal{M}')) & \text{resolve}_\Gamma(\mathcal{M}^\circ) = \mathcal{M}^\circ \qquad \text{otherwise} \\
\text{force}_\Gamma(\mathcal{M}^\circ) = \mathcal{M}^\circ & \\
\\
\text{normalize}_\Gamma(\mathcal{P}) = \begin{cases} \text{normalize}_\Gamma(\mathcal{P}') & \text{when } \Gamma(\mathcal{P}) = (= \mathcal{P}') \\ \mathcal{P} & \text{otherwise} \end{cases}
\end{array}$$