

Hazel PHI: 10-modules

July 2, 2021

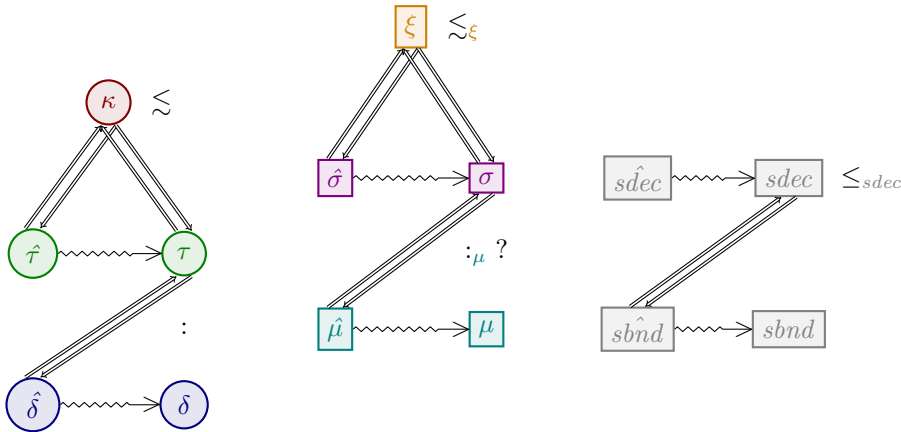
prerequisites

- Hazel PHI: 9-type-aliases-redux
 - github
 - current commit: 4410cd565ce717707e580e44f64868d3175fe2a6
- (optional) Hazel PHI: 1-labeled-tuples
 - github
 - current commit: 0a7d0b53ee7286d03ea3be13a7ac91a86f1c90b1

how to read

800000	kinds	D08000	signature kind
008000	types (constructors)	800080	signatures
000080	terms	008080	modules

notes



external typ/sig/mod syntax not written out yet (waiting for construction dust to settle); patterns not handled yet– will be left till end.

syntax

kind	κ	$::=$	Type	kind of types
			$S(\tau)$	singleton kind
			KHole	kind hole
			$\Pi_{t::\kappa_1}.\kappa_2$	dependent function kind

HType	τ	::=	t bse $\tau_1 \oplus \tau_2$ $[\tau]$ $\lambda t :: \kappa. \tau$ $\tau_1 \tau_2$ $\{lab_1 \hookrightarrow \tau_1, \dots lab_n \hookrightarrow \tau_n\}$ $\mu.lab$ $\langle \rangle$ $\langle \tau \rangle$	type variable base type type binop list type type function type application labelled product type (record) module type projection empty type hole nonempty type hole
base type	bse	::=	\mathbf{Int} \mathbf{Float} \mathbf{Bool}	
HType BinOp	\oplus	::=	\times $+$ \rightarrow	
external expression	$\hat{\delta}$::=	\dots x $\mathbf{signature} \ s = \hat{\sigma} \ \mathbf{in} \ \hat{\delta}$ $\mathbf{module} \ m = \hat{\mu} \ \mathbf{in} \ \hat{\delta}$ $\mathbf{module} \ m :_{\mu} s = \hat{\mu} \ \mathbf{in} \ \hat{\delta}$ $\mathbf{functor} \ \mathbf{something} = \mathbf{something} \ \mathbf{in} \ \hat{\delta}$ $\hat{\mu}.lab$	module term projection
internal expression	δ	::=	\dots x $\mathbf{signature} \ s = \sigma \ \mathbf{in} \ \delta$ $\mathbf{module} \ m :_{\mu} s = \mu \ \mathbf{in} \ \delta$ $\mathbf{functor} \ \mathbf{something} = \mathbf{something} \ \mathbf{in} \ \delta$ $\mu.lab$	module term projection
signature kind	ξ	::=	$\mathbf{SSigKind}(\sigma)$ $\mathbf{SigKHole}$	
external signature	$\hat{\sigma}$::=	s $\{s\hat{decs}\}$ $\Pi_{m :_{\mu} \sigma_1. \hat{\sigma}_2}$ $\langle \rangle^u$ $\langle s \rangle^u$	signature variable structure signature functor signature empty signature hole nonempty signature hole
signature	σ	::=	s $\{sdecs\}$ $\Pi_{m :_{\mu} \sigma_1. \sigma_2}$ $\langle \rangle^u$ $\langle s \rangle^u$ \underline{s}	signature variable structure signature functor signature empty signature hole nonempty signature hole unbound signature variable
module	μ	::=	m $\{sbnds\}$ $\lambda m :_{\mu} \sigma. \mu$ $\mu_1 \ \mu_2$ $\mu.lab$ $\langle \rangle$ $\langle \mu \rangle$	module variable structure functor functor application submodule projection empty module hole nonempty module hole
signature declarations	$sdecs$::=	\cdot $sdec, sdecs$	
signature declaration	$sdec$::=	$\mathbf{type} \ lab :: \kappa$ $\mathbf{val} \ lab : \tau$ $\mathbf{module} \ lab :_{\mu} \sigma$ $\mathbf{functor} \ lab :_{\mu} \sigma$	
structure bindings	$sbnds$::=	\cdot	

		$sbnd, sbnds$
structure binding	$sbnd$	$::=$
		$\text{type } t = \tau$
		$\text{let } x:\tau = \delta$
		$\text{module } m = \mu$
		$\text{module } m:\mu s = \mu$
		$\text{functor } m:\mu s = \mu$

context definitions

$\Delta; , ?; \Gamma; , x:\tau; \Phi; , t::\kappa; \Xi; , m:\mu\sigma; \Psi; , s::\sigma\xi$

declarative statics

scratch

$\Delta; \Phi; \Xi; \Psi; \vdash \Delta; , \Phi; \lesssim_{\kappa_1 \kappa_2}$ κ_1 is a consistent subkind of κ_2

KCSubsumption

$\frac{test}{test}$

$\frac{test}{test}$

$\Delta; , \Phi; , \Xi; , \Psi; \vdash \xi_1 \lesssim_{\xi} \xi_2$ ξ_1 is a consistent sub signature kind of ξ_2

nameMe

$$\frac{\begin{array}{c} \exists sdec_x \in sdec_{s_1} \text{ st } \Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\{sdec_x\}) \lesssim_{\xi} \text{SSigKind}(\{sdec_2\}) \\ \Delta; , \Phi; , \text{type}(\Delta; , \Phi; , \Xi; , \Psi; , sdec_2); \Xi; , \text{submodule}(sdec_2); \Psi; \vdash \{sdec_{s_1}\} \lesssim_{\xi} \{sdec_{s_2}\} \end{array}}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\{sdec_{11}, sdec_{12}, sdec_{13} \text{ as } sdec_{s_1}\}) \lesssim_{\xi} \text{SSigKind}(\{sdec_2, sdec_{s_2}\})}$$

single

$$\frac{\Delta; , \Phi; , \Xi; , \Psi; \vdash sdec_1 \leq_{sdec} sdec_2}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\{sdec_1\}) \lesssim_{\xi} \text{SSigKind}(\{sdec_2\})} \quad \frac{\text{nil}}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\{sdec_{s_1}\}) \lesssim_{\xi} \text{SSigKind}(\{sdec_{s_2}\})}$$

varprop

$$\frac{s::\sigma\xi \in \Psi;}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(s) \lesssim_{\xi} \xi}$$

nameMe?delete?

$$\frac{\sigma_1 \neq s \quad \Delta; , \Phi; , \Xi; , \Psi; \vdash \sigma_1 \leftarrow \text{SSigKind}(\sigma_2)}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\sigma_1) \lesssim_{\xi} \text{SSigKind}(\sigma_2)}$$

funct

$$\frac{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\sigma_{21}) \lesssim_{\xi} \text{SSigKind}(\sigma_{11}) \quad \Delta; , \Phi; , \Xi; , m:\mu\sigma_{11}; \Psi; \vdash \text{SSigKind}(\sigma_{12}) \lesssim_{\xi} \text{SSigKind}(\sigma_{22})}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\Pi_{m:\mu\sigma_{11}}.\sigma_{12}) \lesssim_{\xi} \text{SSigKind}(\Pi_{m:\mu\sigma_{21}}.\sigma_{22})}$$

holes

$$\frac{\Delta; , \Phi; , \Xi; , \Psi; \vdash \langle \emptyset \rangle^u \leftarrow \xi}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\langle \emptyset \rangle^u) \lesssim_{\xi} \xi}$$

neholes

$$\frac{\Delta; , \Phi; , \Xi; , \Psi; \vdash \langle s \rangle^u \leftarrow \xi}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SSigKind}(\langle s \rangle^u) \lesssim_{\xi} \xi}$$

CSubSigKindHoleL

$$\frac{}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \text{SigKHole} \lesssim_{\xi} \xi}$$

CSubSigKindHoleR

$$\frac{}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \xi \lesssim_{\xi} \text{SigKHole}}$$

$\Delta; , \Phi; , \Xi; , \Psi; \vdash \sigma \Rightarrow \xi$ σ synthesizes signature kind ξ

SynSigKndVar

$$\frac{s::\sigma\xi \in \Psi;}{\Delta; , \Phi; , \Xi; , \Psi; \vdash s \Rightarrow \text{SSigKind}(s)}$$

SynSigKndVarFail

$$\frac{s \notin \text{dom}(\Psi;)}{\Delta; , \Phi; , \Xi; , \Psi; \vdash s \Rightarrow \text{SigKHole}}$$

$\{sdec_{s_1}\} \text{ well formed?}$

$$\vdash \{sdec_{s_1}\} \Rightarrow \text{SSigKind}(\{sdec_{s_1}\})$$

SynSigKndSigHole

$$\frac{u::\sigma\xi \in \Delta;}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \langle \emptyset \rangle^u \Rightarrow \xi}$$

SynSigKndSigHole

$$\frac{u::\sigma\xi \in \Delta; \quad \Delta; , \Phi; , \Xi; , \Psi; \vdash s \Rightarrow \xi_1}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \langle s \rangle^u \Rightarrow \xi}$$

$$\frac{\Delta; , \Phi; , \Xi; , m:\mu\sigma_1; \Psi; \vdash \sigma_2 \Rightarrow \xi}{\Delta; , \Phi; , \Xi; , \Psi; \vdash \Pi_{m:\mu\sigma_1}.\sigma_2 \Rightarrow \text{SSigKind}(\Pi_{m:\mu\sigma_1}.\sigma_2)}$$

$\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \sigma \leftarrow \xi$ σ analyzes against signature kind ξ

$$\frac{\text{Sub} \quad \Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \sigma \Rightarrow \xi_1 \quad \Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \xi_1 \lesssim_{\xi} \xi}{\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \sigma \leftarrow \xi}$$

$\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash sdec_1 \leq_{sdec} sdec_2$ $sdec_1$ is a subsdec of $sdec_2$

$$\begin{array}{c} \text{singleType} \\ \frac{\Delta; \Phi; \Xi; \Psi; \vdash \lesssim_{\kappa_1 \kappa_2}}{\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \text{type } lab::\kappa_1 \leq_{sdec} \text{type } lab::\kappa_2} \end{array} \quad \begin{array}{c} \text{singleVa} \\ \frac{\Delta; \Phi; \Xi; \Psi; \vdash \Delta; ; \Phi; ; \Xi; ; \Psi; \equiv \tau_1 \tau_2}{\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \text{val } lab:\tau_1 \leq_{sdec} \text{val } lab:\tau_2} \end{array}$$

$$\begin{array}{c} \text{singleMod} \\ \frac{\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \sigma_1 \leftarrow \text{SSigKind}(\sigma_2)}{\Delta; ; \Phi; ; \Xi; ; \Psi; \vdash \text{module } lab:_\mu \sigma_1 \leq_{sdec} \text{module } lab:_\mu \sigma_2} \end{array}$$

elaboration

$\Gamma; ; \Phi; ; \Xi; \vdash \hat{\delta} \Rightarrow \tau \rightsquigarrow \delta \dashv \Delta;$ $\hat{\delta}$ synthesizes type τ and elaborates to δ with hole context $\Delta;$

$$\begin{array}{c} \dots \\ \text{SynElabLetMod} \\ \frac{\Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Rightarrow \sigma \rightsquigarrow \mu \dashv \Delta;_1 \quad \Gamma; ; \Phi; ; \Xi; , m:_\mu \sigma \vdash \hat{\delta} \Rightarrow \tau \rightsquigarrow \delta \dashv \Delta;_2}{\Gamma; ; \Phi; ; \Xi; \vdash \text{module } m = \hat{\mu} \text{ in } \hat{\delta} \Rightarrow \tau \rightsquigarrow \text{module } m = \mu \text{ in } \delta \dashv \Delta;_1 \cup \Delta;_2} \end{array}$$

$$\begin{array}{c} \text{SynElabLetModAnn} \\ \frac{\Phi; ; \Xi; \vdash \hat{\sigma} \Rightarrow \xi \rightsquigarrow \sigma \dashv \Delta;_1 \quad \Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \leftarrow \sigma \rightsquigarrow \mu \dashv \Delta;_2 \quad \Gamma; ; \Phi; ; \Xi; , m:_\mu \sigma \vdash \hat{\delta} \Rightarrow \tau \rightsquigarrow \delta \dashv \Delta;_3}{\Gamma; ; \Phi; ; \Xi; \vdash \text{module } m:_\mu \hat{\sigma} = \hat{\mu} \text{ in } \hat{\delta} \Rightarrow \tau \rightsquigarrow \text{module } m:_\mu \sigma = \mu \text{ in } \delta \dashv \Delta;_1 \cup \Delta;_2 \cup \Delta;_3} \end{array}$$

$$\begin{array}{c} \text{SynElabModTermPrj} \\ \frac{\Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Rightarrow \sigma \rightsquigarrow \mu \dashv \Delta; \quad \Phi; ; \Xi; \vdash \sigma \Rightarrow \xi}{\Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu}.lab \Rightarrow \tau \rightsquigarrow \mu.lab \dashv \Delta;} \end{array}$$

$\Phi; ; \Xi; \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta;$ $\hat{\tau}$ synthesizes kind κ and elaborates to τ with hole context $\Delta;$

$$\begin{array}{c} \dots \\ \text{SynElabModTypPrj} \\ \frac{\Phi; ; \Xi; \vdash m \Rightarrow \sigma \rightsquigarrow m \dashv \Delta; \quad \text{something} \sigma \kappa}{\Phi; ; \Xi; \vdash m.lab \Rightarrow \kappa \rightsquigarrow m.lab \dashv \Delta;} \end{array}$$

$\Phi; ; \Xi; \vdash \hat{\tau} \leftarrow \kappa \rightsquigarrow \tau \dashv \Delta;$ $\hat{\tau}$ analyzes against kind κ and elaborates to τ with hole context $\Delta;$

$\Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Rightarrow \sigma \rightsquigarrow \mu \dashv \Delta;$ $\hat{\mu}$ synthesizes signature σ and elaborates to μ with hole context $\Delta;$

$$\begin{array}{c} \text{SynElabModVar} \\ \frac{m:_\mu \sigma \in \Xi;}{\Gamma; ; \Phi; ; \Xi; \vdash m \Rightarrow \sigma \rightsquigarrow m \dashv \cdot} \end{array} \quad \begin{array}{c} \text{SynElabModVarFail} \\ \frac{m \notin \text{dom}(\Xi;)}{\Gamma; ; \Phi; ; \Xi; \vdash m \Rightarrow \emptyset \rightsquigarrow \langle m \rangle^u \dashv u:_\mu \emptyset} \end{array}$$

SynElabConsStruct

$$\frac{\Gamma; ; \Phi; ; \Xi; \vdash sbnd \Rightarrow sdec \rightsquigarrow sbnd \dashv \Delta;_1 \quad \Gamma; , \text{val}(sdec); \Phi; , \text{type}(\Delta;_1; \Phi; ; \Xi; , sdec); \Xi; , \text{submodule}(sdec) \vdash \{sbnds\} \Rightarrow \{sdec\} \rightsquigarrow \{sbnds\} \dashv \Delta;_2}{\Gamma; ; \Phi; ; \Xi; \vdash \{sbnd, sbnds\} \Rightarrow \{sdec, sdec\} \rightsquigarrow \{sbnd, sbnds\} \dashv \Delta;_1 \cup \Delta;_2}$$

SynElabNilStruct

$$\Gamma; ; \Phi; ; \Xi; \vdash \{\cdot\} \Rightarrow \{\cdot\} \rightsquigarrow \{\cdot\} \dashv \cdot$$

SynElabEmptyModHole

$$\Gamma; ; \Phi; ; \Xi; \vdash \emptyset^u \Rightarrow \emptyset \rightsquigarrow \emptyset^u \dashv u:_\mu \emptyset$$

SynElabNonemptyModHole

$$\Gamma; ; \Phi; ; \Xi; \vdash \langle m \rangle^u \Rightarrow \emptyset \rightsquigarrow \langle m \rangle^u \dashv u:_\mu \emptyset$$

functor stuff

$\Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Leftarrow \sigma \rightsquigarrow \mu \dashv \Delta; \quad \hat{\mu} \text{ analyzes against signature } \sigma \text{ and elaborates to } \mu \text{ with hole context } \Delta;$

$$\begin{array}{c} \text{AnaElabModSubsumption} \\ \Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Rightarrow \sigma \rightsquigarrow \mu \dashv \Delta; \\ \hline \Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Leftarrow \sigma \rightsquigarrow \mu \dashv \Delta; \end{array}$$

$\Gamma; ; \Phi; ; \Xi; \vdash \hat{s}nd \Rightarrow sdec \rightsquigarrow sbnd \dashv \Delta; \quad \hat{s}nd \text{ synthesizes declaration } sdec \text{ and elaborates to } sbnd \text{ with hole context } \Delta;$

SynElabTypeSbnd

$$\frac{\Phi; ; \Xi; \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta;}{\Gamma; ; \Phi; ; \Xi; \vdash \text{type } t = \hat{\tau} \Rightarrow \text{type } t::\kappa \rightsquigarrow \text{type } t = \tau \dashv \Delta;}$$

SynElabValSbnd

$$\frac{\Phi; ; \Xi; \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta;_1 \quad \Gamma; ; \Phi; ; \Xi; \vdash \hat{\delta} \Leftarrow \tau \rightsquigarrow \delta \dashv \Delta;_2}{\Gamma; ; \Phi; ; \Xi; \vdash \text{let } x:\hat{\tau} = \hat{\delta} \Rightarrow \text{val } x:\tau \rightsquigarrow \text{let } x:\tau = \delta \dashv \Delta;_1 \cup \Delta;_2}$$

SynElabModSbnd

$$\frac{\Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Rightarrow \sigma \rightsquigarrow \mu \dashv \Delta;}{\Gamma; ; \Phi; ; \Xi; \vdash \text{module } m = \hat{\mu} \Rightarrow \text{module } m:_{\mu}\sigma \rightsquigarrow \text{module } m:_{\mu}\sigma = \mu \dashv \Delta;}$$

SynElabModAnnSbnd

$$\frac{\Phi; ; \Xi; \vdash \hat{\sigma} \Rightarrow \xi \rightsquigarrow \sigma_1 \dashv \Delta;_1 \quad \Gamma; ; \Phi; ; \Xi; \vdash \hat{\mu} \Rightarrow \sigma_2 \rightsquigarrow \mu \dashv \Delta;_2 \quad \Phi; ; \Xi; ; \Psi; \vdash \sigma_2 \Leftarrow \xi}{\Gamma; ; \Phi; ; \Xi; \vdash \text{module } m:_{\mu}\hat{\sigma} = \hat{\mu} \Rightarrow \text{module } m:_{\mu}\sigma_1 \rightsquigarrow \text{module } m:_{\mu}\sigma_1 = \mu \dashv \Delta;_1 \cup \Delta;_2}$$

$\Gamma; ; \Phi; ; \Xi; \vdash \hat{s}nd \Leftarrow sdec \rightsquigarrow sbnd \dashv \Delta; \quad \hat{s}nd \text{ analyzes against declaration } sdec \text{ and elaborates to } sbnd \text{ with hole context } \Delta;$

subsump

$$\frac{\Gamma; ; \Phi; ; \Xi; ; l\Psi; \vdash \hat{s}nd \Rightarrow sdec_1 \rightsquigarrow sbnd \dashv \Delta; \quad \Delta; ; \Phi; ; \Xi; ; \Psi; \vdash sdec_1 \leq_{sdec} sdec}{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \hat{s}nd \Leftarrow sdec \rightsquigarrow sbnd \dashv \Delta;}$$

$\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \hat{s}ec \rightsquigarrow sdec \dashv \Delta; \quad \hat{s}ec \text{ elaborates to } sdec \text{ with hole context } \Delta;$

opq

$$\frac{}{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \text{type } lab \rightsquigarrow \text{type } lab::\text{Type} \dashv \cdot}$$

trn

$$\frac{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta;}{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \text{type } lab::\hat{\tau} \rightsquigarrow \text{type } lab::\kappa \dashv \Delta;}$$

val

$$\frac{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta;}{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \text{val } lab:\hat{\tau} \rightsquigarrow \text{val } lab:\tau \dashv \Delta;}$$

mod

$$\frac{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \hat{\sigma} \Rightarrow \xi \rightsquigarrow \sigma \dashv \Delta;}{\Gamma; ; \Phi; ; \Xi; ; \Psi; \vdash \text{module } lab:_{\mu}\hat{\sigma} \rightsquigarrow \text{module } lab:_{\mu}\sigma \dashv \Delta;}$$

we're going to need `HOFunctions` so we don't need to preclude users from typing a functor into a module and vice versa

$\Phi; ; \Xi; ; \Psi; \vdash \hat{\sigma} \Rightarrow \xi \rightsquigarrow \sigma \dashv \Delta; \quad \hat{\sigma} \text{ synthesizes signature kind } \xi \text{ and elaborates to } \sigma \text{ with hole context } \Delta;$

SynSigEmptyHole

$$\frac{}{\Phi; ; \Xi; ; \Psi; \vdash \mathbb{0}^u \Rightarrow \text{SigKHole} \rightsquigarrow \mathbb{0}^u \dashv u::_{\sigma}\text{SigKHole}}$$

SynSigNonEmptyHole

$\Phi; ; \Xi; \vdash \hat{\sigma} \Leftarrow \xi \rightsquigarrow \sigma \dashv \Delta; \quad \hat{\sigma} \text{ analyzes against signature kind } \xi \text{ and elaborates to } \sigma \text{ with hole context } \Delta;$

misc functions

$$\text{val}(sdec) = \begin{cases} lab:\tau & sdec = \text{val } lab:\tau \\ \cdot & \text{otherwise} \end{cases}$$

$$\text{type}(cntxts, sdec) = \begin{cases} lab::\kappa & sdec = \text{type } lab::\kappa \\ \cdot & \text{otherwise} \end{cases}$$

$$\text{submodule}(sdec) = \begin{cases} lab:_{\mu}\sigma & sdec = \text{module } lab:_{\mu}\sigma \\ \cdot & \text{otherwise} \end{cases}$$

$$\begin{aligned} S_{\text{Type}}(\tau) &:= S(\tau) \\ S_{S(\tau_I)}(\tau) &:= S(\tau) \\ S_{\text{KHole}}(\tau) &:= \text{KHole} \end{aligned}$$

theorems

Kind Synthesis Precision

If $\text{cntxts} \vdash \tau \Rightarrow \kappa$ then $\forall \kappa_I. \Delta; \Phi; \Xi; \Psi; \vdash \text{cntxts}::\tau\kappa_I \implies \Delta; \Phi; \Xi; \Psi; \vdash \text{cntxts} \lesssim \kappa\kappa_I$