Hazel PHI: 10-modules

how to read

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

syntax

kind	κ	::=	$\texttt{Type}$	kind of types
		|	$\texttt{S}(\tau)$	singleton kind
		|	$\texttt{KHole}$	kind hole
		|	$\Pi_{t::\kappa_1}.\kappa_2$	dependent function kind
		|	$\Sigma_{t::\kappa_1}.\kappa_2$	dependent product kind
internal HTyp	τ	::=	t	type variable
		|	bse	
		|	$\lambda t :: \kappa.\tau$	
		|	$\tau_1\ \tau_2$	
		|	$\tau_1 \oplus \tau_2$	
		|	$\langle \tau_1, \tau_2 \rangle$	
		|	$\pi_1\ \tau$	
		|	$\pi_2\ \tau$	
		|	$\{lab_1 \hookrightarrow \tau_1, \ldots\ lab_n \hookrightarrow \tau_n\}$	
base type	bse	::=	$\texttt{Int}$	
		|	$\texttt{Float}$	
		|	$\texttt{Bool}$	
HTyp BinOp	$\oplus$	::=	$\times$	
		|	$+$	
		|	$\rightarrow$	
		|	$mod.lab$	
		|	$(\!\|\|\!)$	
		|	$(\!\|\tau\|\!)$	
internal expression	δ	::=	x	
		|	$\texttt{module}\ m = mod\ \texttt{in}\ \delta$	
		|	$\texttt{module}\ m :: sig = mod\ \texttt{in}\ \delta$	
		|	$\texttt{signature}\ s = sig\ \texttt{in}\ \delta$	
		|	$mod.lab$	module term projection
		|	$elided$	
signature	sig	::=	s	signature variable
		|	$\{sdecs\}$	structure signature
		|	$\Pi_{m::sig_1}.sig_2$	functor signature
module	mod	::=	m	module variable
		|	$\{sbnds\}$	structure
		|	$\lambda m :: sig.mod$	functor
		|	$mod_1\ \ mod_2$	functor application
		|	$mod.lab$	submodule projection
structure bindings	$sbnds$	::=	ϵ	
		|	$sbnd, sbnds$	
signature declarations	$sdecs$	::=	ϵ	
		|	$sdec, sdecs$	