single numbers   stone, harper 2006

43.X   PFPL

BinOp	$\oplus$	$::=$	$\texttt{Product} \mid \texttt{Sum} \mid \texttt{Arrow}$
Kind	κ	$::=$	$\texttt{Ty} \mid \texttt{KHole} \mid \texttt{S}(\tau)$
ConstantTypes	c	$::=$	$\texttt{Int} \mid \texttt{Float} \mid \texttt{Bool}$
UserHTyp	$\hat{\tau}$	$::=$	$c \mid \hat{\tau}_1 \oplus \hat{\tau}_2 \mid \texttt{list}(\hat{\tau}) \mid \llparenthesis\rrparenthesis^u \mid \llparenthesis\hat{\tau}\rrparenthesis^u$
InternalHTyp	τ	$::=$	$c \mid \tau_1 \oplus \tau_2 \mid \texttt{list}(\tau) \mid \llparenthesis\rrparenthesis^u \mid \llparenthesis\tau\rrparenthesis^u$ $\mid t$
TypeVars	t		
TypePattern	ρ	$::=$	$t \mid \llparenthesis\rrparenthesis \mid \llparenthesis t\rrparenthesis$
UserExpression	e	$::=$	$\texttt{type}\ \rho\ \texttt{=}\ \hat{\tau}\ \texttt{in}\ e \mid \textit{elided}$
InternalExpression	τ	$::=$	$\texttt{type}\ \rho\ \texttt{=}\ \tau : \kappa\ \texttt{in}\ d \mid \textit{elided}$

$\boxed{\Delta; \Phi \vdash \kappa_1 \lesssim \kappa_2}$ κ_1 is a consistent subkind of κ_2

$$\texttt{KCHoleL}\ \frac{}{\Delta; \Phi \vdash \texttt{KHole} \lesssim \kappa} \qquad \texttt{KCHoleR}\ \frac{}{\Delta; \Phi \vdash \kappa \lesssim \texttt{KHole}} \qquad \texttt{KCRespectEquiv}\ \frac{\Delta; \Phi \vdash \kappa_1 \equiv \kappa_2}{\Delta; \Phi \vdash \kappa_1 \lesssim \kappa_2}$$

$$\texttt{KCSubsumption}\ \frac{\Delta; \Phi \vdash \tau \Leftarrow \texttt{Ty}}{\Delta; \Phi \vdash \texttt{S}(\tau) \lesssim \texttt{Ty}}$$ 7

$\boxed{t\ \texttt{valid}}$ t is a valid type variable

t is valid if it is not a builtin-type or keyword, begins with an alpha char or underscore, and only contains alphanumeric characters, underscores, and primes.

$\boxed{\Delta; \Phi \vdash \kappa\ \texttt{kind}}$ κ forms a kind

$$\texttt{KFTy}\ \frac{}{\Delta; \Phi \vdash \texttt{Ty}\ \texttt{kind}} \qquad \texttt{KFHole}\ \frac{}{\Delta; \Phi \vdash \texttt{KHole}\ \texttt{kind}} \qquad \texttt{KFSing}\ \frac{\Delta; \Phi \vdash \tau \Leftarrow \texttt{Ty}}{\Delta; \Phi \vdash \texttt{S}(\tau)\ \texttt{kind}}$$ 43.2a

$\boxed{\Delta;\Phi \vdash \kappa_1 \equiv \kappa_2}$ κ_1 is equivalent to κ_2

$$\texttt{KERefl}\quad \frac{}{\Delta;\Phi \vdash \kappa \equiv \kappa}
\qquad
\texttt{KESymm}\quad \frac{\Delta;\Phi \vdash \kappa_1 \equiv \kappa_2}{\Delta;\Phi \vdash \kappa_2 \equiv \kappa_1}
\qquad
\texttt{KETrans}\quad \frac{\Delta;\Phi \vdash \kappa_1 \equiv \kappa_2 \qquad \Delta;\Phi \vdash \kappa_2 \equiv \kappa_3}{\Delta;\Phi \vdash \kappa_1 \equiv \kappa_3}$$

$$\texttt{KESingEquiv}\quad \frac{\Delta;\Phi \vdash \tau_1 \equiv \tau_2}{\Delta;\Phi \vdash \mathtt{S}(\tau_1) \equiv \mathtt{S}(\tau_2)}$$

$\boxed{\Delta;\Phi \vdash \tau \Rightarrow \kappa}$ τ synthesizes kind κ

would need to define labelled singletons for KHole

P.687 St, Harp 2006

$$\texttt{KSConst}\quad \frac{}{\Delta;\Phi \vdash c \Rightarrow \mathtt{S}(c)}
\qquad
\texttt{KSVar}\quad \frac{t : \kappa \in \Phi}{\Delta;\Phi \vdash t \Rightarrow \mathtt{S}_{\kappa}(t)}
\qquad
\texttt{KSUVar}\quad \frac{t \notin \mathsf{dom}(\Phi)}{\Delta;\Phi \vdash t \Rightarrow \mathtt{KHole}}$$

$$\texttt{KSBinOp}\quad \frac{\Delta;\Phi \vdash \tau_1 \Leftarrow \overset{\text{Ty}}{\cancel{\mathtt{S}(\tau_1)}} \qquad \Delta;\Phi \vdash \tau_2 \Leftarrow \overset{\text{Ty}}{\cancel{\mathtt{S}(\tau_2)}}}{\Delta;\Phi \vdash \tau_1 \oplus \tau_2 \Rightarrow \mathtt{S}(\tau_1 \oplus \tau_2)}$$

$$\texttt{KSList}\quad \frac{\Delta;\Phi \vdash \tau \Leftarrow \overset{\text{Ty}}{\cancel{\mathtt{S}(\tau)}}}{\Delta;\Phi \vdash \mathtt{list}(\tau) \Rightarrow \mathtt{S}(\mathtt{list}(\tau))}
\qquad
\texttt{KSEHole}\quad \frac{u :: \kappa \in \Delta}{\Delta;\Phi \vdash (\!|\,|\!)^u \Rightarrow \kappa}$$

$$\texttt{KSNEHole}\quad \frac{u :: \kappa \in \Delta \qquad \Delta;\Phi \vdash \tau \Rightarrow \kappa'}{\Delta;\Phi \vdash (\!|\tau|\!)^u \Rightarrow \kappa}$$

$\boxed{\Delta;\Phi \vdash \tau \Leftarrow \kappa}$ τ analyzes against kind κ

$$\texttt{KAASubsume}\quad \frac{\Phi \vdash \tau \Rightarrow \kappa' \qquad \Delta;\Phi \vdash \kappa' \lesssim \kappa}{\Delta;\Phi \vdash \tau \Leftarrow \kappa}
\qquad
\text{KAVar}\quad \frac{t :: \kappa_1 \in \Phi \qquad \kappa_1 \lesssim \kappa}{\Delta;\Phi \vdash t \Leftarrow \kappa}$$

$\boxed{\Delta; \Phi \vdash \tau_1 \equiv \tau_2}$ τ_1 is equivalent to τ_2

KCESymm
$$\frac{\Delta; \Phi \vdash \tau_1 \equiv \tau_2}{\Delta; \Phi \vdash \tau_2 \equiv \tau_1}$$

KCETrans
$$\frac{\Delta; \Phi \vdash \tau_1 \equiv \tau_2 \qquad \Delta; \Phi \vdash \tau_2 \equiv \tau_3}{\Delta; \Phi \vdash \tau_1 \equiv \tau_3}$$

KCESingEquiv
$$\frac{\Delta; \Phi \vdash \tau_1 \Leftarrow \mathtt{S}(\tau_2)}{\Delta; \Phi \vdash \tau_1 \equiv \tau_2}$$

KCERefl
$$\frac{}{\Delta; \Phi \vdash \tau \equiv \tau}$$

~~KCEConst~~
$$\cancel{\frac{}{\Delta; \Phi \vdash c \equiv c}}$$

~~KCEVar~~
$$\cancel{\frac{t : \kappa \in \Phi}{\Delta; \Phi \vdash t \equiv t}}$$

KCEBinOp
$$\frac{\Delta; \Phi \vdash \tau_1 \equiv \tau_2 \qquad \Delta; \Phi \vdash \tau_3 \equiv \tau_4}{\Delta; \Phi \vdash \tau_1 \oplus \tau_3 \equiv \tau_2 \oplus \tau_4}$$

KCEList
$$\frac{\Delta; \Phi \vdash \tau_1 \equiv \tau_2}{\Delta; \Phi \vdash \mathtt{list}(\tau_1) \equiv \mathtt{list}(\tau_2)}$$

KCEEHole
$$\frac{u :: \kappa \in \Delta}{\Delta; \Phi \vdash (\!|\,|\!)^u \equiv (\!|\,|\!)^u}$$

KCENEHole
$$\frac{u :: \kappa \in \Delta \qquad \Delta; \Phi \vdash \tau \Leftarrow \kappa'}{\Delta; \Phi \vdash (\!|\tau|\!)^u \equiv (\!|\tau|\!)^u}$$

$\boxed{\Delta; \Phi \vdash \tau :: \kappa}$ τ is well formed at kind κ

$$\frac{t :: \kappa \in \Phi}{\Delta; \Phi \vdash t :: \kappa}\ 17$$

$$\frac{\Delta; \Phi \vdash \tau :: Ty}{\Delta; \Phi \vdash \mathrm{list}(\tau) :: Ty}$$

$$\frac{\Delta; \Phi \vdash \tau :: \kappa_1 \qquad \Delta; \Phi \vdash \kappa_1 \lesssim \kappa}{\Delta; \Phi \vdash \tau :: \kappa}\ 43.2d$$

$$\boxed{\frac{\Delta; \Phi \vdash \tau :: Ty}{\Delta; \Phi \vdash \tau :: S(\tau)}\ \begin{array}{l}23\\43.2b\end{array}}$$

$$\frac{u :: \kappa \in \Delta}{\Delta; \Phi \vdash (\!|\,|\!)^u :: \kappa}$$

$$\boxed{\frac{\Delta; \Phi \vdash \tau :: \kappa}{\Delta; \Phi \vdash \tau :: S_\kappa(\tau)}\ \begin{array}{l}\text{replaces}\\ 23?\end{array}}$$

$$\frac{}{\Delta; \Phi \vdash c :: Ty}\ 16$$

$$\frac{\Delta; \Phi \vdash u :: \kappa \in \Delta \qquad \Delta; \Phi \vdash \tau :: \kappa_1}{\Delta; \Phi \vdash (\!|\tau|\!)^u :: \kappa}$$

$$\frac{\Delta; \Phi \vdash \tau_1 :: Ty \qquad \Delta; \Phi \vdash \tau_2 :: Ty}{\Delta; \Phi \vdash \tau_1 \oplus \tau_2 :: Ty}$$

$$\frac{t \notin \mathrm{dom}(\Phi)}{\Delta; \Phi \vdash \underset{\sim}{t} :: KHole}$$

$$S_{Ty}(\tau) := S(\tau)$$
$$S_{S(\tau_1)}(\tau) := S(\tau)$$
$$S_{KHole}(\tau) := KHole$$

$\boxed{\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta}$ $\hat{\tau}$ synthesizes kind κ and elaborates to τ

TElabSConst
$$\frac{}{\Phi \vdash c \Rightarrow \mathtt{S}(c) \rightsquigarrow c \dashv \cdot}$$

TElabSBinOp
$$\frac{\Phi \vdash \hat{\tau}_1 \Leftarrow \mathtt{Ty} \rightsquigarrow \tau_1 \dashv \Delta_1 \qquad \Phi \vdash \hat{\tau}_2 \Leftarrow \mathtt{Ty} \rightsquigarrow \tau_2 \dashv \Delta_2}{\Phi \vdash \hat{\tau}_1 \oplus \hat{\tau}_2 \Rightarrow \mathtt{S}(\tau_1 \oplus \tau_2) \rightsquigarrow \tau_1 \oplus \tau_2 \dashv \Delta_1 \cup \Delta_2}$$

TElabSList
$$\frac{\Phi \vdash \hat{\tau} \Leftarrow \mathtt{Ty} \rightsquigarrow \tau \dashv \Delta}{\Phi \vdash \mathtt{list}(\hat{\tau}) \Rightarrow \mathtt{S}(\mathtt{list}(\tau)) \rightsquigarrow \mathtt{list}(\tau) \dashv \Delta}$$

TElabSVar
$$\frac{t : \kappa \in \Phi}{\Phi \vdash t \Rightarrow \mathtt{S}(t) \rightsquigarrow t \dashv \cdot}$$

TElabSUVar
$$\frac{t \notin \mathsf{dom}(\Phi)}{\Phi \vdash t \Rightarrow \mathtt{KHole} \rightsquigarrow (\!| t |\!)^u \dashv u :: \mathtt{KHole}}$$

TElabSHole
$$\frac{}{\Phi \vdash (\!|\,|\!)^u \Rightarrow \mathtt{KHole} \rightsquigarrow (\!|\,|\!)^u \dashv u :: \mathtt{KHole}}$$

TElabSNEHole
$$\frac{\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta}{\Phi \vdash (\!| \hat{\tau} |\!)^u \Rightarrow \mathtt{KHole} \rightsquigarrow (\!| \tau |\!)^u \dashv \Delta, u :: \mathtt{KHole}}$$

$\boxed{\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau \dashv \Delta}$ $\hat{\tau}$ analyzes against kind κ_1 and elaborates to τ

TElabASubsume
$$\frac{\hat{\tau} \neq (\!|\,|\!)^u \qquad \hat{\tau} \neq (\!| \hat{\tau}' |\!)^u \qquad \Phi \vdash \hat{\tau} \Rightarrow \kappa' \rightsquigarrow \tau \dashv \Delta \qquad \Delta; \Phi \vdash \kappa' \lesssim \kappa}{\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau \dashv \Delta}$$

TElabAEHole
$$\frac{}{\Phi \vdash (\!|\,|\!)^u \Leftarrow \kappa \rightsquigarrow (\!|\,|\!)^u \dashv u :: \kappa}$$

TElabANEHole
$$\frac{\Phi \vdash \hat{\tau} \Rightarrow \kappa' \rightsquigarrow \tau \dashv \Delta}{\Phi \vdash (\!| \hat{\tau} |\!)^u \Leftarrow \kappa \rightsquigarrow (\!| \tau |\!)^u \dashv \Delta, u :: \kappa}$$

$\boxed{\Phi_1 \vdash \tau : \kappa \rhd \rho \dashv \Phi_2}$ ρ matches against $\tau : \kappa$ extending Φ if necessary

RESVar
$$\frac{t \;\; \mathtt{valid}}{\Phi \vdash \tau : \kappa \rhd t \dashv \Phi, t :: \kappa}$$

RESEHole
$$\frac{}{\Phi \vdash \tau : \kappa \rhd (\!|\,|\!) \dashv \Phi}$$

RESVarHole
$$\frac{\neg(t \;\; \mathtt{valid})}{\Phi \vdash \tau : \kappa \rhd (\!| t |\!) \dashv \Phi}$$

$\boxed{\Gamma; \Phi \vdash e \Rightarrow \hat{\tau} \rightsquigarrow d \dashv \Delta}$ e synthesizes type τ and elaborates to d

ESDefine

$$\frac{\begin{array}{c}\Phi_1 \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta_1 \\ \Phi_1 \vdash \tau : \kappa \rhd \rho \dashv \Phi_2 \qquad \Gamma; \Phi_2 \vdash e \Rightarrow \tau_1 \rightsquigarrow d \dashv \Delta_2\end{array}}{\Gamma; \Phi_1 \vdash \texttt{type}\ \rho\ \texttt{=}\ \hat{\tau}\ \texttt{in}\ e \Rightarrow \tau_1 \rightsquigarrow \texttt{type}\ \rho\ \texttt{=}\ \tau : \kappa\ \texttt{in}\ d \dashv \Delta_1 \cup \Delta_2}$$

$\boxed{\Delta; \Gamma; \Phi \vdash d : \tau}$ d is assigned type τ

DEDefine

$$\frac{\Phi_1 \vdash \tau_1 : \kappa \rhd \rho \dashv \Phi_2 \qquad \Delta; \Gamma; \Phi_2 \vdash d : \tau_2}{\Delta; \Gamma; \Phi_1 \vdash \texttt{type}\ \rho\ \texttt{=}\ \tau_1 : \kappa\ \texttt{in}\ d : \tau_2}$$

Theorem 1 (Well-Kinded Elaboration)
(1) If $\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta$ *then* $\Delta; \Phi \vdash \tau \Rightarrow \kappa$
(2) If $\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau \dashv \Delta$ *then* $\Delta; \Phi \vdash \tau \Leftarrow \kappa$

This is like the Typed Elaboration theorem in the POPL19 paper.

Theorem 2 (Elaborability)
(1) $\exists \Delta$ *s.t. if* $\Delta; \Phi \vdash \tau \Rightarrow \kappa$ *then* $\exists \hat{\tau}$ *such that* $\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta$
(2) $\exists \Delta$ *s.t. if* $\Delta; \Phi \vdash \tau \Leftarrow \kappa$ *then* $\exists \hat{\tau}$ *such that* $\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau \dashv \Delta$

This is similar but a little different from Elaborability theorem in the POPL19 paper. Choose the Δ that is emitted from elaboration and then there's an $\hat{\tau}$ that elaborates to any of the τ forms. Elaborability and Well-Kinded Elaboration implies we can just rely on the elaboration forms for the premises of any rules that demand kind synthesis/analysis.

Theorem 3 (Type Elaboration Unicity)
(1) If $\Phi \vdash \hat{\tau} \Rightarrow \kappa_1 \rightsquigarrow \tau_1 \dashv \Delta_1$ *and* $\Phi \vdash \hat{\tau} \Rightarrow \kappa_2 \rightsquigarrow \tau_2 \dashv \Delta_2$ *then* $\kappa_1 = \kappa_2$, $\tau_1 = \tau_2$, $\Delta_1 = \Delta_2$
(2) If $\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau_1 \dashv \Delta_1$ *and* $\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau_2 \dashv \Delta_2$ *then* $\tau_1 = \tau_2$, $\Delta_1 = \Delta_2$

This is like the Elaboration Unicity theorem in the POPL19 paper.

Theorem 4 (Kind Synthesis Precision)
If $\Delta; \Phi \vdash \tau \Rightarrow \kappa_1$ *and* $\Delta; \Phi \vdash \tau \Leftarrow \kappa_2$ *then* $\Delta; \Phi \vdash \kappa_1 \lesssim \kappa_2$

Kind Synthesis Precision says that synthesis finds the most precise kappa possible for a given input type. This is somewhat trivial, but interesting to note because it means we can expect singletons wherever possible.

Theorem 5.1 (Kind Analysis Soundness)

If $\Delta; \Phi \vdash \tau \Leftarrow \kappa$ then $\Delta; \Phi \vdash \tau :: \kappa$

Theorem 5.2 (Kind Analysis Completeness)

If $\Delta; \Phi \vdash \tau :: \kappa$ then $\Delta; \Phi \vdash \tau \Leftarrow \kappa$

5.1 Induction on complexity

Base cases

$c :: Ty$ & $\kappa = Ty \Rightarrow Ty \lesssim Ty \Rightarrow c :: \kappa$
or

$c \Leftarrow \kappa \Rightarrow c \Rightarrow \kappa'$ & $\kappa' \lesssim \kappa \Rightarrow \kappa' = S(c) \Rightarrow S(c) \lesssim \kappa \Rightarrow S(c) \equiv \kappa \Rightarrow c :: S(c)$ & $S(c) \leq \kappa \Rightarrow c :: \kappa$

$t \Leftarrow \kappa \Rightarrow t \Rightarrow \kappa'$ & $\kappa' \lesssim \kappa \Rightarrow \kappa' = S(t)$ & $t :: \kappa'' \in \Phi \Rightarrow t :: \kappa'' \Rightarrow \kappa = Ty \Rightarrow t :: Ty \Rightarrow t :: \kappa$
or
$\kappa \equiv S(t) \Rightarrow \kappa \lesssim S(t) \Rightarrow$

$\underset{\sim}{t} \Leftarrow \kappa$

$()^u \Leftarrow \kappa$