$$
\begin{array}{rcl}
\text{BinOp} & \oplus & ::= \ \texttt{Product} \mid \texttt{Sum} \mid \texttt{Arrow} \\
\text{Kind} & \kappa & ::= \ \texttt{Ty} \mid \texttt{KHole} \mid \texttt{S}(\tau) \\
\text{ConstantTypes} & c & ::= \ \texttt{Int} \mid \texttt{Float} \mid \texttt{Bool} \\
\text{UserHTyp} & \hat{\tau} & ::= \ c \mid \hat{\tau}_1 \oplus \hat{\tau}_2 \mid \texttt{list}(\hat{\tau}) \mid (\!|\,|\!)^u \mid (\!|\hat{\tau}|\!)^u \\
\text{InternalHTyp} & \tau & ::= \ c \mid \tau_1 \oplus \tau_2 \mid \texttt{list}(\tau) \mid (\!|\,|\!)^u \mid (\!|\tau|\!)^u \ \ |(\!|\, t \,|\!)^u \\
\text{TypeVars} & t & \\
\text{TypePattern} & \rho & ::= \ t \mid (\!|\,|\!) \mid (\!|t|\!) \\
\text{UserExpression} & e & ::= \ \texttt{type } \rho \texttt{ = } \hat{\tau} \texttt{ in } e \mid elided \\
\text{InternalExpression} & \tau & ::= \ \texttt{type } \rho \texttt{ = } \tau : \kappa \texttt{ in } d \mid elided
\end{array}
$$

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

KCHoleL
$$\overline{\Delta; \Phi \vdash \texttt{KHole} \lesssim \kappa}$$

KCHoleR
$$\overline{\Delta; \Phi \vdash \kappa \lesssim \texttt{KHole}}$$

KCRespectEquiv
$$\frac{\Delta; \Phi \vdash \kappa_1 \equiv \kappa_2}{\Delta; \Phi \vdash \kappa_1 \lesssim \kappa_2}$$

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

$\boxed{t \ \texttt{valid}}$ 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

KFTy
$$\overline{\Delta; \Phi \vdash \texttt{Ty} \ \texttt{kind}}$$

KFHole
$$\overline{\Delta; \Phi \vdash \texttt{KHole} \ \texttt{kind}}$$

KFSing
$$\frac{\Delta; \Phi \vdash \tau \overset{\leftrightarrow}{\cdot} \texttt{Ty}}{\Delta; \Phi \vdash \texttt{S}(\tau) \ \texttt{kind}} \ \textbf{43.2a}$$

1

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

KERefl
$$\Delta; \Phi \vdash \kappa \equiv \kappa$$

KESymm
$$\frac{\Delta; \Phi \vdash \kappa_1 \equiv \kappa_2}{\Delta; \Phi \vdash \kappa_2 \equiv \kappa_1}$$

KETrans
$$\frac{\Delta; \Phi \vdash \kappa_1 \equiv \kappa_2 \qquad \Delta; \Phi \vdash \kappa_2 \equiv \kappa_3}{\Delta; \Phi \vdash \kappa_1 \equiv \kappa_3}$$

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

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

would need to define labelled singletons for KHole
p.697 St, Harp 2006

KSConst
$$\Delta; \Phi \vdash c \Rightarrow \mathtt{S}(c)$$

KSVar
$$\frac{t : \kappa \in \Phi}{\Delta; \Phi \vdash t \Rightarrow \mathtt{S}(t)} \quad _{\kappa}$$

KSUVar
$$\frac{t \notin \mathsf{dom}(\Phi) \quad u :: \kappa \in \Delta}{\Delta; \Phi \vdash \cancel{t} \; t^u \Rightarrow \cancel{\mathtt{KHole}} \; \kappa}$$

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

KSList
$$\frac{\Delta; \Phi \vdash \tau \Leftarrow \cancel{\mathtt{S}(\tau)} \; Ty}{\Delta; \Phi \vdash \mathtt{list}(\tau) \Rightarrow \mathtt{S}(\mathtt{list}(\tau))}$$

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

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

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

KAASubsume
$$\frac{\Phi \vdash \tau \Rightarrow \kappa' \qquad \Delta; \Phi \vdash \kappa' \lesssim \kappa}{\Delta; \Phi \vdash \tau \Leftarrow \kappa}$$

KAVar
$$\frac{t :: k_1 \in \Phi \qquad k_1 \lesssim k}{\Delta; \Phi \vdash t \Leftarrow k}$$

$$\boxed{\Delta; \Phi \vdash \tau_1 \equiv \tau_2} \quad \tau_1 \text{ is equivalent to } \tau_2$$

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
$$\frac{\Delta; \Phi \vdash \tau_1 \leftrightsquigarrow S_\kappa(\tau_2)}{\Delta; \Phi \vdash \tau_1 \equiv \tau_2}$$

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

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

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

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

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

* if $u :: k \notin \Delta$
KCERefl saus OK

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

* sim

$$\boxed{\Delta; \Phi \vdash \tau :: k} \quad \tau \text{ is well formed at kind } k$$

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

$$\frac{\Delta; \Phi \vdash \tau :: T_y}{\Delta; \Phi \vdash \text{list}(\tau) :: T_y}$$

$$\frac{\Delta; \Phi \vdash \tau :: k_1 \quad \Delta; \Phi \vdash k_1 \leqslant k}{\Delta; \Phi \vdash \tau :: k} \; 43.2d$$

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

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

$$\frac{\Delta; \Phi \vdash \tau :: k}{\Delta; \Phi \vdash \tau :: S_\kappa(\tau)} \; \text{replaces } 23?$$

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

$$\frac{\Delta; \Phi \vdash u :: k \in \Delta \quad \Delta; \Phi \vdash \tau :: k_1}{\Delta; \Phi \vdash (\!|\tau|\!)^u :: k}$$

$$\frac{\Delta; \Phi \vdash \tau_1 :: T_y \quad \Delta; \Phi \vdash \tau_2 :: T_y}{\Delta; \Phi \vdash \tau_1 \oplus \tau_2 :: T_y}$$

$$\frac{t \notin \text{dom}(\Phi) \quad u :: k \in \Delta}{\Delta; \Phi \vdash (\!|t|\!)^u :: k}$$

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

3

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

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

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

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

RESVar
$$\frac{t \text{ valid}}{\Phi \vdash \tau : \kappa \triangleright t \dashv \Phi, t :: \kappa}$$

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

RESVarHole
$$\frac{\neg(t \text{ valid})}{\Phi \vdash \tau : \kappa \triangleright (\!|t|\!) \dashv \Phi}$$

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

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

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

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 $\Delta; \Phi \vdash \tau \Leftarrow k$ then $\Delta; \Phi \vdash \tau :: k$

Theorem 5.2 (Kind Analysis Completeness)

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

5.1 Induction on complexity

Base cases

$$c :: T_g \ \& \ k > T_g \Rightarrow T_g \lesssim T_g \Rightarrow c :: k$$
$$\text{or}$$

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

$t \Leftarrow k \Rightarrow t \Rightarrow k' \ \& \ k' \lesssim k \Rightarrow k' = S(t) \ \& \ t :: k'' \& \lesssim \Rightarrow t :: k'' \Rightarrow k = T_g \Rightarrow t :: T_g \Rightarrow t :: k$

$$\quad \underset{\sim}{t} \Leftarrow k$$
$$\text{or}$$

$$k \lesssim S(t) \Rightarrow S(t) \lesssim k \Rightarrow t :: S(t)$$

$\langle D^u \Leftarrow k$

$t :: S_{k''}(t) \quad S_{k''}(t) \lesssim k$
$t :: k$