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

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

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

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

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

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

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

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

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

$\boxed{\Delta; \Phi \vdash \tau : \kappa}$ τ is assigned non-singleton kind κ

$$\texttt{KAConst}\ \frac{}{\Delta; \Phi \vdash c : \texttt{Ty}} \qquad \texttt{KAVar}\ \frac{t : \kappa_1 \in \Phi \qquad \Delta; \Phi \vdash \kappa_1 \Rrightarrow \lceil \kappa_2 \rceil}{\Delta; \Phi \vdash t : \kappa_2}$$

$$\texttt{KABinOp}\ \frac{\Delta; \Phi \vdash \tau_1 : \texttt{Ty} \qquad \Delta; \Phi \vdash \tau_2 : \texttt{Ty}}{\Delta; \Phi \vdash \tau_1 \oplus \tau_2 : \texttt{Ty}} \qquad \texttt{KAList}\ \frac{\Delta; \Phi \vdash \tau : \texttt{Ty}}{\Delta; \Phi \vdash \texttt{list(}\tau\texttt{)} : \texttt{Ty}}$$

$$\texttt{KAEHole}\ \frac{u :: \kappa[\Phi'] \in \Delta \qquad \Delta; \Phi \vdash \sigma : \Phi'}{\Delta; \Phi \vdash (\!|\!)^u_\sigma : \kappa}$$

$$\texttt{KANEHole}\ \frac{\Delta; \Phi \vdash \tau : \kappa' \qquad u :: \kappa[\Phi'] \in \Delta \qquad \Delta; \Phi \vdash \sigma : \Phi'}{\Delta; \Phi \vdash (\!|\tau|\!)^u_\sigma : \kappa}$$

$\boxed{\Delta; \Phi \vdash \tau :_s \kappa}$ τ is assigned kind (with singleton affinity) κ

$$\texttt{KAConst}\quad \frac{}{\Delta; \Phi \vdash c :_s \mathtt{S}(c)}$$

$$\texttt{KAVar}\quad \frac{t : \kappa \in \Phi}{\Delta; \Phi \vdash t :_s \kappa}$$

$$\texttt{KABinOp}\quad \frac{\Delta; \Phi \vdash \tau_1 :_s \mathtt{S}(\tau_1') \qquad \Delta; \Phi \vdash \tau_2 :_s \mathtt{S}(\tau_2')}{\Delta; \Phi \vdash \tau_1 \oplus \tau_2 :_s \mathtt{S}(\tau_1' \oplus \tau_2')}$$

$$\texttt{KAList}\quad \frac{\Delta; \Phi \vdash \tau :_s \mathtt{S}(\tau')}{\Delta; \Phi \vdash \mathtt{list}(\tau) :_s \mathtt{S}(\mathtt{list}(\tau'))}$$

$$\texttt{KAEHole}\quad \frac{u :: \kappa[\Phi'] \in \Delta \qquad \Delta; \Phi \vdash \sigma :_s \Phi'}{\Delta; \Phi \vdash (\!|\,|\!)^u_\sigma :_s \kappa}$$

$$\texttt{KANEHole}\quad \frac{\Delta; \Phi \vdash \tau :_s \kappa' \qquad u :: \kappa[\Phi'] \in \Delta \qquad \Delta; \Phi \vdash \sigma :_s \Phi'}{\Delta; \Phi \vdash (\!|\tau|\!)^u_\sigma :_s \kappa}$$

$\boxed{\Delta; \Phi \vdash \kappa_1 \Rrightarrow \lceil \kappa_2 \rceil}$ κ_1 is unrecognized to consistent superkind κ_2

$$\texttt{KUSing}\quad \frac{\Delta; \Phi \vdash \tau :_s \mathtt{S}(\tau')}{\Delta; \Phi \vdash \mathtt{S}(\tau') \Rrightarrow \lceil \mathtt{Ty} \rceil}$$

$$\texttt{KUTy}\quad \frac{}{\Delta; \Phi \vdash \mathtt{Ty} \Rrightarrow \lceil \mathtt{Ty} \rceil}$$

$$\texttt{KUHole}\quad \frac{}{\Delta; \Phi \vdash \mathtt{KHole} \Rrightarrow \lceil \mathtt{KHole} \rceil}$$

$\boxed{\Delta; \Phi \vdash \tau_1 \equiv \tau_2 : \kappa}$ τ_1 is equivalent to τ_2 and has kind κ_2

$$\texttt{KCERefl}\quad \frac{\Delta; \Phi \vdash \tau : \kappa}{\Delta; \Phi \vdash \tau \equiv \tau : \kappa}$$

$$\texttt{KCESymm}\quad \frac{\Delta; \Phi \vdash \tau_1 \equiv \tau_2 : \kappa}{\Delta; \Phi \vdash \tau_2 \equiv \tau_1 : \kappa}$$

$$\texttt{KCETrans}\quad \frac{\Delta; \Phi \vdash \tau_1 \equiv \tau_2 : \kappa \qquad \Delta; \Phi \vdash \tau_2 \equiv \tau_3 : \kappa}{\Delta; \Phi \vdash \tau_1 \equiv \tau_3 : \kappa}$$

$$\texttt{KCESingEquiv}\quad \frac{\Delta; \Phi \vdash \tau_1 :_s \mathtt{S}(\tau_2)}{\Delta; \Phi \vdash \tau_1 \equiv \tau_2 : \mathtt{Ty}}$$

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

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

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

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

$$\texttt{TElabSVar}\quad \frac{t : \kappa \in \Phi}{\Phi \vdash t \Rightarrow \kappa \rightsquigarrow t \dashv \cdot}$$

$$\texttt{TElabSUVar}\quad \frac{t \notin \Phi}{\Phi \vdash t \Rightarrow \texttt{KHole} \rightsquigarrow (\!|t|\!)^u_{\mathsf{id}(\Phi)} \dashv u :: (\!|\,|\!)[\Phi]}$$

$$\texttt{TElabSHole}\quad \frac{}{\Phi \vdash (\!|\,|\!)^u \Rightarrow \texttt{KHole} \rightsquigarrow (\!|\,|\!)^u_{\mathsf{id}(\Phi)} \dashv u :: (\!|\,|\!)[\Phi]}$$

$$\texttt{TElabSNEHole}\quad \frac{\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta}{\Phi \vdash (\!|\hat{\tau}|\!)^u \Rightarrow \texttt{KHole} \rightsquigarrow (\!|\tau|\!)^u_{\mathsf{id}(\Phi)} \dashv \Delta, u :: (\!|\,|\!)[\Phi]}$$

$\boxed{\Phi \vdash \hat{\tau} \Leftarrow \kappa_1 \rightsquigarrow \tau : \kappa_2 \dashv \Delta}$ $\hat{\tau}$ analyzes against kind κ_1 and elaborates to τ of consistent subkind κ_2

TElabASubsume

$$\frac{\hat{\tau} \neq t \text{ where } t \notin \Phi \quad \hat{\tau} \neq (\!|\!)^u \quad \hat{\tau} \neq (\!|\hat{\tau}'|\!)^u \quad \Phi \vdash \hat{\tau} \Rightarrow \kappa' \rightsquigarrow \tau \dashv \Delta \quad \Delta; \Phi \vdash \kappa' \lesssim \kappa}{\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau : \kappa' \dashv \Delta}$$

TElabAUVar

$$\frac{t \notin \Phi}{\Phi \vdash t \Leftarrow \texttt{KHole} \rightsquigarrow (\!|t|\!)^u_{\mathsf{id}(\Phi)} : \texttt{KHole} \dashv u :: (\!|\!)[\Phi]}$$

TElabAEHole

$$\frac{}{\Phi \vdash (\!|\!)^u \Leftarrow \kappa \rightsquigarrow (\!|\!)^u_{\mathsf{id}(\Phi)} : \kappa \dashv u :: \kappa[\Phi]}$$

TElabANEHole

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

$\boxed{\Delta_1; \Phi_1 \vdash \tau \rhd \hat{\rho} \dashv \Phi_2; \Delta_2}$ $\hat{\rho}$ analyzes against τ yielding new tyvar and hole bindings

RESVar

$$\frac{t \ \texttt{valid} \quad \Delta; \Phi \vdash \tau :_s \kappa}{\Delta; \Phi \vdash \tau \rhd t \dashv t :: \kappa; \cdot}$$

RESEHole

$$\frac{}{\Delta; \Phi \vdash \tau \rhd (\!|\!)^u \dashv \cdot; u :: (\!|\!)[\Phi]}$$

RESVarHole

$$\frac{\neg(t \ \texttt{valid})}{\Delta; \Phi \vdash \tau \rhd (\!|t|\!)^u \dashv \cdot; u :: (\!|\!)[\Phi]}$$

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

ESDefine

$$\frac{\Phi_1 \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta_1 \qquad \Delta_1; \Phi_1 \vdash \tau \rhd \hat{\rho} \dashv \Phi_2; \Delta_2 \quad \Gamma; \Phi_1 \cup \Phi_2 \vdash e \Rightarrow \tau_1 \rightsquigarrow d \dashv \Delta_3}{\Gamma; \Phi_1 \vdash \texttt{type}\ \hat{\rho}\ \texttt{=}\ \hat{\tau}\ \texttt{in}\ e \Rightarrow \tau_1 \rightsquigarrow \texttt{type}\ \hat{\rho}\ \texttt{=}\ \tau\ \texttt{in}\ d \dashv \Delta_1 \cup \Delta_2 \cup \Delta_3}$$

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

DEDefine

$$\frac{\Delta; \Phi_1 \vdash \tau \rhd \hat{\rho} \dashv \Phi_2; \Delta \qquad \Delta; \Gamma; \Phi_1 \cup \Phi_2 \vdash d : \tau}{\Delta; \Gamma; \Phi_1 \vdash \texttt{type } \hat{\rho} \texttt{ = } \tau \texttt{ in } d : \tau}$$

Theorem 1 (Well-Kinded Elaboration)
(1) If $\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta$ and $\Delta; \Phi \vdash \tau :_s \kappa$
(2) If $\Phi \vdash \hat{\tau} \Leftarrow \kappa_1 \rightsquigarrow \tau : \kappa_2 \dashv \Delta$ then $\Delta; \Phi \vdash \kappa_2 \lesssim \kappa_1$ and $\Delta; \Phi \vdash \tau :_s \kappa_2$

This is like the Typed Elaboration theorem in the POPL19 paper. Note that analysis produces the most precise kind (preferring singletons) even when analyzing against unprecise `Ty`. This is because the relevant rules are handled by synthesis.

Theorem 2 (Type Elaboration Singleton Affinity)
(1) If $\Phi \vdash \hat{\tau} \Rightarrow \kappa \rightsquigarrow \tau \dashv \Delta$ then $\kappa = \texttt{S}(\tau')$ or $\kappa = \texttt{KHole}$.
(2) If $\Phi \vdash \hat{\tau} \Leftarrow \kappa_1 \rightsquigarrow \tau : \kappa_2 \dashv \Delta$ then $\kappa_2 = \texttt{S}(\tau')$ or $\kappa_2 = \texttt{KHole}$.

Type Elaboration always elaborates to the most precise kind – it never elaborates to `Ty` directly even when analyzed against `Ty`.

Theorem 3 (Kind Assignment Unicity)
(1) If $\Delta; \Phi \vdash \tau : \kappa$ and $\Delta; \Phi \vdash \tau : \kappa'$ then $\kappa = \kappa'$
(2) If $\Delta; \Phi \vdash \tau :_s \kappa$ and $\Delta; \Phi \vdash \tau :_s \kappa'$ then $\kappa = \kappa'$

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

Theorem 4 (Kind Assignment Ty Affinity)
If $\Delta; \Phi \vdash \tau : \kappa$ then $\Delta; \Phi \vdash \kappa \Rrightarrow \lceil \kappa \rceil$

Kind assignment assigns `Ty` rather than a singleton. When the kind needs to be a singleton rather than `Ty`, instead use: $\Delta; \Phi \vdash \tau :_s \kappa$.

Theorem 5 (Kind AssignmentS Singleton Affinity)
If $\Delta; \Phi \vdash \tau :_s \kappa$ then $\kappa = \texttt{S}(\tau')$ or $\kappa = \texttt{KHole}$.

Kind assignment assigns a singleton rather than `Ty`. When the kind needs to be a `Ty`, instead use: $\Delta; \Phi \vdash \tau : \kappa$.

Theorem 6 (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 $\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau_1 : \kappa_1 \dashv \Delta_1$ *and* $\Phi \vdash \hat{\tau} \Leftarrow \kappa \rightsquigarrow \tau_2 : \kappa_2 \dashv \Delta_2$ *then* $\tau_1 = \tau_2$, $\kappa_1 = \kappa_2$, $\Delta_1 = \Delta_2$

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

Theorem 7 (Kind Assignment Consistency)
If $\Delta; \Phi \vdash \tau :_s \kappa_1$ *and* $\Delta; \Phi \vdash \tau : \kappa_2$ *then* $\Delta; \Phi \vdash \kappa_1 \lesssim \kappa_2$ *and* κ_1, κ_2 *are either both* `KHole` *or neither is.*

Kind Assignment Consistency says that both the singleton form and non-singleton form of kind assignment differ only in that the singleton form is more precise. But both must agree on holes.