1 Grammar

$$
\begin{array}{rcll}
e & ::= & x & \textit{exprs.} \\
& | & r & \\
& | & \lambda x : \tau . e & \\
& | & e\ e & \\
& | & e.\pi & \\
\\
\hat{e} & ::= & x & \textit{labelled exprs.} \\
& | & r & \\
& | & \lambda x : \hat{\tau} . \hat{e} & \\
& | & \hat{e}\ \hat{e} & \\
& | & \hat{e}.\pi & \\
& | & \texttt{import}(\varepsilon)\ x = \hat{e}\ \texttt{in}\ e & \\
\\
v & ::= & r & \textit{values.} \\
& | & \lambda x : \tau . e & \\
\\
\hat{v} & ::= & r & \textit{labelled values} \\
& | & \lambda x : \hat{\tau} . \hat{e} &
\end{array}
$$

$$
\begin{array}{rcll}
\varepsilon & ::= & \{\overline{r.\pi}\} & \textit{effects} \\
\\
\tau & ::= & \{\bar{r}\} & \textit{types} \\
& | & \tau \to \tau & \\
\\
\hat{\tau} & ::= & \{\bar{r}\} & \textit{labelled types} \\
& | & \hat{\tau} \to_{\varepsilon} \hat{\tau} & \\
\\
\Gamma & ::= & \varnothing & \textit{type ctx.} \\
& | & \Gamma, x : \tau & \\
\\
\hat{\Gamma} & ::= & \varnothing & \textit{labelled type ctx.} \\
& | & \hat{\Gamma}, x : \hat{\tau} &
\end{array}
$$

2 Functions

Definition ($\texttt{annot} :: \boldsymbol{\tau} \times \boldsymbol{\varepsilon} \to \hat{\boldsymbol{\tau}}$)

1. $\texttt{annot}(\{\bar{r}\}, \_) = \{\bar{r}\}$
2. $\texttt{annot}(\tau_1 \to \tau_2, \varepsilon) = \texttt{annot}(\tau_1, \varepsilon) \to_{\varepsilon} \texttt{annot}(\tau_2, \varepsilon)$

Definition ($\texttt{annot} :: \mathsf{e} \times \boldsymbol{\varepsilon} \to \hat{\mathsf{e}}$)

1. $\texttt{annot}(x, \_) = e$
2. $\texttt{annot}(r, \_) = r$
3. $\texttt{annot}(e_1 e_2, \varepsilon) = \texttt{annot}(e_1)\texttt{annot}(e_2)$
4. $\texttt{annot}(e.\pi, \varepsilon) = \texttt{annot}(e).\pi$
5. $\texttt{annot}(\lambda x : \tau.e, \varepsilon) = \lambda x : \texttt{annot}(\tau, \varepsilon).\texttt{annot}(e, \varepsilon)$

Definition ($\texttt{annot} :: \Gamma \times \boldsymbol{\varepsilon} \to \hat{\Gamma}$)

1. $\texttt{annot}(\varnothing, \_) = \varnothing$
2. $\texttt{annot}((\Gamma, x : \tau), \varepsilon) = \texttt{annot}(\Gamma, \varepsilon), x : \texttt{annot}(\tau, \varepsilon)$

Definition ($\texttt{erase} :: \hat{\boldsymbol{\tau}} \to \boldsymbol{\tau}$)

1. $\texttt{erase}(\{\bar{r}\}, \_) = \{\bar{r}\}$
2. $\texttt{erase}(\hat{\tau}_1 \to_{\varepsilon} \hat{\tau}_2) = \texttt{erase}(\hat{\tau}_1) \to \texttt{erase}(\hat{\tau}_2)$

Definition ($\texttt{erase} :: \hat{\mathsf{e}} \to \mathsf{e}$)

1. $\texttt{erase}(x) = x$
2. $\texttt{erase}(r) = r$
3. $\texttt{erase}(e_1 e_2) = \texttt{erase}(e_1)\texttt{erase}(e_2)$
4. $\texttt{erase}(e.\pi) = \texttt{erase}(e).\pi$
5. $\texttt{erase}(\lambda x : \hat{\tau}.\hat{e}) = \lambda x : \texttt{erase}(\hat{\tau}).\texttt{erase}(\hat{e})$

Definition ($\texttt{effects} :: \boldsymbol{\tau} \rightarrow \boldsymbol{\varepsilon}$)

1. $\texttt{effects}(\{\bar{r}\}) = \{r.\pi \mid r \in \bar{r}, \pi \in \Pi\}$
2. $\texttt{effects}(\hat{\tau}_1 \rightarrow_\varepsilon \hat{\tau}_2) = \texttt{ho-effects}(\hat{\tau}_1) \cup \varepsilon \cup \texttt{effects}(\hat{\tau}_2)$

This function computes those effects:

- Directly invoked by a function of type $\hat{\tau}$.
- Captured by functions created/returned by $\hat{\tau}$.

Definition ($\texttt{ho-effects} :: \boldsymbol{\tau} \rightarrow \boldsymbol{\varepsilon}$)

1. $\texttt{ho-effects}(\{\bar{r}\}) = \varnothing$
2. $\texttt{ho-effects}(\hat{\tau}_1 \rightarrow_\varepsilon \hat{\tau}_2) = \texttt{effects}(\hat{\tau}_1) \cup \texttt{ho-effects}(\hat{\tau}_2)$

This function computes those effects:

- Captured by functions passed into $\hat{\tau}$.

Examples

Suppose a is a base type that captures no effects.

Consider $\hat{\tau} = (a \rightarrow_b (a \rightarrow_c a)) \rightarrow_d (a \rightarrow_e a)$.
$\texttt{effects}(\hat{\tau}) = \{d, e\}$
$\texttt{ho-effects}(\hat{\tau}) = \{b, c\}$

Consider $\hat{\tau} = ((a \rightarrow_b a) \rightarrow_c (a \rightarrow_d a)) \rightarrow_e ((a \rightarrow_f a) \rightarrow_g (a \rightarrow_h a))$
$\texttt{effects}(\hat{\tau}) = \{e, b, g, h\}$
$\texttt{ho-effects}(\hat{\tau}) = \{c, d, f\}$

3 Static Rules

$\boxed{\Gamma \vdash e : \tau}$

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ (\text{T-Var}) \qquad \frac{}{\Gamma, r : \{r\} \vdash r : \{r\}}\ (\text{T-Resource}) \qquad \frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . e : \tau_1 \rightarrow \tau_2}\ (\text{T-Abs})$$

$$\frac{\Gamma \vdash e_1 : \tau_2 \rightarrow \tau_3 \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash e_1\ e_2 : \tau_3}\ (\text{T-App}) \qquad \frac{\Gamma \vdash e : \{\bar{r}\} \quad \forall r \in \bar{r} \mid r \in R \quad \pi \in \Pi}{\Gamma \vdash e.\pi : \texttt{Unit}}\ (\text{T-OperCall})$$

$\boxed{\hat{\Gamma} \vdash \hat{e} : \hat{\tau} \texttt{ with } \varepsilon}$

$$\frac{}{\hat{\Gamma}, x : \tau \vdash x : \tau \texttt{ with } \varnothing}\ (\varepsilon\text{-Var}) \qquad \frac{}{\hat{\Gamma}, r : \{r\} \vdash r : \{r\} \texttt{ with } \varnothing}\ (\varepsilon\text{-Resource})$$

$$\frac{\hat{\Gamma}, x : \hat{\tau}_2 \vdash \hat{e} : \hat{\tau}_3 \texttt{ with } \varepsilon_3}{\hat{\Gamma} \vdash \lambda x : \tau_2 . \hat{e} : \hat{\tau}_2 \rightarrow_{\varepsilon_3} \hat{\tau}_3 \texttt{ with } \varnothing}\ (\varepsilon\text{-Abs}) \qquad \frac{\hat{\Gamma} \vdash \hat{e}_1 : \hat{\tau}_2 \rightarrow_\varepsilon \hat{\tau}_3 \texttt{ with } \varepsilon_1 \quad \hat{\Gamma} \vdash \hat{e}_2 : \hat{\tau}_2 \texttt{ with } \varepsilon_2}{\hat{\Gamma} \vdash \hat{e}_1 \hat{e}_2 : \hat{\tau}_3 \texttt{ with } \varepsilon_1 \cup \varepsilon_2 \cup \varepsilon}\ (\varepsilon\text{-App})$$

$$\frac{\hat{\Gamma} \vdash \hat{e} : \{\bar{r}\} \quad \forall r \in \bar{r} \mid r : \{r\} \in \Gamma \quad \pi \in \Pi}{\hat{\Gamma} \vdash \hat{e}.\pi : \texttt{Unit with } \{\bar{r}.\pi\}}\ (\varepsilon\text{-OperCall}) \qquad \frac{\hat{\Gamma} \vdash e : \tau \texttt{ with } \varepsilon \quad \tau' <: \tau \quad \varepsilon' \subseteq \varepsilon}{\hat{\Gamma} \vdash e : \tau' \texttt{ with } \varepsilon'}\ (\varepsilon\text{-Subsume})$$

$$\frac{\begin{array}{cc} \hat{\Gamma} \vdash \hat{e} : \hat{\tau} \texttt{ with } \varepsilon_1 & \varepsilon = \texttt{effects}(\hat{\tau}) \\ \texttt{ho-safe}(\hat{\tau}, \varepsilon) & x : \texttt{erase}(\hat{\tau}) \vdash e : \tau \end{array}}{\hat{\Gamma} \vdash \texttt{import}(\varepsilon)\ x = \hat{e} \texttt{ in } e : \texttt{annot}(\tau, \varepsilon) \texttt{ with } \varepsilon \cup \varepsilon_1}\ (\varepsilon\text{-Module})$$

$\boxed{\texttt{safe}(\tau, \varepsilon)}$

$$\frac{}{\texttt{safe}(\{\bar{r}\}, \varepsilon)}\ (\textsc{Safe-Resource}) \qquad \frac{}{\texttt{safe}(\texttt{Unit}, \varepsilon)}\ (\textsc{Safe-Unit})$$

$$\frac{\varepsilon \subseteq \varepsilon' \quad \texttt{ho-safe}(\hat{\tau}_1, \varepsilon) \quad \texttt{safe}(\hat{\tau}_2, \varepsilon)}{\texttt{safe}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2, \varepsilon)}\ (\textsc{Safe-Arrow})$$

$\boxed{\texttt{ho-safe}(\hat{\tau}, \varepsilon)}$

$$\frac{}{\texttt{ho-safe}(\{\bar{r}\}, \varepsilon)}\ (\text{HOS}\textsc{afe-Resource}) \qquad \frac{}{\texttt{ho-safe}(\texttt{Unit}, \varepsilon)}\ (\text{HOS}\textsc{afe-Unit})$$

$$\frac{\texttt{safe}(\hat{\tau}_1, \varepsilon) \quad \texttt{ho-safe}(\hat{\tau}_2, \varepsilon)}{\texttt{ho-safe}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2, \varepsilon)}\ (\text{HOS}\textsc{afe-Arrow})$$

$\boxed{\hat{\tau} <: \hat{\tau}}$

$$\frac{\varepsilon \subseteq \varepsilon' \quad \hat{\tau}_2 <: \hat{\tau}_2' \quad \hat{\tau}_1' <: \hat{\tau}_1}{\hat{\tau}_1 \rightarrow_{\varepsilon} \hat{\tau}_2 <: \hat{\tau}_1' \rightarrow_{\varepsilon'} \hat{\tau}_2'}\ (\text{S-}\textsc{Effects})$$

4 Dynamic Rules

$\boxed{\hat{e} \longrightarrow \hat{e} \mid \varepsilon}$

$$\frac{\hat{e}_1 \longrightarrow \hat{e}_1' \mid \varepsilon}{\hat{e}_1 \hat{e}_2 \longrightarrow \hat{e}_1' \hat{e}_2 \mid \varepsilon}\ (\text{E-}\textsc{App1}) \qquad \frac{\hat{e}_2 \longrightarrow \hat{e}_2' \mid \varepsilon}{\hat{v}_1 \hat{e}_2 \longrightarrow \hat{v}_1 \hat{e}_2' \mid \varepsilon}\ (\text{E-}\textsc{App2}) \qquad \frac{}{(\lambda x : \hat{\tau}.\hat{e})\hat{v}_2 \longrightarrow [\hat{v}_2/x]\hat{e} \mid \varnothing}\ (\text{E-}\textsc{App3})$$

$$\frac{\hat{e} \rightarrow \hat{e}' \mid \varepsilon}{\hat{e}.\pi \longrightarrow \hat{e}'.\pi \mid \varepsilon}\ (\text{E-}\textsc{OperCall1}) \qquad \frac{r \in R \quad \pi \in \mathit{\Pi}}{r.\pi \longrightarrow \texttt{unit} \mid \{r.\pi\}}\ (\text{E-}\textsc{OperCall2})$$

$$\frac{\hat{e} \longrightarrow \hat{e}' \mid \varepsilon'}{\texttt{import}(\varepsilon)\ x = \hat{e}\ \texttt{in}\ e \longrightarrow \texttt{import}(\varepsilon)\ x = \hat{e}'\ \texttt{in}\ e \mid \varepsilon'}\ (\text{E-}\textsc{Module1})$$

$$\frac{}{\texttt{import}(\varepsilon)\ x = \hat{v}\ \texttt{in}\ e \longrightarrow [\hat{v}/x]\texttt{annot}(e, \varepsilon) \mid \varnothing}\ (\text{E-}\textsc{Module2})$$

5 Encodings

5.1 $\perp$

We can define the bottom type as $\perp = \{\}$, because there is no empty-set literal.

5.2 `unit`, `Unit`

Define `unit` $= \lambda$`x` $: \{\}.$`x`, i.e. the function which takes an empty set of resources and returns it. We shall refer to its type, which is $\{\} \rightarrow_{\varnothing} \{\}$, as `Unit`. It has various properties befitting `unit`.

1. `unit` cannot be invoked, as $\{\}$ is uninhabited.
2. `unit` is a value.
3. The only term with type `Unit` is `unit`.
4. $\vdash$ `unit` : `Unit`, by using ε-Abs and ε-Var.
5. $\texttt{effects}(\texttt{Unit}) = \texttt{ho-effects}(\texttt{Unit}) = \varnothing$
6. $\texttt{safe}(\texttt{Unit}, \varepsilon)$ and $\texttt{ho-safe}(\texttt{Unit}, \varepsilon)$

6 Proofs

Theorem 1 (Progress). *If* $\hat{\Gamma} \vdash \hat{e}_A : \hat{\tau}_A$ `with` ε_A *then* $\hat{e}_A$ *is a value or* $\hat{e}_A \longrightarrow \hat{e}_B \mid \varepsilon$*.*

Proof. By induction on $\hat{\Gamma} \vdash \hat{e}_A : \hat{\tau}_A$ `with` ε_A.

$\boxed{\text{Case: } \varepsilon\text{-Resource, } \varepsilon\text{-Unit, } \varepsilon\text{-Abs}}$ Then $\hat{e}_A$ is a value.

$\boxed{\text{Case: } \varepsilon\text{-Subsume}}$ Then $\hat{\Gamma} \vdash e : \tau'$ `with` ε', and $\hat{\Gamma} \vdash e : \tau$ `with` ε, where $\tau' <: \tau$ and $\varepsilon' \subseteq \varepsilon$ are subderivations. The theorem conclusion holds by inductive assumption applied to $\hat{\Gamma} \vdash e : \tau$ `with` ε.

$\boxed{\text{Case: } \varepsilon\text{-App}}$ Then $\hat{e}_A = \hat{e}_1\ \hat{e}_2$. We consider the cases in which $\hat{e}_1$ and $\hat{e}_2$ are values.

If $\hat{e}_1$ is not a value then by inductive assumption there is a reduction $\hat{e}_1 \longrightarrow \hat{e}_1' \mid \varepsilon$. Then $\hat{e}_1\ \hat{e}_2$ reduces by the rule E-App1, giving $\hat{e}_1\ \hat{e}_2 \longrightarrow \hat{e}_1'\ \hat{e}_2 \mid \varepsilon$.

If $\hat{e}_2$ is not a value then WLOG $\hat{e}_1$ is a value. By inductive assumption $\hat{e}_2 \longrightarrow \hat{e}_2' \mid \varepsilon$. Then $\hat{v}_1\ \hat{e}_2$ reduces by the rule E-App2, giving $\hat{v}_1\ \hat{e}_2 \longrightarrow \hat{v}_1\ \hat{e}_2' \mid \varepsilon$.

If $\hat{e}_1$ and $\hat{e}_2$ are both values then by canonical forms $\hat{e}_1 = \hat{v}_1 = \lambda x : \tau_2.e$. Then $\hat{v}_1\ \hat{v}_2$ reduces by the rule E-App3, giving $\hat{v}_1\ \hat{v}_2 \longrightarrow [\hat{v}_2/x]\hat{e} \mid \varnothing$.

$\boxed{\text{Case: } \varepsilon\text{-OperCall}}$ Then $\hat{e}_A = \hat{e}_1.\pi$. We consider whether $\hat{e}_1$ is a value.

If $\hat{e}_1$ is not a value then by inductive assumption there is a reduction $\hat{e}_1 \longrightarrow \hat{e}_1' \mid \varepsilon$. Then $\hat{e}_1.\pi$ reduces by the rule E-OperCall1, giving $\hat{e}_1.\pi \longrightarrow \hat{e}_1'.\pi \mid \varepsilon$.

If $\hat{e}_1$ is a value then $\hat{e}_1 = r$ by canonical forms. By the assumption that $r.\pi$ is closed under Γ, we know $r \in R$ and $\pi \in \Pi$. Then $\hat{e}_1.\pi$ reduces by the rule E-OperCall2, giving $r.\pi \longrightarrow$ `unit` $\mid \varepsilon$.

$\boxed{\text{Case: } \varepsilon\text{-Module}}$ Then $e_A =$ `import`$(\varepsilon)\ x = \hat{e}$ `in` e. If $\hat{e}$ is an expression then it can be reduced, so $\hat{e} \longrightarrow \hat{e}' \mid \varepsilon'$, and so by E-Module1 we get `import`$(\varepsilon)\ x = \hat{e}$ `in` $e \longrightarrow$ `import`$(\varepsilon)\ x = \hat{e}'$ `in` $e \mid \varepsilon'$. Otherwise $\hat{e} = \hat{v}$ is a value. Then by E-Module2 we get `import`$(\varepsilon)\ x = \hat{v} \longrightarrow [\hat{v}/x]$`annot`$(e, \varepsilon) \mid \varnothing$.

Lemma 1 (Substitution). *If* $\hat{\Gamma}, x : \hat{\tau}' \vdash e : \hat{\tau}$ `with` ε *and* $\hat{\Gamma} \vdash v : \hat{\tau}'$ `with` $\varnothing$ *then* $\hat{\Gamma} \vdash [v/x]e : \hat{\tau}$ `with` ε*.*

Lemma 2. $\texttt{ho-safe}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2, \varepsilon) \implies \varepsilon \subseteq \texttt{ho-effects}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2)$

Lemma 3. $\texttt{safe}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2, \varepsilon) \implies \varepsilon \subseteq \texttt{effects}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2)$

Proof. By simultaneous induction on derivations.

$\boxed{\texttt{ho-safe}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2, \varepsilon)}$ By definition, $\texttt{safe}(\hat{\tau}_1, \varepsilon)$ and $\texttt{ho-safe}(\hat{\tau}_2, \varepsilon)$. By applying inductive assumptions, $\varepsilon \subseteq \texttt{effects}(\hat{\tau}_1)$ and $\varepsilon \subseteq \texttt{ho-effects}(\hat{\tau}_2)$. Then $\varepsilon \subseteq \texttt{effects}(\hat{\tau}_1) \cup \texttt{ho-effects}(\hat{\tau}_2) = \texttt{ho-effects}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2)$.

$\boxed{\texttt{safe}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2, \varepsilon)}$ By definition, $\varepsilon \subseteq \varepsilon'$ and $\texttt{ho-safe}(\hat{\tau}_1, \varepsilon)$ and $\texttt{safe}(\hat{\tau}_2, \varepsilon)$. By applying inductive assumptions, $\varepsilon \subseteq \texttt{ho-effects}(\hat{\tau}_1)$ and $\varepsilon \subseteq \texttt{effects}(\hat{\tau}_2)$. Then $\varepsilon \subseteq \varepsilon' \cup \texttt{ho-effects}(\hat{\tau}_1) \cup \texttt{effects}(\hat{\tau}_2) = \texttt{effects}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2)$.

Lemma 4. *If* $\varepsilon \subseteq \texttt{effects}(\hat{\tau})$ *and* $\texttt{ho-safe}(\hat{\tau}, \varepsilon)$ *then* $\hat{\tau} <: \texttt{annot}(\texttt{erase}(\hat{\tau}), \varepsilon)$*.*

Lemma 5. *If* $\varepsilon \subseteq \texttt{ho-effects}(\hat{\tau})$ *and* $\texttt{safe}(\hat{\tau}, \varepsilon)$ *then* $\texttt{annot}(\texttt{erase}(\hat{\tau}), \varepsilon) <: \hat{\tau}$*.*

Proof. By simultaneous induction on derivations.

$\boxed{\text{Case: } \hat{\tau} = \{\bar{r}\}}$ Then $\texttt{annot}(\texttt{erase}(\hat{\tau}), \varepsilon) = \hat{\tau}$.

$\boxed{\text{Case: } \hat{\tau} = \hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2,\ \varepsilon \subseteq \texttt{effects}(\hat{\tau}),\ \texttt{ho-safe}(\hat{\tau}, \varepsilon)}$ By definition of `annot` and `erase` we know:
$\texttt{annot}(\texttt{erase}(\hat{\tau}_1 \rightarrow_{\varepsilon'} \hat{\tau}_2), \varepsilon = \texttt{annot}(\texttt{erase}(\hat{\tau}_1), \varepsilon) \rightarrow_{\varepsilon} \texttt{annot}(\texttt{erase}(\hat{\tau}_2), \varepsilon)$

By definition we have the following subderivations.

1. $\texttt{effects}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2) = \texttt{ho-effects}(\hat{\tau}_1) \cup \varepsilon' \cup \texttt{effects}(\hat{\tau}_2)$
2. $\texttt{ho-safe}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2, \varepsilon) = \texttt{safe}(\hat{\tau}_1, \varepsilon) \wedge \texttt{ho-safe}(\hat{\tau}_2, \varepsilon)$

$\texttt{safe}(\hat{\tau}_1, \varepsilon)$ is a subderivation of $\texttt{ho-safe}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2)$. By **Lemma 2**, $\texttt{ho-safe}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2)$ implies $\varepsilon \subseteq \texttt{ho-effects}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2) = \texttt{effects}(\hat{\tau}_1) \cup \texttt{ho-effects}(\hat{\tau}_2)$.

$\boxed{\text{Case: } \hat{\tau} = \hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2,\ \varepsilon \subseteq \texttt{ho-effects}(\hat{\tau}),\ \texttt{safe}(\hat{\tau}, \varepsilon)}$

$\texttt{ho-effects}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2) = \texttt{effects}(\hat{\tau}_1) \cup \texttt{ho-effects}(\hat{\tau}_2)$
$\texttt{safe}(\hat{\tau}_1 \to_{\varepsilon'} \hat{\tau}_2) = \varepsilon \subseteq \varepsilon' \wedge \texttt{ho-safe}(\hat{\tau}_1, \varepsilon) \wedge \texttt{safe}(\hat{\tau}_2, \varepsilon)$

Theorem 2 (Preservation). *If $\hat{\Gamma} \vdash \hat{e}_A : \hat{\tau}_A$ $\texttt{with}$ ε_A and $e_A \longrightarrow e_B \mid \varepsilon$, then $\hat{\Gamma} \vdash e_B : \tau_B$ $\texttt{with}$ ε_B, where $e_B <: e_B$ and $\varepsilon \cup \varepsilon_B \subseteq \varepsilon_A$.*

Proof. By induction on $\hat{\Gamma} \vdash \hat{e}_A : \tau_A$ $\texttt{with}$ ε_A, and then on $e_A \longrightarrow e_B \mid \varepsilon$.

$\boxed{\varepsilon\text{-Var, } \varepsilon\text{-Resource, } \varepsilon\text{-Unit, } \varepsilon\text{-Abs}}$ Then e_A cannot be reduced and so the theorem statement vacuously holds.

$\boxed{\varepsilon\text{-App}}$ Then $e_A = \hat{e}_1 \hat{e}_2$ and $\hat{e}_1 : \hat{\tau}_2 \to_\varepsilon \hat{\tau}_3$ $\texttt{with}$ ε_1 and $\hat{\Gamma} \vdash \hat{e}_2 : \hat{\tau}_2$ $\texttt{with}$ ε_2. If the reduction rule used was E-App1 or E-App2, then the result follows by applying the inductive hypothesis to $\hat{e}_1$ and $\hat{e}_2$ respectively.

Otherwise the rule used was E-App3. Then $(\lambda x : \hat{\tau}_2.\hat{e})\hat{v}_2 \longrightarrow [\hat{v}_2/x]\hat{e} \mid \varnothing$. By inversion on the typing rule for $\lambda x : \hat{\tau}_2.\hat{e}$ we know $\Gamma, x : \hat{\tau}_2 \vdash \hat{e} : \hat{\tau}_3$ $\texttt{with}$ ε_3. By canonical forms, $\varepsilon_2 = \varnothing$ because $\hat{e}_2 = \hat{v}_2$ is a value. Then by the substitution lemma, $\hat{\Gamma} \vdash [\hat{v}_2/x]\hat{e} : \hat{\tau}_3$ $\texttt{with}$ ε_3. By canonical forms, $\varepsilon_1 = \varepsilon_2 = \varnothing = \varepsilon$. Therefore $\varepsilon_A = \varepsilon_3 = \varepsilon_B \cup \varepsilon$.

$\boxed{\varepsilon\text{-OperCall}}$ Then $e_A = e_1.\pi$ and $\hat{\Gamma} \vdash e_1 : \{\bar{r}\}$ $\texttt{with}$ ε_1. If the reduction rule used was E-OperCall1 then the result follows by applying the inductive hypothesis to $\hat{e}_1$.

Otherwise the reduction rule used was E-OperCall2 and $v_1.\pi \longrightarrow \texttt{unit} \mid \{r.\pi\}$. By canonical forms, $\hat{\Gamma} \vdash v_1 : \texttt{unit}$ $\texttt{with}$ $\{r.\pi\}$. Also, $\hat{\Gamma} \vdash \texttt{unit} : \texttt{Unit}$ $\texttt{with}$ $\varnothing$. Then $\tau_B = \tau_A$. Also, $\varepsilon \cup \varepsilon_B = \{r.\pi\} = \varepsilon_A$.

$\boxed{\varepsilon\text{-Module}}$ Then $e_A = \texttt{import}(\varepsilon)\ x = \hat{e}$ $\texttt{in}$ e. If the reduction rule used was E-ModuleCall1 then the result follows by applying the inductive hypothesis to $\hat{e}$.

Otherwise the following are true:

1. $e_A = \texttt{import}(\varepsilon)\ x = \hat{v}$ $\texttt{in}$ e
2. $\hat{\Gamma} \vdash e_A : \texttt{annot}(\tau, \varepsilon)$ $\texttt{with}$ $\varepsilon \cup \varepsilon_1$
3. $\texttt{import}(\varepsilon)\ x = \hat{v}$ $\texttt{in}$ $e \longrightarrow [\hat{v}/x]\texttt{annot}(e, \varepsilon) \mid \varnothing$
4. $\hat{\Gamma} \vdash \hat{v} : \hat{\tau}$ $\texttt{with}$ $\varnothing$
5. $\varepsilon = \texttt{effects}(\hat{\tau})$
6. $\texttt{ho-safe}(\hat{\tau}, \varepsilon)$
7. $x : \texttt{erase}(\hat{\tau}) \vdash e : \tau$

Use the **annotation lemma** to get $\hat{\Gamma}, x : \hat{\tau} \vdash \texttt{annot}(e, \varepsilon) : \texttt{annot}(\tau, \varepsilon)$ $\texttt{with}$ ε.

By **3** we have $\hat{\Gamma} \vdash \hat{v} : \hat{\tau}$ $\texttt{with}$ $\varnothing$.

By **substitution lemma**, $\hat{\Gamma} \vdash [\hat{v}/x]\texttt{annot}(e, \varepsilon) : \texttt{annot}(\tau, \varepsilon)$ $\texttt{with}$ ε.

Lemma 6 (Annotation). *If the following are true for every Γ:*

- $\hat{\Gamma} \vdash \hat{v} : \hat{\tau}$ $\texttt{with}$ $\varnothing$
- $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash e : \tau$
- $\varepsilon = \texttt{effects}(\hat{\tau})$
- $\texttt{ho-safe}(\hat{\tau}, \varepsilon)$

Then $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash \texttt{annot}(e, \varepsilon) : \texttt{annot}(\tau, \varepsilon)$ $\texttt{with}$ $\varepsilon \cup \texttt{effects}(\texttt{annot}(\Gamma, \varepsilon))$.

Proof. By induction on $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash e : \tau$.

$\boxed{\text{Case: T-VAR}}$ Then $e = x$ and $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash x : \tau$. There are two cases: either $x = y$ or $x \neq y$.

Subcase 1: $x = y$. Then by ε-VAR we get $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash x : \hat{\tau}$ $\texttt{with}$ $\varnothing$. **Need to justify why** $\texttt{annot}(\tau, \varepsilon) = \hat{\tau}$**. Note that** $\varepsilon = \texttt{effects}(\hat{\tau})$**. Maybe show that** $\texttt{annot}(\tau, \texttt{effects}(\tau)) <: \tau$**. Can you use the lemma here?**

Subcase 2: $x \neq y$. Then $x : \tau \in \Gamma$.

$\boxed{\text{Case: T-RESOURCE}}$ Then $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash r : \{r\}$. By definition, $\texttt{annot}(r, \varepsilon) = r$ and $\texttt{annot}(\{r\}, \varepsilon)$. By ε-RESOURCE $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash r : \{r\}$ $\texttt{with}$ $\varnothing$. By ε-SUBSUME, $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash r : \{r\}$ $\texttt{with}$ $\varepsilon \cup \texttt{effects}(\texttt{annot}(\Gamma, \varepsilon))$.

$\boxed{\text{Case: T-ABS}}$ Then $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash \lambda x : \tau_1.e_{body} : \tau_1 \to \tau_2$.

By inversion, we get the sub-derivation $\Gamma, y : \texttt{erase}(\hat{\tau}), x : \tau_1 \vdash e_2 : \tau_2$.

By definition, $\texttt{annot}(e, \varepsilon) = \texttt{annot}(\lambda x : \tau_1.e_2, \varepsilon) = \lambda x : \texttt{annot}(\tau_1, \varepsilon).\texttt{annot}(e_2, \varepsilon)$.

By definition, $\texttt{annot}(\tau, \varepsilon) = \texttt{annot}(\tau_1 \to \tau_2, \varepsilon) = \texttt{annot}(\tau_1, \varepsilon) \to_\varepsilon \texttt{annot}(\tau_2, \varepsilon)$.

To apply the inductive assumption to e_2 we use the unlabelled context $\Gamma, x : \tau_1$. The inductive assumption tells us $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau}, x : \texttt{annot}(\tau_1, \varepsilon) \vdash \texttt{annot}(e_2, \varepsilon) : \texttt{annot}(\tau_2, \varepsilon)$ $\texttt{with}$ $\varepsilon \cup \texttt{effects}(\texttt{annot}(\Gamma, \varepsilon)) \cup \texttt{effects}(\texttt{annot}(\tau_1, \varepsilon))$. Call this last effect-set ε'.

By ε-ABS, we get $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash \lambda x : \texttt{annot}(\tau_1, \varepsilon).\texttt{annot}(e_2, \varepsilon) : \texttt{annot}(\hat{\tau}_1) \to_{\varepsilon'} \texttt{annot}(\hat{\tau}_2)$ $\texttt{with}$ $\varnothing$.

By ε-SUBSUME, we get $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash \texttt{annot}(e, \varepsilon) : \texttt{annot}(\hat{\tau}_1) \to_\varepsilon \texttt{annot}(\hat{\tau}_2)$ $\texttt{with}$ $\varepsilon \cup \texttt{effects}(\texttt{annot}(\Gamma), \varepsilon)$.

$\boxed{\text{Case: T-APP}}$ TODO

$\boxed{\text{Case: T-OPERCALL}}$ Then $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash e_1.\pi : \texttt{Unit}$.

By inversion we get the sub-derivation $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash e_1 : \{\bar{r}\}$.

By definition, $\texttt{annot}(\{\bar{r}\}, \varepsilon) = \{\bar{r}\}$.

By inductive assumption, $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash e_1 : \{\bar{r}\}$ $\texttt{with}$ $\varepsilon \cup \texttt{effects}(\texttt{annot}(\Gamma, \varepsilon))$.

By ε-OPERCALL, $\hat{\Gamma}, \texttt{annot}(\Gamma, \varepsilon), y : \hat{\tau} \vdash e_1.\pi : \{\bar{r}\}$ $\texttt{with}$ $\varepsilon \cup \{\bar{r}.\pi\}$.

It remains to show $\{\bar{r}.\pi\} \subseteq \varepsilon$. We shall do this by considering where r must have come from (which subcontext left of the turnstile).

Subcase 1. $r = \hat{\tau}$. As $\varepsilon = \texttt{effects}(\hat{\tau})$, then $r.\pi \in \texttt{effects}(\hat{\tau})$.

Subcase 2. $r : \{r\} \in \Gamma$. As $\texttt{annot}(r, \varepsilon) = r$, then $r.\pi \in \texttt{annot}(\Gamma, \varepsilon)$.

Subcase 3. $r : \{r\} \in \hat{\Gamma}$. Then because $\Gamma, y : \texttt{erase}(\hat{\tau}) \vdash e_1 : \{\bar{r}\}$, then $r \in \Gamma$ or $r = \texttt{erase}(\hat{\tau}) = \hat{\tau}$ and one of the above subcases must also hold.