Extrinsic Type Inference SUPP (DRAFT 2025-06-03)

THOMAS LOGAN

ACM Reference Format:
Thomas Logan. 2025. Extrinsic Type Inference SUPP (DRAFT 2025-06-03). 1, 1 (August 2025), 22 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ANCILLARIES

TODO: add collection concatentation definition (oplus)
TODO: read through definitions and update syntax
TODO: organize descriptions
TODO: describe the important properties in depth

Definition 1.1. Expression Congruence $\boxed{e_l \cong e_r}$

$$\frac{\forall \delta, \tau \;.\; \delta \vDash e_l : \tau \iff \delta \vDash e_r : \tau}{e_l \cong e_r}$$

Definition 1.2. Union Drop $\boxed{\alpha \Vdash \tau_r :> \tau_r'}$

$$\frac{\alpha \notin \mathbf{ftv}(\tau)}{\alpha \Vdash \tau :> \tau} \qquad \frac{\alpha \in \mathbf{ftv}(\tau) \qquad \alpha \Vdash \tau_l :> \tau_l' \qquad \alpha \Vdash \tau_r :> \tau_r'}{\alpha \Vdash \tau_l \mid \tau_r :> \tau_l' \mid \tau_r'} \qquad \frac{\alpha \in \mathbf{ftv}(\tau) \qquad \nexists \tau_l, \tau_r \;.\; \tau = \tau_l \mid \tau_r}{\alpha \Vdash \tau :> \texttt{BOT}}$$

Definition 1.3. Type Negatability $\boxed{\Theta, \Delta \Vdash \eta}$

$$\frac{\mathbf{ftv}(\rho) \subseteq \emptyset}{\Vdash \rho} \qquad \frac{\mathbf{ftv}(\texttt{EXI}\,[\Theta]\,\Omega : \eta) \subseteq \emptyset \qquad \Theta, \Delta \Vdash \Omega \qquad \Theta, \Delta \Vdash \eta}{\Theta, \Delta \Vdash \texttt{EXI}\,[\Theta]\,\Omega : \eta}$$

Definition 1.4. Subtyping Sequence Negatability $\boxed{\Theta, \Delta \Vdash \Omega}$

$$\frac{}{\Theta, \Delta \Vdash \epsilon} \qquad \frac{\Theta, \Delta \Vdash \Omega \qquad \Theta, \Delta \Vdash \eta}{\Theta, \Delta \Vdash \Omega\ \tau <: \eta} \qquad \frac{\alpha \notin \Theta \qquad \Theta, \Delta \Vdash \Omega \qquad (\forall \tau' \;.\; \alpha <: \tau' \in \Delta \implies \tau' = \texttt{TOP})}{\Theta, \Delta \Vdash \Omega\ \alpha <: \tau}$$

Definition 1.5. Typing Consolidation $\boxed{\Theta, \Delta \vdash e : \Pi}$

$$\frac{(\forall \Theta', \Delta' \;.\; \Theta' \backslash \Theta, \Delta' \backslash \Delta, \tau \in \Pi \implies \Theta, \Delta \vdash e : \tau \dashv \Theta', \Delta') \qquad \Theta \cup \mathbf{ftv}(\Delta) \vdash \Pi \searrow \Pi'}{\Theta, \Delta \vdash e : \Pi'}$$

Author's address: Thomas Logan.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM XXXX-XXXX/2025/8-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Definition 1.6. Subtyping Guarded List Locale Static $\boxed{\Theta, \Delta \vdash \tau <: \alpha \rightarrow \Pi}$

$$\frac{(\forall \Theta', \Delta', \tau' \,.\, (\Theta' \backslash \Theta, \Delta' \backslash \Delta, \tau') \in \Pi \implies \Theta, \Delta \vdash \tau <: \alpha \text{->} \tau' \dashv \Theta', \Delta') \quad \textbf{ftv}(\Delta) \vdash \Pi \searrow \Pi'}{\Theta, \Delta \vdash \tau <: \alpha \rightarrow \Pi'}$$

TODO: ...

Definition 1.7. Function Typing Consolidation $\boxed{\Theta, \Delta, \Gamma \vdash f : \Pi \sim \Xi}$

$$\frac{}{\Theta, \Delta, \Gamma \vdash \epsilon : \epsilon \sim \epsilon} \qquad \frac{\Theta, \Delta, \Gamma \vdash f : \Pi \sim \Xi \quad \textbf{lift}(\Delta, \Gamma \vdash p) = (\Delta', \Gamma', \rho) \quad \textbf{diff}(\rho, \Xi) = \tau_l \quad (\forall \Theta', \Delta'', \tau_r \,.\, \Theta' \backslash \Theta, \Delta'' \backslash \Delta', \tau_l \text{->} \tau_r \in \Pi' \implies \Theta, \Delta', \Gamma' \vdash e : \tau_r \dashv \Theta', \Delta'') \quad \textbf{capture}(\rho) = \eta \quad \textbf{ftv}(\Delta) \vdash \Pi' \searrow \Pi''}{\Theta, \Delta, \Gamma \vdash f \ ?p \text{=>} e : \Pi \cup \Pi'' \sim \Xi\, \eta}$$

Path sequence typing $(\Theta, \Delta, \Gamma \vdash f : \Pi \sim \Xi)$ handles the details of constructing types for each path of a function. Progression of application tries each path of the applied function is in order. That is, the application will progress along the first path whose pattern matches the argument. To account for this ordering and short-circuiting of pattern matching, the types for each path of the function must subtract the types of patterns from earlier paths Ξ when constructing antecedents of path types. Each path of the function could branch out, represented by the typing $(\Gamma \oplus \Gamma' \vdash e : \tau_r \dashv ...)$, so each function path can produce multiple path types $(\tau_l \text{->} \tau_r)$, which are collected in π'.
TODO: explain the tidy up relation Keeps things tidy by compacting the subtyping environments and making the path types more readable.

Theorem 1.1. Schema Sequence Duality Correspondence

$$\frac{\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu \qquad \delta \vDash \Delta}{\begin{array}{c}(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \textbf{dom}(\delta^\dagger) \cong \Theta^\dagger \wedge \\ (\forall \delta' \,.\, \textbf{dom}(\delta') \cap \textbf{ftv}(\Delta) = \epsilon \implies \\ \delta\ \alpha_\nu/\texttt{BOT} \oplus \delta' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta\ \alpha_\nu/\texttt{BOT} \oplus \delta' \oplus \delta^\dagger \vDash e_f : \tau^\dagger)) \\ \Longleftrightarrow \\ (\forall e_a, e_r \,.\, (\exists \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\mu \wedge (\forall \delta^\dagger \,.\, \textbf{dom}(\delta^\dagger) \cong \Theta^\dagger \implies \\ \exists \delta' \,.\, \textbf{dom}(\delta') \cap \textbf{ftv}(\Delta) = \epsilon \wedge \\ \delta\ \alpha_\mu/\texttt{TOP} \oplus \delta' \oplus \delta^\dagger \vDash \Delta^\dagger \wedge \delta\ \alpha_\mu/\texttt{TOP} \oplus \delta' \oplus \delta^\dagger \vDash (e_a, e_r) : \tau^\dagger)) \implies e_r \cong e_f\ (e_a))\end{array}}$$

Duality correspondence $\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu$ rewrites schemas with path types Π_ν into schemas with pair types Π_ν such that for any interpretation δ, a given function e_f inhabits all interpretations of every path type schema if and only if there's an argument e_a paired with the function applied to it that inhabits at least one of the pair type schemas.

Definition 1.8. Schema Sequence Duality $\boxed{\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu}$

$$\frac{}{\alpha_\nu \downarrow \epsilon \coloneqq \alpha_\mu \uparrow \epsilon} \qquad \frac{\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu \qquad \alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu}{\alpha_\nu \downarrow \Pi_\nu\ (\Theta_c, \Delta_\nu, \tau_l \text{->} \tau_r) \coloneqq \alpha_\mu \uparrow \Pi_\mu\ (\Theta_c, \Delta_\mu, \tau_l * \tau_r)}$$

The duality correspondence predicate unrolls the two sequences of schemas simultaneously. In the case where both sequence are empty $\alpha_\nu \downarrow \epsilon \coloneqq \alpha_\mu \uparrow \epsilon$, duality correspondence simply holds. Otherwise, duality correspondence requires that the left sequence's last schema has a path type $(\Theta_c, \Delta_\nu, \tau_l \text{->} \tau_r)$ and the right sequence's last schema has a pair type $(\Theta_c, \Delta_\nu, \tau_l * \tau_r)$; the subtyping environments correspond $\alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu$; and the earlier sequences of schemas correspond $\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu$.

Theorem 1.2. Subtyping Environment Duality Correspondence

$$\frac{\alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu}{\begin{array}{c}(\forall \tau_\nu, \Delta_\nu \,.\, \delta\ \alpha_\nu/\tau_\nu \vDash \Delta_\nu \implies \delta\ \alpha_\nu/\tau_\nu \vDash e_f : \tau_\nu) \\ \iff \\ (\exists \tau_\mu, \Delta_\mu, e_a \,.\, \delta\ \alpha_\mu/\tau_\mu \vDash \Delta_\mu \wedge \delta\ \alpha_\mu/\tau_\mu \vDash (e_a\,,\, e_f\ (e_a)) : \tau_\mu))\end{array}}$$

Subtyping duality correspondence $\alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu$ rewrites a subtyping environment Δ_ν containing path type upper bounds on variable α_ν into a subtyping environment Δ_μ containing pair type lower bounds on variable α_μ, such that for any interpretation δ, a given function e_f inhabits all types τ_ν constrained by the environment's path type upper bounds if and only if there's an argument e_a paired with the function applied to it that inhabits at least one type τ_μ constrained by the environment's pair type lower bounds.

Definition 1.9. Subtyping Environment Duality $\boxed{\alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu}$

$$\frac{}{\alpha_\nu \downarrow \epsilon \coloneqq \alpha_\mu \uparrow \epsilon} \qquad \frac{\alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu}{\alpha_\nu \downarrow \Delta_\nu\ \alpha_\nu <: \alpha_l -> \alpha_r \coloneqq \alpha_\mu \uparrow \Delta_\mu\ \alpha_l * \alpha_r <: \alpha_\mu}$$

The subtyping duality correspondence predicate unrolls the two subtyping environments simultaneously. In the case where both environments are empty $\alpha_\nu \downarrow \epsilon \coloneqq \alpha_\mu \uparrow \epsilon$, subtyping duality correspondence simply holds. In the case where the subtyping constraints are the same, the upper bounded variable α_ν must not appear in the subtype τ_l; the lower bounded variable α_μ must not appear in the supertype τ_r; and the correspondence must hold for the previous constraints in the environment $\alpha_\nu \downarrow \Delta_\nu \coloneqq \alpha_\mu \uparrow \Delta_\mu$. Lastly, subtyping duality correspondence holds if the last subtyping of each subtyping environment contains the path type upper bounds $\alpha_\nu <: \alpha_l -> \alpha_r$ and pair type lower bounds $\alpha_l * \alpha_r <: \alpha_\mu$ on the specified variables, and the the correspondence holds for the previous constraints in the environment

Theorem 1.3. Type Interpretation Soundness

$$\frac{[\![\Theta \vdash \Delta, \tau]\!]^{\pm} = (\Delta', \tau') \qquad \Theta \subseteq \mathbf{dom}(\delta')}{\exists \delta.\ (\delta \oplus \delta' \vDash \Delta \wedge \delta \oplus \delta' \vDash e : \tau) \iff (\delta' \vDash \Delta' \wedge \delta' \vDash e : \tau')}$$

Type interpretation $[\![\Theta_\varnothing, \Theta \vdash \Delta, \tau]\!]^{\pm} = (\Delta', \tau')$ constructs a type τ', such that τ' is behaves like the input type τ for some interpretation δ. The positive form $[\![...]\!]^{+}$ constructs the strongest type or tries to get close to the strongest type possible. The negative form $[\![...]\!]^{-}$ constructs the weakest type or tries to get close to the weakest type possible. The type variables $\Theta_\varnothing$ and Θ must be irrelevant, meaning the constructed type τ' must behave like τ for all interpretations over those variables δ'. Additionally, any interpretation δ' for the resulting subtyping environment Δ' must be part of some interpretation for the original subtyping environment Δ.

The details of finding the type interpretation involves numerous tedious details, so we leave the definition in the appendix (Section **??**).

Packing ($\mathbf{pack}^{\pm}(\Theta_\varnothing \vdash \Pi) = \tau'$) constructs a type τ' from the schemas Π such that for each schema (Θ, Δ, τ), it binds the variables of the type τ within some combination of existentials and/or universals, where indexes are qualified by the constraints of the subtyping environment Δ. The choice of binding in an existential or universal is determined by whether the variable is open ($\alpha \notin \Theta$) or closed ($\alpha \in \Theta$) and whether the interpretation is positive ($\mathbf{pack}^{+}(...)$) or negative ($\mathbf{pack}^{-}(...)$). For the positive form, the construction (including how the closed variables are bound) must result in a type τ' that is weaker than the type of every schema in every interpretation of the closed

variables δ', for some interpretation of the open variables δ . For the negative form, the construction must result in a type τ' that is stronger than the type of every schema in every interpretation of the closed variables δ', for some interpretation of the open variables δ .

The interested reader can find the actual the definition of packing in the appendix (Section **??**).

Definition 1.10. Type Interpretation $\boxed{[\![\Theta, \Delta \vdash \tau]\!]^{\pm} = \tau'}$

$$
\begin{array}{rll}
[\![\Theta, \Delta \vdash \alpha]\!]^{\pm} = [\![\Delta \star \alpha]\!]^{\pm} & \text{if} & \alpha \notin \Theta \\
\alpha & \text{if} & \alpha \in \Theta \\
[\![\Theta, \Delta \vdash \texttt{@}]\!]^{\pm} = \texttt{@} & & \\
[\![\Theta, \Delta \vdash <l>\tau]\!]^{\pm} = <l>[\![\Theta, \Delta \vdash \tau]\!]^{\pm} & & \\
[\![\Theta, \Delta \vdash \tau_l \texttt{->} \tau_r]\!]^{\pm} = [\![\Theta, \Delta \vdash \tau_l]\!]^{\text{flip}(\pm)} \texttt{->} [\![\Theta, \Delta \vdash \tau_r]\!]^{\pm} & & \\
[\![\Theta, \Delta \vdash \tau_l \mid \tau_r]\!]^{\pm} = [\![\Theta, \Delta \vdash \tau_l]\!]^{\pm} \mid [\![\Theta, \Delta \vdash \tau_r]\!]^{\pm} & & \\
[\![\Theta, \Delta \vdash \tau_l \& \tau_r]\!]^{\pm} = [\![\Theta, \Delta \vdash \tau_l]\!]^{\pm} \& [\![\Theta, \Delta \vdash \tau_r]\!]^{\pm} & & \\
[\![\Theta, \Delta \vdash \texttt{EXI}\,[\Theta_q]\,\Delta_q\ \tau]\!]^{\pm} = \texttt{EXI}\,[\Theta_q]\,[\![\Theta \cup \Theta_q, \Delta \vdash \Delta_q]\!]^{\pm}\ [\![\Theta \cup \Theta_q, \Delta \vdash \tau]\!]^{\pm} & & \\
[\![\Theta, \Delta \vdash \texttt{ALL}\,[\Theta_q]\,\Delta_q\ \tau]\!]^{\pm} = \texttt{ALL}\,[\Theta_q]\,[\![\Theta \cup \Theta_q, \Delta \vdash \Delta_q]\!]^{\pm}\ [\![\Theta \cup \Theta_q, \Delta \vdash \tau]\!]^{\pm} & & \\
[\![\Theta, \Delta \vdash \tau \backslash \eta]\!]^{\pm} = [\![\Theta, \Delta \vdash \tau]\!]^{\pm} \backslash \eta & & \\
[\![\Theta, \Delta \vdash \texttt{LFP}\,[\alpha]\,\tau]\!]^{\pm} = \texttt{LFP}\,[\alpha]\,[\![\Theta\ \alpha, \Delta \vdash \tau]\!]^{\pm} & &
\end{array}
$$

Definition 1.11. Qualification Interpretation $\boxed{[\![\Theta, \Delta \vdash \Delta_q]\!]^{\pm} = \Delta_q'}$

$$
\begin{array}{rl}
[\![\Theta, \Delta \vdash \epsilon]\!]^{\pm} = \epsilon & \\
[\![\Theta, \Delta \vdash \Delta_q\ \tau_l <: \tau_r]\!]^{\pm} = [\![\Theta, \Delta \vdash \Delta_q]\!]^{\pm}\ [\![\Theta, \Delta \vdash \tau_l]\!]^{\text{flip}(\pm)} <: [\![\Theta, \Delta \vdash \tau_r]\!]^{\pm} &
\end{array}
$$

TODO: ...

Definition 1.12. Variable Interpretation $\boxed{[\![\Delta \star \alpha]\!]^{\pm} = \tau}$

$$
\begin{array}{rll}
[\![\epsilon \star \alpha]\!]^{-} = \texttt{TOP} & & \\
[\![\Delta\ \alpha <: \tau \star \alpha]\!]^{-} = [\![\Delta \star \alpha]\!]^{-} \& \tau & & \\
[\![\Delta\ \tau_l <: \tau_r \star \alpha]\!]^{-} = [\![\Delta, \alpha]\!]^{-} & \text{if} & \alpha \neq \tau_l \\
[\![\epsilon \star \alpha]\!]^{+} = \texttt{BOT} & & \\
[\![\Delta\ \tau <: \alpha \star \alpha]\!]^{+} = [\![\Delta \star \alpha]\!]^{+} \mid \tau & & \\
[\![\Delta\ \tau_l <: \tau_r \star \alpha]\!]^{+} = [\![\Delta \star \alpha]\!]^{+} & \text{if} & \alpha \neq \tau_r
\end{array}
$$

TODO: ...
TODO: update type packing to place constraints on inside if they have neither skolem nor learnable variables TODO: the inner constraints will not be required to have conform to a decidable syntax
TODO: update implementation to check restriction on skolem constraints
TODO: double check that tests still pass

Definition 1.13. Type Packing $\boxed{\textbf{pack}^{\pm}(\dot{\Theta} \vdash \Theta, \Delta, \tau) = \tau'}$

$$
\begin{array}{rll}
\textbf{pack}^{\pm}(\dot{\Theta} \vdash \Theta, \Delta, \tau) = \text{wrap}^{\pm}(\Theta_o, \Delta_o, \Theta_i, \Delta_i, \tau) & \text{if} & \textbf{ftv}(\tau) = \Theta_p \\
 & & \dot{\Theta}, \Theta \vdash \Delta \rightrightarrows \Delta_o, \Delta_i \\
 & & (\textbf{ftv}(\Delta) \cup \Theta_p) \cap \Theta = \Theta_o \\
 & & \textbf{ftv}(\Delta_i) \cup \Theta_p \backslash \Theta \backslash \dot{\Theta} = \Theta_i
\end{array}
$$

Definition 1.14. Multiple Type Packing $\boxed{\mathbf{pack}^{\pm}(\Theta \vdash \Pi) = \tau}$

$$\begin{array}{rll} \mathbf{pack}^{+}(\Theta_{\varnothing} \vdash \epsilon) = & \texttt{TOP} & \\ \mathbf{pack}^{+}(\Theta_{\varnothing} \vdash \Pi\ (\Theta, \Delta, \tau)) = & \tau_l \,\&\, \tau_r & \text{if} \quad \mathbf{pack}^{+}(\Theta_{\varnothing} \vdash \Pi) = \tau_l \\ & & \quad\ \ \mathbf{pack}^{+}(\Theta_{\varnothing} \vdash \Theta, \Delta, \tau) = \tau_r \\ \mathbf{pack}^{-}(\Theta_{\varnothing} \vdash \epsilon) = & \texttt{BOT} & \\ \mathbf{pack}^{-}(\Theta_{\varnothing} \vdash \Pi\ (\Theta, \Delta, \tau)) = & \tau_l \mid \tau_r & \text{if} \quad \mathbf{pack}^{-}(\Theta_{\varnothing} \vdash \Pi) = \tau_l \\ & & \quad\ \ \mathbf{pack}^{-}(\Theta_{\varnothing} \vdash \Theta, \Delta, \tau) = \tau_r \end{array}$$

Definition 1.15. Polar Subtyping Environment Substitution $\boxed{\Delta[\alpha/\tau]^{\pm} = \Delta}$

TODO: ...

Definition 1.16. Subtyping Environment Filter $\boxed{\Theta \vdash \Delta \dashrightarrow \Delta'}$

$$\frac{}{\Theta \vdash \epsilon \dashrightarrow \epsilon} \qquad \frac{\Theta \cap (\mathbf{ftv}(\tau_l) \cup \mathbf{ftv}(\tau_r)) \neq \epsilon \qquad \Theta \vdash \Delta \dashrightarrow \Delta'}{\Theta \vdash \Delta\ \tau_l <: \tau_r \dashrightarrow \Delta'\ \tau_l <: \tau_r}$$

$$\frac{\Theta \cap (\mathbf{ftv}(\tau_l) \cup \mathbf{ftv}(\tau_r)) = \epsilon \qquad \Theta \vdash \Delta \dashrightarrow \Delta'}{\Theta \vdash \Delta\ \tau_l <: \tau_r \dashrightarrow \Delta'}$$

TODO: need lemma for correctness

Definition 1.17. Schema Sequence Compaction $\boxed{\dot{\Theta} \vdash \Pi \searrow \Pi'}$

$$\frac{}{\dot{\Theta} \vdash \epsilon \searrow \epsilon} \qquad \frac{\dot{\Theta} \vdash \Pi \searrow \Pi' \qquad \dot{\Theta} \cup \Theta \vdash \Delta \dashrightarrow \Delta' \qquad [\![\dot{\Theta} \cup \Theta, \Delta \vdash \tau]\!]^{\pm} = \tau'}{\dot{\Theta} \vdash \Pi\ (\Theta, \Delta, \tau) \searrow \Pi'\ (\Theta, \Delta', \tau')}$$

TODO: need lemma for correctness

Definition 1.18. Outer $\boxed{\text{outer}(\pm) = \texttt{ALL}|\texttt{EXI}}$

$$\begin{array}{rl} \text{outer}(+) = & \texttt{EXI} \\ \text{outer}(-) = & \texttt{ALL} \end{array}$$

Definition 1.19. Inner $\boxed{\text{inner}(\pm) = \texttt{ALL}|\texttt{EXI}}$

$$\begin{array}{rl} \text{inner}(+) = & \texttt{ALL} \\ \text{inner}(-) = & \texttt{EXI} \end{array}$$

Definition 1.20. Wrap $\boxed{\text{wrap}^{\pm}(\Theta, \Delta, \Theta, \Delta, \tau) = \tau}$

$$\begin{array}{rl} \text{wrap}^{\pm}(\epsilon, \epsilon, \epsilon, \epsilon, \tau) = & \tau \\ \text{wrap}^{\pm}(\epsilon, \epsilon, \Theta_i, \Delta_i, \tau) = & \text{inner}(\pm)\,[\,\Theta_i \quad \Delta_i\,]\,\tau \\ \text{wrap}^{\pm}(\Theta_o, \Delta_o, \epsilon, \epsilon, \tau) = & \text{outer}(\pm)\,[\,\Theta_o \quad \Delta_o\,]\,\tau \\ \text{wrap}^{\pm}(\Theta_o, \Delta_o, \Theta_i, \Delta_i, \tau) = & \text{outer}(\pm)\,[\,\Theta_o \quad \Delta_o\,]\,\text{inner}(\pm)\,[\,\Theta_i \quad \Delta_i\,]\,\tau \end{array}$$

Definition 1.21. Partition $\boxed{\Theta_\varnothing, \Theta \vdash \Delta \rightrightarrows \Delta_o, \Delta_i}$

$$\frac{}{\Theta_\varnothing, \Theta \vdash \epsilon \rightrightarrows \epsilon, \epsilon} \qquad \frac{\Theta_\varnothing, \Theta \vdash \Delta \rightrightarrows \Delta_o, \Delta_i \quad \exists \alpha.\ \alpha \in (\mathbf{ftv}(\tau_l) \cup \mathbf{ftv}(\tau_r)) \backslash \Theta_\varnothing \backslash \Theta}{\Theta_\varnothing, \Theta \vdash \Delta\ \tau_l <: \tau_r \rightrightarrows \Delta_o, \Delta_i\ \tau_l <: \tau_r}$$

$$\frac{\Theta_\varnothing, \Theta \vdash \Delta \rightrightarrows \Delta_o, \Delta_i \quad \exists \alpha.\ \alpha \in (\mathbf{ftv}(\tau_l) \cup \mathbf{ftv}(\tau_r)) \cap \Theta \qquad \mathbf{ftv}(\tau_l) \cup \mathbf{ftv}(\tau_r) \subseteq \Theta_\varnothing \cup \Theta}{\Theta_\varnothing, \Theta \vdash \Delta\ \tau_l <: \tau_r \rightrightarrows \Delta_o\ \tau_l <: \tau_r, \Delta_i}$$

2 MODEL THEORY

Theorem 2.1. Record Extension Preservation

$$\frac{\delta \vDash r : \tau \qquad \nexists e'.\ <l>e' \in r}{\delta \vDash r\ <l>e : \tau}$$

Theorem 2.2. Entry Type Introduction

$$\frac{\delta \vDash e : \tau \qquad \nexists e'.\ <l>e' \in r}{\delta \vDash r\ <l>e : <l>\tau}$$

Theorem 2.3. Model Typing Entry Elimination

$$\frac{\delta \vDash e : <l>\tau}{\delta \vDash e.l : \tau}$$

Theorem 2.4. Entry Type Preservation

$$\frac{\delta \vDash \tau_l <: \tau_r}{\delta \vDash <l>\tau_l <: <l>\tau_r}$$

Theorem 2.5. Model Subtyping Union Type Path Antecedent

$$\frac{}{\delta \vDash (\tau_a -> \tau_r) \& (\tau_b -> \tau_r) <: (\tau_a \mid \tau_b) -> \tau_r)}$$

Theorem 2.6. Model Subtyping Intersection Type Path Consequent

$$\frac{}{\delta \vDash (\tau_r -> \tau_a) \& (\tau_r -> \tau_b) <: \tau_r -> (\tau_a \& \tau_b)}$$

Theorem 2.7. Model Subtyping Intersection Type Entry Consequent

$$\frac{}{\delta \vDash (<l>\tau_a) \& (<l>\tau_b) <: <l>(\tau_a \& \tau_b)}$$

Theorem 2.8. Model Subtyping Transitivity

$$\frac{\delta \vDash \tau <: \tau' \qquad \delta \vDash \tau' <: \tau''}{\delta \vDash \tau <: \tau''}$$

Theorem 2.9. Model Typing Path Elimination

$$\frac{\delta \vDash e_f : \tau \text{->} \tau' \qquad \delta \vDash e_a : \tau}{\delta \vDash e_f\ (e_a)\ : \tau'}$$

Theorem 2.10. Path Type Preservation

$$\frac{\delta \vDash \tau_x <: \tau_p \qquad \delta \vDash \tau_q <: \tau_y}{\delta \vDash \tau_p \text{->} \tau_q <: \tau_x \text{->} \tau_y}$$

Theorem 2.11. Annotated Binding Introduction

$$\frac{\delta \vDash e : \tau \qquad \delta \vDash e'[x/e] : \tau'}{\delta \vDash \texttt{def}\ \ x \texttt{:} \tau\ \texttt{=}\ e\ \texttt{in}\ e' : \tau'}$$

Theorem 2.12. Model Subtyping Sequence Introduction

$$\frac{\delta \vDash \Delta \qquad \delta \vDash \Delta'}{\delta \vDash \Delta \cup \Delta'}$$

Theorem 2.13. Model Subtyping Sequence Elimination

$$\frac{\delta \vDash \Delta' \qquad \Delta \subseteq \Delta'}{\delta \vDash \Delta}$$

Theorem 2.14. Model Subtyping Sequence Interpretation Independence

$$\frac{\mathbf{dom}(\delta') \cap \mathbf{ftv}(\Delta) = \emptyset}{\delta \vDash \Delta \iff \delta \oplus \delta' \vDash \Delta}$$

Theorem 2.15. Model Typing Interpretation Independence

$$\frac{\mathbf{dom}(\delta') \cap \mathbf{ftv}(\tau) = \emptyset}{\delta \vDash e : \tau \iff \delta \oplus \delta' \vDash e : \tau}$$

Theorem 2.16. Subtyping Interpretation Independence

$$\frac{\mathbf{dom}(\delta') \cap \mathbf{ftv}(\tau) = \emptyset \qquad \mathbf{dom}(\delta') \cap \mathbf{ftv}(\tau') = \emptyset}{\delta \vDash \tau <: \tau' \iff \delta \oplus \delta' \vDash \tau <: \tau'}$$

Theorem 2.17. Subtyping Interpretation Reordering

$$\frac{\mathbf{dom}(\delta_1) \cap \mathbf{dom}(\delta_2) = \emptyset}{\delta_0 \oplus \delta_1 \oplus \delta_2 \oplus \delta_3 \vDash \tau <: \tau' \iff \delta_0 \oplus \delta_2 \oplus \delta_1 \oplus \delta_3 \vDash \tau <: \tau'}$$

Theorem 2.18. Subtyping Environment Interpretation Reordering

$$\frac{\mathbf{dom}(\delta_1) \cap \mathbf{dom}(\delta_2) = \emptyset}{\delta_0 \oplus \delta_1 \oplus \delta_2 \oplus \delta_3 \vDash \Delta \iff \delta_0 \oplus \delta_2 \oplus \delta_1 \oplus \delta_3 \vDash \Delta}$$

Theorem 2.19. Subtyping Environment Interpretation Independence

$$\frac{\mathbf{dom}(\delta') \cap \mathbf{ftv}(\Delta) = \emptyset}{\delta \vDash \Delta \iff \delta \oplus \delta' \vDash \Delta}$$

Theorem 2.20. Model Typing Sequence Interpretation Independence

$$\frac{\mathbf{dom}(\delta') \cap \mathbf{ftv}(\Gamma) = \emptyset}{\delta, \sigma \vDash \Gamma \iff \delta \oplus \delta', \sigma \vDash \Gamma}$$

Theorem 2.21. Domain Extension

$$\frac{\mathbf{dom}(\delta_l) = \Theta_l \qquad \mathbf{dom}(\delta_r) = \Theta_r}{\mathbf{dom}(\delta_l \oplus \delta_r) = \Theta_l \cup \Theta_r}$$

Theorem 2.22. Subsumption

$$\frac{\delta \vDash \tau <: \tau' \qquad \delta \vDash e : \tau}{\delta \vDash e : \tau'}$$

Theorem 2.23. Model Subtyping Union Type Elimination

$$\frac{\delta \vDash \tau_l <: \tau \qquad \delta \vDash \tau_r <: \tau}{\delta \vDash \tau_l \mid \tau_r <: \tau}$$

Theorem 2.24. Model Subtyping Indexed Union Type Elimination

$$\frac{(\forall \delta' \,.\, \mathbf{dom}(\delta') \subseteq \Theta \implies \delta \oplus \delta' \vDash \Delta \implies \delta \oplus \delta' \vDash \tau <: \tau') \qquad \Theta \cap \mathbf{ftv}(\tau') = \emptyset}{\delta \vDash \mathtt{EXI}\,[\Theta]\,\Delta\,\tau <: \tau'}$$

Theorem 2.25. Model Subtyping Union Type Elimination

$$\frac{\delta \vDash \tau <: \tau_l \qquad \delta \vDash \tau <: \tau_r}{\delta \vDash \tau <: \tau_l \,\&\, \tau_r}$$

Theorem 2.26. Model Subtyping Indexed Intersection Type Introduction

$$\frac{(\forall \delta' \,.\, \mathbf{dom}(\delta') \subseteq \Theta \implies \delta \oplus \delta' \vDash \Delta \implies \delta \oplus \delta' \vDash \tau <: \tau') \qquad \Theta \cap \mathbf{ftv}(\tau') = \emptyset}{\delta \vDash \tau <: \mathtt{ALL}\,[\Theta]\,\Delta\,\tau}$$

Theorem 2.27. Least Fixed Point Type Elimination

$$\frac{\delta \vDash e : \texttt{LFP}\,[\alpha]\,\tau}{\delta\ \alpha/\texttt{TOP} \vDash e : \tau}$$

Theorem 2.28. Model Subtyping Induction

$$\frac{(\forall \tau\ .\ \delta \vDash \tau <: \tau_r \implies \delta\ \alpha/\tau \vDash \tau_l <: \tau_r) \qquad \alpha \notin \mathbf{ftv}(\tau_r)}{\delta \vDash \texttt{LFP}\,[\alpha]\,\tau_l <: \tau_r}$$

Theorem 2.29. Left Congruence

$$\frac{}{(e_l, e_r)\texttt{.left} \cong e_l}$$

Theorem 2.30. Right Congruence

$$\frac{}{(e_l, e_r)\texttt{.right} \cong e_r}$$

Theorem 2.31. Path Type Elimination Loop

$$\frac{\delta \vDash e : \alpha \texttt{->} \tau}{\delta \vDash \texttt{loop}\,(e)\ : \tau}$$

Theorem 2.32. Path Type Introduction

$$\frac{\forall e_a\ .\ \delta \vDash e_a : \tau_l \implies \delta \vDash e_f\,(e_a)\ : \tau_r}{\delta \vDash e_f : \tau_l \texttt{->} \tau_r}$$

Theorem 2.33. Model Subtyping Difference Elimination

$$\frac{\delta \vDash \tau_0 <: \tau_1 \mid \tau_2}{\delta \vDash \tau_0 \setminus \tau_1 <: \tau_2}$$

Theorem 2.34. Model Subtyping Union Right Introduction

$$\frac{}{\delta \vDash \tau_l <: \tau_l \mid \tau_r}$$

Theorem 2.35. Model Subtyping Union Left Introduction

$$\frac{}{\delta \vDash \tau_r <: \tau_l \mid \tau_r}$$

Theorem 2.36. Model Subtyping Intersection Left Elimination

$$\frac{}{\delta \vDash \tau_l \& \tau_r <: \tau_r}$$

Theorem 2.37. Model Subtyping Intersection Right Elimination

$$\frac{}{\delta \vDash \tau_l \& \tau_r <: \tau_l}$$

Theorem 2.38. Model Typing Union Reorder

$$\frac{}{\delta \vDash e : \tau_a \mid \tau_b \iff \delta \vDash e : \tau_b \mid \tau_a}$$

Theorem 2.39. Model Subtyping Difference Introduction

$$\frac{(\nexists e \,.\, \delta \vDash e : \tau_0 \ \wedge \ \delta \vDash e : \tau_2) \qquad \delta \vDash \tau_0 <: \tau_1}{\delta \vDash \tau_0 <: \tau_1 \setminus \tau_2}$$

Theorem 2.40. Model Subtyping Indexed Union Introduction

$$\frac{\delta \vDash \tau_l <: \tau_r \qquad \delta \vDash \Delta}{\delta \vDash \tau_l <: \texttt{EXI}\,[\Theta]\,\Delta\ \tau_r}$$

Theorem 2.41. Model Subtyping Indexed Intersection Elimination

$$\frac{\delta \vDash \tau_l <: \tau_r \qquad \delta \vDash \Delta}{\delta \vDash \texttt{ALL}\,[\Theta]\,\Delta\ \tau_l <: \tau_r}$$

3 PROOF THEORY

Theorem 3.1. Type Substitution Correspondence

$$\frac{}{\delta \vDash e : \tau[\alpha/\tau'] \iff \delta\ \alpha/\tau' \vDash e : \tau}$$

Theorem 3.2. Proof Typing Containment

$$\frac{\Theta, \Delta, \Gamma \vdash e : \tau \dashv \Theta', \Delta'}{\mathbf{ftv}(\Gamma) \subseteq \mathbf{ftv}(\Delta) \ \wedge \ \Delta \subseteq \Delta' \ \wedge \ \mathbf{ftv}(\tau) \subseteq \mathbf{ftv}(\Delta')}$$

Theorem 3.3. *proof subtyping containment*

$$\frac{\Theta, \Delta \vdash \tau <: \tau' \dashv \Theta', \Delta'}{\Delta \subseteq \Delta' \ \wedge \ \mathbf{ftv}(\tau) \subseteq \mathbf{ftv}(\Delta') \ \wedge \ \mathbf{ftv}(\tau') \subseteq \mathbf{ftv}(\Delta')}$$

Theorem 3.4. Proof Subtyping Environment Containment

$$\frac{\Theta, \Delta \vdash \Delta_q \dashv \Theta', \Delta'}{\Delta \subseteq \Delta' \ \wedge \ \mathbf{ftv}(\Delta_q) \subseteq \mathbf{ftv}(\Delta')}$$

Theorem 3.5. Proof Typing Skolem Freshness

$$\frac{\Theta, \Delta, \Gamma \vdash e : \tau \dashv \Theta', \Delta'}{\Theta' \backslash \Theta \cap \mathbf{ftv}(\Delta) = \emptyset}$$

Theorem 3.6. Proof Subtyping Skolem Freshness

$$\frac{\Theta, \Delta \vdash \tau <: \tau' \dashv \Theta', \Delta'}{\Theta' \backslash \Theta \cap \mathbf{ftv}(\Delta) = \emptyset}$$

Theorem 3.7. Proof Subtyping Sequence Skolem Freshness

$$\frac{\Theta, \Delta \vdash \Delta_q \dashv \Theta', \Delta'}{\Theta' \backslash \Theta \cap \mathbf{ftv}(\Delta) = \emptyset}$$

Theorem 3.8. Function Type Explosion Soundness

$$\frac{\Theta, \Delta \vdash \tau <: \alpha \to \Pi}{\begin{array}{l}\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi \implies \\ (\exists \delta \,.\, \mathbf{dom}(\delta) \cong \Theta^\dagger \;\wedge\; (\forall \delta' \,.\, \delta' \oplus \delta \vDash \Delta \cup \Delta^\dagger \implies \delta' \oplus \delta \vDash \tau <: \alpha \text{->} \tau^\dagger))\end{array}}$$

Theorem 3.9. Function Type Explosion Skolem Freshness

$$\frac{\Theta, \Delta \vdash \tau <: \alpha \to \Pi}{\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi \implies \Theta^\dagger \cap \Theta = \emptyset \;\wedge\; \Theta^\dagger \cap \mathbf{ftv}(\Delta) = \emptyset}$$

Theorem 3.10. Path Sequence Typing Soundness

$$\frac{\Theta, \Delta, \Gamma \vdash f : \Pi \sim \Xi}{\begin{array}{l}\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi \implies \\ (\exists \delta \,.\, \mathbf{dom}(\delta) \cong \Theta^\dagger \;\wedge\; (\forall \delta', \sigma \,.\, \delta' \oplus \delta \vDash \Delta \cup \Delta^\dagger \implies \delta', \sigma \vDash \Gamma \implies \delta' \oplus \delta \vDash f[\sigma] : \tau^\dagger))\end{array}}$$

TODO: proof of this depends on consistency of subtyping environment when checking skolem variables

TODO: to prevent a path expression from having an empty type

Theorem 3.11. Function Lifting Skolem Freshness

$$\frac{\Theta, \Delta, \Gamma \vdash f : \Pi \sim \Xi}{(\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi \implies \Theta^\dagger \cap \Theta = \emptyset \;\wedge\; \Theta^\dagger \cap \mathbf{ftv}(\Delta) = \emptyset \;\wedge\; \Theta^\dagger \cap \mathbf{ftv}(\Gamma) = \emptyset}$$

Theorem 3.12. Schema Sequence Duality Correspondence

$$\frac{\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu \qquad \delta \vDash \Delta}{\begin{array}{c}(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \wedge \\ (\forall \delta' \,.\, \mathbf{dom}(\delta') \cap \mathbf{ftv}(\Delta) = \epsilon \implies \\ \delta\ \alpha_\nu/\texttt{BOT} \oplus \delta' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta\ \alpha_\nu/\texttt{BOT} \oplus \delta' \oplus \delta^\dagger \vDash e_f : \tau^\dagger)) \\ \Longleftrightarrow \\ (\forall e_a, e_r \,.\, (\exists \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\mu \wedge (\forall \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \implies \\ \exists \delta' \,.\, \mathbf{dom}(\delta') \cap \mathbf{ftv}(\Delta) = \epsilon \wedge \\ \delta\ \alpha_\mu/\texttt{TOP} \oplus \delta' \oplus \delta^\dagger \vDash \Delta^\dagger \wedge \delta\ \alpha_\mu/\texttt{TOP} \oplus \delta' \oplus \delta^\dagger \vDash (e_a, e_r) : \tau^\dagger)) \implies e_r \cong e_f\ (e_a))\end{array}}$$

Theorem 3.13. Packing Correspondence

$$\frac{\mathbf{pack}^+(\dot\Theta \vdash \Pi) = \tau \qquad \delta \vDash \Delta \qquad \dot\Theta \subseteq \mathbf{dom}(\delta)}{\begin{array}{c}(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi \implies \\ (\exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \backslash \dot\Theta \wedge \\ (\forall \delta' \,.\, \mathbf{dom}(\delta') \cap \mathbf{ftv}(\Delta) = \emptyset \implies \delta \oplus \delta' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta \oplus \delta' \oplus \delta^\dagger \vDash e : \tau^\dagger)) \\ \Longleftrightarrow \\ \delta \vDash e : \tau\end{array}}$$

$$\frac{\mathbf{pack}^-(\dot\Theta \vdash \Pi) = \tau \qquad \delta \vDash \Delta \qquad \dot\Theta \subseteq \mathbf{dom}(\delta)}{\begin{array}{c}(\exists \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi \wedge \\ (\forall \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \backslash \dot\Theta \implies \\ (\exists \delta' \,.\, \mathbf{dom}(\delta') \cap \mathbf{ftv}(\Delta) = \emptyset \wedge \delta \oplus \delta' \oplus \delta^\dagger \vDash \Delta^\dagger \wedge \delta \oplus \delta' \oplus \delta^\dagger \vDash e : \tau^\dagger)) \\ \Longleftrightarrow \\ \delta \vDash e : \tau\end{array}}$$

Theorem 3.14. Subtyping Factorization Correspondence

$$\frac{\Vdash \tau <: <l>\tau'}{\forall e' \,.\, \delta \vDash e' : \tau' \iff (\exists e \,.\, e\,.\,l \cong e' \wedge \delta \vDash e : \tau)}$$

Theorem 3.15. Proof Subtyping Sequence Soundness

$$\frac{\Theta, \Delta \vdash \Delta_q \dashv \Theta', \Delta'}{\exists \delta \,.\, \mathbf{dom}(\delta) \subseteq \Theta' \backslash \Theta \wedge (\forall \delta' \,.\, \delta' \oplus \delta \vDash \Delta' \implies \delta' \oplus \delta \vDash \Delta_q)}$$

Theorem 3.16. Proof Decidable Subtyping Sequence Uniqueness

$$\frac{\Theta, \Delta \vdash \Omega \dashv \Theta', \Delta'}{\forall \Theta'', \Delta'' \,.\, \Theta, \Delta \vdash \Omega \dashv \Theta'', \Delta'' \implies (\Theta'', \Delta'') = (\Theta', \Delta')}$$

Theorem 3.17. Proof Decidable Subtyping Sequence Completeness

$$\frac{\Theta, \Delta \vdash \Omega \dashv \Theta', \Delta' \qquad \delta' \vDash \Delta'}{\forall \delta^\dagger \,.\, \delta' \oplus \delta^\dagger \vDash \Omega \implies \delta' \oplus \delta^\dagger \vDash \Delta'}$$

Theorem 3.18. Proof Decidable Subtyping Solution Completeness

$$\frac{\Theta, \Delta \vdash \tau <: \eta \dashv \Theta', \Delta' \qquad \mathbf{ftv}(\eta) \subseteq \emptyset \qquad \delta' \vDash \Delta'}{\forall \delta^\dagger \,.\, \delta' \oplus \delta^\dagger \vDash \tau <: \eta \implies \delta' \oplus \delta^\dagger \vDash \Delta'}$$

Theorem 3.19. Proof Decidable Subtyping Existential Completeness

$$\frac{\delta \vDash \tau <: \eta \qquad \delta \vDash \Delta}{\exists \Theta', \Delta' \,.\, \Theta, \Delta \vdash \tau <: \eta \dashv \Theta', \Delta'}$$

Theorem 3.20. Proof Subtyping Interpretation Independence

$$\frac{\Theta, \Delta \vdash \tau_l <: \tau_r \dashv \Theta', \Delta' \qquad \delta' \vDash \Delta'}{\forall \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \mathbf{ftv}(\Delta) \implies \delta' \oplus \delta^\dagger \vDash \Delta \implies \delta' \oplus \delta^\dagger \vDash \Delta'}$$

Theorem 3.21. Proof Typing Annotation Soudness

$$\frac{\Theta, \Delta, \Gamma \vdash \texttt{def}\ \ x : \tau_a\ \ =\ \ e\ \ \texttt{in}\ \ e' : \tau' \dashv \Theta', \Delta'}{\forall \delta, \sigma \,.\, \delta, \sigma \vDash \Gamma \implies \delta \vDash e : \tau_a}$$

Theorem 3.22. Proof Typing Substance

$$\frac{\Theta, \Delta \vdash e : \tau \dashv \Theta', \Delta'}{\forall \delta \,.\, \mathbf{dom}(\delta) \subseteq \Theta \implies \exists \delta' \,.\, \delta \oplus \delta' \vDash \Delta \implies \exists \delta' \,.\, \delta \oplus \delta' \vDash \Delta'}$$

Theorem 3.23. Proof Subtyping Substance

$$\frac{\Theta, \Delta \vdash \tau_l <: \tau_r \dashv \Theta', \Delta'}{\forall \delta \,.\, \mathbf{dom}(\delta) \subseteq \Theta \implies \exists \delta' \,.\, \delta \oplus \delta' \vDash \Delta \implies \exists \delta' \,.\, \delta \oplus \delta' \vDash \Delta'}$$

TODO: note how this is necessary for safety of function type inference
TODO: so that bottom can't be inferred for a function expression

Theorem 3.24. Polar Substitution Consistency

$$\frac{\delta \vDash \Delta[\alpha/\tau]^+}{\exists \delta' \,.\, \delta' \vDash \Delta\ \alpha <: \tau} \qquad \frac{\delta \vDash \Delta[\alpha/\tau]^-}{\exists \delta' \,.\, \delta' \vDash \Delta\ \tau <: \alpha}$$

TODO: note how this is necessary to prove subtyping consistency
TODO: note how consitency is necessary to prove parts of subtyping soundness

Theorem 3.25. Type Substitution (Positive) Elimination

$$\frac{\tau_l[\alpha/\tau_r]^+ = \tau^\dagger \qquad \delta \vDash \tau^\dagger <: \tau_r \qquad \alpha \notin \mathbf{ftv}(\tau^\dagger) \qquad \alpha \notin \mathbf{ftv}(\tau_r)}{\forall \tau \,.\, \delta \vDash \tau <: \tau_r \implies \delta\ \alpha/\tau \vDash \tau_l <: \tau_r}$$

Theorem 3.26. Proof Subtyping Skolem Variable Elimination Soundness

$$\frac{\Theta, \Delta \vdash \alpha <: \tau\ \checkmark}{\forall \delta \,.\, \delta \vDash \Delta \implies \delta \vDash \alpha <: \tau}$$

Theorem 3.27. Proof Subtyping Soundness

$$\frac{\Theta, \Delta \vdash \tau_l <: \tau_r \dashv \Theta', \Delta'}{\exists \delta' \,.\, \mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta \ \wedge\ (\forall \delta \,.\, \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_l <: \tau_r)}$$

Induct over Definition **??**.
· Assume $(\Theta, \Delta \vdash \tau <: \tau \dashv \Theta, \Delta)$
  · Derive $(\mathbf{dom}(\epsilon) \subseteq \Theta \backslash \Theta)$
  · For δ, assume $(\delta \oplus \epsilon \vDash \Delta)$

- Derive $(\forall e\ .\ \delta \oplus \epsilon \vDash e : \tau \implies \delta \oplus \epsilon \vDash e : \tau)$
- Derive $(\delta \oplus \epsilon \vDash \tau <: \tau)$ [Def. **??**]

· Assume $(\Theta, \Delta \vdash <l>\tau_l <: <l>\tau_r \dashv \Theta', \Delta')$
- Derive $(\Theta, \Delta \vdash \tau_l <: \tau_r \dashv \Theta', \Delta')$ [Def. **??**]
- Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
- And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_l <: \tau_r)$ [IH]
- For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: \tau_r)$ [P]
  - Derive $(\delta \oplus \delta' \vDash <l>\tau_l <: <l>\tau_r)$ [Thm. 2.4]

· Assume $(\Theta, \Delta \vdash \tau_p \text{->} \tau_q <: \tau_x \text{->} \tau_y \dashv \Theta'', \Delta'')$
- Derive $(\Theta, \Delta \vdash \tau_x <: \tau_p \dashv \Theta', \Delta')$ [Def. **??**]
- And $(\Theta', \Delta' \vdash \tau_q <: \tau_y \dashv \Theta'', \Delta'')$ [Def. **??**]
- Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
- And P0: $(\forall \delta\ .\ \delta \oplus \delta_0 \vDash \Delta' \implies \delta \oplus \delta_0 \vDash \tau_x <: \tau_p)$ [IH]
- Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [IH]
- And P1: $(\forall \delta\ .\ \delta \oplus \delta_1 \vDash \Delta' \implies \delta \oplus \delta_1 \vDash \tau_q <: \tau_y)$ [IH]
- Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$
- For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.12]
  - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.6, Thm. 2.19]
  - Derive $(\delta \oplus \delta_0 \vDash \tau_x <: \tau_p)$ [P0]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_x <: \tau_p)$ [Thm. 3.3, Thm. 2.16]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_q <: \tau_y)$ [P1]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_p \text{->} \tau_q <: \tau_x \text{->} \tau_y)$ [Thm 2.10]

· Assume $(\Theta, \Delta \vdash \tau_l <: (\tau_a \mid \tau_b) \text{->} \tau_r \dashv \Theta', \Delta')$
- Derive $(\Theta, \Delta \vdash \tau_l <: (\tau_a \text{->} \tau_r) \& (\tau_b \text{->} \tau_r) \dashv \Theta', \Delta')$ [Def. **??**]
- Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
- And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_l <: (\tau_a \text{->} \tau_r) \& (\tau_b \text{->} \tau_r))$ [IH]
- For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: (\tau_a \text{->} \tau_r) \& (\tau_b \text{->} \tau_r))$ [P]
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: (\tau_a \mid \tau_b) \text{->} \tau_r))$ [Thm. 2.5, Thm. 2.8]

· Assume $(\Theta, \Delta \vdash \tau_l <: \tau_r \text{->} (\tau_a \& \tau_b) \dashv \Theta', \Delta')$
- Derive $(\Theta, \Delta \vdash \tau_l <: (\tau_r \text{->} \tau_a) \& (\tau_r \text{->} \tau_b) \dashv \Theta', \Delta')$ [Def. **??**]
- Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
- And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_l <: (\tau_r \text{->} \tau_a) \& (\tau_r \text{->} \tau_b))$ [IH]
- For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: (\tau_r \text{->} \tau_a) \& (\tau_r \text{->} \tau_b))$ [P]
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: \tau_r \text{->} (\tau_a \& \tau_b)))$ [Thm. 2.6, Thm. 2.8]

· Assume $(\Theta, \Delta \vdash \tau <: <l>(\tau_a \& \tau_b) \dashv \Theta', \Delta')$
- Derive $(\Theta, \Delta \vdash \tau <: (<l>\tau_a) \& (<l>\tau_b) \dashv \Theta', \Delta')$ [Def. **??**]
- Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
- And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau <: (<l>\tau_a) \& (<l>\tau_b))$ [IH]
- For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
  - Derive $(\delta \oplus \delta' \vDash \tau <: (<l>\tau_a) \& (<l>\tau_b))$ [P]
  - Derive $(\delta \oplus \delta' \vDash \tau <: <l>(\tau_a \& \tau_b)))$ [Thm. 2.7, Thm. 2.8]

· Assume $(\Theta, \Delta \vdash \tau_a \mid \tau_b <: \tau \dashv \Theta'', \Delta'')$
- Derive $(\Theta, \Delta \vdash \tau_a <: \tau \dashv \Theta', \Delta')$ [Def. **??**]
- And $(\Theta', \Delta' \vdash \tau_b <: \tau \dashv \Theta'', \Delta'')$ [Def. **??**]
- Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
- And P0: $(\forall \delta\ .\ \delta \oplus \delta_0 \vDash \Delta' \implies \delta \oplus \delta_0 \vDash \tau_a <: \tau)$ [IH]
- Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [IH]
- And P1: $(\forall \delta\ .\ \delta \oplus \delta_1 \vDash \Delta' \implies \delta \oplus \delta_1 \vDash \tau_b <: \tau)$ [IH]

  - Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$
  - For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.12]
    - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.6, Thm. 2.19]
    - Derive $(\delta \oplus \delta_0 \vDash \tau_a <: \tau)$ [P0]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_a <: \tau)$ [Thm. 3.3, Thm. 2.16]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_b <: \tau)$ [P1]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_a \mid \tau_b <: \tau)$ [Thm 2.23]
- Assume $(\Theta, \Delta \vdash \texttt{EXI[}\Theta_l\texttt{]}\,\Omega\texttt{:}\,\tau_l <: \tau_r \dashv \Theta'', \Delta'')$
  - Derive $(\Theta_l \cap \mathbf{ftv}(\Delta) \subseteq \emptyset)$ [Def. **??**]
  - And $(\Theta_l \cap \mathbf{ftv}(\tau_r) \subseteq \emptyset)$ [Def. **??**]
  - And $(\forall \tau, \eta \,.\, (\tau \texttt{<:} \eta) \in \Omega \implies \mathbf{ftv}(\eta) \subseteq \emptyset)$ [Def. **??**]
  - And $(\Theta, \Delta \vdash \Omega \dashv \Theta', \Delta')$ [Def. **??**]
  - And $(\Theta' \cup \Theta_l, \Delta' \vdash \tau_l <: \tau_r \dashv \Theta'', \Delta'')$ [Def. **??**]
  - Let δ_0 be, such that $(\mathbf{dom}(\delta) \subseteq \Theta' \backslash \Theta)$ [Thm. 3.15]
  - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash (\Theta' \cup \Theta_l))$ [IH]
  - And P1: $(\forall \delta \,.\, \delta \oplus \delta_1 \vDash \Delta'' \implies \delta \oplus \delta_1 \vDash \tau_l <: \tau_r)$ [IH]
  - Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta \backslash \Theta_l)$
  - Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$
  - For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 2.12]
    - Derive $(\forall \delta_l \,.\, \mathbf{dom}(\delta_l) \subseteq \mathbf{ftv}(\Delta') \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Delta' \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Delta'')$ [Thm 3.20]
    - Derive $(\forall \delta_l \,.\, \mathbf{dom}(\delta_l) \subseteq \mathbf{ftv}(\Delta') \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Delta'')$ [Thm 3.17]
    - Derive $(\Theta_l \subseteq \mathbf{ftv}(\Delta''))$ [Thm. 3.4]
    - Derive $(\forall \delta_l \,.\, \mathbf{dom}(\delta_l) \subseteq \Theta_l \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Delta'')$
    - Derive $(\forall \delta_l \,.\, \mathbf{dom}(\delta_l) \subseteq \Theta_l \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_l \oplus \delta_1 \vDash \Delta'')$ [Thm. Thm. 2.17]
    - Derive $(\forall \delta_l \,.\, \mathbf{dom}(\delta_l) \subseteq \Theta_l \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_l \oplus \delta_1 \vDash \tau_l <: \tau_r)$ [P1]
    - Derive $(\forall \delta_l \,.\, \mathbf{dom}(\delta_l) \subseteq \Theta_l \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_l \vDash \tau_l <: \tau_r)$ [Thm. 2.17]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \texttt{EXI[}\Theta_l\texttt{]}\,\Omega\texttt{:}\,\tau_l <: \tau_r)$ [Thm. 2.24]
- Assume $(\Theta, \Delta \vdash \tau <: \tau_a \texttt{\&} \tau_b \dashv \Theta'', \Delta'')$
  - Derive $(\Theta, \Delta \vdash \tau <: \tau_a \dashv \Theta', \Delta')$ [Def. **??**]
  - Derive $(\Theta', \Delta' \vdash \tau <: \tau_b \dashv \Theta'', \Delta'')$ [Def. **??**]
  - Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P0: $(\forall \delta \,.\, \delta \oplus \delta_0 \vDash \Delta' \implies \delta \oplus \delta_0 \vDash \tau <: \tau_a)$ [IH]
  - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [IH]
  - And P1: $(\forall \delta \,.\, \delta \oplus \delta_1 \vDash \Delta' \implies \delta \oplus \delta_1 \vDash \tau <: \tau_b)$ [IH]
  - Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$
  - For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
    - Derive $(\delta' \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.12]
    - Derive $(\delta' \oplus \delta_0 \vDash \Delta')$ [Thm. 3.6, Thm. 2.19]
    - Derive $(\delta \oplus \delta_0 \vDash \tau <: \tau_a)$ [P0]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau <: \tau_a)$ [Thm. 3.3, Thm. 2.16]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau <: \tau_b)$ [P1]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau <: \tau_a \texttt{\&} \tau_b)$ [Thm 2.25]
- Assume $(\Theta, \Delta \vdash \tau_l <: \texttt{ALL[}\Theta_r\texttt{]}\,\Omega\texttt{:}\,\tau_r \dashv \Theta'', \Delta'')$
  - Derive $(\Theta_r \cap \mathbf{ftv}(\Delta) \subseteq \emptyset)$ [Def. **??**]
  - And $(\Theta_r \cap \mathbf{ftv}(\tau_l) \subseteq \emptyset)$ [Def. **??**]
  - And $(\forall \tau, \eta \,.\, (\tau \texttt{<:} \eta) \in \Omega \implies \mathbf{ftv}(\eta) \subseteq \emptyset)$ [Def. **??**]
  - And $(\Theta, \Delta \vdash \Omega \dashv \Theta', \Delta')$ [Def. **??**]
  - And $(\Theta' \cup \Theta_r, \Delta' \vdash \tau_l <: \tau_r \dashv \Theta'', \Delta'')$ [Def. **??**]
  - Let δ_0 be, such that $(\mathbf{dom}(\delta) \subseteq \Theta' \backslash \Theta)$ [Thm. 3.15]
  - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash (\Theta' \cup \Theta_r))$ [IH]

- And P1: $(\forall \delta\ .\ \delta \oplus \delta_1 \vDash \Delta'' \implies \delta \oplus \delta_1 \vDash \tau_l <: \tau_r)$ [IH]
- Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta''\backslash\Theta\backslash\Theta_r)$
- Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta''\backslash\Theta)$
- For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 2.12]
  - Derive $(\forall \delta_r\ .\ \mathbf{dom}(\delta_r) \subseteq \mathbf{ftv}(\Delta') \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Delta' \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Delta'')$ [Thm 3.20]
  - Derive $(\forall \delta_r\ .\ \mathbf{dom}(\delta_r) \subseteq \mathbf{ftv}(\Delta') \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Delta'')$ [Thm 3.17]
  - Derive $(\Theta_r \subseteq \mathbf{ftv}(\Delta'))$ [Thm. 3.4]
  - Derive $(\forall \delta_r\ .\ \mathbf{dom}(\delta_r) \subseteq \Theta_r \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Delta'')$
  - Derive $(\forall \delta_r\ .\ \mathbf{dom}(\delta_r) \subseteq \Theta_r \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_r \oplus \delta_1 \vDash \Delta'')$ [Thm. Thm. 2.17]
  - Derive $(\forall \delta_r\ .\ \mathbf{dom}(\delta_r) \subseteq \Theta_r \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_r \oplus \delta_1 \vDash \tau_l <: \tau_r)$ [P1]
  - Derive $(\forall \delta_r\ .\ \mathbf{dom}(\delta_r) \subseteq \Theta_r \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \Omega \implies \delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_r \vDash \tau_l <: \tau_r)$ [Thm. 2.17]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_l <: \texttt{ALL}\,[\Theta_r]\,\Omega : \tau_r)$ [Thm. 2.26]
- Assume $(\Theta, \Delta \vdash \alpha <: \tau \dashv \Theta', \Delta'\ \alpha <: \tau)$ and $(\alpha \notin \Theta)$
  - Derive $(\Theta, \Delta \vdash \Delta[\alpha/\tau]^+\backslash\Delta \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ be, such that $(\mathbf{dom}(\delta) \subseteq \Theta'\backslash\Theta)$ [Def. **??**]
  - For δ', assume $(\delta \oplus \delta \vDash \Delta'\ \alpha <: \tau)$
    - Derive $(\delta \oplus \delta \vDash \alpha <: \tau)$ [Thm. **??**]
- Assume $(\Theta, \Delta \vdash \alpha <: \tau \dashv \Theta, \Delta\ \alpha <: \tau)$ and $(\alpha \in \Theta)$
  - Derive $(\Theta, \Delta \vdash \alpha <: \tau\ \checkmark)$ [Def. **??**]
  - And $(\nexists \alpha'\ .\ \tau = \alpha')$ [Def. **??**]
  - For δ', assume $(\delta \oplus \epsilon \vDash \Delta\ \alpha <: \tau)$
    - Derive $(\delta \oplus \epsilon \vDash \alpha <: \tau)$ [Def. **??**]
- Assume $(\Theta, \Delta \vdash \tau <: \alpha \dashv \Theta', \Delta'\ \tau <: \alpha)$ and $(\alpha \notin \Theta)$
  - Derive $(\Theta, \Delta \vdash \Delta[\alpha/\tau]^-\backslash\Delta \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ be, such that $(\mathbf{dom}(\delta) \subseteq \Theta'\backslash\Theta)$ [Def. **??**]
  - For δ', assume $(\delta \oplus \delta \vDash \Delta'\ \tau <: \alpha)$
    - Derive $(\delta \oplus \delta \vDash \tau <: \alpha)$ [Thm. **??**]
- Assume $(\Theta, \Delta \vdash \tau <: \alpha \dashv \Theta, \Delta)$ and $(\alpha \in \Theta)$
  - Derive $(\Theta, \Delta \vdash \tau <: \alpha\ \checkmark)$ [Def. **??**]
  - And $(\nexists \alpha'\ .\ \tau = \alpha')$ [Def. **??**]
  - For δ', assume $(\delta \oplus \epsilon \vDash \Delta\ \tau <: \alpha)$
    - Derive $(\delta \oplus \epsilon \vDash \tau <: \alpha)$ [Def. **??**]
- Assume $(\Theta, \Delta \vdash \texttt{LFP}\,[\alpha]\,\tau_l <: \tau_r \dashv \Theta', \Delta')$
  - Derive $(\tau_l[\alpha/\tau_r]^+ = \tau^\dagger)$ [Def. **??**]
  - And $(\Theta, \Delta \vdash \tau_l[\alpha/\tau_r]^+ <: \tau_r \dashv \Theta', \Delta')$ [Def. **??**]
  - And $(\alpha \notin \mathbf{ftv}(\tau^\dagger))$ [Def. **??**]
  - And $(\alpha \notin \mathbf{ftv}(\tau_r))$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta'\backslash\Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta \vDash \Delta' \implies \delta \oplus \delta \vDash \tau^\dagger <: \tau_r)$ [IH]
  - For δ, Assume $(\delta \oplus \delta \vDash \Delta')$
    - Derive $(\delta \oplus \delta' \vDash \tau^\dagger <: \tau_r)$ [P]
    - Derive $(\forall \tau\ .\ \delta \oplus \delta' \vDash \tau <: \tau_r \implies \delta' \oplus \delta\ \alpha/\tau \vDash \tau_l <: \tau_r)$ [Thm. 3.25]
    - Derive $(\delta \oplus \delta' \vDash \texttt{LFP}\,[\alpha]\,\tau_l <: \tau_r)$ [Thm. 2.28]
- Assume $(\Theta, \Delta \vdash \tau <: \rho\backslash\eta \dashv \Theta', \Delta')$
  - **TODO: REDO**
- Assume $(\Theta, \Delta \vdash \tau_l <: \texttt{LFP}\,[\alpha]\,\tau_r \dashv \Theta', \Delta')$
  - Derive $(\Theta, \Delta \star \tau_l <: \texttt{LFP}\,[\alpha]\,\tau_r)$ [Def. **??**]
  - And $(\Theta, \Delta \vdash \tau_l <: \tau_r[\alpha/\texttt{LFP}\,[\alpha]\,\tau_r] \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta'\backslash\Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_l <: \tau_r[\alpha/\texttt{LFP}\,[\alpha]\,\tau_r])$ [IH]

- For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: \tau_r[\alpha/\texttt{LFP}[\alpha]\,\tau_r])$ [P]
  - Derive $(\forall e\ .\ \delta \oplus \delta' \vDash e : \tau_r[\alpha/\texttt{LFP}[\alpha]\,\tau_r] \implies \delta \oplus \delta'\ \alpha/\texttt{LFP}[\alpha]\,\tau_r \vDash e : \tau_r)$ [Thm. 3.1]
  - Derive $(\forall e\ .\ \delta \oplus \delta' \vDash e : \tau_r[\alpha/\texttt{LFP}[\alpha]\,\tau_r] \implies \delta \oplus \delta' \vDash e : \texttt{LFP}[\alpha]\,\tau_r)$ [Thm. **??**]
  - Derive $(\delta \oplus \delta' \vDash \tau_r[\alpha/\texttt{LFP}[\alpha]\,\tau_r] <: \texttt{LFP}[\alpha]\,\tau_r)$ [Def. **??**]
  - Derive $(\delta \oplus \delta' \vDash \tau_l <: \texttt{LFP}[\alpha]\,\tau_r)$ [Thm. 2.8]
- Assume $(\Theta, \Delta \vdash \rho \backslash \eta <: \tau \dashv \Theta', \Delta')$
  - Derive $(\Theta, \Delta \vdash \rho <: \eta \mid \tau \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \rho <: \eta \mid \tau)$ [IH]
  - For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
    - Derive $(\delta \oplus \delta' \vDash \rho <: \eta \mid \tau)$ [P]
    - Derive $(\delta \oplus \delta' \vDash \rho \backslash \eta <: \tau)$ [Thm. 2.33]
- Assume $(\Theta, \Delta \vdash \tau <: \tau_l \mid \tau_r \dashv \Theta', \Delta')$
  - Derive $(\Theta, \Delta \vdash \tau <: \tau_l \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau <: \tau_l)$ [IH]
  - For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
    - Derive $(\delta \oplus \delta' \vDash \tau <: \tau_l)$ [P]
    - Derive $(\delta \oplus \delta' \vDash \tau <: \tau_l \mid \tau_r)$ [Thm. 2.34, Thm. 2.8]
- Assume $(\Theta, \Delta \vdash \tau <: \tau_l \mid \tau_r \dashv \Theta', \Delta')$
  - Derive $(\Theta, \Delta \vdash \tau <: \tau_r \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau <: \tau_r)$ [IH]
  - For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
    - Derive $(\delta \oplus \delta' \vDash \tau <: \tau_r)$ [P]
    - Derive $(\delta \oplus \delta' \vDash \tau <: \tau_l \mid \tau_r)$ [Thm. 2.35, Thm. 2.8]
- Assume $(\Theta, \Delta \vdash \tau_l <: \texttt{EXI}[\Theta_r]\,\Delta_r\ \tau_r \dashv \Theta'', \Delta'')$
  - Derive $(\Theta, \Delta \vdash \tau_l <: \tau_r \dashv \Theta', \Delta')$ [Def. **??**]
  - And $(\Theta', \Delta' \vdash \Delta_r \dashv \Theta'', \Delta'')$ [Def. **??**]
  - Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P0: $(\forall \delta'\ .\ \delta \oplus \delta_0 \vDash \Delta' \implies \delta \oplus \delta_0 \vDash \tau_l <: \tau_r)$ [IH]
  - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [Inductive Thm. 3.15]
  - And P1: $(\forall \delta'\ .\ \delta \oplus \delta_1 \vDash \Delta'' \implies \delta \oplus \delta_1 \vDash \Delta_r)$ [Inductive Thm. 3.15]
  - For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.13]
    - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.7, Thm. 2.19]
    - Derive $(\delta \oplus \delta_0 \vDash \tau_l <: \tau_r)$ [P0]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_l <: \tau_r)$ [Thm. 3.3, Thm. 2.16]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta_r)$ [P1]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_l <: \texttt{EXI}[\Theta_r]\,\Delta_r\ \tau_r)$ [Thm 2.40]
- Assume $(\Theta, \Delta \vdash \tau_l \& \tau_r <: \tau \dashv \Theta', \Delta')$
  - Derive $(\Theta, \Delta \vdash \tau_l <: \tau \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_l <: \tau)$ [IH]
  - For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
    - Derive $(\delta \oplus \delta' \vDash \tau_l <: \tau)$ [P]
    - Derive $(\delta \oplus \delta' \vDash \tau_l \& \tau_r <: \tau)$ [Thm. 2.37, Thm. 2.8]
- Assume $(\Theta, \Delta \vdash \tau_l \& \tau_r <: \tau \dashv \Theta', \Delta')$
  - Derive $(\Theta, \Delta \vdash \tau_r <: \tau \dashv \Theta', \Delta')$ [Def. **??**]
  - Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P: $(\forall \delta\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta \oplus \delta' \vDash \tau_r <: \tau)$ [IH]

- For δ, assume $(\delta \oplus \delta' \vDash \Delta')$
  - Derive $(\delta \oplus \delta' \vDash \tau_r <: \tau)$ [P]
  - Derive $(\delta \oplus \delta' \vDash \tau_l \& \tau_r <: \tau)$ [Thm. 2.36, Thm. 2.8]
- Assume $(\Theta, \Delta \vdash \texttt{ALL}[\Theta_l]\Delta_l\ \tau_l <: \tau_r \dashv \Theta'', \Delta'')$
  - Derive $(\Theta, \Delta \vdash \tau_l <: \tau_r \dashv \Theta', \Delta')$ [Def. **??**]
  - And $(\Theta', \Delta' \vdash \Delta_l \dashv \Theta'', \Delta'')$ [Def. **??**]
  - Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P0: $(\forall \delta\ .\ \delta \oplus \delta_0 \vDash \Delta' \implies \delta \oplus \delta_0 \vDash \tau_l <: \tau_r)$ [IH]
  - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [Inductive Thm. 3.15]
  - And P1: $(\forall \delta\ .\ \delta \oplus \delta_1 \vDash \Delta'' \implies \delta \oplus \delta_1 \vDash \Delta_l)$ [Inductive Thm. 3.15]
  - For δ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.13]
    - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.7, Thm. 2.19]
    - Derive $(\delta \oplus \delta_0 \vDash \tau_l <: \tau_r)$ [P0]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau_l <: \tau_r)$ [Thm. 3.3, Thm. 2.16]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta_l)$ [P1]
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \texttt{ALL}[\Theta_l]\Delta_l\ \tau_l <: \tau_r)$ [Thm. 2.41]

□

Theorem 3.28. Proof Typing Soundness

$$\frac{\Theta, \Delta, \Gamma \vdash e : \tau \dashv \Theta', \Delta'}{\exists \delta'\ .\ \mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta\ \wedge\ (\forall \delta, \sigma\ .\ \delta \oplus \delta' \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta' \vDash e[\sigma] : \tau)}$$

Induct over Definition **??**

- Assume $(\Theta, \Delta, \Gamma \vdash \texttt{@} : \texttt{@} \dashv \Theta, \Delta)$
  - Derive $(\mathbf{dom}(\epsilon) \subseteq \Theta \backslash \Theta)$
  - For δ, σ, assume $(\delta \oplus \epsilon \vDash \Delta')$ and $(\delta, \sigma \vDash \Gamma)$
    - Derive $(\delta \oplus \epsilon \vDash \texttt{@} : \texttt{@})$ [Def. **??**]
    - Derive $(\delta \oplus \epsilon \vDash \texttt{@}[\sigma] : \texttt{@})$ [Def. **??**]
- Assume $(\Theta, \Delta, \Gamma \vdash x : \tau \dashv \Theta, \Delta)$
  - Derive $(x : \tau \in \Gamma)$ [Def. **??**]
  - Derive $(\mathbf{dom}(\epsilon) \subseteq \Theta \backslash \Theta)$
  - For δ, σ, assume $(\delta \oplus \epsilon \vDash \Delta')$ and $(\delta, \sigma \vDash \Gamma)$
    - Derive $(\delta \oplus \epsilon \vDash x[\sigma] : \tau)$ [Def. **??**]
- Assume $(\Theta, \Delta, \Gamma \vdash \epsilon : \texttt{TOP} \dashv \Theta, \Delta)$
  - Derive $(\mathbf{dom}(\epsilon) \subseteq \Theta \backslash \Theta)$
  - For δ, σ, assume $(\delta \oplus \epsilon \vDash \Delta')$ and $(\delta, \sigma \vDash \Gamma)$
    - Derive $(\delta \oplus \epsilon \vDash \epsilon : \texttt{TOP})$ [Def. **??**]
    - Derive $(\delta \oplus \epsilon \vDash \epsilon[\sigma] : \texttt{TOP})$ [Def. **??**]
- Assume $(r <l> e : (\tau \ \& <l> \tau') \dashv \Theta'', \Delta'')$
  - Derive $(\Theta, \Delta, \Gamma \vdash r : \tau \dashv \Theta', \Delta')$ [Def. **??**]
  - Derive $(\Theta', \Delta', \Gamma \vdash e : \tau' \dashv \Theta'', \Delta'')$ [Def. **??**]
  - Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
  - And P0: $(\forall \delta\ .\ \delta \oplus \delta_0 \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta_0 \vDash r : \tau)$ [IH]
  - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [IH]
  - And P1: $(\forall \delta\ .\ \delta \oplus \delta_1 \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta_1 \vDash e : \tau')$ [IH]
  - Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$
  - For δ, σ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$ and $(\delta, \sigma \vDash \Gamma)$
    - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.2, Thm. 2.13]

        - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.5, Thm. 2.19]
        - Derive $(\delta \oplus \delta_0 \vDash r : \tau)$ [P0]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash r[\sigma] : \tau)$ [Thm. 3.5, Thm. 3.2, Thm. 2.15]
        - Derive $(\delta \oplus \delta_0 \vDash \Gamma)$ [Thm. 3.5, Thm. 3.2, Thm. 2.20]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e[\sigma] : \tau')$ [P1]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash r[\sigma]<l>e[\sigma] : \tau)$ [Thm. 2.1]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash r[\sigma]<l>e[\sigma] : <l>\tau')$ [Thm. 2.2]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash r[\sigma]<l>e[\sigma] : \tau \& <l>\tau')$ [Def. **??**]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash (r<l>e)[\sigma] : \tau \& <l>\tau')$ [Def. **??**]
- Assume $(\Theta, \Delta, \Gamma \vdash f : \tau \dashv \Theta, \Delta)$
    - Derive $(\exists \Xi \,.\, \Theta, \Delta, \Gamma \vdash f : \Pi \sim \Xi)$ [Def. **??**]
    - Derive $(\Theta \cup \mathbf{ftv}(\Delta) \cup \mathbf{ftv}(\Gamma) = \dot{\Theta})$ [Def. **??**]
    - Derive $(\mathbf{pack}^{+}(\dot{\Theta} \vdash \Pi) = \tau)$ [Def. **??**]
    - Derive $(\mathbf{dom}(\epsilon) \subseteq \Theta \backslash \Theta)$
    - For δ, σ, assume $(\delta \oplus \epsilon \vDash \Delta)$ and $(\delta, \sigma \vDash \Gamma)$
        - Derive $(\delta \vDash \Delta')$
        - Derive $(\delta \vDash f[\sigma] : \tau)$ [Ind. Thm. 3.10, Thm. 3.11, Thm. 3.13]
        - Derive $(\delta \oplus \epsilon \vDash f[\sigma] : \tau)$
- Assume $(\Theta, \Delta, \Gamma \vdash e\,.\,l : \alpha \dashv \Theta'', \Delta'')$
    - Derive $(\Theta, \Delta, \Gamma \vdash e : \tau \dashv \Theta', \Delta')$ [Def. **??**]
    - Derive $(\Theta', \Delta' \vdash \tau <: <l>\alpha \dashv \Theta'', \Delta'')$ [Def. **??**]
    - Let δ_0 be, such that $(\mathbf{dom}(\delta_a) \subseteq \Theta' \backslash \Theta)$ [IH]
    - And P0: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_0 \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta_0 \vDash e[\sigma] : \tau)$ [IH]
    - Let δ_1 be, such that $(\mathbf{dom}(\delta_a) \subseteq \Theta'' \backslash \Theta')$ [Ind. Thm. 3.27]
    - And P1: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_1 \vDash \Delta'' \implies \delta \oplus \delta_1 \vDash \tau <: <l>\alpha)$ [Ind. Thm. 3.27]
    - Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$ [Thm. 2.21]
    - For δ, σ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$ and $(\delta, \sigma \vDash \Gamma)$
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.13]
        - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.6, Thm. 2.19]
        - Derive $(\delta \oplus \delta_0 \vDash e[\sigma] : \tau)$ [P0]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e[\sigma] : \tau)$ [Thm. 3.6, Thm. 3.2, Thm. 2.15]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \tau <: <l>\alpha)$ [P1]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e[\sigma] : <l>\alpha)$ [Thm. 2.22]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash (e[\sigma])\,.\,l)$ [Thm. 2.3]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash (e\,.\,l)[\sigma])$ [Def. **??**]
- Assume $(\Theta, \Delta, \Gamma \vdash e_f\ (e_a)\ : \alpha \dashv \Theta''', \Delta''')$
    - Derive $(\Theta, \Delta, \Gamma \vdash e_f : \tau_f \dashv \Theta', \Delta')$ [Def. **??**]
    - Derive $(\Theta', \Delta', \Gamma \vdash e_a : \tau_a \dashv \Theta'', \Delta'')$ [Def. **??**]
    - Derive $(\Theta'', \Delta'', \tau_f <: \tau_a -> \alpha \dashv \Theta''', \Delta''')$ [Def. **??**]
    - Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
    - And P0: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_0 \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta_0 \vDash e_f[\sigma] : \tau_f)$ [IH]
    - Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [IH]
    - And P1: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_1 \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta_1 \vDash e_a[\sigma] : \tau_a)$ [IH]
    - Let δ_2 be, such that $(\mathbf{dom}(\delta_2) \subseteq \Theta''' \backslash \Theta'')$ [Ind. Thm. 3.27]
    - And P2: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_2 \vDash \Delta' \implies \delta \oplus \delta_2 \vDash \tau_f <: \tau_a -> \alpha)$ [Ind. Thm. 3.27]
    - Derive $(\mathrm{dom}(\delta_0 \oplus \delta_1 \oplus \delta_2) \subseteq \Theta''' \backslash \Theta)$ [Thm. 2.21]
    - For δ, σ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash \Delta''')$ and $(\delta, \sigma \vDash \Gamma)$
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash \Delta'')$ [Thm. 3.3, Thm. 2.13]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$ [Thm. 3.6, Thm. 2.19]
        - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.2, Thm. 2.13]
        - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.5, Thm. 2.19]
        - Derive $(\delta \oplus \delta_0 \vDash e_f[\sigma] : \tau_f)$ [P0]

- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e_f[\sigma] : \tau_f)$ [Thm. 3.5, Thm. 3.2, Thm. 2.15]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash e_f[\sigma] : \tau_f)$ [Thm. 3.6, Thm. 3.2, Thm. 2.15]
- Derive $(\delta \oplus \delta_0 \vDash \Gamma)$ [Thm. 3.5, Thm. 3.2, Thm. 2.20]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e_a[\sigma] : \tau_a)$ [P1]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash e_a[\sigma] : \tau_a)$ [Thm. 3.6, Thm. 3.2, Thm. 2.15]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash \tau_f <: \tau_a \texttt{->} \alpha)$ [P2]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash e_f[\sigma] : \tau_a \texttt{->} \alpha)$ [Thm. 2.22]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash e_f[\sigma]\ (e_a[\sigma]) : \alpha)$ [Thm 2.9]
- Derive $(\delta \oplus \delta_0 \oplus \delta_1 \oplus \delta_2 \vDash (e_f\ (e_a))[\sigma] : \alpha)$ [Def. **??**]

· Assume $((\Theta, \Delta, \Gamma \vdash \texttt{def}\ x \texttt{:} \tau_a\ \texttt{=}\ e\ \texttt{in}\ e' : \tau' \dashv \Theta'', \Delta''))$
- Derive $(\Theta, \Delta, \Gamma \vdash e : \tau \dashv \Theta', \Delta')$ [Def. **??**]
- Derive $(\mathbf{ftv}(\tau_a) \subseteq \emptyset)$ [Def. **??**]
- Derive $(\exists \Theta^\dagger, \Delta^\dagger \,.\, \Theta', \Delta' \vdash \tau <: \tau_a \dashv \Theta^\dagger, \Delta^\dagger)$ [Def. **??**]
- Derive $(\Theta', \Delta', \Gamma, x : \tau_a \vdash e' : \tau' \dashv \Theta'', \Delta'')$ [Def. **??**]
- Let δ_0 be, such that $(\mathbf{dom}(\delta_0) \subseteq \Theta' \backslash \Theta)$ [IH]
- And P0: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_0 \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta_0 \vDash e[\sigma] : \tau)$ [IH]
- Let $\Theta^\dagger, \Delta^\dagger$ be, such that $(\Theta', \Delta' \vdash \tau <: \tau_a \dashv \Theta^\dagger, \Delta^\dagger)$
- Let $\delta^\dagger$ be, such that $(\mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \backslash \Theta')$ [Ind. Thm. 3.27]
- And P†: $(\forall \delta, \sigma \,.\, \delta \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta \oplus \delta^\dagger \vDash \tau <: \tau_a)$ [Ind. Thm. 3.27]
- Let δ_1 be, such that $(\mathbf{dom}(\delta_1) \subseteq \Theta'' \backslash \Theta')$ [IH]
- And P1: $(\forall \delta, \sigma \,.\, \delta \oplus \delta_1 \vDash \Delta'' \implies \delta, \sigma \vDash \Gamma\ x : \tau_a \implies \delta \oplus \delta_1 \vDash e'[\sigma] : \tau')$ [IH]
- Derive $(\mathbf{dom}(\delta_0 \oplus \delta^\dagger) \subseteq \Theta^\dagger \backslash \Theta)$ [Thm. 2.21]
- Derive $(\mathbf{dom}(\delta_0 \oplus \delta_1) \subseteq \Theta'' \backslash \Theta)$ [Thm. 2.21]
- For δ, σ, assume $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta'')$ and $(\delta, \sigma \vDash \Gamma)$
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash \Delta')$ [Thm. 3.3, Thm. 2.13]
  - Derive $(\delta \oplus \delta_0 \vDash \Delta')$ [Thm. 3.6, Thm. 2.19]
  - Derive $(\delta \oplus \delta_0 \vDash e[\sigma] : \tau)$ [P0]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e[\sigma] : \tau)$ [Thm. 3.6, Thm. 3.2, Thm. 2.15]
  - Derive $(\delta \oplus \delta_0 \oplus \delta^\dagger \vDash \Delta^\dagger)$ [Ind. Thm. NonVac **TODO: ...**]
  - Derive $(\delta \oplus \delta_0 \oplus \delta^\dagger \vDash \tau <: \tau_a)$ [P†]
  - Derive $(\delta \oplus \delta_0 \vDash \tau <: \tau_a)$ [Because ftv is empty]
  - Derive $(\delta \oplus \delta_0 \vDash e[\sigma] : \tau_a)$ [Thm. 2.22]
  - Derive $(\delta \oplus \delta_0 \vDash \Gamma)$ [Thm. 3.5, Thm. 3.2, Thm. 2.20]
  - Derive $(\delta \oplus \delta_0\ x/e[\sigma] \vDash \Gamma\ x/\tau_a)$ [Def. **??**]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e'[\sigma\ x/[\sigma]] : \tau')$ [P2]
  - Derive $(\delta \oplus \delta_0 \oplus \delta_1 \vDash e'[\sigma][x/[\sigma]] : \tau')$ [Def. **??**]
  - Derive $(\delta \oplus \delta \vdash \texttt{def}\ x \texttt{:} \tau_a\ \texttt{=}\ e[\sigma]\ \texttt{in}\ e'[\sigma] : \tau')$ [Thm. 2.11]
  - Derive $(\delta \oplus \delta \vdash (\texttt{def}\ x \texttt{:} \tau_a\ \texttt{=}\ e\ \texttt{in}\ e')[\sigma] : \tau')$ [Def. **??**]

· Assume $(\Theta, \Delta, \Gamma \vdash \texttt{loop}(e) : \tau_l \texttt{->} \tau_r \dashv \Theta', \Delta')$
- Derive $(\Theta, \Delta, \Gamma \vdash e : \tau \dashv \Theta', \Delta')$ [Def. **??**]
- Derive $(\Theta', \Delta' \vdash \tau <: \alpha_\nu \to \Pi_\nu)$ [Def. **??**]
- Derive $(\alpha_\nu \downarrow \Pi_\nu \coloneqq \alpha_\mu \uparrow \Pi_\mu)$ [Def. **??**]
- Derive $(\mathbf{pack}^-(\mathbf{ftv}(\Delta)\ \alpha_\mu \vdash \Pi_\mu) = \tau_\mu)$ [Def. **??**]
- Derive $(\Vdash \texttt{LFP}[\alpha_\mu]\tau_\mu <: \texttt{<left>}\ \tau_l)$ [Def. **??**]
- Derive $(\Vdash \texttt{LFP}[\alpha_\mu]\tau_\mu <: \texttt{<right>}\ \tau_r)$ [Def. **??**]
- Let δ' be, such that $(\mathbf{dom}(\delta') \subseteq \Theta' \backslash \Theta)$ [IH]
- And P0: $(\forall \delta, \sigma \,.\, \delta \oplus \delta' \vDash \Delta' \implies \delta, \sigma \vDash \Gamma \implies \delta \oplus \delta' \vDash e[\sigma] : \tau)$ [P0]
- For δ, σ, assume $(\delta \oplus \delta' \vDash \Delta')$ and $(\delta, \sigma \vDash \Gamma)$
  - Derive $(\delta \oplus \delta' \vDash e[\sigma] : \tau)$ [P0]
  - Derive P1: $(\forall \delta'' \,.\, \mathbf{dom}(\delta'') \cap \mathbf{ftv}(\Delta') = \emptyset \implies \delta \oplus \delta' \oplus \delta'' \vDash \Delta')$ [Thm. 2.19]
  - Derive P2: $(\forall \delta'' \,.\, \mathbf{dom}(\delta'') \cap \mathbf{ftv}(\Delta') = \emptyset \implies \delta \oplus \delta' \oplus \delta'' \vDash e[\sigma] : \tau)$ [Thm. 3.2, Thm. 2.15]

· Derive P3: $(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \wedge (\forall \delta \,.\, \delta \oplus \delta^\dagger \vDash \Delta' \implies \delta \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta \oplus \delta^\dagger \vDash \tau <: \alpha{->}\tau^\dagger))$ [Thm. 3.8, Thm. 2.12]

· Derive $(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \wedge (\forall \delta'' \,.\, \delta \oplus \delta' \oplus \delta'' \oplus \delta^\dagger \vDash \Delta' \implies \delta \oplus \delta' \oplus \delta'' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta \oplus \delta' \oplus \delta'' \oplus \delta^\dagger \vDash \tau <: \alpha{->}\tau^\dagger))$ [P3]

· Derive $(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \wedge (\forall \delta'' \,.\, \mathbf{dom}(\delta'') \cap \mathbf{ftv}(\Delta') = \emptyset \implies \delta' \oplus \delta \oplus \delta'' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta' \oplus \delta \oplus \delta'' \oplus \delta^\dagger \vDash e[\sigma] : \alpha{->}\tau^\dagger))$ [P1, P2, Thm. 2.22]

· Derive $(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \wedge (\forall \delta'' \,.\, \mathbf{dom}(\delta'') \cap \mathbf{ftv}(\Delta') = \emptyset \implies \delta \oplus \delta' \; \alpha/\texttt{BOT} \oplus \delta'' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta \oplus \delta' \; \alpha/\texttt{BOT} \oplus \delta'' \oplus \delta^\dagger \vDash e[\sigma] : \alpha{->}\tau^\dagger))$ [Thm. ?? **TODO: weakest interpretation of alpha**]

· Derive $(\forall \Theta^\dagger, \Delta^\dagger, \tau^\dagger \,.\, (\Theta^\dagger, \Delta^\dagger, \tau^\dagger) \in \Pi_\nu \implies \exists \delta^\dagger \,.\, \mathbf{dom}(\delta^\dagger) \subseteq \Theta^\dagger \wedge (\forall \delta'' \,.\, \mathbf{dom}(\delta'') \cap \mathbf{ftv}(\Delta') = \emptyset \implies \delta \oplus \delta' \oplus \delta'' \oplus \delta^\dagger \vDash \Delta^\dagger \implies \delta \oplus \delta' \oplus \delta'' \oplus \delta^\dagger \vDash \texttt{loop}\,(e[\sigma])\,\texttt{:}\,\tau^\dagger))$ [Thm. 2.31]

· Derive $(\forall e_a, e_r \,.\, \delta \oplus \delta' \; \alpha_\mu/\texttt{TOP} \vDash (e_a\,\texttt{,}\,e_r) : \tau_\mu \implies e_r \cong \texttt{loop}\,(e[\sigma])\;(e_a))$ [Thm. 3.13]

· Derive $(\forall e_a, e_r \,.\, \delta \oplus \delta' \vDash (e_a\,\texttt{,}\,e_r) : \texttt{LFP}\,[\alpha_\mu]\,\tau_\mu \implies e_r \cong \texttt{loop}\,(e[\sigma])\;(e_a))$ [Thm. 2.27]

· Derive $(\forall e_a, e_r \,.\, \delta \oplus \delta' \vDash (e_a\,\texttt{,}\,e_r) : \texttt{LFP}\,[\alpha_\mu]\,\tau_\mu \implies \delta \oplus \delta' \vDash e_r : \tau_r \wedge e_r \cong \texttt{loop}\,(e[\sigma])\;(e_a))$ [Thm. 3.14, Thm. 2.22, Thm. 2.3, Thm. 2.30]

· Derive $(\forall e_a, e_r \,.\, \delta \oplus \delta' \vDash e_a : \tau_l \implies \delta \oplus \delta' \vDash e_r : \tau_r \wedge e_r \cong \texttt{loop}\,(e[\sigma])\;(e_a))$ [Thm. 2.29, Thm. 3.14]

· Derive $(\forall e_a \,.\, \delta \oplus \delta' \vDash e_a : \tau_l \implies \delta \oplus \delta' \vDash \texttt{loop}\,(e[\sigma])\;(e_a) : \tau_r)$ [Thm. ?? **TODO: ...**]

· Derive $(\delta \oplus \delta' \vDash \texttt{loop}\,(e[\sigma])\,\texttt{:}\,\tau_l{->}\tau_r)$ [Thm. 2.31]

· Derive $(\delta \oplus \delta' \vDash \texttt{loop}\,(e)\;[\sigma]\,\texttt{:}\,\tau_l{->}\tau_r)$ [Thm. ??]

□

TODO: update implementation to be consistent with paper.

Theorem 3.29. (Model typing record elimination)

$$\frac{\delta, \Gamma \vDash e : l{->}\tau}{\delta, \Gamma \vDash e\,.\,l : \tau}$$

Proof:

assume $\delta, \Gamma \vDash e : l{->}\tau$

. **induct on** $\delta, \Gamma \vDash e : l{->}\tau$

. **case** $e = G \quad ?l{=>}v \in G \quad \delta, \Gamma \vDash v : \tau \quad \forall v' .\; ?l{=>}v' \in G \implies v' = v$

. **wrt** $G\; v$

. . $\delta, \Gamma \vDash G : l{->}\tau$ by substitution

. . $G\,.\,l \rightsquigarrow v$ by definition

. . **def** σ **s.t.** $\delta, \Gamma \vDash \sigma$ by theorem **??**

. . $G\,.\,l[\sigma] \rightsquigarrow v$ by definition

. . $v = v[\sigma \cup \epsilon]$ by definition

. . $G\,.\,l[\sigma] \rightsquigarrow v[\sigma \cup \epsilon]$ by substitution

. . $\delta, \epsilon \vDash \epsilon$ by definition

. . $\delta, \Gamma \cup \epsilon \vDash v : \tau$ by definition

. . $\delta, \Gamma \vDash G\,.\,l : \tau$ by definition

. . $\delta, \Gamma \vDash e\,.\,l : \tau$ by substitution

. **case** $\delta, \sigma \vDash \Gamma \quad e[\sigma] \rightsquigarrow e'[\sigma \cup \sigma'] \quad \delta, \sigma' \vDash \Gamma' \quad \delta, \Gamma \cup \Gamma' \vDash e' : l{->}\tau$

. **hypo** $\delta, \Gamma \cup \Gamma' \vDash e' : l{->}\tau \implies \delta, \Gamma \cup \Gamma' \vDash e'\,.\,l : \tau$

. **wrt** $\sigma\; e'\; \sigma'\; \Gamma'$

. . $\delta, \Gamma \cup \Gamma' \vDash e'\,.\,l : \tau$ by application

. . $e[\sigma]\,.\,l \rightsquigarrow e'[\sigma \cup \sigma']\,.\,l$ by definition

. . $e[\sigma]\,.\,l = e\,.\,l[\sigma]$ by definition

. . $e'[\sigma \cup \sigma']\,.\,l = e'\,.\,l[\sigma \cup \sigma']$ by definition

. . $e\,.\,l[\sigma] \rightsquigarrow e'\,.\,l[\sigma \cup \sigma']$ by substitution
. . $\delta, \Gamma \vDash e\,.\,l : \tau$ by definition
. $\delta, \Gamma \vDash e\,.\,l : \tau$ by induction
□

Theorem 3.30. Model typing reduced implication elimination

$$\frac{\delta, \Gamma \vDash (F\,?p{=}{>}e) : \tau_l{-}{>}\tau_r \qquad \delta, \Gamma \vDash e_1 : \tau_l \qquad F = \epsilon \vee \delta, \Gamma \vDash F\,(e_1) : \tau_r}{\delta, \Gamma \vDash (F\,?p{=}{>}e)\,(e_1) : \tau_r}$$

assume $\Vdash e_1[\sigma]$
. **def** σ **s.t.** $\delta, \sigma \vDash \Gamma$ by theorem **??**
. $\Vdash e_1[\sigma]$ by theorem **??**
. **induct on** $\Vdash e_1[\sigma]$
. **case** $e_1[\sigma] = v_1$
. **wrt** v_1
. . $\delta, \Gamma \vDash (F\,?p{=}{>}e)\,(e_1) : \tau_r$ by theorem **??**
. **case** $e_1[\sigma] \rightsquigarrow e_1'$ $\Vdash e_1'$
. **hypo** $\Vdash e_1' \implies \delta, \Gamma \vDash (F\,?p{=}{>}e)\,(e_1') : \tau_r$
. **wrt** e_1'
. . $\delta, \Gamma \vDash (F\,?p{=}{>}e)\,(e_1') : \tau_r$ by application
. . $(F\,?p{=}{>}e)\,[\sigma]\,(e_1[\sigma]) \rightsquigarrow (F\,?p{=}{>}e)\,[\sigma]\,(e_1')$ by definition
. . $\forall x.\, x \notin \mathbf{FV}(e_1')$ by theorem **??**
. . $e_1' = e_1'[\sigma]$ by by theorem **??**
. . $(F\,?p{=}{>}e)\,[\sigma]\,(e_1[\sigma]) \rightsquigarrow (F\,?p{=}{>}e)\,[\sigma]\,(e_1'[\sigma])$ by substitution
. . $((F\,?p{=}{>}e)\,(e_1))[\sigma] \rightsquigarrow ((F\,?p{=}{>}e)\,(e_1'))[\sigma]$ by definition
. . $\sigma \cup \epsilon = \sigma$ by definition
. . $((F\,?p{=}{>}e)\,(e_1))[\sigma] \rightsquigarrow ((F\,?p{=}{>}e)\,(e_1'))[\sigma \cup \epsilon]$ by substitution
. . $\Gamma \cup \epsilon = \Gamma$ by definition
. . $\delta, \epsilon \vDash \epsilon$ by definition
. . $\delta, \Gamma \cup \epsilon \vDash (F\,?p{=}{>}e)\,(e_1')$ by substitution
. . $\delta, \Gamma \vDash (F\,?p{=}{>}e)\,(e_1) : \tau_r$ by definition
. $\delta, \Gamma \vDash (F\,?p{=}{>}e)\,(e_1) : \tau_r$ by induction
□ by implication