Sound and Complete Bidirectional Typechecking for Higher-Rank Polymorphism with Existentials and Indexed Types: Full definitions, lemmas and proofs

(Anonymous Authors)

March 16, 2018

The first part (Sections 1–2) of this supplementary material contains rules, figures and definitions omitted in the main paper for space reasons, and a list of judgment forms (Section 2).

The remainder (Sections A–K′) includes statements of all lemmas and theorems, along with full proofs. as well as statements of theorems and a few selected lemmas.

Contents

1 Figures

We repeat some figures from the main paper. In Figures 5a and 12a, we include rules omitted from the main paper for space reasons.

$\boxed{\Psi \vdash P\ \textit{true}}$ Under context Ψ, check P

$$\frac{}{\Psi \vdash (t = t)\ \textit{true}}\ \textsf{DeclCheckpropEq}$$

$\boxed{\begin{array}{l}\Psi \vdash e \Leftarrow A\ p \\ \Psi \vdash e \Rightarrow A\ p\end{array}}$ Under context Ψ, expression e checks against input type A
Under context Ψ, expression e synthesizes output type A

$\boxed{\begin{array}{l}\Psi \vdash s : A\ p \gg C\ q \\ \Psi \vdash s : A\ p \gg C\ \lceil q \rceil\end{array}}$ Under context Ψ,
passing spine s to a function of type A synthesizes type C;
in the $\lceil q \rceil$ form, recover principality in q if possible

$$\frac{x : A\ p \in \Psi}{\Psi \vdash x \Rightarrow A\ p}\ \textsf{DeclVar} \qquad \frac{\Psi \vdash e \Rightarrow A\ q \qquad \Psi \vdash A \leq^{\mathrm{pol}(B)} B}{\Psi \vdash e \Leftarrow B\ p}\ \textsf{DeclSub}$$

$$\frac{\Psi \vdash A\ \textit{type} \qquad \Psi \vdash e \Leftarrow A\ !}{\Psi \vdash (e : A) \Rightarrow A\ !}\ \textsf{DeclAnno} \qquad \frac{\Psi, x : A\ p \vdash v \Leftarrow A\ p}{\Psi \vdash \mathtt{rec}\ x.\, v \Leftarrow A\ p}\ \textsf{DeclRec} \qquad \frac{}{\Psi \vdash () \Leftarrow 1\ p}\ \textsf{Decl1I}$$

$$\frac{v\ \textit{chk-I} \qquad \Psi, \alpha : \kappa \vdash v \Leftarrow A\ p}{\Psi \vdash v \Leftarrow (\forall \alpha : \kappa.\, A)\ p}\ \textsf{Decl}\forall\textsf{I} \qquad \frac{\Psi \vdash \tau : \kappa \qquad \Psi \vdash e\ s : [\tau/\alpha]A\ \not p \gg C\ q}{\Psi \vdash e\ s : (\forall \alpha : \kappa.\, A)\ p \gg C\ q}\ \textsf{Decl}\forall\textsf{Spine}$$

$$\frac{\Psi \vdash P\ \textit{true} \qquad \Psi \vdash e \Leftarrow A\ p}{\Psi \vdash e \Leftarrow (A \wedge P)\ p}\ \textsf{Decl}\wedge\textsf{I}$$

$$\frac{v\ \textit{chk-I} \qquad \Psi\ /\ P \vdash v \Leftarrow A\ !}{\Psi \vdash v \Leftarrow (P \supset A)\ !}\ \textsf{Decl}\supset\textsf{I} \qquad \frac{\Psi \vdash P\ \textit{true} \qquad \Psi \vdash e\ s : A\ p \gg C\ q}{\Psi \vdash e\ s : (P \supset A)\ p \gg C\ q}\ \textsf{Decl}\supset\textsf{Spine}$$

$$\frac{\Psi, x : A\ p \vdash e \Leftarrow B\ p}{\Psi \vdash \lambda x.\, e \Leftarrow A \to B\ p}\ \textsf{Decl}{\to}\textsf{I} \qquad \frac{\Psi \vdash e \Rightarrow A\ p \qquad \Psi \vdash s : A\ p \gg C\ \lceil q \rceil}{\Psi \vdash e\ s \Rightarrow C\ q}\ \textsf{Decl}{\to}\textsf{E}$$

$$\frac{\Psi \vdash s : A\ ! \gg C\ \not p \qquad \begin{array}{l}\text{for all } C'.\\ \quad \text{if } \Psi \vdash s : A\ ! \gg C'\ \not p\\ \quad \text{then } C' = C\end{array}}{\Psi \vdash s : A\ ! \gg C\ \lceil ! \rceil}\ \textsf{DeclSpineRecover} \qquad \frac{\Psi \vdash s : A\ p \gg C\ q}{\Psi \vdash s : A\ p \gg C\ \lceil q \rceil}\ \textsf{DeclSpinePass}$$

$$\frac{}{\Psi \vdash \cdot : A\ p \gg A\ p}\ \textsf{DeclEmptySpine} \qquad \frac{\Psi \vdash e \Leftarrow A\ p \qquad \Psi \vdash s : B\ p \gg C\ q}{\Psi \vdash e\ s : A \to B\ p \gg C\ q}\ \textsf{Decl}{\to}\textsf{Spine}$$

$$\frac{\Psi \vdash e \Leftarrow A_k\ p}{\Psi \vdash \mathsf{inj}_k\ e \Leftarrow A_1 + A_2\ p}\ \textsf{Decl+I}_k \qquad \frac{\Psi \vdash e_1 \Leftarrow A_1\ p \qquad \Psi \vdash e_2 \Leftarrow A_2\ p}{\Psi \vdash \langle e_1, e_2 \rangle \Leftarrow A_1 \times A_2\ p}\ \textsf{Decl}\times\textsf{I}$$

$$\frac{\Psi \vdash t = \mathsf{zero}\ \textit{true}}{\Psi \vdash [] \Leftarrow (\mathsf{Vec}\ t\ A)\ p}\ \textsf{DeclNil} \qquad \frac{\Psi \vdash t = \mathsf{succ}(t_2)\ \textit{true} \qquad \begin{array}{l}\Psi \vdash e_1 \Leftarrow A\ p\\ \Psi \vdash e_2 \Leftarrow (\mathsf{Vec}\ t_2\ A)\ \not p\end{array}}{\Psi \vdash e_1 :: e_2 \Leftarrow (\mathsf{Vec}\ t\ A)\ p}\ \textsf{DeclCons}$$

$$\frac{\Psi \vdash e \Rightarrow A\ ! \qquad \Psi \vdash \Pi :: A \Leftarrow C\ p \qquad \Psi \vdash \Pi\ \textit{covers}\ A}{\Psi \vdash \mathsf{case}(e, \Pi) \Leftarrow C\ p}\ \textsf{DeclCase}$$

$\boxed{\Psi\ /\ P \vdash e \Leftarrow C\ p}$ Under context Ψ, incorporate proposition P and check e against C

$$\frac{\mathsf{mgu}(\sigma, \tau) = \bot}{\Psi\ /\ (\sigma = \tau) \vdash e \Leftarrow C\ p}\ \textsf{DeclCheck}\bot \qquad \frac{\begin{array}{l}\mathsf{mgu}(\sigma, \tau) = \theta\\ \theta(\Psi) \vdash \theta(e) \Leftarrow \theta(C)\ p\end{array}}{\Psi\ /\ (\sigma = \tau) \vdash e \Leftarrow C\ p}\ \textsf{DeclCheckUnify}$$

Figure 5a: Declarative typing, including rules omitted from main paper

$\Gamma \vdash e \Leftarrow A\ p \dashv \Delta$	Under input context Γ, expression e checks against input type A, with output context Δ
$\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$	Under input context Γ, expression e synthesizes output type A, with output context Δ
$\Gamma \vdash s : A\ p \gg C\ q \dashv \Delta$ $\Gamma \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta$	Under input context Γ, passing spine s to a function of type A synthesizes type C; in the $\lceil q \rceil$ form, recover principality in q if possible

$$\dfrac{(x : A\ p) \in \Gamma}{\Gamma \vdash x \Rightarrow [\Gamma]A\ p \dashv \Gamma}\ \mathsf{Var} \qquad \dfrac{\Gamma, x : A\ p \vdash v \Leftarrow A\ p \dashv \Delta, x : A\ p, \Theta}{\Gamma \vdash \mathsf{rec}\ x.\, v \Leftarrow A\ p \dashv \Delta}\ \mathsf{Rec}$$

$$\dfrac{\Gamma \vdash e \Rightarrow A\ q \dashv \Theta \qquad \Theta \vdash A <:^{\mathsf{pol}(B)} B \dashv \Delta}{\Gamma \vdash e \Leftarrow B\ p \dashv \Delta}\ \mathsf{Sub} \qquad \dfrac{\Gamma \vdash A\ !\ \mathit{type} \qquad \Gamma \vdash e \Leftarrow [\Gamma]A\ ! \dashv \Delta}{\Gamma \vdash (e : A) \Rightarrow [\Delta]A\ ! \dashv \Delta}\ \mathsf{Anno}$$

$$\dfrac{}{\Gamma \vdash () \Leftarrow 1\ p \dashv \Gamma}\ 1\mathsf{I} \qquad \dfrac{}{\Gamma[\hat\alpha : \star] \vdash () \Leftarrow \hat\alpha \ \dashv \Gamma[\hat\alpha : \star = 1]}\ 1\mathsf{I}\hat\alpha$$

$$\dfrac{v\ \mathit{chk\text{-}I} \qquad \Gamma, \alpha : \kappa \vdash v \Leftarrow A\ p \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma \vdash v \Leftarrow \forall \alpha : \kappa.\, A\ p \dashv \Delta}\ \forall\mathsf{I} \qquad \dfrac{\Gamma, \hat\alpha : \kappa \vdash e\ s : [\hat\alpha/\alpha]A \ \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s : \forall \alpha : \kappa.\, A\ p \gg C\ q \dashv \Delta}\ \forall\mathsf{Spine}$$

$$\dfrac{e \text{ not a case} \qquad \begin{array}{l}\Gamma \vdash P\ \mathit{true} \dashv \Theta \\ \Theta \vdash e \Leftarrow [\Theta]A\ p \dashv \Delta\end{array}}{\Gamma \vdash e \Leftarrow A \wedge P\ p \dashv \Delta}\ \wedge\mathsf{I} \qquad \dfrac{v\ \mathit{chk\text{-}I} \qquad \begin{array}{l}\Gamma, \blacktriangleright_P \ /\ P \dashv \Theta \\ \Theta \vdash v \Leftarrow [\Theta]A\ ! \dashv \Delta, \blacktriangleright_P, \Delta'\end{array}}{\Gamma \vdash v \Leftarrow P \supset A\ ! \dashv \Delta}\ \supset\mathsf{I}$$

$$\dfrac{v\ \mathit{chk\text{-}I} \qquad \Gamma, \blacktriangleright_P \ /\ P \dashv \bot}{\Gamma \vdash v \Leftarrow P \supset A\ ! \dashv \Gamma}\ \supset\mathsf{I}\bot \qquad \dfrac{\Gamma \vdash P\ \mathit{true} \dashv \Theta \qquad \Theta \vdash e\ s : [\Theta]A\ p \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s : P \supset A\ p \gg C\ q \dashv \Delta}\ \supset\mathsf{Spine}$$

$$\dfrac{\Gamma, x : A\ p \vdash e \Leftarrow B\ p \dashv \Delta, x : A\ p, \Theta}{\Gamma \vdash \lambda x.\, e \Leftarrow A \to B\ p \dashv \Delta}\ \to\mathsf{I} \qquad \dfrac{\begin{array}{l}\Gamma \vdash e \Rightarrow A\ p \dashv \Theta \\ \Theta \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta\end{array}}{\Gamma \vdash e\, s \Rightarrow C\ q \dashv \Delta}\ \to\mathsf{E}$$

$$\dfrac{\Gamma[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 {\to} \hat\alpha_2], x : \hat\alpha_1 \ \vdash e \Leftarrow \hat\alpha_2 \ \dashv \Delta, x : \hat\alpha_1, \Delta'}{\Gamma[\hat\alpha : \star] \vdash \lambda x.\, e \Leftarrow \hat\alpha \ \dashv \Delta}\ \to\mathsf{I}\hat\alpha$$

$$\dfrac{\begin{array}{l}\Gamma \vdash s : A\ ! \gg C\ \not{!} \dashv \Delta \\ \mathsf{FEV}(C) = \emptyset\end{array}}{\Gamma \vdash s : A\ ! \gg C\ \lceil ! \rceil \dashv \Delta}\ \mathsf{SpineRecover} \qquad \dfrac{\begin{array}{l}\Gamma \vdash s : A\ p \gg C\ q \dashv \Delta \\ \big((p = \not{!}) \text{ or } (q = !) \text{ or } (\mathsf{FEV}(C) \neq \emptyset)\big)\end{array}}{\Gamma \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta}\ \mathsf{SpinePass}$$

$$\dfrac{}{\Gamma \vdash \cdot : A\ p \gg A\ p \dashv \Gamma}\ \mathsf{EmptySpine} \qquad \dfrac{\Gamma \vdash e \Leftarrow A\ p \dashv \Theta \qquad \Theta \vdash s : [\Theta]B\ p \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s : A \to B\ p \gg C\ q \dashv \Delta}\ \to\mathsf{Spine}$$

$$\dfrac{\Gamma \vdash e \Leftarrow A_k\ p \dashv \Delta}{\Gamma \vdash \mathsf{inj}_k\, e \Leftarrow A_1 + A_2\ p \dashv \Delta}\ +\mathsf{I}_k \qquad \dfrac{\Gamma[\hat\alpha_1 : \star, \hat\alpha_2 : \star, \hat\alpha : \star = \hat\alpha_1 {+} \hat\alpha_2] \vdash e \Leftarrow \hat\alpha_k \ \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash \mathsf{inj}_k\, e \Leftarrow \hat\alpha \ \dashv \Delta}\ +\mathsf{I}\hat\alpha_k$$

$$\dfrac{\Gamma \vdash e_1 \Leftarrow A_1\ p \dashv \Theta \qquad \Theta \vdash e_2 \Leftarrow [\Theta]A_2\ p \dashv \Delta}{\Gamma \vdash \langle e_1, e_2 \rangle \Leftarrow A_1 \times A_2\ p \dashv \Delta}\ \times\mathsf{I} \qquad \dfrac{\begin{array}{l}\Gamma[\hat\alpha_2{:}\star, \hat\alpha_1{:}\star, \hat\alpha{:}\star = \hat\alpha_1 {\times} \hat\alpha_2] \vdash e_1 \Leftarrow \hat\alpha_1 \dashv \Theta \\ \Theta \vdash e_2 \Leftarrow [\Theta]\hat\alpha_2 \dashv \Delta\end{array}}{\Gamma[\hat\alpha : \star] \vdash \langle e_1, e_2 \rangle \Leftarrow \hat\alpha \ \dashv \Delta}\ \times\mathsf{I}\hat\alpha$$

$$\dfrac{\Gamma \vdash t = \mathsf{zero}\ \mathit{true} \dashv \Delta}{\Gamma \vdash [] \Leftarrow (\mathsf{Vec}\ t\ A)\ p \dashv \Delta}\ \mathsf{Nil} \qquad \dfrac{\begin{array}{l}\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} \vdash t = \mathsf{succ}(\hat\alpha)\ \mathit{true} \dashv \Gamma' \\ \Gamma' \vdash e_1 \Leftarrow [\Gamma']A\ p \dashv \Theta \\ \Theta \vdash e_2 \Leftarrow [\Theta](\mathsf{Vec}\ \hat\alpha\ A)\ \not{!} \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Delta'\end{array}}{\Gamma \vdash e_1 :: e_2 \Leftarrow (\mathsf{Vec}\ t\ A)\ p \dashv \Delta}\ \mathsf{Cons}$$

$$\dfrac{\begin{array}{l}\Gamma[\hat\alpha_2{:}\star, \hat\alpha_1{:}\star, \hat\alpha{:}\star = \hat\alpha_1 {\to} \hat\alpha_2] \vdash \\ \quad e\ s : (\hat\alpha_1 \to \hat\alpha_2) \ \gg C \ \dashv \Delta\end{array}}{\Gamma[\hat\alpha : \star] \vdash e\ s : \hat\alpha \ \gg C \ \dashv \Delta}\ \hat\alpha\mathsf{Spine} \qquad \dfrac{\Gamma \vdash e \Rightarrow A\ ! \dashv \Theta \qquad \begin{array}{l}\Theta \vdash \Pi :: [\Theta]A \Leftarrow [\Theta]C\ p \dashv \Delta \\ \Delta \vdash \Pi\ \mathit{covers}\ [\Delta]A\end{array}}{\Gamma \vdash \mathsf{case}(e, \Pi) \Leftarrow C\ p \dashv \Delta}\ \mathsf{Case}$$

Figure 12a: Algorithmic typing, including rules omitted from main paper

$\boxed{\Psi \vdash t : \kappa}$ Under context Ψ, term t has sort κ

$$\frac{(\alpha : \kappa) \in \Psi}{\Psi \vdash \alpha : \kappa}\ \mathsf{UvarSort} \qquad \frac{}{\Psi \vdash 1 : \star}\ \mathsf{UnitSort} \qquad \frac{\Psi \vdash t_1 : \star \qquad \Psi \vdash t_2 : \star}{\Psi \vdash t_1 \oplus t_2 : \star}\ \mathsf{BinSort}$$

$$\frac{}{\Psi \vdash \mathsf{zero} : \mathbb{N}}\ \mathsf{ZeroSort} \qquad \frac{\Psi \vdash t : \mathbb{N}}{\Psi \vdash \mathsf{succ}(t) : \mathbb{N}}\ \mathsf{SuccSort}$$

$\boxed{\Psi \vdash P\ \textit{prop}}$ Under context Ψ, proposition P is well-formed

$$\frac{\Psi \vdash t : \mathbb{N} \qquad \Psi \vdash t' : \mathbb{N}}{\Psi \vdash t = t'\ \textit{prop}}\ \mathsf{EqDeclProp}$$

$\boxed{\Psi \vdash A\ \textit{type}}$ Under context Ψ, type A is well-formed

$$\frac{(\alpha : \star) \in \Psi}{\Psi \vdash \alpha\ \textit{type}}\ \mathsf{DeclUvarWF} \qquad \frac{}{\Psi \vdash 1\ \textit{type}}\ \mathsf{DeclUnitWF}$$

$$\frac{\Psi \vdash A\ \textit{type} \qquad \Psi \vdash B\ \textit{type} \qquad \oplus \in \{\rightarrow, \times, +\}}{\Psi \vdash A \oplus B\ \textit{type}}\ \mathsf{DeclBinWF} \qquad \frac{\Gamma \vdash t : \mathbb{N} \qquad \Gamma \vdash A\ \textit{type}}{\Gamma \vdash \mathsf{Vec}\ t\ A\ \textit{type}}\ \mathsf{DeclVecWF}$$

$$\frac{\Psi, \alpha : \kappa \vdash A\ \textit{type}}{\Psi \vdash (\forall \alpha : \kappa.\ A)\ \textit{type}}\ \mathsf{DeclAllWF} \qquad \frac{\Psi, \alpha : \kappa \vdash A\ \textit{type}}{\Psi \vdash (\exists \alpha : \kappa.\ A)\ \textit{type}}\ \mathsf{DeclExistsWF}$$

$$\frac{\Psi \vdash P\ \textit{prop} \qquad \Psi \vdash A\ \textit{type}}{\Psi \vdash P \supset A\ \textit{type}}\ \mathsf{DeclImpliesWF} \qquad \frac{\Psi \vdash P\ \textit{prop} \qquad \Psi \vdash A\ \textit{type}}{\Psi \vdash A \wedge P\ \textit{type}}\ \mathsf{DeclWithWF}$$

$\boxed{\Psi \vdash \vec{A}\ \textit{types}}$ Under context Ψ, types in $\vec{A}$ are well-formed

$$\frac{\text{for all } A \in \vec{A}.\quad \Psi \vdash A\ \textit{type}}{\Psi \vdash \vec{A}\ \textit{types}}\ \mathsf{DeclTypevecWF}$$

$\boxed{\Psi\ \textit{ctx}}$ Declarative context Ψ is well-formed

$$\frac{}{\cdot\ \textit{ctx}}\ \mathsf{EmptyDeclCtx} \qquad \frac{\Psi\ \textit{ctx} \qquad x \notin \mathsf{dom}(\Psi) \qquad \Psi \vdash A\ \textit{type}}{\Psi, x : A\ \textit{ctx}}\ \mathsf{HypDeclCtx}$$

$$\frac{\Psi\ \textit{ctx} \qquad \alpha \notin \mathsf{dom}(\Psi)}{\Psi, \alpha : \kappa\ \textit{ctx}}\ \mathsf{VarDeclCtx}$$

Figure 14: Sorting; well-formedness of propositions, types, and contexts in the declarative system

$\boxed{\Gamma \vdash \tau : \kappa}$ Under context Γ, term τ has sort κ

$$\frac{(u : \kappa) \in \Gamma}{\Gamma \vdash u : \kappa}\ \mathsf{VarSort} \qquad \frac{(\hat{\alpha} : \kappa = \tau) \in \Gamma}{\Gamma \vdash \hat{\alpha} : \kappa}\ \mathsf{SolvedVarSort} \qquad \frac{}{\Gamma \vdash 1 : \star}\ \mathsf{UnitSort}$$

$$\frac{\Gamma \vdash \tau_1 : \star \qquad \Gamma \vdash \tau_2 : \star}{\Gamma \vdash \tau_1 \oplus \tau_2 : \star}\ \mathsf{BinSort} \qquad \frac{}{\Gamma \vdash \mathsf{zero} : \mathbb{N}}\ \mathsf{ZeroSort} \qquad \frac{\Gamma \vdash t : \mathbb{N}}{\Gamma \vdash \mathsf{succ}(t) : \mathbb{N}}\ \mathsf{SuccSort}$$

$\boxed{\Gamma \vdash P\ \mathit{prop}}$ Under context Γ, proposition P is well-formed

$$\frac{\Gamma \vdash t : \mathbb{N} \qquad \Gamma \vdash t' : \mathbb{N}}{\Gamma \vdash t = t'\ \mathit{prop}}\ \mathsf{EqProp}$$

$\boxed{\Gamma \vdash A\ \mathit{type}}$ Under context Γ, type A is well-formed

$$\frac{(u : \star) \in \Gamma}{\Gamma \vdash u\ \mathit{type}}\ \mathsf{VarWF} \qquad \frac{(\hat{\alpha} : \star = \tau) \in \Gamma}{\Gamma \vdash \hat{\alpha}\ \mathit{type}}\ \mathsf{SolvedVarWF} \qquad \frac{}{\Gamma \vdash 1\ \mathit{type}}\ \mathsf{UnitWF}$$

$$\frac{\Gamma \vdash A\ \mathit{type} \qquad \Gamma \vdash B\ \mathit{type} \qquad \oplus \in \{\rightarrow, \times, +\}}{\Gamma \vdash A \oplus B\ \mathit{type}}\ \mathsf{BinWF} \qquad \frac{\Gamma \vdash t : \mathbb{N} \qquad \Gamma \vdash A\ \mathit{type}}{\Gamma \vdash \mathsf{Vec}\ t\ A\ \mathit{type}}\ \mathsf{VecWF}$$

$$\frac{\Gamma, \alpha : \kappa \vdash A\ \mathit{type}}{\Gamma \vdash \forall \alpha : \kappa.\ A\ \mathit{type}}\ \mathsf{ForallWF} \qquad \frac{\Gamma, \alpha : \kappa \vdash A\ \mathit{type}}{\Gamma \vdash \exists \alpha : \kappa.\ A\ \mathit{type}}\ \mathsf{ExistsWF}$$

$$\frac{\Gamma \vdash P\ \mathit{prop} \qquad \Gamma \vdash A\ \mathit{type}}{\Gamma \vdash P \supset A\ \mathit{type}}\ \mathsf{ImpliesWF} \qquad \frac{\Gamma \vdash P\ \mathit{prop} \qquad \Gamma \vdash A\ \mathit{type}}{\Gamma \vdash A \wedge P\ \mathit{type}}\ \mathsf{WithWF}$$

$\boxed{\Gamma \vdash A\ p\ \mathit{type}}$ Under context Γ, type A is well-formed and respects principality p

$$\frac{\Gamma \vdash A\ \mathit{type} \qquad \mathsf{FEV}([\Gamma]A) = \emptyset}{\Gamma \vdash A\ !\ \mathit{type}}\ \mathsf{PrincipalWF} \qquad \frac{\Gamma \vdash A\ \mathit{type}}{\Gamma \vdash A\ \not{!}\ \mathit{type}}\ \mathsf{NonPrincipalWF}$$

$\boxed{\Gamma \vdash \vec{A}\ [p]\ \mathit{types}}$ Under context Γ, types in $\vec{A}$ are well-formed [with principality p]

$$\frac{\text{for all } A \in \vec{A}.\ \ \Gamma \vdash A\ \mathit{type}}{\Gamma \vdash \vec{A}\ \mathit{types}}\ \mathsf{TypevecWF} \qquad \frac{\text{for all } A \in \vec{A}.\ \ \Gamma \vdash A\ p\ \mathit{type}}{\Gamma \vdash \vec{A}\ p\ \mathit{types}}\ \mathsf{PrincipalTypevecWF}$$

$\boxed{\Gamma\ \mathit{ctx}}$ Algorithmic context Γ is well-formed

$$\frac{}{\cdot\ \mathit{ctx}}\ \mathsf{EmptyCtx} \qquad \frac{\begin{array}{c} x \notin \mathsf{dom}(\Gamma) \\ \Gamma\ \mathit{ctx} \qquad \Gamma \vdash A\ \mathit{type} \end{array}}{\Gamma, x : A\ \not{!}\ \mathit{ctx}}\ \mathsf{HypCtx} \qquad \frac{\begin{array}{c} x \notin \mathsf{dom}(\Gamma) \\ \Gamma\ \mathit{ctx} \qquad \Gamma \vdash A\ \mathit{type} \qquad \mathsf{FEV}([\Gamma]A) = \emptyset \end{array}}{\Gamma, x : A\ !\ \mathit{ctx}}\ \mathsf{Hyp!Ctx}$$

$$\frac{\Gamma\ \mathit{ctx} \qquad u \notin \mathsf{dom}(\Gamma)}{\Gamma, u : \kappa\ \mathit{ctx}}\ \mathsf{VarCtx} \qquad \frac{\Gamma\ \mathit{ctx} \qquad \hat{\alpha} \notin \mathsf{dom}(\Gamma) \qquad \Gamma \vdash t : \kappa}{\Gamma, \hat{\alpha} : \kappa = t\ \mathit{ctx}}\ \mathsf{SolvedCtx}$$

$$\frac{\Gamma\ \mathit{ctx} \qquad \alpha : \kappa \in \Gamma \qquad (\alpha = -) \notin \Gamma \qquad \Gamma \vdash \tau : \kappa}{\Gamma, \alpha = \tau\ \mathit{ctx}}\ \mathsf{EqnVarCtx} \qquad \frac{\Gamma\ \mathit{ctx} \qquad \blacktriangleright_u \notin \Gamma}{\Gamma, \blacktriangleright_u\ \mathit{ctx}}\ \mathsf{MarkerCtx}$$

Figure 15: Well-formedness of types and contexts in the algorithmic system

$\boxed{\Gamma \vdash P\ \mathit{true} \dashv \Delta}$ Under context Γ, check P, with output context Δ

$$\frac{\Gamma \vdash t_1 \mathrel{\mathring{=}} t_2 : \mathbb{N} \dashv \Delta}{\Gamma \vdash t_1 = t_2\ \mathit{true} \dashv \Delta}\ \textsf{CheckpropEq}$$

$\boxed{\Gamma \mathrel{/} P \dashv \Delta^{\perp}}$ Incorporate hypothesis P into Γ, producing Δ or inconsistency $\perp$

$$\frac{\Gamma \mathrel{/} t_1 \mathrel{\mathring{=}} t_2 : \mathbb{N} \dashv \Delta^{\perp}}{\Gamma \mathrel{/} t_1 = t_2 \dashv \Delta^{\perp}}\ \textsf{ElimpropEq}$$

Figure 16: Checking and assuming propositions

$\boxed{\Gamma \vdash t_1 \mathrel{\mathring{=}} t_2 : \kappa \dashv \Delta}$ Check that t_1 equals t_2, taking Γ to Δ

$$\frac{}{\Gamma \vdash u \mathrel{\mathring{=}} u : \kappa \dashv \Gamma}\ \textsf{CheckeqVar} \qquad \frac{}{\Gamma \vdash 1 \mathrel{\mathring{=}} 1 : \star \dashv \Gamma}\ \textsf{CheckeqUnit}$$

$$\frac{\Gamma \vdash \tau_1 \mathrel{\mathring{=}} \tau_1' : \star \dashv \Theta \qquad \Theta \vdash [\Theta]\tau_2 \mathrel{\mathring{=}} [\Theta]\tau_2' : \star \dashv \Delta}{\Gamma \vdash (\tau_1 \oplus \tau_2) \mathrel{\mathring{=}} (\tau_1' \oplus \tau_2') : \star \dashv \Delta}\ \textsf{CheckeqBin}$$

$$\frac{}{\Gamma \vdash \mathsf{zero} \mathrel{\mathring{=}} \mathsf{zero} : \mathbb{N} \dashv \Gamma}\ \textsf{CheckeqZero} \qquad \frac{\Gamma \vdash t_1 \mathrel{\mathring{=}} t_2 : \mathbb{N} \dashv \Delta}{\Gamma \vdash \mathsf{succ}(t_1) \mathrel{\mathring{=}} \mathsf{succ}(t_2) : \mathbb{N} \dashv \Delta}\ \textsf{CheckeqSucc}$$

$$\frac{\Gamma[\hat{\alpha} : \kappa] \vdash \hat{\alpha} := t : \kappa \dashv \Delta \qquad \hat{\alpha} \notin FV(t)}{\Gamma[\hat{\alpha} : \kappa] \vdash \hat{\alpha} \mathrel{\mathring{=}} t : \kappa \dashv \Delta}\ \textsf{CheckeqInstL}$$

$$\frac{\Gamma[\hat{\alpha} : \kappa] \vdash \hat{\alpha} := t : \kappa \dashv \Delta \qquad \hat{\alpha} \notin FV(t)}{\Gamma[\hat{\alpha} : \kappa] \vdash t \mathrel{\mathring{=}} \hat{\alpha} : \kappa \dashv \Delta}\ \textsf{CheckeqInstR}$$

Figure 17: Checking equations

$\boxed{t_1 \# t_2}$ t_1 and t_2 have incompatible head constructors

$$\frac{}{\mathsf{zero} \# \mathsf{succ}(t)} \qquad \frac{}{\mathsf{succ}(t) \# \mathsf{zero}} \qquad \frac{}{1 \# (\tau_1 \oplus \tau_2)} \qquad \frac{}{(\tau_1 \oplus \tau_2) \# 1} \qquad \frac{\oplus_1 \neq \oplus_2}{(\sigma_1 \oplus_1 \tau_1) \# (\sigma_2 \oplus_2 \tau_2)}$$

Figure 18: Head constructor clash

$\boxed{\Gamma \;/\; \sigma \mathrel{\mathring{=}} \tau : \kappa \dashv \Delta^{\perp}}$ Unify σ and τ, taking Γ to Δ, or to inconsistency $\perp$

$$\frac{}{\Gamma \;/\; \alpha \mathrel{\mathring{=}} \alpha : \kappa \dashv \Gamma}\ \textsf{ElimeqUvarRefl}$$

$$\frac{}{\Gamma \;/\; \mathsf{zero} \mathrel{\mathring{=}} \mathsf{zero} : \mathbb{N} \dashv \Gamma}\ \textsf{ElimeqZero} \qquad \frac{\Gamma \;/\; \sigma \mathrel{\mathring{=}} \tau : \mathbb{N} \dashv \Delta^{\perp}}{\Gamma \;/\; \mathsf{succ}(\sigma) \mathrel{\mathring{=}} \mathsf{succ}(\tau) : \mathbb{N} \dashv \Delta^{\perp}}\ \textsf{ElimeqSucc}$$

$$\frac{\alpha \notin FV(\tau) \qquad (\alpha = -) \notin \Gamma}{\Gamma \;/\; \alpha \mathrel{\mathring{=}} \tau : \kappa \dashv \Gamma, \alpha = \tau}\ \textsf{ElimeqUvarL} \qquad \frac{\alpha \notin FV(\tau) \qquad (\alpha = -) \notin \Gamma}{\Gamma \;/\; \tau \mathrel{\mathring{=}} \alpha : \kappa \dashv \Gamma, \alpha = \tau}\ \textsf{ElimeqUvarR}$$

$$\frac{t \neq \alpha \qquad \alpha \in FV(\tau)}{\Gamma \;/\; \alpha \mathrel{\mathring{=}} \tau : \kappa \dashv \perp}\ \textsf{ElimeqUvarL}\perp \qquad \frac{t \neq \alpha \qquad \alpha \in FV(\tau)}{\Gamma \;/\; \tau \mathrel{\mathring{=}} \alpha : \kappa \dashv \perp}\ \textsf{ElimeqUvarR}\perp$$

$$\frac{}{\Gamma \;/\; 1 \mathrel{\mathring{=}} 1 : \star \dashv \Gamma}\ \textsf{ElimeqUnit} \qquad \frac{\Gamma \;/\; \tau_1 \mathrel{\mathring{=}} \tau_1' : \star \dashv \Theta \qquad \Theta \;/\; [\Theta]\tau_2 \mathrel{\mathring{=}} [\Theta]\tau_2' : \star \dashv \Delta^{\perp}}{\Gamma \;/\; (\tau_1 \oplus \tau_2) \mathrel{\mathring{=}} (\tau_1' \oplus \tau_2') : \star \dashv \Delta^{\perp}}\ \textsf{ElimeqBin}$$

$$\frac{\Gamma \;/\; \tau_1 \mathrel{\mathring{=}} \tau_1' : \star \dashv \perp}{\Gamma \;/\; (\tau_1 \oplus \tau_2) \mathrel{\mathring{=}} (\tau_1' \oplus \tau_2') : \star \dashv \perp}\ \textsf{ElimeqBinBot}$$

$$\frac{\sigma \mathrel{\#} \tau}{\Gamma \;/\; \sigma \mathrel{\mathring{=}} \tau : \kappa \dashv \perp}\ \textsf{ElimeqClash}$$

Figure 19: Eliminating equations

$\boxed{\Gamma \vdash A <:^{\pm} B \dashv \Delta}$ Under input context Γ, type A is a subtype of B, with output context Δ

$$\frac{\begin{array}{l}A \text{ not headed by } \forall/\exists \\ B \text{ not headed by } \forall/\exists \qquad \Gamma \vdash A \equiv B \dashv \Delta\end{array}}{\Gamma \vdash A <:^{\pm} B \dashv \Delta}\;{<:}\mathsf{Equiv}$$

$$\frac{\begin{array}{l}B \text{ not headed by } \forall \\ \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa \vdash [\hat{\alpha}/\alpha]A <:^{-} B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta\end{array}}{\Gamma \vdash \forall \alpha : \kappa.\, A <:^{-} B \dashv \Delta}\;{<:}\forall\mathsf{L} \qquad \frac{\Gamma, \beta : \kappa \vdash A <:^{-} B \dashv \Delta, \beta : \kappa, \Theta}{\Gamma \vdash A <:^{-} \forall \beta : \kappa.\, B \dashv \Delta}\;{<:}\forall\mathsf{R}$$

$$\frac{\Gamma, \alpha : \kappa \vdash A <:^{+} B \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma \vdash \exists \alpha : \kappa.\, A <:^{+} B \dashv \Delta}\;{<:}\exists\mathsf{L} \qquad \frac{\begin{array}{l}A \text{ not headed by } \exists \\ \Gamma, \blacktriangleright_{\hat{\beta}}, \hat{\beta} : \kappa \vdash A <:^{+} [\hat{\beta}/\beta]B \dashv \Delta, \blacktriangleright_{\hat{\beta}}, \Theta\end{array}}{\Gamma \vdash A <:^{+} \exists \beta : \kappa.\, B \dashv \Delta}\;{<:}\exists\mathsf{R}$$

$$\frac{\Gamma \vdash A <:^{-} B \dashv \Delta \qquad \begin{array}{l}\mathit{neg}(A) \\ \mathit{nonpos}(B)\end{array}}{\Gamma \vdash A <:^{+} B \dashv \Delta}\;{<:}^{-}_{+}\mathsf{L} \qquad \frac{\Gamma \vdash A <:^{-} B \dashv \Delta \qquad \begin{array}{l}\mathit{nonpos}(A) \\ \mathit{neg}(B)\end{array}}{\Gamma \vdash A <:^{+} B \dashv \Delta}\;{<:}^{-}_{+}\mathsf{R}$$

$$\frac{\Gamma \vdash A <:^{+} B \dashv \Delta \qquad \begin{array}{l}\mathit{pos}(A) \\ \mathit{nonneg}(B)\end{array}}{\Gamma \vdash A <:^{-} B \dashv \Delta}\;{<:}^{+}_{-}\mathsf{L} \qquad \frac{\Gamma \vdash A <:^{+} B \dashv \Delta \qquad \begin{array}{l}\mathit{nonneg}(A) \\ \mathit{pos}(B)\end{array}}{\Gamma \vdash A <:^{-} B \dashv \Delta}\;{<:}^{+}_{-}\mathsf{R}$$

$\boxed{\Gamma \vdash P \equiv Q \dashv \Delta}$ Under input context Γ, check that P is equivalent to Q with output context Δ

$$\frac{\Gamma \vdash t_1 \mathrel{\mathring{=}} t_2 : \mathbb{N} \dashv \Theta \qquad \Theta \vdash [\Theta]t_1' \mathrel{\mathring{=}} [\Theta]t_2' : \mathbb{N} \dashv \Delta}{\Gamma \vdash (t_1 = t_1') \equiv (t_2 = t_2') \dashv \Delta}\;{\equiv}\mathsf{PropEq}$$

$\boxed{\Gamma \vdash A \equiv B \dashv \Delta}$ Under input context Γ, check that A is equivalent to B with output context Δ

$$\frac{}{\Gamma \vdash \alpha \equiv \alpha \dashv \Gamma}\;{\equiv}\mathsf{Var} \qquad \frac{}{\Gamma \vdash \hat{\alpha} \equiv \hat{\alpha} \dashv \Gamma}\;{\equiv}\mathsf{Exvar} \qquad \frac{}{\Gamma \vdash 1 \equiv 1 \dashv \Gamma}\;{\equiv}\mathsf{Unit}$$

$$\frac{\Gamma \vdash A_1 \equiv B_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta}{\Gamma \vdash (A_1 \oplus A_2) \equiv (B_1 \oplus B_2) \dashv \Delta}\;{\equiv}{\oplus} \qquad \frac{\Gamma \vdash t_1 \equiv t_2 \dashv \Theta \qquad \Theta \vdash [\Theta]A_1 \equiv [\Theta]A_2 \dashv \Delta}{\Gamma \vdash (\mathsf{Vec}\ t_1\ A_1) \equiv (\mathsf{Vec}\ t_2\ A_2) \dashv \Delta}\;{\equiv}\mathsf{Vec}$$

$$\frac{\Gamma, \alpha : \kappa \vdash A \equiv B \dashv \Delta, \alpha : \kappa, \Delta'}{\Gamma \vdash (\forall \alpha : \kappa.\, A) \equiv (\forall \alpha : \kappa.\, B) \dashv \Delta}\;{\equiv}\forall \qquad \frac{\Gamma, \alpha : \kappa \vdash A \equiv B \dashv \Delta, \alpha : \kappa, \Delta'}{\Gamma \vdash (\exists \alpha : \kappa.\, A) \equiv (\exists \alpha : \kappa.\, B) \dashv \Delta}\;{\equiv}\exists$$

$$\frac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A \equiv [\Theta]B \dashv \Delta}{\Gamma \vdash (P \supset A) \equiv (Q \supset B) \dashv \Delta}\;{\equiv}{\supset} \qquad \frac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A \equiv [\Theta]B \dashv \Delta}{\Gamma \vdash (A \wedge P) \equiv (B \wedge Q) \dashv \Delta}\;{\equiv}{\wedge}$$

$$\frac{\hat{\alpha} \notin \mathit{FV}(\tau) \qquad \Gamma[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \equiv \tau \dashv \Delta}\;{\equiv}\mathsf{InstantiateL} \qquad \frac{\hat{\alpha} \notin \mathit{FV}(\tau) \qquad \Gamma[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta}{\Gamma[\hat{\alpha}] \vdash \tau \equiv \hat{\alpha} \dashv \Delta}\;{\equiv}\mathsf{InstantiateR}$$

Figure 20: Algorithmic subtyping and equivalence

$\boxed{\Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta}$ Under input context Γ, instantiate $\hat{\alpha}$ such that $\hat{\alpha} = t$ with output context Δ

$$\frac{\Gamma_0 \vdash \tau : \kappa}{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \vdash \hat{\alpha} := \tau : \kappa \dashv \Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1}\ \textsf{InstSolve}$$

$$\frac{\hat{\beta} \in \mathsf{unsolved}(\Gamma[\hat{\alpha} : \kappa][\hat{\beta} : \kappa])}{\Gamma[\hat{\alpha} : \kappa][\hat{\beta} : \kappa] \vdash \hat{\alpha} := \hat{\beta} : \kappa \dashv \Gamma[\hat{\alpha} : \kappa][\hat{\beta} : \kappa = \hat{\alpha}]}\ \textsf{InstReach}$$

$$\frac{\Gamma[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2] \vdash \hat{\alpha}_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat{\alpha}_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma[\hat{\alpha} : \star] \vdash \hat{\alpha} := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \textsf{InstBin}$$

$$\frac{}{\Gamma[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{zero} : \mathbb{N} \dashv \Gamma[\hat{\alpha} : \mathbb{N} = \mathsf{zero}]}\ \textsf{InstZero} \qquad \frac{\Gamma[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)] \vdash \hat{\alpha}_1 := t_1 : \mathbb{N} \dashv \Delta}{\Gamma[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \textsf{InstSucc}$$

Figure 21: Instantiation

$$\boxed{\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta}$$

Under context Γ, check branches Π with patterns of type $\vec{A}$ and bodies of type C

$$\frac{}{\Gamma \vdash \cdot :: \vec{A} \Leftarrow C\ p \dashv \Gamma}\ \textsf{MatchEmpty} \qquad \frac{\Gamma \vdash \pi :: \vec{A} \Leftarrow C\ p \dashv \Theta \qquad \Theta \vdash \Pi' :: \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash \pi \mid \Pi' :: \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchSeq}$$

$$\frac{\Gamma \vdash e \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash (\cdot \Rightarrow e) :: \cdot \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchBase} \qquad \frac{\Gamma \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash (), \vec{\rho} \Rightarrow e :: 1, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchUnit}$$

$$\frac{\Gamma, \alpha : \kappa \vdash \vec{\rho} \Rightarrow e :: A, \vec{A} \Leftarrow C\ p \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma \vdash \vec{\rho} \Rightarrow e :: (\exists \alpha : \kappa.\ A), \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{Match}\exists \qquad \frac{\Gamma \,/\, P \vdash \vec{\rho} \Rightarrow e :: A, \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash \vec{\rho} \Rightarrow e :: A \wedge P, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{Match}\wedge$$

$$\frac{\Gamma \vdash \rho_1, \rho_2, \vec{\rho} \Rightarrow e :: A_1, A_2, \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash \langle \rho_1, \rho_2 \rangle, \vec{\rho} \Rightarrow e :: A_1 \times A_2, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{Match}\times$$

$$\frac{\Gamma \vdash \rho, \vec{\rho} \Rightarrow e :: A_k, \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash (\mathsf{inj}_k\ \rho), \vec{\rho} \Rightarrow e :: A_1 + A_2, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{Match}+_k$$

$$\frac{A \text{ not headed by } \wedge \text{ or } \exists \qquad \Gamma, z : A\,! \vdash \vec{\rho} \Rightarrow e' :: \vec{A} \Leftarrow C\ p \dashv \Delta, z : A\,!, \Delta'}{\Gamma \vdash z, \vec{\rho} \Rightarrow e :: A, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchNeg}$$

$$\frac{A \text{ not headed by } \wedge \text{ or } \exists \qquad \Gamma \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash \_, \vec{\rho} \Rightarrow e :: A, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchWild}$$

$$\frac{\Gamma \,/\, (t = \mathsf{zero}) \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash [\,], \vec{\rho} \Rightarrow e :: (\mathsf{Vec}\ t\ A), \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchNil}$$

$$\frac{\Gamma, \alpha : \mathbb{N} \,/\, (t = \mathsf{succ}(\alpha)) \vdash \rho_1, \rho_2, \vec{\rho} \Rightarrow e :: A, (\mathsf{Vec}\ \alpha\ A), \vec{A} \Leftarrow C\ p \dashv \Delta, \alpha : \mathbb{N}, \Theta}{\Gamma \vdash (\rho_1 :: \rho_2), \vec{\rho} \Rightarrow e :: (\mathsf{Vec}\ t\ A), \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchCons}$$

$$\boxed{\Gamma \,/\, P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta}$$

Under context Γ, incorporate proposition P while checking branches Π with patterns of type $\vec{A}$ and bodies of type C

$$\frac{\Gamma \,/\, \sigma \mathrel{\dot{=}} \tau : \kappa \dashv \bot}{\Gamma \,/\, \sigma = \tau \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Gamma}\ \textsf{Match}\bot$$

$$\frac{\Gamma, \blacktriangleright_P \,/\, \sigma \mathrel{\dot{=}} \tau : \kappa \dashv \Theta \qquad \Theta \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta, \blacktriangleright_P, \Delta'}{\Gamma \,/\, \sigma = \tau \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchUnify}$$

Figure 22: Algorithmic pattern matching

$$\boxed{\begin{array}{l}\Gamma \vdash \Pi \ \textit{covers}\ \vec{A} \\ \Gamma \ / \ P \vdash \Pi \ \textit{covers}\ \vec{A}\end{array}}\quad \begin{array}{l}\text{Under context } \Gamma\text{, patterns } \Pi \text{ cover the types } \vec{A} \\ \text{Under context } \Gamma\text{, patterns } \Pi \text{ cover the types } \vec{A} \text{ assuming } P\end{array}$$

$$\frac{}{\Gamma \vdash (\cdot \Rightarrow e_1) \mid \Pi \ \textit{covers}\ \cdot}\ \textsf{CoversEmpty} \qquad \frac{\Pi \overset{\text{var}}{\rightsquigarrow} \Pi' \qquad \Gamma \vdash \Pi' \ \textit{covers}\ \vec{A}}{\Gamma \vdash \Pi \ \textit{covers}\ A, \vec{A}}\ \textsf{CoversVar}$$

$$\frac{\Pi \overset{1}{\rightsquigarrow} \Pi' \qquad \Gamma \vdash \Pi' \ \textit{covers}\ \vec{A}}{\Gamma \vdash \Pi \ \textit{covers}\ 1, \vec{A}}\ \textsf{Covers1} \qquad \frac{\Pi \overset{\times}{\rightsquigarrow} \Pi' \qquad \Gamma \vdash \Pi' \ \textit{covers}\ A_1, A_2, \vec{A}}{\Gamma \vdash \Pi \ \textit{covers}\ (A_1 \times A_2), \vec{A}}\ \textsf{Covers}\times$$

$$\frac{\Pi \overset{+}{\rightsquigarrow} \Pi_L \parallel \Pi_R \qquad \Gamma \vdash \Pi_L \ \textit{covers}\ A_1, \vec{A} \qquad \Gamma \vdash \Pi_R \ \textit{covers}\ A_2, \vec{A}}{\Gamma \vdash \Pi \ \textit{covers}\ (A_1 + A_2), \vec{A}}\ \textsf{Covers}+$$

$$\frac{\Gamma, \alpha : \kappa \vdash \Pi \ \textit{covers}\ \vec{A}}{\Gamma \vdash \Pi \ \textit{covers}\ (\exists \alpha : \kappa.\ A), \vec{A}}\ \textsf{Covers}\exists \qquad \frac{\Gamma \ / \ t_1 = t_2 \vdash \Pi \ \textit{covers}\ A_0, \vec{A}}{\Gamma \vdash \Pi \ \textit{covers}\ \big(A_0 \wedge (t_1 = t_2)\big), \vec{A}}\ \textsf{Covers}\wedge$$

$$\frac{\Pi \overset{\textsf{Vec}}{\rightsquigarrow} \Pi_{[]} \parallel \Pi_{::} \qquad \begin{array}{l}\Gamma \ / \ t = \textsf{zero} \vdash \Pi_{[]} \ \textit{covers}\ \vec{A} \\ \Gamma, n : \mathbb{N} \ / \ t = \textsf{succ}(n) \vdash \Pi_{::} \ \textit{covers}\ (A, \textsf{Vec}\ n\ A, \vec{A})\end{array}}{\Gamma \vdash \Pi \ \textit{covers}\ \textsf{Vec}\ t\ A, \vec{A}}\ \textsf{CoversVec}$$

$$\frac{\Gamma \ / \ [\Gamma]t_1 \mathrel{\mathring{=}} [\Gamma]t_2 : \kappa \dashv \Delta \qquad \Delta \vdash [\Delta]\Pi \ \textit{covers}\ [\Delta]\vec{A}}{\Gamma \ / \ t_1 = t_2 \vdash \Pi \ \textit{covers}\ \vec{A}}\ \textsf{CoversEq} \qquad \frac{\Gamma \ / \ [\Gamma]t_1 \mathrel{\mathring{=}} [\Gamma]t_2 : \kappa \dashv \bot}{\Gamma \ / \ t_1 = t_2 \vdash \Pi \ \textit{covers}\ \vec{A}}\ \textsf{CoversEqBot}$$

Figure 23: Algorithmic match coverage

2 List of Judgments

For convenience, we list all the judgment forms:

Judgment	*Description*	*Location*
$\Psi \vdash t : \kappa$	Index term/monotype is well-formed	Figure 14
$\Psi \vdash P\ \mathit{prop}$	Proposition is well-formed	Figure 14
$\Psi \vdash A\ \mathit{type}$	Type is well-formed	Figure 14
$\Psi \vdash \vec{A}\ \mathit{types}$	Type vector is well-formed	Figure 14
$\Psi\ \mathit{ctx}$	Declarative context is well-formed	Figure 14
$\Psi \vdash A \leq^{\pm} B$	Declarative subtyping	Figure 3
$\Psi \vdash P\ \mathit{true}$	Declarative truth	Figure 5
$\Psi \vdash e \Leftarrow A\ p$	Declarative checking	Figure 5
$\Psi \vdash e \Rightarrow A\ p$	Declarative synthesis	Figure 5
$\Psi \vdash s : A\ p \gg C\ q$	Declarative spine typing	Figure 5
$\Psi \vdash s : A\ p \gg C\ \lceil q \rceil$	Declarative spine typing, recovering principality	Figure 5
$\Psi \vdash \Pi :: \vec{A} \Leftarrow C\ p$	Declarative pattern matching	Figure 6
$\Psi\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p$	Declarative proposition assumption	Figure 6
$\Psi \vdash \Pi\ \mathit{covers}\ \vec{A}$	Declarative match coverage	Figure 7
$\Gamma \vdash \tau : \kappa$	Index term/monotype is well-formed	Figure 15
$\Gamma \vdash P\ \mathit{prop}$	Proposition is well-formed	Figure 15
$\Gamma \vdash A\ \mathit{type}$	Polytype is well-formed	Figure 15
$\Gamma\ \mathit{ctx}$	Algorithmic context is well-formed	Figure 15
$[\Gamma]A$	Applying a context, as a substitution, to a type	Figure 10
$\Gamma \vdash P\ \mathit{true} \dashv \Delta$	Check proposition	Figure 16
$\Gamma\ /\ P \dashv \Delta^{\perp}$	Assume proposition	Figure 16
$\Gamma \vdash s \stackrel{\circ}{=} t : \kappa \dashv \Delta$	Check equation	Figure 17
$s \# t$	Head constructors clash	Figure 18
$\Gamma\ /\ s \stackrel{\circ}{=} t : \kappa \dashv \Delta^{\perp}$	Assume/eliminate equation	Figure 19
$\Gamma \vdash A <:^{\pm} B \dashv \Delta$	Algorithmic subtyping	Figure 20
$\Gamma\ /\ P \vdash A <: B \dashv \Delta$	Assume/eliminate proposition	Figure 20
$\Gamma \vdash P \equiv Q \dashv \Delta$	Equivalence of propositions	Figure 20
$\Gamma \vdash A \equiv B \dashv \Delta$	Equivalence of types	Figure 20
$\Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta$	Instantiate	Figure 21
$e\ \mathit{chk\text{-}I}$	Checking intro form	Figure 4
$\Gamma \vdash e \Leftarrow A\ p \dashv \Delta$	Algorithmic checking	Figure 12
$\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$	Algorithmic synthesis	Figure 12
$\Gamma \vdash s : A\ p \gg C\ q \dashv \Delta$	Algorithmic spine typing	Figure 12
$\Gamma \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta$	Algorithmic spine typing, recovering principality	Figure 12
$\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$	Algorithmic pattern matching	Figure 22
$\Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$	Algorithmic pattern matching (assumption)	Figure 22
$\Gamma \vdash \Pi\ \mathit{covers}\ \vec{A}$	Algorithmic match coverage	Figure 23
$\Gamma \longrightarrow \Delta$	Context extension	Figure 13
$[\Omega]\Gamma$	Apply complete context	Figure 11

A Properties of the Declarative System

Lemma 1 (Declarative Well-foundedness). *Go to proof*
The inductive definition of the following judgments is well-founded:

(i) *synthesis* $\Psi \vdash e \Rightarrow B\ p$

(ii) *checking* $\Psi \vdash e \Leftarrow A\ p$

(iii) *checking, equality elimination* $\Psi \ /\ P \vdash e \Leftarrow C\ p$

(iv) *ordinary spine* $\Psi \vdash s : A\ p \gg B\ q$

(v) *recovery spine* $\Psi \vdash s : A\ p \gg B\ \lceil q \rceil$

(vi) *pattern matching* $\Psi \vdash \Pi :: \vec{A} \Leftarrow C\ p$

(vii) *pattern matching, equality elimination* $\Psi \ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p$

Lemma 2 (Declarative Weakening). *Go to proof*

(i) *If* $\Psi_0, \Psi_1 \vdash t : \kappa$ *then* $\Psi_0, \Psi, \Psi_1 \vdash t : \kappa$.

(ii) *If* $\Psi_0, \Psi_1 \vdash P$ *prop* *then* $\Psi_0, \Psi, \Psi_1 \vdash P$ *prop*.

(iii) *If* $\Psi_0, \Psi_1 \vdash P$ *true* *then* $\Psi_0, \Psi, \Psi_1 \vdash P$ *true*.

(iv) *If* $\Psi_0, \Psi_1 \vdash A$ *type* *then* $\Psi_0, \Psi, \Psi_1 \vdash A$ *type*.

Lemma 3 (Declarative Term Substitution). *Go to proof*
Suppose $\Psi \vdash t : \kappa$. *Then:*

1. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash t' : \kappa$ *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]t' : \kappa$.
2. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash P$ *prop* *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]P$ *prop*.
3. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash A$ *type* *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]A$ *type*.
4. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash A \leq^{\pm} B$ *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]A \leq^{\pm} [t/\alpha]B$.
5. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash P$ *true* *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]P$ *true*.

Lemma 4 (Reflexivity of Declarative Subtyping). *Go to proof*
Given $\Psi \vdash A$ *type, we have that* $\Psi \vdash A \leq^{\pm} A$.

Lemma 5 (Subtyping Inversion). *Go to proof*

- *If* $\Psi \vdash \exists \alpha : \kappa.\ A \leq^{+} B$ *then* $\Psi, \alpha : \kappa \vdash A \leq^{+} B$.
- *If* $\Psi \vdash A \leq^{-} \forall \beta : \kappa.\ B$ *then* $\Psi, \beta : \kappa \vdash A \leq^{-} B$.

Lemma 6 (Subtyping Polarity Flip). *Go to proof*

- *If* *nonpos*(A) *and* *nonpos*(B) *and* $\Psi \vdash A \leq^{+} B$
  then $\Psi \vdash A \leq^{-} B$ *by a derivation of the same or smaller size.*
- *If* *nonneg*(A) *and* *nonneg*(B) *and* $\Psi \vdash A \leq^{-} B$
  then $\Psi \vdash A \leq^{+} B$ *by a derivation of the same or smaller size.*
- *If* *nonpos*(A) *and* *nonneg*(A) *and* *nonpos*(B) *and* *nonneg*(B) *and* $\Psi \vdash A \leq^{\pm} B$
  then $A = B$.

Lemma 7 (Transitivity of Declarative Subtyping). *Go to proof*
Given $\Psi \vdash A$ *type and* $\Psi \vdash B$ *type and* $\Psi \vdash C$ *type:*

(i) If $\mathcal{D}_1 :: \Psi \vdash A \leq^{\pm} B$ *and* $\mathcal{D}_2 :: \Psi \vdash B \leq^{\pm} C$
then $\Psi \vdash A \leq^{\pm} C$.

Property 1. *We assume that all types mentioned in annotations in expressions have no free existential variables. By the grammar, it follows that all expressions have no free existential variables, that is,* $\mathsf{FEV}(e) = \emptyset$.

B Substitution and Well-formedness Properties

Definition 1 (Softness). *A context* Θ *is* soft *iff it consists only of* $\hat{\alpha} : \kappa$ *and* $\hat{\alpha} : \kappa = \tau$ *declarations.*

Lemma 8 (Substitution—Well-formedness). *Go to proof*

(i) If $\Gamma \vdash A\ p$ *type and* $\Gamma \vdash \tau\ p$ *type then* $\Gamma \vdash [\tau/\alpha]A\ p$ *type.*

(ii) If $\Gamma \vdash P$ *prop and* $\Gamma \vdash \tau\ p$ *type then* $\Gamma \vdash [\tau/\alpha]P$ *prop.*
Moreover, if $p = \mathbf{!}$ *and* $\mathsf{FEV}([\Gamma]P) = \emptyset$ *then* $\mathsf{FEV}([\Gamma][\tau/\alpha]P) = \emptyset$.

Lemma 9 (Uvar Preservation). *Go to proof*
If $\Delta \longrightarrow \Omega$ *then:*

(i) If $(\alpha : \kappa) \in \Omega$ *then* $(\alpha : \kappa) \in [\Omega]\Delta$.

(ii) If $(x : A\ p) \in \Omega$ *then* $(x : [\Omega]A\ p) \in [\Omega]\Delta$.

Lemma 10 (Sorting Implies Typing). *Go to proof* *If* $\Gamma \vdash t : \star$ *then* $\Gamma \vdash t$ *type.*

Lemma 11 (Right-Hand Substitution for Sorting). *Go to proof* *If* $\Gamma \vdash t : \kappa$ *then* $\Gamma \vdash [\Gamma]t : \kappa$.

Lemma 12 (Right-Hand Substitution for Propositions). *Go to proof* *If* $\Gamma \vdash P$ *prop then* $\Gamma \vdash [\Gamma]P$ *prop.*

Lemma 13 (Right-Hand Substitution for Typing). *Go to proof* *If* $\Gamma \vdash A$ *type then* $\Gamma \vdash [\Gamma]A$ *type.*

Lemma 14 (Substitution for Sorting). *Go to proof* *If* $\Omega \vdash t : \kappa$ *then* $[\Omega]\Omega \vdash [\Omega]t : \kappa$.

Lemma 15 (Substitution for Prop Well-Formedness). *Go to proof*
If $\Omega \vdash P$ *prop then* $[\Omega]\Omega \vdash [\Omega]P$ *prop.*

Lemma 16 (Substitution for Type Well-Formedness). *Go to proof* *If* $\Omega \vdash A$ *type then* $[\Omega]\Omega \vdash [\Omega]A$ *type.*

Lemma 17 (Substitution Stability). *Go to proof*
If (Ω, Ω_Z) *is well-formed and* Ω_Z *is soft and* $\Omega \vdash A$ *type then* $[\Omega]A = [\Omega, \Omega_Z]A$.

Lemma 18 (Equal Domains). *Go to proof*
If $\Omega_1 \vdash A$ *type and* $\mathsf{dom}(\Omega_1) = \mathsf{dom}(\Omega_2)$ *then* $\Omega_2 \vdash A$ *type.*

C Properties of Extension

Lemma 19 (Declaration Preservation). *Go to proof* *If* $\Gamma \longrightarrow \Delta$ *and* u *is declared in* Γ*, then* u *is declared in* Δ.

Lemma 20 (Declaration Order Preservation). *Go to proof* *If* $\Gamma \longrightarrow \Delta$ *and* u *is declared to the left of* v *in* Γ*, then* u *is declared to the left of* v *in* Δ.

Lemma 21 (Reverse Declaration Order Preservation). *Go to proof* *If* $\Gamma \longrightarrow \Delta$ *and* u *and* v *are both declared in* Γ *and* u *is declared to the left of* v *in* Δ*, then* u *is declared to the left of* v *in* Γ.

An older paper had a lemma

"Substitution Extension Invariance"
If $\Theta \vdash A$ *type* and $\Theta \longrightarrow \Gamma$ then $[\Gamma]A = [\Gamma]([\Theta]A)$ and $[\Gamma]A = [\Theta]([\Gamma]A)$.

For the second part, $[\Gamma]A = [\Theta]([\Gamma]A)$, use Lemma 29 (Substitution Monotonicity) (i) or (iii) instead. The first part $[\Gamma]A = [\Gamma][\Theta]A$ hasn't been proved in this system.

Lemma 22 (Extension Inversion). *Go to proof*

(i) *If* $\mathcal{D} :: \Gamma_0, \alpha : \kappa, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0 *and* Δ_1
such that $\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$.

Moreover, if Γ_1 *is soft, then* Δ_1 *is soft.*

(ii) *If* $\mathcal{D} :: \Gamma_0, \blacktriangleright_u, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0 *and* Δ_1
such that $\Delta = (\Delta_0, \blacktriangleright_u, \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$.

Moreover, if Γ_1 *is soft, then* Δ_1 *is soft.*

Moreover, if $\mathsf{dom}(\Gamma_0, \blacktriangleright_u, \Gamma_1) = \mathsf{dom}(\Delta)$ *then* $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Delta_0)$.

(iii) *If* $\mathcal{D} :: \Gamma_0, \alpha = \tau, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0, τ', *and* Δ_1
such that $\Delta = (\Delta_0, \alpha = \tau', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *and* $[\Delta_0]\tau = [\Delta_0]\tau'$ *where* $\mathcal{D}' < \mathcal{D}$.

(iv) *If* $\mathcal{D} :: \Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0, τ', *and* Δ_1
such that $\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *and* $[\Delta_0]\tau = [\Delta_0]\tau'$ *where* $\mathcal{D}' < \mathcal{D}$.

(v) *If* $\mathcal{D} :: \Gamma_0, x : A, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0, A', *and* Δ_1
such that $\Delta = (\Delta_0, x : A', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *and* $[\Delta_0]A = [\Delta_0]A'$ *where* $\mathcal{D}' < \mathcal{D}$.

Moreover, if Γ_1 *is soft, then* Δ_1 *is soft.*

Moreover, if $\mathsf{dom}(\Gamma_0, x : A, \Gamma_1) = \mathsf{dom}(\Delta)$ *then* $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Delta_0)$.

(vi) *If* $\mathcal{D} :: \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \longrightarrow \Delta$ *then either*

- *there exist unique* Δ_0, τ', *and* Δ_1
  such that $\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$,
  or
- *there exist unique* Δ_0 *and* Δ_1
  such that $\Delta = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$.

Lemma 23 (Deep Evar Introduction). *Go to proof*

(i) *If* Γ_0, Γ_1 *is well-formed and* $\hat{\alpha}$ *is not declared in* Γ_0, Γ_1 *then* $\Gamma_0, \Gamma_1 \longrightarrow \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1$.

(ii) *If* $\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1$ *is well-formed and* $\Gamma \vdash t : \kappa$ *then* $\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1$.

(iii) *If* Γ_0, Γ_1 *is well-formed and* $\Gamma \vdash t : \kappa$ *then* $\Gamma_0, \Gamma_1 \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1$.

Lemma 24 (Soft Extension). *Go to proof*
If $\Gamma \longrightarrow \Delta$ *and* Γ, Θ *ctx and* Θ *is soft, then there exists* Ω *such that* $\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega)$ *and* $\Gamma, \Theta \longrightarrow \Delta, \Omega$.

Definition 2 (Filling). *The filling of a context* $|\Gamma|$ *solves all unsolved variables:*

$$
\begin{array}{lcl}
|\cdot| & = & \cdot \\
|\Gamma, x : A| & = & |\Gamma|, x : A \\
|\Gamma, \alpha : \kappa| & = & |\Gamma|, \alpha : \kappa \\
|\Gamma, \alpha = t| & = & |\Gamma|, \alpha = t \\
|\Gamma, \hat{\alpha} : \kappa = t| & = & |\Gamma|, \hat{\alpha} : \kappa = t \\
|\Gamma, \blacktriangleright_{\hat{\alpha}}| & = & |\Gamma|, \blacktriangleright_{\hat{\alpha}} \\
|\Gamma, \hat{\alpha} : \star| & = & |\Gamma|, \hat{\alpha} : \star = 1 \\
|\Gamma, \hat{\alpha} : \mathbb{N}| & = & |\Gamma|, \hat{\alpha} : \mathbb{N} = \mathsf{zero}
\end{array}
$$

Lemma 25 (Filling Completes). *If $\Gamma \longrightarrow \Omega$ and (Γ, Θ) is well-formed, then $\Gamma, \Theta \longrightarrow \Omega, |\Theta|$.*

Proof. By induction on Θ, following the definition of $|{-}|$ and applying the rules for $\longrightarrow$. □

Lemma 26 (Parallel Admissibility). Go to proof
If $\Gamma_L \longrightarrow \Delta_L$ and $\Gamma_L, \Gamma_R \longrightarrow \Delta_L, \Delta_R$ then:

(i) $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa, \Delta_R$

(ii) If $\Delta_L \vdash \tau' : \kappa$ then $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$.

(iii) If $\Gamma_L \vdash \tau : \kappa$ and $\Delta_L \vdash \tau'$ type and $[\Delta_L]\tau = [\Delta_L]\tau'$, then $\Gamma_L, \hat{\alpha} : \kappa = \tau, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$.

Lemma 27 (Parallel Extension Solution). Go to proof
If $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$ and $\Gamma_L \vdash \tau : \kappa$ and $[\Delta_L]\tau = [\Delta_L]\tau'$
then $\Gamma_L, \hat{\alpha} : \kappa = \tau, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$.

Lemma 28 (Parallel Variable Update). Go to proof
If $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau_0, \Delta_R$ and $\Gamma_L \vdash \tau_1 : \kappa$ and $\Delta_L \vdash \tau_2 : \kappa$ and $[\Delta_L]\tau_0 = [\Delta_L]\tau_1 = [\Delta_L]\tau_2$
then $\Gamma_L, \hat{\alpha} : \kappa = \tau_1, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau_2, \Delta_R$.

Lemma 29 (Substitution Monotonicity). Go to proof

(i) If $\Gamma \longrightarrow \Delta$ and $\Gamma \vdash t : \kappa$ then $[\Delta][\Gamma]t = [\Delta]t$.

(ii) If $\Gamma \longrightarrow \Delta$ and $\Gamma \vdash P$ prop then $[\Delta][\Gamma]P = [\Delta]P$.

(iii) If $\Gamma \longrightarrow \Delta$ and $\Gamma \vdash A$ type then $[\Delta][\Gamma]A = [\Delta]A$.

Lemma 30 (Substitution Invariance). Go to proof

(i) If $\Gamma \longrightarrow \Delta$ and $\Gamma \vdash t : \kappa$ and $\mathsf{FEV}([\Gamma]t) = \emptyset$ then $[\Delta][\Gamma]t = [\Gamma]t$.

(ii) If $\Gamma \longrightarrow \Delta$ and $\Gamma \vdash P$ prop and $\mathsf{FEV}([\Gamma]P) = \emptyset$ then $[\Delta][\Gamma]P = [\Gamma]P$.

(iii) If $\Gamma \longrightarrow \Delta$ and $\Gamma \vdash A$ type and $\mathsf{FEV}([\Gamma]A) = \emptyset$ then $[\Delta][\Gamma]A = [\Gamma]A$.

Definition 3 (Canonical Contexts). *A (complete) context Ω is* canonical *iff, for all $(\hat{\alpha} : \kappa = t)$ and $(\alpha = t) \in \Omega$, the solution t is ground ($\mathsf{FEV}(t) = \emptyset$).*

Lemma 31 (Split Extension). Go to proof
If $\Delta \longrightarrow \Omega$
and $\hat{\alpha} \in \mathsf{unsolved}(\Delta)$
and $\Omega = \Omega_1[\hat{\alpha} : \kappa = t_1]$
and Ω is canonical (Definition 3)
and $\Omega \vdash t_2 : \kappa$
then $\Delta \longrightarrow \Omega_1[\hat{\alpha} : \kappa = t_2]$.

C.1 Reflexivity and Transitivity

Lemma 32 (Extension Reflexivity). *Go to proof*
If Γ *ctx then* $\Gamma \longrightarrow \Gamma$.

Lemma 33 (Extension Transitivity). *Go to proof*
If $\mathcal{D} :: \Gamma \longrightarrow \Theta$ *and* $\mathcal{D}' :: \Theta \longrightarrow \Delta$ *then* $\Gamma \longrightarrow \Delta$.

C.2 Weakening

The "suffix weakening" lemmas take a judgment under Γ and produce a judgment under (Γ, Θ). They do *not* require $\Gamma \longrightarrow \Gamma, \Theta$.

Lemma 34 (Suffix Weakening). *Go to proof* *If* $\Gamma \vdash t : \kappa$ *then* $\Gamma, \Theta \vdash t : \kappa$.

Lemma 35 (Suffix Weakening). *Go to proof* *If* $\Gamma \vdash A$ *type then* $\Gamma, \Theta \vdash A$ *type*.

The following proposed lemma is false.

> "Extension Weakening (Truth)"
> If $\Gamma \vdash P$ *true* $\dashv \Delta$ and $\Gamma \longrightarrow \Gamma'$ then there exists Δ' such that $\Delta \longrightarrow \Delta'$ and $\Gamma' \vdash P$ *true* $\dashv \Delta'$.

Counterexample: Suppose $\hat{\alpha} \vdash \hat{\alpha} = 1$ *true* $\dashv \hat{\alpha} = 1$ and $\hat{\alpha} \longrightarrow (\hat{\alpha} = (1 \to 1))$. Then there does *not* exist such a Δ'.

Lemma 36 (Extension Weakening (Sorts)). *Go to proof* *If* $\Gamma \vdash t : \kappa$ *and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash t : \kappa$.

Lemma 37 (Extension Weakening (Props)). *Go to proof* *If* $\Gamma \vdash P$ *prop and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash P$ *prop*.

Lemma 38 (Extension Weakening (Types)). *Go to proof* *If* $\Gamma \vdash A$ *type and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash A$ *type*.

C.3 Principal Typing Properties

Lemma 39 (Principal Agreement). *Go to proof*

(i) *If* $\Gamma \vdash A$ **!** *type and* $\Gamma \longrightarrow \Delta$ *then* $[\Delta]A = [\Gamma]A$.

(ii) *If* $\Gamma \vdash P$ *prop and* $\mathsf{FEV}(P) = \emptyset$ *and* $\Gamma \longrightarrow \Delta$ *then* $[\Delta]P = [\Gamma]P$.

Lemma 40 (Right-Hand Subst. for Principal Typing). *Go to proof* *If* $\Gamma \vdash A\ p$ *type then* $\Gamma \vdash [\Gamma]A\ p$ *type*.

Lemma 41 (Extension Weakening for Principal Typing). *Go to proof* *If* $\Gamma \vdash A\ p$ *type and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash A\ p$ *type*.

Lemma 42 (Inversion of Principal Typing). *Go to proof*

(1) *If* $\Gamma \vdash (A \to B)\ p$ *type then* $\Gamma \vdash A\ p$ *type and* $\Gamma \vdash B\ p$ *type*.

(2) *If* $\Gamma \vdash (P \supset A)\ p$ *type then* $\Gamma \vdash P$ *prop and* $\Gamma \vdash A\ p$ *type*.

(3) *If* $\Gamma \vdash (A \wedge P)\ p$ *type then* $\Gamma \vdash P$ *prop and* $\Gamma \vdash A\ p$ *type*.

C.4 Instantiation Extends

Lemma 43 (Instantiation Extension). *Go to proof*
If $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

C.5 Equivalence Extends

Lemma 44 (Elimeq Extension). *Go to proof*
If $\Gamma \ / \ s \mathrel{\mathring{=}} t : \kappa \dashv \Delta$ *then there exists* Θ *such that* $\Gamma, \Theta \longrightarrow \Delta$.

Lemma 45 (Elimprop Extension). *Go to proof*
If $\Gamma \ / \ P \dashv \Delta$ *then there exists* Θ *such that* $\Gamma, \Theta \longrightarrow \Delta$.

Lemma 46 (Checkeq Extension). *Go to proof*
If $\Gamma \vdash A \equiv B \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Lemma 47 (Checkprop Extension). *Go to proof*
If $\Gamma \vdash P$ *true* $\dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Lemma 48 (Prop Equivalence Extension). *Go to proof*
If $\Gamma \vdash P \equiv Q \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Lemma 49 (Equivalence Extension). *Go to proof*
If $\Gamma \vdash A \equiv B \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

C.6 Subtyping Extends

Lemma 50 (Subtyping Extension). *Go to proof* *If* $\Gamma \vdash A <:^{\mp} B \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

C.7 Typing Extends

Lemma 51 (Typing Extension). *Go to proof*
If $\Gamma \vdash e \Leftarrow A \ p \dashv \Delta$
or $\Gamma \vdash e \Rightarrow A \ p \dashv \Delta$
or $\Gamma \vdash s : A \ p \gg B \ q \dashv \Delta$
or $\Gamma \vdash \Pi :: \vec{A} \Leftarrow C \ p \dashv \Delta$
or $\Gamma \ / \ P \vdash \Pi :: \vec{A} \Leftarrow C \ p \dashv \Delta$
then $\Gamma \longrightarrow \Delta$.

C.8 Unfiled

Lemma 52 (Context Partitioning). *Go to proof*
If $\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta \longrightarrow \Omega, \blacktriangleright_{\hat{\alpha}}, \Omega_Z$ *then there is a* Ψ *such that* $[\Omega, \blacktriangleright_{\hat{\alpha}}, \Omega_Z](\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta) = [\Omega]\Delta, \Psi$.

Lemma 53 (Softness Goes Away).
If $\Delta, \Theta \longrightarrow \Omega, \Omega_Z$ *where* $\Delta \longrightarrow \Omega$ *and* Θ *is soft, then* $[\Omega, \Omega_Z](\Delta, \Theta) = [\Omega]\Delta$.

Proof. By induction on Θ, following the definition of $[\Omega]\Gamma$. □

Lemma 54 (Completing Stability). *Go to proof*
If $\Gamma \longrightarrow \Omega$ *then* $[\Omega]\Gamma = [\Omega]\Omega$.

Lemma 55 (Completing Completeness). *Go to proof*

(i) If $\Omega \longrightarrow \Omega'$ *and* $\Omega \vdash t : \kappa$ *then* $[\Omega]t = [\Omega']t$.

(ii) If $\Omega \longrightarrow \Omega'$ *and* $\Omega \vdash A$ *type then* $[\Omega]A = [\Omega']A$.

(iii) If $\Omega \longrightarrow \Omega'$ *then* $[\Omega]\Omega = [\Omega']\Omega'$.

Lemma 56 (Confluence of Completeness). *Go to proof*
If $\Delta_1 \longrightarrow \Omega$ *and* $\Delta_2 \longrightarrow \Omega$ *then* $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

Lemma 57 (Multiple Confluence). *Go to proof*
If $\Delta \longrightarrow \Omega$ *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta' \longrightarrow \Omega'$ *then* $[\Omega]\Delta = [\Omega']\Delta'$.

Lemma 58 (Bundled Substitution for Sorting). *If* $\Gamma \vdash t : \kappa$ *and* $\Gamma \longrightarrow \Omega$ *then* $[\Omega]\Gamma \vdash [\Omega]t : \kappa$.

Proof.

	$\Gamma \vdash t : \kappa$	Given
	$\Omega \vdash t : \kappa$	By Lemma 36 (Extension Weakening (Sorts))
	$[\Omega]\Omega \vdash [\Omega]t : \kappa$	By Lemma 14 (Substitution for Sorting)
	$\Omega \longrightarrow \Omega$	By Lemma 32 (Extension Reflexivity)
	$[\Omega]\Omega = [\Omega]\Gamma$	By Lemma 56 (Confluence of Completeness)
☞	$[\Omega]\Gamma \vdash [\Omega]t : \kappa$	By above equality

□

Lemma 59 (Canonical Completion). *Go to proof*
If $\Gamma \longrightarrow \Omega$
then there exists Ω_{canon} *such that* $\Gamma \longrightarrow \Omega_{canon}$ *and* $\Omega_{canon} \longrightarrow \Omega$ *and* $\mathsf{dom}(\Omega_{canon}) = \mathsf{dom}(\Gamma)$ *and, for all* $\hat{\alpha} : \kappa = \tau$ *and* $\alpha = \tau$ *in* Ω_{canon}*, we have* $\mathsf{FEV}(\tau) = \emptyset$.

The completion Ω_{canon} is "canonical" because (1) its domain exactly matches Γ and (2) its solutions τ have no evars. Note that it follows from Lemma 57 (Multiple Confluence) that $[\Omega_{canon}]\Gamma = [\Omega]\Gamma$.

Lemma 60 (Split Solutions). *Go to proof*
If $\Delta \longrightarrow \Omega$ *and* $\hat{\alpha} \in \mathsf{unsolved}(\Delta)$
then there exists $\Omega_1 = \Omega_1'[\hat{\alpha} : \kappa = t_1]$ *such that* $\Omega_1 \longrightarrow \Omega$ *and* $\Omega_2 = \Omega_1'[\hat{\alpha} : \kappa = t_2]$ *where* $\Delta \longrightarrow \Omega_2$ *and* $t_2 \neq t_1$ *and* Ω_2 *is canonical.*

D Internal Properties of the Declarative System

Lemma 61 (Interpolating With and Exists). *Go to proof*

(1) *If* $\mathcal{D} :: \Psi \vdash \Pi :: \vec{A} \Leftarrow C\ p$ *and* $\Psi \vdash P_0$ *true*
then $\mathcal{D}' :: \Psi \vdash \Pi :: \vec{A} \Leftarrow C \wedge P_0\ p$.

(2) *If* $\mathcal{D} :: \Psi \vdash \Pi :: \vec{A} \Leftarrow [\tau/\alpha]C_0\ p$ *and* $\Psi \vdash \tau : \kappa$
then $\mathcal{D}' :: \Psi \vdash \Pi :: \vec{A} \Leftarrow (\exists \alpha : \kappa.\, C_0)\ p$.

In both cases, the height of $\mathcal{D}'$ *is one greater than the height of* $\mathcal{D}$.
Moreover, similar properties hold for the eliminating judgment $\Psi \,/\, P \vdash \Pi :: \vec{A} \Leftarrow C\ p$.

Lemma 62 (Case Invertibility). *Go to proof*
If $\Psi \vdash \mathsf{case}(e_0, \Pi) \Leftarrow C\ p$
then $\Psi \vdash e_0 \Rightarrow A\ !$ *and* $\Psi \vdash \Pi :: A \Leftarrow C\ p$ *and* $\Psi \vdash \Pi$ *covers* A
where the height of each resulting derivation is strictly less than the height of the given derivation.

E Miscellaneous Properties of the Algorithmic System

Lemma 63 (Well-Formed Outputs of Typing). *Go to proof*

(Spines) *If* $\Gamma \vdash s : A\ q \gg C\ p \dashv \Delta$ *or* $\Gamma \vdash s : A\ q \gg C\ \lceil p \rceil \dashv \Delta$
and $\Gamma \vdash A\ q$ *type*
then $\Delta \vdash C\ p$ *type.*

(Synthesis) *If* $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$
then $A \vdash p$ *type.*

F Decidability of Instantiation

Lemma 64 (Left Unsolvedness Preservation). *Go to proof*
If $\underbrace{\Gamma_0, \hat{\alpha}, \Gamma_1}_{\Gamma} \vdash \hat{\alpha} := A : \kappa \dashv \Delta$ *and* $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0)$ *then* $\hat{\beta} \in \mathsf{unsolved}(\Delta)$.

Lemma 65 (Left Free Variable Preservation). *Go to proof* *If* $\overbrace{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1}^{\Gamma} \vdash \hat{\alpha} := t : \kappa \dashv \Delta$ *and* $\Gamma \vdash s : \kappa'$ *and* $\hat{\alpha} \notin FV([\Gamma]s)$ *and* $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0)$ *and* $\hat{\beta} \notin FV([\Gamma]s)$, *then* $\hat{\beta} \notin FV([\Delta]s)$.

Lemma 66 (Instantiation Size Preservation). *Go to proof* *If* $\overbrace{\Gamma_0, \hat{\alpha}, \Gamma_1}^{\Gamma} \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *and* $\Gamma \vdash s : \kappa'$ *and* $\hat{\alpha} \notin FV([\Gamma]s)$, *then* $|[\Gamma]s| = |[\Delta]s|$, *where* $|C|$ *is the plain size of the term* C.

Lemma 67 (Decidability of Instantiation). *Go to proof* *If* $\Gamma = \Gamma_0[\hat{\alpha} : \kappa']$ *and* $\Gamma \vdash t : \kappa$ *such that* $[\Gamma]t = t$ *and* $\hat{\alpha} \notin FV(t)$, *then:*

(1) Either there exists Δ *such that* $\Gamma_0[\hat{\alpha} : \kappa'] \vdash \hat{\alpha} := t : \kappa \dashv \Delta$, *or not.*

G Separation

Definition 4 (Separation).
An algorithmic context Γ *is* separable *and written* $\Gamma_L * \Gamma_R$ *if (1)* $\Gamma = (\Gamma_L, \Gamma_R)$ *and (2) for all* $(\hat{\alpha} : \kappa = \tau) \in \Gamma_R$ *it is the case that* $\mathsf{FEV}(\tau) \subseteq \mathsf{dom}(\Gamma_R)$.

Any context Γ is separable into, at least, $\cdot * \Gamma$ and $\Gamma * \cdot$.

Definition 5 (Separation-Preserving Extension).
The separated context $\Gamma_L * \Gamma_R$ *extends to* $\Delta_L * \Gamma_R$, *written*

$$(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$$

if $(\Gamma_L, \Gamma_R) \longrightarrow (\Delta_L, \Delta_R)$ *and* $\mathsf{dom}(\Gamma_L) \subseteq \mathsf{dom}(\Delta_L)$ *and* $\mathsf{dom}(\Gamma_R) \subseteq \mathsf{dom}(\Delta_R)$.

Separation-preserving extension says that variables from one half don't "cross" into the other half. Thus, Δ_L may add existential variables to Γ_L, and Δ_R may add existential variables to Γ_R, but no variable from Γ_L ends up in Δ_R and no variable from Γ_R ends up in Δ_L.

It is necessary to write $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$ rather than $(\Gamma_L * \Gamma_R) \longrightarrow (\Delta_L * \Delta_R)$, because only $\underset{*}{\longrightarrow}$ includes the domain conditions. For example, $(\hat{\alpha} * \hat{\beta}) \longrightarrow (\hat{\alpha}, \hat{\beta} = \hat{\alpha}) * \cdot$, but the variable $\hat{\beta}$ has "crossed over" to the left of $*$ in the context $(\hat{\alpha}, \hat{\beta} = \hat{\alpha}) * \cdot$.

Lemma 68 (Transitivity of Separation). *Go to proof*
If $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Theta_L * \Theta_R)$ *and* $(\Theta_L * \Theta_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$
then $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

Lemma 69 (Separation Truncation). *Go to proof*
If H *has the form* $\alpha : \kappa$ *or* $\blacktriangleright_{\hat{\alpha}}$ *or* $\blacktriangleright_P$ *or* $x : A\, p$
and $(\Gamma_L * (\Gamma_R, H)) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$
then $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_0)$ *where* $\Delta_R = (\Delta_0, H, \Theta)$.

Lemma 70 (Separation for Auxiliary Judgments). *Go to proof*

(i) If $\Gamma_L * \Gamma_R \vdash \sigma \mathrel{\mathring{=}} \tau : \kappa \dashv \Delta$
and $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(\tau) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(ii) If $\Gamma_L * \Gamma_R \vdash P$ *true* $\dashv \Delta$
and $\mathsf{FEV}(P) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(iii) *If* $\Gamma_L * \Gamma_R \;/\; \sigma \mathrel{\mathring{=}} \tau : \kappa \dashv \Delta$
and $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(\tau) = \emptyset$
then $\Delta = (\Delta_L * (\Delta_R, \Theta))$ *and* $(\Gamma_L * (\Gamma_R, \Theta)) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(iv) *If* $\Gamma_L * \Gamma_R \;/\; P \dashv \Delta$
and $\mathsf{FEV}(P) = \emptyset$
then $\Delta = (\Delta_L * (\Delta_R, \Theta))$ *and* $(\Gamma_L * (\Gamma_R, \Theta)) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(v) *If* $\Gamma_L * \Gamma_R \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$
and $(\mathsf{FEV}(\tau) \cup \{\hat{\alpha}\}) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(vi) *If* $\Gamma_L * \Gamma_R \vdash P \equiv Q \dashv \Delta$
and $\mathsf{FEV}(P) \cup \mathsf{FEV}(Q) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(vii) *If* $\Gamma_L * \Gamma_R \vdash A \equiv B \dashv \Delta$
and $\mathsf{FEV}(A) \cup \mathsf{FEV}(B) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

Lemma 71 (Separation for Subtyping). *Go to proof*
If $\Gamma_L * \Gamma_R \vdash A <:^{\pm} B \dashv \Delta$
and $\mathsf{FEV}(A) \subseteq \mathsf{dom}(\Gamma_R)$
and $\mathsf{FEV}(B) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

Lemma 72 (Separation—Main). *Go to proof*

(Spines) *If* $\Gamma_L * \Gamma_R \vdash s : A\ p \gg C\ q \dashv \Delta$
or $\Gamma_L * \Gamma_R \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta$
and $\Gamma_L * \Gamma_R \vdash A\ p$ *type*
and $\mathsf{FEV}(A) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$ *and* $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R)$.

(Checking) *If* $\Gamma_L * \Gamma_R \vdash e \Leftarrow C\ p \dashv \Delta$
and $\Gamma_L * \Gamma_R \vdash C\ p$ *type*
and $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(Synthesis) *If* $\Gamma_L * \Gamma_R \vdash e \Rightarrow A\ p \dashv \Delta$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(Match) *If* $\Gamma_L * \Gamma_R \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$
and $\mathsf{FEV}(\vec{A}) = \emptyset$
and $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

(Match Elim.) *If* $\Gamma_L * \Gamma_R \;/\; P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$
and $\mathsf{FEV}(P) = \emptyset$
and $\mathsf{FEV}(\vec{A}) = \emptyset$
and $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \underset{*}{\longrightarrow} (\Delta_L * \Delta_R)$.

H Decidability of Algorithmic Subtyping

Definition 6. *The following connectives are* large*:*

$$\forall \qquad \supset \qquad \wedge$$

A type is large *iff its head connective is large. (Note that a non-large type may contain large connectives, provided they are not in head position.)*

The number of these connectives in a type A *is denoted by* $\#\mathsf{large}(A)$.

H.1 Lemmas for Decidability of Subtyping

Lemma 73 (Substitution Isn't Large). Go to proof
For all contexts Θ, *we have* $\#\mathsf{large}([\Theta]A) = \#\mathsf{large}(A)$.

Lemma 74 (Instantiation Solves). Go to proof
If $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *and* $[\Gamma]\tau = \tau$ *and* $\hat{\alpha} \notin FV([\Gamma]\tau)$ *then* $|\mathsf{unsolved}(\Gamma)| = |\mathsf{unsolved}(\Delta)| + 1$.

Lemma 75 (Checkeq Solving). Go to proof *If* $\Gamma \vdash s \mathrel{\mathring{=}} t : \kappa \dashv \Delta$ *then either* $\Delta = \Gamma$ *or* $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

Lemma 76 (Prop Equiv Solving). Go to proof
If $\Gamma \vdash P \equiv Q \dashv \Delta$ *then either* $\Delta = \Gamma$ *or* $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

Lemma 77 (Equiv Solving). Go to proof
If $\Gamma \vdash A \equiv B \dashv \Delta$ *then either* $\Delta = \Gamma$ *or* $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

Lemma 78 (Decidability of Propositional Judgments). Go to proof
The following judgments are decidable, with Δ *as output in (1)–(3), and* $\Delta^{\perp}$ *as output in (4) and (5).*

We assume $\sigma = [\Gamma]\sigma$ *and* $t = [\Gamma]t$ *in (1) and (4). Similarly, in the other parts we assume* $P = [\Gamma]P$ *and (in part (3))* $Q = [\Gamma]Q$.

(1) $\Gamma \vdash \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta$

(2) $\Gamma \vdash P \textit{ true} \dashv \Delta$

(3) $\Gamma \vdash P \equiv Q \dashv \Delta$

(4) $\Gamma \,/\, \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta^{\perp}$

(5) $\Gamma \,/\, P \dashv \Delta^{\perp}$

Lemma 79 (Decidability of Equivalence). Go to proof
Given a context Γ *and types* A, B *such that* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type and* $[\Gamma]A = A$ *and* $[\Gamma]B = B$, *it is decidable whether there exists* Δ *such that* $\Gamma \vdash A \equiv B \dashv \Delta$.

H.2 Decidability of Subtyping

Theorem 1 (Decidability of Subtyping). Go to proof
Given a context Γ *and types* A, B *such that* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type and* $[\Gamma]A = A$ *and* $[\Gamma]B = B$, *it is decidable whether there exists* Δ *such that* $\Gamma \vdash A <:^{\pm} B \dashv \Delta$.

H.3 Decidability of Matching and Coverage

Lemma 80 (Decidability of Expansion Judgments). Go to proof
Given branches Π, *it is decidable whether:*

(1) there exists Π' *such that* $\Pi \overset{\times}{\rightsquigarrow} \Pi'$;

(2) there exist Π_L *and* Π_R *such that* $\Pi \overset{+}{\rightsquigarrow} \Pi_L \parallel \Pi_R$;

(3) there exists Π' *such that* $\Pi \overset{\mathrm{var}}{\rightsquigarrow} \Pi'$;

(4) there exists Π' *such that* $\Pi \overset{1}{\rightsquigarrow} \Pi'$.

Theorem 2 (Decidability of Coverage). Go to proof
Given a context Γ, *branches* Π *and types* $\vec{A}$, *it is decidable whether* $\Gamma \vdash \Pi$ *covers* $\vec{A}$ *is derivable.*

H.4 Decidability of Typing

Theorem 3 (Decidability of Typing). *Go to proof*

(i) Synthesis: *Given a context* Γ*, a principality* p*, and a term* e*, it is decidable whether there exist a type* A *and a context* Δ *such that* $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$.

(ii) Spines: *Given a context* Γ*, a spine* s*, a principality* p*, and a type* A *such that* $\Gamma \vdash A$ *type, it is decidable whether there exist a type* B*, a principality* q *and a context* Δ *such that* $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$.

(iii) Checking: *Given a context* Γ*, a principality* p*, a term* e*, and a type* B *such that* $\Gamma \vdash B$ *type, it is decidable whether there is a context* Δ *such that* $\Gamma \vdash e \Leftarrow B\ p \dashv \Delta$.

(iv) Matching: *Given a context* Γ*, branches* Π*, a list of types* $\vec{A}$*, a type* C*, and a principality* p*, it is decidable whether there exists* Δ *such that* $\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$.

Also, if given a proposition P *as well, it is decidable whether there exists* Δ *such that* $\Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$.

I Determinacy

Lemma 81 (Determinacy of Auxiliary Judgments). *Go to proof*

(1) Elimeq: *Given* $\Gamma, \sigma, t, \kappa$ *such that* $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(t) = \emptyset$ *and* $\mathcal{D}_1 :: \Gamma\ /\ \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta_1^{\perp}$ *and* $\mathcal{D}_2 :: \Gamma\ /\ \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta_2^{\perp}$*, it is the case that* $\Delta_1^{\perp} = \Delta_2^{\perp}$.

(2) Instantiation: *Given* $\Gamma, \hat{\alpha}, t, \kappa$ *such that* $\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$ *and* $\Gamma \vdash t : \kappa$ *and* $\hat{\alpha} \notin FV(t)$ *and* $\mathcal{D}_1 :: \Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta_2$ *it is the case that* $\Delta_1 = \Delta_2$.

(3) Symmetric instantiation:

Given $\Gamma, \hat{\alpha}, \hat{\beta}, \kappa$ *such that* $\hat{\alpha}, \hat{\beta} \in \mathsf{unsolved}(\Gamma)$ *and* $\hat{\alpha} \neq \hat{\beta}$ *and* $\mathcal{D}_1 :: \Gamma \vdash \hat{\alpha} := \hat{\beta} : \kappa \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \hat{\beta} := \hat{\alpha} : \kappa \dashv \Delta_2$ *it is the case that* $\Delta_1 = \Delta_2$.

(4) Checkeq: *Given* $\Gamma, \sigma, t, \kappa$ *such that* $\mathcal{D}_1 :: \Gamma \vdash \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta_2$ *it is the case that* $\Delta_1 = \Delta_2$.

(5) Elimprop: *Given* Γ, P *such that* $\mathcal{D}_1 :: \Gamma\ /\ P \dashv \Delta_1^{\perp}$ *and* $\mathcal{D}_2 :: \Gamma\ /\ P \dashv \Delta_2^{\perp}$ *it is the case that* $\Delta_1 = \Delta_2$.

(6) Checkprop: *Given* Γ, P *such that* $\mathcal{D}_1 :: \Gamma \vdash P$ *true* $\dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash P$ *true* $\dashv \Delta_2$*, it is the case that* $\Delta_1 = \Delta_2$.

Lemma 82 (Determinacy of Equivalence). *Go to proof*

(1) Propositional equivalence: *Given* Γ, P, Q *such that* $\mathcal{D}_1 :: \Gamma \vdash P \equiv Q \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash P \equiv Q \dashv \Delta_2$*, it is the case that* $\Delta_1 = \Delta_2$.

(2) Type equivalence: *Given* Γ, A, B *such that* $\mathcal{D}_1 :: \Gamma \vdash A \equiv B \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash A \equiv B \dashv \Delta_2$*, it is the case that* $\Delta_1 = \Delta_2$.

Theorem 4 (Determinacy of Subtyping). *Go to proof*

(1) Subtyping: *Given* Γ, e, A, B *such that* $\mathcal{D}_1 :: \Gamma \vdash A <:^{\pm} B \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash A <:^{\pm} B \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

Theorem 5 (Determinacy of Typing). *Go to proof*

(1) Checking: *Given* Γ, e, A, p *such that* $\mathcal{D}_1 :: \Gamma \vdash e \Leftarrow A\ p \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash e \Leftarrow A\ p \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

(2) Synthesis: *Given* Γ, e *such that* $\mathcal{D}_1 :: \Gamma \vdash e \Rightarrow B_1\ p_1 \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash e \Rightarrow B_2\ p_2 \dashv \Delta_2$, *it is the case that* $B_1 = B_2$ *and* $p_1 = p_2$ *and* $\Delta_1 = \Delta_2$.

(3) Spine judgments:

Given Γ, e, A, p *such that* $\mathcal{D}_1 :: \Gamma \vdash e : A\ p \gg C_1\ q_1 \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash e : A\ p \gg C_2\ q_2 \dashv \Delta_2$, *it is the case that* $C_1 = C_2$ *and* $q_1 = q_2$ *and* $\Delta_1 = \Delta_2$.

The same applies for derivations of the principality-recovering judgments $\Gamma \vdash e : A\ p \gg C_k\ \lceil q_k \rceil \dashv \Delta_k$.

(4) Match judgments:

Given Γ, Π, $\vec{A}$, p, C *such that* $\mathcal{D}_1 :: \Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

Given Γ, P, Π, $\vec{A}$, p, C
such that $\mathcal{D}_1 :: \Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

J Soundness

J.1 Soundness of Instantiation

Lemma 83 (Soundness of Instantiation). *Go to proof*
If $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *and* $\hat{\alpha} \notin FV([\Gamma]\tau)$ *and* $[\Gamma]\tau = \tau$ *and* $\Delta \longrightarrow \Omega$ *then* $[\Omega]\hat{\alpha} = [\Omega]\tau$.

J.2 Soundness of Checkeq

Lemma 84 (Soundness of Checkeq). *Go to proof*
If $\Gamma \vdash \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta$ *where* $\Delta \longrightarrow \Omega$ *then* $[\Omega]\sigma = [\Omega]t$.

J.3 Soundness of Equivalence (Propositions and Types)

Lemma 85 (Soundness of Propositional Equivalence). *Go to proof*
If $\Gamma \vdash P \equiv Q \dashv \Delta$ *where* $\Delta \longrightarrow \Omega$ *then* $[\Omega]P = [\Omega]Q$.

Lemma 86 (Soundness of Algorithmic Equivalence). *Go to proof*
If $\Gamma \vdash A \equiv B \dashv \Delta$ *where* $\Delta \longrightarrow \Omega$ *then* $[\Omega]A = [\Omega]B$.

J.4 Soundness of Checkprop

Lemma 87 (Soundness of Checkprop). *Go to proof*
If $\Gamma \vdash P$ *true* $\dashv \Delta$ *and* $\Delta \longrightarrow \Omega$ *then* $\Psi \vdash [\Omega]P$ *true*.

J.5 Soundness of Eliminations (Equality and Proposition)

Lemma 88 (Soundness of Equality Elimination). *Go to proof*
If $[\Gamma]\sigma = \sigma$ *and* $[\Gamma]t = t$ *and* $\Gamma \vdash \sigma : \kappa$ *and* $\Gamma \vdash t : \kappa$ *and* $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(t) = \emptyset$, *then:*

(1) *If* $\Gamma \;/\; \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta$
then $\Delta = (\Gamma, \Theta)$ *where* $\Theta = (\alpha_1 = t_1, \ldots, \alpha_n = t_n)$ *and*
for all Ω *such that* $\Gamma \longrightarrow \Omega$
and all t' *such that* $\Omega \vdash t' : \kappa'$,
it is the case that $[\Omega, \Theta]t' = [\theta][\Omega]t'$, *where* $\theta = \mathsf{mgu}(\sigma, t)$.

(2) *If* $\Gamma \;/\; \sigma \mathrel{\mathring{=}} t : \kappa \dashv \bot$ *then* $\mathsf{mgu}(\sigma, t) = \bot$ *(that is, no most general unifier exists).*

J.6 Soundness of Subtyping

Theorem 6 (Soundness of Algorithmic Subtyping). *Go to proof*
If $[\Gamma]A = A$ *and* $[\Gamma]B = B$ *and* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type and* $\Delta \longrightarrow \Omega$ *and* $\Gamma \vdash A <:^{\pm} B \dashv \Delta$ *then* $[\Omega]\Delta \vdash [\Omega]A \leq^{\pm} [\Omega]B$.

J.7 Soundness of Typing

Theorem 7 (Soundness of Match Coverage). *Go to proof*

1. *If* $\Gamma \vdash \Pi$ *covers* $\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *then* $[\Omega]\Gamma \vdash \Pi$ *covers* $\vec{A}$.
2. *If* $\Gamma \;/\; P \vdash \Pi$ *covers* $\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *and* $[\Gamma]P = P$ *then* $[\Omega]\Gamma \;/\; P \vdash \Pi$ *covers* $\vec{A}$.

Lemma 89 (Well-formedness of Algorithmic Typing). *Go to proof*
Given Γ *ctx:*

(i) *If* $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$ *then* $\Delta \vdash A\ p$ *type.*

(ii) *If* $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $\Delta \vdash B\ q$ *type.*

Definition 7 (Measure). *Let measure* $\mathcal{M}$ *on typing judgments be a lexicographic ordering:*

1. *first, the subject expression* e*, spine* s*, or matches* Π*—regarding all types in annotations as equal in size;*
2. *second, the partial order on judgment forms where an ordinary spine judgment is* smaller *than a principality-recovering spine judgment—and with all other judgment forms considered equal in size; and,*
3. *third, the derivation height.*

$$\left\langle e/s/\Pi, \quad \begin{array}{c}\text{ordinary spine judgment}\\ < \\ \text{recovering spine judgment}\end{array}, \quad \mathsf{height}(\mathcal{D}) \right\rangle$$

Note that this definition doesn't take notice of whether a spine judgment is declarative or algorithmic.

This measure works to show soundness and completeness. We list each rule below, along with a 3-tuple. For example, for Sub we write $\langle =, =, < \rangle$, meaning that each judgment to which we need to apply the i.h. has a subject of the same size ($=$), a judgment form of the same size ($=$), and a smaller derivation height. We write $-$ when a part of the measure need not be considered because a lexicographically more significant part is smaller, as in the Anno rule, where the premise has a smaller subject: $\langle <, -, - \rangle$.

Algorithmic rules (soundness cases):

- Var, 1I, 1I$\hat{\alpha}$, EmptySpine and Nil have no premises, or only auxiliary judgments as premises.
- Sub: $\langle =, =, < \rangle$

- Anno: $\langle <, -, - \rangle$
- $\forall$I, $\forall$Spine, $\wedge$I: $\langle =, =, < \rangle$
- $\supset$I: $\langle =, =, < \rangle$
- $\supset$I$\bot$ has only an auxiliary judgment, to which we need not apply the i.h., putting it in the same class as the rules with no premises.
- $\supset$Spine: $\langle =, =, < \rangle$
- $\rightarrow$I, $\rightarrow$I$\hat{\alpha}$, $\rightarrow$E, Rec: $\langle <, -, - \rangle$
- SpineRecover: $\langle =, <, - \rangle$
- SpinePass: $\langle =, <, - \rangle$
- $\rightarrow$Spine, $+$I$_k$, $+$I$\hat{\alpha}_k$, $\times$I, $\times$I$\hat{\alpha}$, Cons: $\langle <, -, - \rangle$
- $\hat{\alpha}$Spine: $\langle =, =, < \rangle$
- Case: $\langle <, -, - \rangle$

Declarative rules (completeness cases):

- DeclVar, Decl1I, DeclEmptySpine and DeclNil have no premises, or only auxiliary judgments as premises.
- DeclSub: $\langle =, =, < \rangle$
- DeclAnno: $\langle <, -, - \rangle$
- Decl$\forall$I, Decl$\forall$Spine, Decl$\wedge$I, Decl$\supset$I, Decl$\supset$Spine: $\langle =, =, < \rangle$
- Decl$\rightarrow$I, Decl$\rightarrow$E, DeclRec: $\langle <, -, - \rangle$
- DeclSpineRecover: $\langle =, <, - \rangle$
- DeclSpinePass: $\langle =, <, - \rangle$
- Decl$\rightarrow$Spine, Decl$+$I$_k$, Decl$\times$I, DeclCase, DeclCons, $\langle <, -, - \rangle$

Theorem 8 (Soundness of Algorithmic Typing). *Go to proof*
Given $\Delta \longrightarrow \Omega$*:*

(i) *If* $\Gamma \vdash e \Leftarrow A\ p \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $[\Omega]\Delta \vdash [\Omega]e \Leftarrow [\Omega]A\ p$.

(ii) *If* $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$ *then* $[\Omega]\Delta \vdash [\Omega]e \Rightarrow [\Omega]A\ p$.

(iii) *If* $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $[\Omega]\Delta \vdash [\Omega]s : [\Omega]A\ p \gg [\Omega]B\ q$.

(iv) *If* $\Gamma \vdash s : A\ p \gg B\ \lceil q \rceil \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $[\Omega]\Delta \vdash [\Omega]s : [\Omega]A\ p \gg [\Omega]B\ \lceil q \rceil$.

(v) *If* $\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *and* $\Gamma \vdash C\ p$ *type*
then $[\Omega]\Delta \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$.

(vi) *If* $\Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$ *and* $\Gamma \vdash P$ *prop and* $\mathsf{FEV}(P) = \emptyset$ *and* $[\Gamma]P = P$
and $\Gamma \vdash \vec{A}$ **!** *types and* $\Gamma \vdash C\ p$ *type*
then $[\Omega]\Delta\ /\ [\Omega]P \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$.

K Completeness

K.1 Completeness of Auxiliary Judgments

Lemma 90 (Completeness of Instantiation). Go to proof
Given $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$ *and* $\Gamma \vdash \tau : \kappa$ *and* $\tau = [\Gamma]\tau$ *and* $\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$ *and* $\hat{\alpha} \notin FV(\tau)$*:*
If $[\Omega]\hat{\alpha} = [\Omega]\tau$
then there are Δ, Ω' *such that* $\Omega \longrightarrow \Omega'$ *and* $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$.

Lemma 91 (Completeness of Checkeq). Go to proof
Given $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$
and $\Gamma \vdash \sigma : \kappa$ *and* $\Gamma \vdash \tau : \kappa$
and $[\Omega]\sigma = [\Omega]\tau$
then $\Gamma \vdash [\Gamma]\sigma \stackrel{\circ}{=} [\Gamma]\tau : \kappa \dashv \Delta$
where $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$.

Lemma 92 (Completeness of Elimeq). Go to proof
If $[\Gamma]\sigma = \sigma$ *and* $[\Gamma]t = t$ *and* $\Gamma \vdash \sigma : \kappa$ *and* $\Gamma \vdash t : \kappa$ *and* $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(t) = \emptyset$ *then:*

(1) *If* $\mathsf{mgu}(\sigma, t) = \theta$
then $\Gamma \mathrel{/} \sigma \stackrel{\circ}{=} t : \kappa \dashv (\Gamma, \Delta)$
where Δ *has the form* $\alpha_1 = t_1, \ldots, \alpha_n = t_n$
and for all u *such that* $\Gamma \vdash u : \kappa$*, it is the case that* $[\Gamma, \Delta]u = \theta([\Gamma]u)$.

(2) *If* $\mathsf{mgu}(\sigma, t) = \bot$ *(that is, no most general unifier exists) then* $\Gamma \mathrel{/} \sigma \stackrel{\circ}{=} t : \kappa \dashv \bot$.

Lemma 93 (Substitution Upgrade). Go to proof
If Δ *has the form* $\alpha_1 = t_1, \ldots, \alpha_n = t_n$
and, for all u *such that* $\Gamma \vdash u : \kappa$*, it is the case that* $[\Gamma, \Delta]u = \theta([\Gamma]u)$*,*
then:

(i) *If* $\Gamma \vdash A$ *type then* $[\Gamma, \Delta]A = \theta([\Gamma]A)$.

(ii) *If* $\Gamma \longrightarrow \Omega$ *then* $[\Omega]\Gamma = \theta([\Omega]\Gamma)$.

(iii) *If* $\Gamma \longrightarrow \Omega$ *then* $[\Omega, \Delta](\Gamma, \Delta) = \theta([\Omega]\Gamma)$.

(iv) *If* $\Gamma \longrightarrow \Omega$ *then* $[\Omega, \Delta]e = \theta([\Omega]e)$.

Lemma 94 (Completeness of Propequiv). Go to proof
Given $\Gamma \longrightarrow \Omega$
and $\Gamma \vdash P$ *prop and* $\Gamma \vdash Q$ *prop*
and $[\Omega]P = [\Omega]Q$
then $\Gamma \vdash [\Gamma]P \equiv [\Gamma]Q \dashv \Delta$
where $\Delta \longrightarrow \Omega'$ *and* $\Omega \longrightarrow \Omega'$.

Lemma 95 (Completeness of Checkprop). Go to proof
If $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$
and $\Gamma \vdash P$ *prop*
and $[\Gamma]P = P$
and $[\Omega]\Gamma \vdash [\Omega]P$ *true*
then $\Gamma \vdash P$ *true* $\dashv \Delta$
where $\Delta \longrightarrow \Omega'$ *and* $\Omega \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$.

K.2 Completeness of Equivalence and Subtyping

Lemma 96 (Completeness of Equiv). Go to proof
If $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type*

and $[\Omega]A = [\Omega]B$
then there exist Δ *and* Ω' *such that* $\Delta \longrightarrow \Omega'$ *and* $\Omega \longrightarrow \Omega'$ *and* $\Gamma \vdash [\Gamma]A \equiv [\Gamma]B \dashv \Delta$.

Theorem 9 (Completeness of Subtyping). Go to proof
If $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$ *and* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type*
and $[\Omega]\Gamma \vdash [\Omega]A \leq^{\pm} [\Omega]B$
then there exist Δ *and* Ω' *such that* $\Delta \longrightarrow \Omega'$
and $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$
and $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash [\Gamma]A <:^{\pm} [\Gamma]B \dashv \Delta$.

K.3 Completeness of Typing

Theorem 10 (Completeness of Match Coverage). Go to proof

1. *If* $[\Omega]\Gamma \vdash [\Omega]\Pi$ *covers* $[\Omega]\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$
   then $\Gamma \vdash \Pi$ *covers* $\vec{A}$.

2. *If* $[\Omega]\Gamma \ / \ [\Omega]P \vdash [\Omega]\Pi$ *covers* $[\Omega]\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *and* $[\Gamma]P = P$
   then $\Gamma \ / \ P \vdash \Pi$ *covers* $\vec{A}$.

Theorem 11 (Completeness of Algorithmic Typing). Go to proof *Given* $\Gamma \longrightarrow \Omega$ *such that* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$:

(i) *If* $\Gamma \vdash A\ p$ *type and* $[\Omega]\Gamma \vdash [\Omega]e \Leftarrow [\Omega]A\ p$ *and* $p' \sqsubseteq p$
then there exist Δ *and* Ω'
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash e \Leftarrow [\Gamma]A\ p' \dashv \Delta$.

(ii) *If* $\Gamma \vdash A\ p$ *type and* $[\Omega]\Gamma \vdash [\Omega]e \Rightarrow A\ p$
then there exist Δ, Ω', A', *and* $p' \sqsubseteq p$
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash e \Rightarrow A'\ p' \dashv \Delta$ *and* $A' = [\Delta]A'$ *and* $A = [\Omega']A'$.

(iii) *If* $\Gamma \vdash A\ p$ *type and* $[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg B\ q$ *and* $p' \sqsubseteq p$
then there exist Δ, Ω', B' *and* $q' \sqsubseteq q$
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash s : [\Gamma]A\ p' \gg B'\ q' \dashv \Delta$ *and* $B' = [\Delta]B'$ *and* $B = [\Omega']B'$.

(iv) *If* $\Gamma \vdash A\ p$ *type and* $[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg B\ \lceil q \rceil$ *and* $p' \sqsubseteq p$
then there exist Δ, Ω', B', *and* $q' \sqsubseteq q$
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash s : [\Gamma]A\ p' \gg B'\ \lceil q' \rceil \dashv \Delta$ *and* $B' = [\Delta]B'$ *and* $B = [\Omega']B'$.

(v) *If* $\Gamma \vdash \vec{A}$ **!** *types and* $\Gamma \vdash C\ p$ *type and* $[\Omega]\Gamma \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$ *and* $p' \sqsubseteq p$
then there exist Δ, Ω', *and* C
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash \Pi :: [\Gamma]\vec{A} \Leftarrow [\Gamma]C\ p' \dashv \Delta$.

(vi) *If* $\Gamma \vdash \vec{A}$ **!** *types and* $\Gamma \vdash P$ *prop and* $\mathsf{FEV}(P) = \emptyset$ *and* $\Gamma \vdash C\ p$ *type*
and $[\Omega]\Gamma \ / \ [\Omega]P \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$
and $p' \sqsubseteq p$
then there exist Δ, Ω', *and* C
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \ / \ [\Gamma]P \vdash \Pi :: [\Gamma]\vec{A} \Leftarrow [\Gamma]C\ p' \dashv \Delta$.

Proofs

In the rest of this document, we prove the results stated above, with the same sectioning.

A′ Properties of the Declarative System

Lemma 1 (Declarative Well-foundedness).
The inductive definition of the following judgments is well-founded:

(i) *synthesis* $\Psi \vdash e \Rightarrow B\ p$

(ii) *checking* $\Psi \vdash e \Leftarrow A\ p$

(iii) *checking, equality elimination* $\Psi \mathrel{/} P \vdash e \Leftarrow C\ p$

(iv) *ordinary spine* $\Psi \vdash s : A\ p \gg B\ q$

(v) *recovery spine* $\Psi \vdash s : A\ p \gg B\ \lceil q \rceil$

(vi) *pattern matching* $\Psi \vdash \Pi :: \vec{A} \Leftarrow C\ p$

(vii) *pattern matching, equality elimination* $\Psi \mathrel{/} P \vdash \Pi :: \vec{A} \Leftarrow C\ p$

Proof. Let $|e|$ be the size of the expression e. Let $|s|$ be the size of the spine s. Let $|\Pi|$ be the size of the branch list Π. Let $\#\mathsf{large}(A)$ be the number of "large" connectives $\forall$, $\exists$, $\supset$, $\wedge$ in A.

First, stratify judgments by the size of the term (expression, spine, or branches), and say that a judgment is *at* n if it types a term of size n. Order the main judgment forms as follows:

$$\begin{array}{rl} & \text{synthesis judgment at } n \\ < & \text{checking judgments at } n \\ < & \text{ordinary spine judgment at } n \\ < & \text{recovery spine judgment at } n \\ < & \text{match judgments at } n \\ < & \text{synthesis judgment at } n+1 \\ & \vdots \end{array}$$

Within the checking judgment forms at n, we compare types lexicographically, first by the number of large connectives, and then by the ordinary size. Within the match judgment forms at n, we compare using a lexicographic order of, first, $\#\mathsf{large}(\vec{A})$; second, the judgment form, considering the match judgment to be smaller than the matchelim judgment; third, the size of $\vec{A}$. These criteria order the judgments as follows:

$$\begin{array}{rl} & \text{synthesis judgment at } n \\ < & \big(\text{checking judgment at } n \text{ with } \#\mathsf{large}(A) = 1 \\ & \quad < \text{checkelim judgment at } n \text{ with } \#\mathsf{large}(A) = 1 \\ & \quad < \text{checking judgment at } n \text{ with } \#\mathsf{large}(A) = 2 \\ & \quad < \text{checkelim judgment at } n \text{ with } \#\mathsf{large}(A) = 2 \\ & \quad < \ldots\big) \\ < & \big(\text{match judgment at } n \text{ with } \#\mathsf{large}(\vec{A}) = 1 \text{ and } \vec{A} \text{ of size } 1 \\ & \quad < \text{match judgment at } n \text{ with } \#\mathsf{large}(\vec{A}) = 1 \text{ and } \vec{A} \text{ of size } 2 \\ & \quad < \text{matchelim judgment at } n \text{ with } \#\mathsf{large}(\vec{A}) = 1 \\ & \quad < \text{match judgment at } n \text{ with } \#\mathsf{large}(\vec{A}) = 2 \text{ and } \vec{A} \text{ of size } 1 \\ & \quad < \text{match judgment at } n \text{ with } \#\mathsf{large}(\vec{A}) = 2 \text{ and } \vec{A} \text{ of size } 2 \\ & \quad < \text{matchelim judgment at } n \text{ with } \#\mathsf{large}(\vec{A}) = 2 \\ & \quad < \ldots\big) \end{array}$$

The class of ordinary spine judgments at 1 need not be refined, because the only ordinary spine rule applicable to a spine of size 1 is DeclEmptySpine, which has no premises; rules Decl$\forall$Spine, Decl$\supset$Spine, and Decl$\to$Spine are restricted to non-empty spines and can only apply to larger terms.

Similarly, the class of match judgments at 1 need not be refined, because only DeclMatchEmpty is applicable.

Note that we distinguish the "checkelim" form $\Psi \mathrel{/} P \vdash e \Leftarrow A\ p$ of the checking judgment. We also define the size of an expression e to consider all types in annotations to be of the same size, that is,

$$|(e : A)| \;=\; |e| + 1$$

Thus, $|\theta(e)| = |e|$, even when e has annotations. This is used for DeclCheckUnify; see below.

We assume that coverage, which does not depend on any other typing judgments, is well-founded. We likewise assume that subtyping, $\Psi \vdash A$ *type*, $\Psi \vdash \tau : \kappa$, and $\Psi \vdash P$ *prop* are well-founded.

We now show that, for each class of judgments, every judgment in that class depends only on smaller judgments.

- **Synthesis judgments**

  Claim: For all n, synthesis at n depends only on judgments at $n-1$ or less.

  Proof. Rule DeclVar has no premises.
  Rule DeclAnno depends on a premise at a strictly smaller term.
  Rule Decl$\to$E depends on (1) a synthesis premise at a strictly smaller term, and (2) a recovery spine judgment at a strictly smaller term.

- **Checking judgments**

  Claim: For all $n \geq 1$, the checking judgment over terms of size n with type of size m depends only on

  (1) synthesis judgments at size n or smaller, and
  (2) checking judgments at size $n-1$ or smaller, and
  (3) checking judgments at size n with fewer large connectives, and
  (4) checkelim judgments at size n with fewer large connectives, and
  (5) match judgments at size $n-1$ or smaller.

  Proof. Rule DeclSub depends on a synthesis judgment of size n. (1)
  Rule Decl1I has no premises.
  Rule Decl$\forall$I depends on a checking judgment at n with fewer large connectives. (3)
  Rule Decl$\wedge$I depends on a checking judgment at n with fewer large connectives. (3)
  Rule Decl$\supset$I depends on a checkelim judgment at n with fewer large connectives. (4)
  Rules Decl$\to$I, DeclRec, Decl$+$I$_k$, Decl$\times$I, and DeclCons depend on checking judgments at size $< n$. (2)
  Rule DeclNil depends only on an auxiliary judgment.
  Rule DeclCase depends on:

  – a synthesis judgment at size n (1),
  – a match judgment at size $< n$ (5), and
  – a coverage judgment.

- **Checkelim judgments**

  Claim: For all $n \geq 1$, the checkelim judgment $\Psi \mathrel{/} P \vdash e \Leftarrow A\ p$ over terms of size n depends only on checking judgments at size n, with a type A' such that $\#\mathsf{large}(A') = \#\mathsf{large}(A)$.

  Proof. Rule DeclCheck$\bot$ has no nontrivial premises.
  Rule DeclCheckUnify depends on a checking judgment: Since $|\theta(e)| = |e|$, this checking judgment is at n. Since the mgu θ is over monotypes, $\#\mathsf{large}(\theta(A)) = \#\mathsf{large}(A)$.

- **Ordinary spine judgments**

  An ordinary spine judgment at 1 depends on no other judgments: the only spine of size 1 is the empty spine, so only DeclEmptySpine applies, and it has no premises.

  Claim: For all $n \geq 2$, the ordinary spine judgment $\Psi \vdash s : A\ p \gg C\ q$ over spines of size n depends only on

  (a) checking judgments at size $n-1$ or smaller, and
  (b) ordinary spine judgments at size $n-1$ or smaller, and
  (c) ordinary spine judgments at size n with strictly smaller $\#\mathsf{large}(A)$.

  Proof. Rule Decl∀Spine depends on an ordinary spine judgment of size n, with a type that has fewer large connectives. (c)
  Rule Decl⊃Spine depends on an ordinary spine judgment of size n, with a type that has fewer large connectives. (c)
  Rule DeclEmptySpine has no premises.
  Rule Decl→Spine depends on a checking judgment of size $n-1$ or smaller (a) and an ordinary spine judgment of size $n-1$ or smaller (b).

- **Recovery spine judgments**

  Claim: For all n, the recovery spine judgment at n depends only on ordinary spine judgments at n.

  Proof. Rules DeclSpineRecover and DeclSpinePass depend only on ordinary spine judgments at n.

- **Match judgments**

  Claim: For all $n \geq 1$, the match judgment $\Psi \vdash \Pi :: \vec{A} \Leftarrow C\ p$ over Π of size n depends only on

  (a) checking judgments at size $n-1$ or smaller, and
  (b) match judgments at size $n-1$ or smaller, and
  (c) match judgments at size n with smaller $\vec{A}$, and
  (d) matchelim judgments at size n with fewer large connectives in $\vec{A}$.

  Proof. Rule DeclMatchEmpty has no premises.
  Rule DeclMatchSeq depends on match judgments at $n-1$ or smaller (b).
  Rule DeclMatchBase depends on a checking judgment at $n-1$ or smaller (a).
  Rules DeclMatchUnit, DeclMatch×, DeclMatch$+_k$, DeclMatchNeg, and DeclMatchWild depend on match judgments at $n-1$ or smaller (b).
  Rule DeclMatch∃ depends on a match judgment at size n with smaller $\vec{A}$ (c).
  Rule DeclMatch∧ depends on an matchelim judgment at n, with fewer large connectives in $\vec{A}$. (d)

- **Matchelim judgments**

  Claim: For all $n \geq 1$, the matchelim judgment $\Psi\ /\ \Pi \vdash P :: \vec{A} \Leftarrow C\ p$ over Ψ of size n depends only on match judgments with the same number of large connectives in $\vec{A}$.

  Proof. Rule DeclMatch⊥ has no nontrivial premises.
  Rule DeclMatchUnify depends on a match judgment with the same number of large connectives (similar to DeclCheckUnify, considered above). ☐

Lemma 2 (Declarative Weakening)**.**

(i) *If* $\Psi_0, \Psi_1 \vdash t : \kappa$ *then* $\Psi_0, \Psi, \Psi_1 \vdash t : \kappa$.

(ii) *If* $\Psi_0, \Psi_1 \vdash P$ *prop then* $\Psi_0, \Psi, \Psi_1 \vdash P$ *prop*.

(iii) *If* $\Psi_0, \Psi_1 \vdash P$ *true then* $\Psi_0, \Psi, \Psi_1 \vdash P$ *true*.

(iv) *If* $\Psi_0, \Psi_1 \vdash A$ *type then* $\Psi_0, \Psi, \Psi_1 \vdash A$ *type*.

Proof. By induction on the derivation. $\square$

Lemma 3 (Declarative Term Substitution)**.** *Suppose* $\Psi \vdash t : \kappa$. *Then:*

1. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash t' : \kappa$ *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]t' : \kappa$.
2. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash P$ *prop then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]P$ *prop*.
3. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash A$ *type then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]A$ *type*.
4. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash A \leq^{\pm} B$ *then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]A \leq^{\pm} [t/\alpha]B$.
5. *If* $\Psi_0, \alpha : \kappa, \Psi_1 \vdash P$ *true then* $\Psi_0, [t/\alpha]\Psi_1 \vdash [t/\alpha]P$ *true*.

Proof. By induction on the derivation of the substitutee. $\square$

Lemma 4 (Reflexivity of Declarative Subtyping)**.**
Given $\Psi \vdash A$ *type, we have that* $\Psi \vdash A \leq^{\pm} A$.

Proof. By induction on A, writing p for the sign of the subtyping judgment.

Our induction metric is the number of quantifiers on the outside of A, plus one if the polarity of A and the subtyping judgment do not match up (that is, if *neg*(A) and $p = +$, or *pos*(A) and $p = -$).

- **Case** *nonpos*(A), *nonneg*(A), $p = \pm$:

  By rule $\leq$Refl$\pm$.

- **Case** $A = \exists b : \kappa.\ B, p = +$:

$\Psi, b : \kappa \vdash B \leq^{+} B$	By i.h. (one less quantifier)
$\Psi, b : \kappa \vdash b : \kappa$	By rule UvarSort
$\Psi, b : \kappa \vdash B \leq^{+} \exists b : \kappa.\ B$	By rule $\leq\exists$R
$\Psi \vdash \exists b : \kappa.\ B \leq^{+} \exists b : \kappa.\ B$	By rule $\leq\exists$L

- **Case** $A = \exists b : \kappa.\ B, p = -$:

$\Psi \vdash \exists b : \kappa.\ B \leq^{+} \exists b : \kappa.\ B$	By i.h. (polarities match)
$\Psi \vdash \exists b : \kappa.\ B \leq^{-} \exists b : \kappa.\ B$	By $\leq^{+}_{-}$

- **Case** $A = \forall b : \kappa.\ B, p = +$:

$\Psi \vdash \forall b : \kappa.\ B \leq^{-} \forall b : \kappa.\ B$	By i.h. (polarities match)
$\Psi \vdash \forall b : \kappa.\ B \leq^{+} \forall b : \kappa.\ B$	By $\leq^{-}_{+}$

- **Case** $A = \forall b : \kappa.\ B, p = -$:

$\Psi, b : \kappa \vdash B \leq^{-} B$	By i.h. (one less quantifier)
$\Psi, b : \kappa \vdash b : \kappa$	By rule UvarSort
$\Psi, b : \kappa \vdash \forall b : \kappa.\ B \leq^{-} B$	By rule $\leq\forall$L
$\Psi \vdash \forall b : \kappa.\ B \leq^{-} \forall b : \kappa.\ B$	By rule $\leq\forall$R

$\square$

Lemma 5 (Subtyping Inversion)**.**

- *If* $\Psi \vdash \exists\alpha : \kappa.\ A \leq^{+} B$ *then* $\Psi, \alpha : \kappa \vdash A \leq^{+} B$.
- *If* $\Psi \vdash A \leq^{-} \forall\beta : \kappa.\ B$ *then* $\Psi, \beta : \kappa \vdash A \leq^{-} B$.

Proof. By a routine induction on the subtyping derivations. $\square$

Lemma 6 (Subtyping Polarity Flip)**.**

- *If nonpos*(A) *and nonpos*(B) *and* $\Psi \vdash A \leq^+ B$
  then $\Psi \vdash A \leq^- B$ *by a derivation of the same or smaller size.*
- *If nonneg*(A) *and nonneg*(B) *and* $\Psi \vdash A \leq^- B$
  then $\Psi \vdash A \leq^+ B$ *by a derivation of the same or smaller size.*
- *If nonpos*(A) *and nonneg*(A) *and nonpos*(B) *and nonneg*(B) *and* $\Psi \vdash A \leq^\pm B$
  then $A = B$.

Proof. By a routine induction on the subtyping derivations. ☐

Lemma 7 (Transitivity of Declarative Subtyping).
Given $\Psi \vdash A$ *type and* $\Psi \vdash B$ *type and* $\Psi \vdash C$ *type:*

(i) If $\mathcal{D}_1 :: \Psi \vdash A \leq^\pm B$ *and* $\mathcal{D}_2 :: \Psi \vdash B \leq^\pm C$
then $\Psi \vdash A \leq^\pm C$.

Proof. By lexicographic induction on (1) the sum of head quantifiers in A, B, and C, and (2) the size of the derivation.

We begin by case analysis on the shape of B, and the polarity of subtyping:

- Case $B = \forall\beta : \kappa_2.\, B'$, polarity $= -$:

  We case-analyze $\mathcal{D}_1$:

  - **Case** $\dfrac{\Psi \vdash \tau : \kappa_1 \qquad \Psi \vdash [\tau/\alpha]A' \leq^- B}{\Psi \vdash \forall\alpha : \kappa_1.\, A' \leq^- B} \leq\!\forall\mathsf{L}$

$\Psi \vdash \tau : \kappa_1$	Subderivation
$\Psi \vdash [\tau/\alpha]A' \leq^- B$	Subderivation
$\Psi \vdash B \leq^- C$	Given
$\Psi \vdash [\tau/\alpha]A' \leq^- C$	By i.h. (A lost a quantifier)
$\Psi \vdash A \leq^- C$	By rule $\leq\forall\mathsf{L}$

  - **Case** $\dfrac{\Psi, \beta : \kappa_2 \vdash A \leq^- B'}{\Psi \vdash A \leq^- \forall\beta : \kappa_2.\, B'} \leq\!\forall\mathsf{R}$

    We case-analyze $\mathcal{D}_2$:

    * **Case** $\dfrac{\Psi \vdash \tau : \kappa_2 \qquad \Psi \vdash [\tau/\beta]B' \leq^- C}{\Psi \vdash \forall\beta : \kappa_2.\, B' \leq^- C} \leq\!\forall\mathsf{L}$

$\Psi, \beta : \kappa_2 \vdash A \leq^- B'$	By Lemma 5 (Subtyping Inversion) on $\mathcal{D}_1$
$\Psi \vdash \tau : \kappa_2$	Subderivation
$\Psi \vdash [\tau/\beta]B' \leq^- C$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^- [\tau/\beta]B'$	By Lemma 3 (Declarative Term Substitution)
$\Psi \vdash A \leq^- C$	By i.h. (B lost a quantifier)

    * **Case** $\dfrac{\Psi, c : \kappa_3 \vdash B \leq^- C'}{\Psi \vdash B \leq^- \forall c : \kappa_3.\, C'} \leq\!\forall\mathsf{R}$

$\Psi \vdash A \leq^- B$	Given
$\Psi, c : \kappa_3 \vdash A \leq^- B$	By Lemma 2 (Declarative Weakening)
$\Psi, c : \kappa_3 \vdash B \leq^- C'$	Subderivation
$\Psi, c : \kappa_3 \vdash A \leq^- C'$	By i.h. (C lost a quantifier)
$\Psi \vdash B \leq^- \forall c : \kappa_3.\, C'$	By $\leq\forall\mathsf{R}$

- Case $nonpos(B)$, polarity $= +$:

  Now we case-analyze $\mathcal{D}_1$:

  – **Case** $$\dfrac{\Psi, \alpha : \tau \vdash A' \leq^+ B}{\Psi \vdash \underbrace{\exists \alpha : \kappa_1.\, A'}_{A} \leq^+ B} \leq\exists\mathsf{L}$$

$\Psi, \alpha : \tau \vdash A' \leq^+ B$	Subderivation
$\Psi, \alpha : \tau \vdash B \leq^+ C$	By Lemma 2 (Declarative Weakening) ($\mathcal{D}_2$)
$\Psi, \alpha : \tau \vdash A' \leq^+ C$	By i.h. (A lost a quantifier)
$\Psi \vdash \exists \alpha : \kappa_1.\, A' \leq^+ C$	By $\leq\exists\mathsf{L}$

  – **Case** $$\dfrac{\Psi \vdash A \leq^- B \qquad nonpos(A) \qquad nonpos(B)}{\Psi \vdash A \leq^+ B} \leq^-_+$$

  Now we case-analyze $\mathcal{D}_2$:

  ∗ **Case** $$\dfrac{\Psi \vdash \tau : \kappa_3 \qquad \Psi \vdash B \leq^+ [\tau/c]C'}{\Psi \vdash B \leq^+ \underbrace{\exists c : \kappa_3.\, C'}_{C}} \leq\exists\mathsf{R}$$

$\Psi \vdash A \leq^+ B$	Given
$\Psi \vdash \tau : \kappa_3$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash B \leq^+ [\tau/c]C'$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^+ [\tau/c]C'$	By i.h. (C lost a quantifier)
$\Psi \vdash A \leq^+ \exists c : \kappa_3.\, C'$	By $\leq\exists\mathsf{R}$

  ∗ **Case** $$\dfrac{\Psi \vdash B \leq^- C \qquad nonpos(B) \qquad nonpos(C)}{\Psi \vdash B \leq^+ C} \leq^-_+$$

$\Psi \vdash A \leq^- B$	Subderivation of $\mathcal{D}_1$
$\Psi \vdash B \leq^- C$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^- C$	By i.h. ($\mathcal{D}_1$ and $\mathcal{D}_2$ smaller)
$nonpos(A)$	Subderivation of $\mathcal{D}_1$
$nonpos(C)$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^+ C$	By $\leq^-_+$

- Case $B = \exists \beta : \kappa_2.\, B'$, polarity $= +$:

  Now we case-analyze $\mathcal{D}_2$:

  – **Case** $$\dfrac{\Psi \vdash \tau : \kappa_3 \qquad \Psi \vdash B \leq^+ [\tau/\alpha]C'}{\Psi \vdash B \leq^+ \underbrace{\exists \alpha : \kappa_3.\, C'}_{C}} \leq\exists\mathsf{R}$$

$\Psi \vdash \tau : \kappa_3$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash B \leq^+ [\tau/\alpha]C'$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^+ B$	Given
$\Psi \vdash A \leq^+ [\tau/\alpha]C'$	By i.h. (C lost a quantifier)
$\Psi \vdash A \leq^+ C$	By rule $\leq\exists\mathsf{R}$

– **Case** $\dfrac{\Psi, \beta : \kappa_2 \vdash B' \leq^+ C}{\Psi \vdash \exists \beta : \kappa_2.\, B' \leq^+ C} \leq\exists\mathsf{L}$

Now we case-analyze $\mathcal{D}_1$:

∗ **Case** $\dfrac{\Psi \vdash \tau : \kappa_2 \qquad \Psi \vdash A \leq^+ [\tau/\beta]B'}{\Psi \vdash A \leq^+ \underbrace{\exists \beta : \kappa_2.\, B'}_{B}} \leq\exists\mathsf{R}$

$\Psi, \beta : \kappa_2 \vdash B' \leq^+ C$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash \tau : \kappa_2$	Subderivation of $\mathcal{D}_1$
$\Psi \vdash A \leq^+ [\tau/\beta]B'$	Subderivation of $\mathcal{D}_1$
$\Psi \vdash [\tau/\beta]B' \leq^+ C$	By Lemma 3 (Declarative Term Substitution)
$\Psi \vdash A \leq^+ C$	By i.h. (B lost a quantifier)

∗ **Case** $\dfrac{\Psi, \alpha : \kappa_1 \vdash A \leq^+ B}{\Psi \vdash \underbrace{\exists \alpha : \kappa_1.\, A'}_{A} \leq^+ B} \leq\exists\mathsf{L}$

$\Psi \vdash B \leq^+ C$	Given
$\Psi, \alpha : \kappa_1 \vdash A' \leq^+ B$	Subderivation of $\mathcal{D}_1$
$\Psi, \alpha : \kappa_1 \vdash A' \leq^+ B$	By Lemma 2 (Declarative Weakening)
$\Psi, \alpha : \kappa_1 \vdash A' \leq^+ C$	By i.h. (A lost a quantifier)
$\Psi \vdash \exists \alpha : \kappa_1.\, A' \leq^+ C$	By $\leq\exists\mathsf{L}$

- Case *nonneg*(B), polarity $= -$:

  We case-analyze $\mathcal{D}_2$:

– **Case** $\dfrac{\Psi, c : \kappa_3 \vdash B \leq^+ C'}{\Psi \vdash B \leq^+ \underbrace{\exists c : \kappa_3.\, C'}_{C}} \leq\forall\mathsf{R}$

$\Psi, c : \kappa_3 \vdash B \leq^+ C'$	Subderivation of $\mathcal{D}_2$
$\Psi, c : \kappa_3 \vdash A \leq^+ B$	By Lemma 2 (Declarative Weakening)
$\Psi, c : \kappa_3 \vdash A \leq^+ C'$	By i.h. (C lost a quantifier)
$\Psi \vdash A \leq^+ \forall c : \kappa_3.\, C'$	By $\leq\forall\mathsf{R}$

– **Case** $\dfrac{\Psi \vdash B \leq^+ C \qquad \textit{nonneg}(B) \qquad \textit{nonneg}(C)}{\Psi \vdash B \leq^- C} \leq^+_-$

We case-analyze $\mathcal{D}_1$:

∗ **Case** $\dfrac{\Psi \vdash \tau : \kappa_1 \qquad \Psi \vdash [\tau/\alpha]A' \leq^- B}{\Psi \vdash \underbrace{\forall \alpha : \kappa_1.\, A'}_{A} \leq^- B} \leq\forall\mathsf{L}$

$\Psi \vdash B \leq^- C$	Given
$\Psi \vdash \tau : \kappa_1$	Subderivation of $\mathcal{D}_1$
$\Psi \vdash [\tau/\alpha]A' \leq^- B$	Subderivation of $\mathcal{D}_1$
$\Psi \vdash [\tau/\alpha]A' \leq^- C$	By i.h. (A lost a quantifier)
$\Psi \vdash \forall \alpha : \kappa_1.\, A' \leq^- C$	By $\leq\forall\mathsf{L}$

* **Case** $\dfrac{\Psi \vdash A \leq^+ B \qquad \textit{nonpos}(A) \qquad \textit{nonpos}(B)}{\Psi \vdash A \leq^- B} \leq^+_-$

$\Psi \vdash A \leq^+ B$	Subderivation of $\mathcal{D}_1$
$\Psi \vdash B \leq^+ C$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^+ C$	By i.h. ($\mathcal{D}_1$ and $\mathcal{D}_2$ smaller)
$\textit{nonneg}(A)$	Subderivation of $\mathcal{D}_2$
$\textit{nonneg}(C)$	Subderivation of $\mathcal{D}_2$
$\Psi \vdash A \leq^- C$	By $\leq^+_-$

□

B′ Substitution and Well-formedness Properties

Lemma 8 (Substitution—Well-formedness)**.**

(i) If $\Gamma \vdash A\ p$ *type and* $\Gamma \vdash \tau\ p$ *type then* $\Gamma \vdash [\tau/\alpha]A\ p$ *type.*

(ii) If $\Gamma \vdash P$ *prop and* $\Gamma \vdash \tau\ p$ *type then* $\Gamma \vdash [\tau/\alpha]P$ *prop.*
Moreover, if $p = \mathbf{!}$ *and* $\mathsf{FEV}([\Gamma]P) = \emptyset$ *then* $\mathsf{FEV}([\Gamma][\tau/\alpha]P) = \emptyset$*.*

Proof. By induction on the derivations of $\Gamma \vdash A\ p$ *type* and $\Gamma \vdash P$ *prop*. □

Lemma 9 (Uvar Preservation)**.**
If $\Delta \longrightarrow \Omega$ *then:*

(i) If $(\alpha : \kappa) \in \Omega$ *then* $(\alpha : \kappa) \in [\Omega]\Delta$*.*

(ii) If $(x : A\ p) \in \Omega$ *then* $(x : [\Omega]A\ p) \in [\Omega]\Delta$*.*

Proof. By induction on Ω, following the definition of context application (Figure 11). □

Lemma 10 (Sorting Implies Typing)**.** *If* $\Gamma \vdash t : \star$ *then* $\Gamma \vdash t$ *type.*

Proof. By induction on the given derivation. All cases are straightforward. □

Lemma 11 (Right-Hand Substitution for Sorting)**.** *If* $\Gamma \vdash t : \kappa$ *then* $\Gamma \vdash [\Gamma]t : \kappa$*.*

Proof. By induction on $|\Gamma \vdash t|$ (the size of t under Γ).

- **Cases** UnitSort: Here $t = 1$, so applying Γ to t does not change it: $t = [\Gamma]t$. Since $\Gamma \vdash t : \kappa$, we have $\Gamma \vdash [\Gamma]t : \kappa$, which was to be shown.
- **Case** VarSort: If t is an existential variable $\hat{\alpha}$, then $\Gamma = \Gamma_0[\hat{\alpha}]$, so applying Γ to t does not change it, and we proceed as in the UnitSort case above.

  If t is a universal variable α and Γ has no equation for it, then proceed as in the UnitSort case.

  Otherwise, $t = \alpha$ and $(\alpha = \tau) \in \Gamma$:

  $$\Gamma \quad = \quad (\Gamma_L, \alpha : \kappa, \Gamma_M, \alpha = \tau, \Gamma_R)$$

  By the implicit assumption that Γ is well-formed, $\Gamma_L, \alpha : \kappa, \Gamma_M \vdash \tau : \kappa$.
  By Lemma 34 (Suffix Weakening), $\Gamma \vdash \tau : \kappa$. Since $|\Gamma \vdash \tau| < |\Gamma \vdash \alpha|$, we can apply the i.h., giving

  $$\Gamma \vdash [\Gamma]\tau : \kappa$$

  By the definition of substitution, $[\Gamma]\tau = [\Gamma]\alpha$, so we have $\Gamma \vdash [\Gamma]\alpha : \kappa$.

- **Case** SolvedVarSort: In this case $t = \hat{\alpha}$ and $\Gamma = (\Gamma_L, \hat{\alpha} = \tau, \Gamma_R)$. Thus $[\Gamma]t = [\Gamma]\hat{\alpha} = [\Gamma_L]\tau$. We assume contexts are well-formed, so all free variables in τ are declared in Γ_L. Consequently, $|\Gamma_L \vdash \tau| = |\Gamma \vdash \tau|$, which is less than $|\Gamma \vdash \hat{\alpha}|$. We can therefore apply the i.h. to τ, yielding $\Gamma \vdash [\Gamma]\tau : \kappa$. By the definition of substitution, $[\Gamma]\tau = [\Gamma]\hat{\alpha}$, so we have $\Gamma \vdash [\Gamma]\hat{\alpha} : \kappa$.

- **Case** BinSort: In this case $t = t_1 \oplus t_2$. By i.h., $\Gamma \vdash [\Gamma]t_1 : \kappa$ and $\Gamma \vdash [\Gamma]t_2 : \kappa$. By BinSort, $\Gamma \vdash ([\Gamma]t_1) \oplus ([\Gamma]t_2) : \kappa$, which by the definition of substitution is $\Gamma \vdash [\Gamma](t_1 \oplus t_2) : \kappa$. □

Lemma 12 (Right-Hand Substitution for Propositions)**.** *If* $\Gamma \vdash P$ *prop then* $\Gamma \vdash [\Gamma]P$ *prop.*

Proof. Use inversion (EqProp), apply Lemma 11 (Right-Hand Substitution for Sorting) to each premise, and apply EqProp again. □

Lemma 13 (Right-Hand Substitution for Typing)**.** *If* $\Gamma \vdash A$ *type then* $\Gamma \vdash [\Gamma]A$ *type.*

Proof. By induction on $|\Gamma \vdash A|$ (the size of A under Γ).

Several cases correspond to cases in the proof of Lemma 11 (Right-Hand Substitution for Sorting):

- the case for UnitWF is like the case for UnitSort;
- the case for SolvedVarSort is like the cases for VarWF and SolvedVarWF,
- the case for VarSort is like the case for VarWF, but in the last subcase, apply Lemma 10 (Sorting Implies Typing) to move from a sorting judgment to a typing judgment.
- the case for BinWF is like the case for BinSort.

Now, the new cases:

- **Case** ForallWF: In this case $A = \forall\alpha : \kappa.\, A_0$. By i.h., $\Gamma, \alpha : \kappa \vdash [\Gamma, \alpha : \kappa]A_0$ *type*. By the definition of substitution, $[\Gamma, \alpha : \kappa]A_0 = [\Gamma]A_0$, so by ForallWF, $\Gamma \vdash \forall\alpha.\, [\Gamma]A_0$ *type*, which by the definition of substitution is $\Gamma \vdash [\Gamma](\forall\alpha.\, A_0)$ *type*.

- **Case** ExistsWF: Similar to the ForallWF case.

- **Case** ImpliesWF, WithWF: Use the i.h. and Lemma 12 (Right-Hand Substitution for Propositions), then apply ImpliesWF or WithWF. □

Lemma 14 (Substitution for Sorting)**.** *If* $\Omega \vdash t : \kappa$ *then* $[\Omega]\Omega \vdash [\Omega]t : \kappa$.

Proof. By induction on $|\Omega \vdash t|$ (the size of t under Ω).

- **Case** $\dfrac{u : \kappa \in \Omega}{\Omega \vdash u : \kappa}$ VarSort

  We have a complete context Ω, so u cannot be an existential variable: it must be some universal variable α.

  If Ω lacks an equation for α, use Lemma 9 (Uvar Preservation) and apply rule UvarSort.

  Otherwise, $(\alpha = \tau \in \Omega$, so we need to show $\Omega \vdash [\Omega]\tau : \kappa$. By the implicit assumption that Ω is well-formed, plus Lemma 34 (Suffix Weakening), $\Omega \vdash \tau : \kappa$. By Lemma 11 (Right-Hand Substitution for Sorting), $\Omega \vdash [\Omega]\tau : \kappa$.

- **Case** $\dfrac{\hat{\alpha} : \kappa = \tau \in \Omega}{\Omega \vdash \hat{\alpha} : \kappa}$ SolvedVarSort

$$\begin{array}{rll}
& \hat{\alpha} : \kappa = \tau \in \Omega & \text{Subderivation} \\
& \Omega = (\Omega_L, \hat{\alpha} : \kappa = \tau, \Omega_R) & \text{Decomposing } \Omega \\
& \Omega_L \vdash \tau : \kappa & \text{By implicit assumption that } \Omega \text{ is well-formed} \\
& \Omega_L, \hat{\alpha} : \kappa = \tau, \Omega_R \vdash \tau : \kappa & \text{By Lemma 34 (Suffix Weakening)} \\
& \Omega \vdash [\Omega]\tau : \kappa & \text{By Lemma 11 (Right-Hand Substitution for Sorting)} \\
☞ & [\Omega]\Omega \vdash [\Omega]\hat{\alpha} : \kappa & [\Omega]\tau = [\Omega]\hat{\alpha}
\end{array}$$

- **Case** $\dfrac{}{\Omega \vdash 1 : \star}$ UnitSort

  Since $1 = [\Omega]1$, applying UnitSort gives the result.

- **Case** $\dfrac{\Omega \vdash \tau_1 : \star \qquad \Omega \vdash \tau_2 : \star}{\Omega \vdash \tau_1 \oplus \tau_2 : \star}$ BinSort

  By i.h. on each premise, rule BinSort, and the definition of substitution.

- **Case** $\dfrac{}{\Omega \vdash \mathsf{zero} : \mathbb{N}}$ ZeroSort

  Since $\mathsf{zero} = [\Omega]\mathsf{zero}$, applying ZeroSort gives the result.

- **Case** $\dfrac{\Omega \vdash t : \mathbb{N}}{\Omega \vdash \mathsf{succ}(t) : \mathbb{N}}$ SuccSort

  By i.h., rule SuccSort, and the definition of substitution. □

Lemma 15 (Substitution for Prop Well-Formedness)**.**
If $\Omega \vdash P$ *prop then* $[\Omega]\Omega \vdash [\Omega]P$ *prop.*

Proof. Only one rule derives this judgment form:

- **Case** $\dfrac{\Omega \vdash t : \mathbb{N} \qquad \Omega \vdash t' : \mathbb{N}}{\Omega \vdash t = t' \textit{ prop}}$ EqProp

$$\begin{array}{rll}
& \Omega \vdash t : \mathbb{N} & \text{Subderivation} \\
& [\Omega]\Omega \vdash [\Omega]t : \mathbb{N} & \text{By Lemma 14 (Substitution for Sorting)} \\
& \Omega \vdash t' : \mathbb{N} & \text{Subderivation} \\
& [\Omega]\Omega \vdash [\Omega]t' : \mathbb{N} & \text{By Lemma 14 (Substitution for Sorting)} \\
& [\Omega]\Omega \vdash ([\Omega]t) = ([\Omega]t') \textit{ prop} & \text{By EqProp} \\
☞ & [\Omega]\Omega \vdash [\Omega](t = t') \textit{ prop} & \text{By def. of subst.}
\end{array}$$

□

Lemma 16 (Substitution for Type Well-Formedness)**.** *If* $\Omega \vdash A$ *type then* $[\Omega]\Omega \vdash [\Omega]A$ *type.*

Proof. By induction on $|\Omega \vdash A|$.

Several cases correspond to those in the proof of Lemma 14 (Substitution for Sorting):

- the UnitWF case is like the UnitSort case (using DeclUnitWF instead of UnitSort);

- the VarWF case is like the VarSort case (using DeclUvarWF instead of UvarSort);
- the SolvedVarWF case is like the SolvedVarSort case.

However, uses of Lemma 11 (Right-Hand Substitution for Sorting) are replaced by uses of Lemma 13 (Right-Hand Substitution for Typing).

Now, the new cases:

- **Case** $$\frac{\Omega, \alpha : \kappa \vdash A_0 \; \textit{type}}{\Omega \vdash \forall \alpha : \kappa.\, A_0 \; \textit{type}} \; \mathsf{ForallWF}$$

	$\Omega, \alpha : \kappa \vdash A_0 : \kappa'$	Subderivation
	$[\Omega, \alpha : \kappa](\Omega, \alpha : \kappa) \vdash [\Omega]A_0 : \kappa'$	By i.h.
	$[\Omega]\Omega, \alpha : \kappa \vdash [\Omega]A_0 : \kappa'$	By definition of completion
	$[\Omega]\Omega \vdash \forall \alpha : \kappa.\, [\Omega]A_0 : \kappa'$	By DeclAllWF
☞	$[\Omega]\Omega \vdash [\Omega](\forall \alpha : \kappa.\, A_0) : \kappa'$	By def. of subst.

- **Case** ExistsWF: Similar to the ForallWF case, using DeclExistsWF instead of DeclAllWF.

- **Case** $$\frac{\Omega \vdash A_1 \; \textit{type} \qquad \Omega \vdash A_2 \; \textit{type}}{\Omega \vdash A_1 \oplus A_2 \; \textit{type}} \; \mathsf{BinWF}$$

  By i.h. on each premise, rule DeclBinWF, and the definition of substitution.

- **Case** VecWF: Similar to the BinWF case.

- **Case** $$\frac{\Omega \vdash P \; \textit{prop} \qquad \Omega \vdash A_0 \; \textit{type}}{\Omega \vdash P \supset A_0 \; \textit{type}} \; \mathsf{ImpliesWF}$$

	$\Omega \vdash P \; \textit{prop}$	Subderivation
	$[\Omega]\Omega \vdash [\Omega]P \; \textit{prop}$	By Lemma 15 (Substitution for Prop Well-Formedness)
	$\Omega \vdash A_0 \; \textit{type}$	Subderivation
	$[\Omega]\Omega \vdash [\Omega]A_0 \; \textit{type}$	By i.h.
	$[\Omega]\Omega \vdash ([\Omega]P) \supset ([\Omega]A_0) \; \textit{type}$	By DeclImpliesWF
☞	$[\Omega]\Omega \vdash [\Omega](P \supset A_0) \; \textit{type}$	By def. of subst.

- **Case** $$\frac{\Omega \vdash P \; \textit{prop} \qquad \Omega \vdash A_0 \; \textit{type}}{\Omega \vdash A_0 \wedge P \; \textit{type}} \; \mathsf{WithWF}$$

  Similar to the ImpliesWF case. □

Lemma 17 (Substitution Stability)**.**
If (Ω, Ω_Z) *is well-formed and* Ω_Z *is soft and* $\Omega \vdash A$ *type then* $[\Omega]A = [\Omega, \Omega_Z]A$.

Proof. By induction on Ω_Z.

Since Ω_Z is soft, either (1) $\Omega_Z = \cdot$ (and the result is immediate) or (2) $\Omega_Z = (\Omega', \hat{\alpha} : \kappa)$ or (3) $\Omega_Z = (\Omega', \hat{\alpha} : \kappa = t)$. However, according to the grammar for complete contexts such as Ω_Z, (2) is impossible. Only case (3) remains.

By i.h., $[\Omega]A = [\Omega, \Omega']A$. Use the fact that $\Omega \vdash A$ *type* implies $FV(A) \cap \mathsf{dom}(\Omega_Z) = \emptyset$. □

Lemma 18 (Equal Domains)**.**
If $\Omega_1 \vdash A$ *type and* $\mathsf{dom}(\Omega_1) = \mathsf{dom}(\Omega_2)$ *then* $\Omega_2 \vdash A$ *type.*

Proof. By induction on the given derivation. □

C′ Properties of Extension

Lemma 19 (Declaration Preservation). *If $\Gamma \longrightarrow \Delta$ and u is declared in Γ, then u is declared in Δ.*

Proof. By induction on the derivation of $\Gamma \longrightarrow \Delta$.

- **Case** $\dfrac{}{\cdot \longrightarrow \cdot}\ \longrightarrow\mathsf{Id}$

  This case is impossible, since by hypothesis u is declared in Γ.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]A = [\Delta]A'}{\Gamma, x : A \longrightarrow \Delta, x : A'}\ \longrightarrow\mathsf{Var}$

  - Case $u = x$: Immediate.
  - Case $u \neq x$: Since u is declared in $(\Gamma, x : A)$, it is declared in Γ. By i.h., u is declared in Δ, and therefore declared in $(\Delta, x : A')$.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \alpha : \kappa \longrightarrow \Delta, \alpha : \kappa}\ \longrightarrow\mathsf{Uvar}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \hat{\alpha} : \kappa \longrightarrow \Delta, \hat{\alpha} : \kappa}\ \longrightarrow\mathsf{Unsolved}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\Gamma, \hat{\alpha} : \kappa = t \longrightarrow \Delta, \hat{\alpha} : \kappa = t'}\ \longrightarrow\mathsf{Solved}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\Gamma, \alpha = t \longrightarrow \Delta, \alpha = t'}\ \longrightarrow\mathsf{Eqn}$

  It is given that u is declared in $(\Gamma, \alpha = t)$. Since $\alpha = t$ is not a declaration, u is declared in Γ.
  By i.h., u is declared in Δ, and therefore declared in $(\Delta, \alpha = t'$.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \blacktriangleright_{\hat{\alpha}} \longrightarrow \Delta, \blacktriangleright_{\hat{\alpha}}}\ \longrightarrow\mathsf{Marker}$

  Similar to the $\longrightarrow$Eqn case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \hat{\beta} : \kappa' \longrightarrow \Delta, \hat{\beta} : \kappa' = t}\ \longrightarrow\mathsf{Solve}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \hat{\alpha} : \kappa} \longrightarrow\mathsf{Add}$

  It is given that u is declared in Γ. By i.h., u is declared in Δ, and therefore declared in $(\Delta, \hat{\alpha} : \kappa)$.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \hat{\alpha} : \kappa = t} \longrightarrow\mathsf{AddSolved}$

  Similar to the $\longrightarrow$Add case. ☐

Lemma 20 (Declaration Order Preservation)**.** *If $\Gamma \longrightarrow \Delta$ and u is declared to the left of v in Γ, then u is declared to the left of v in Δ.*

Proof. By induction on the derivation of $\Gamma \longrightarrow \Delta$.

- **Case** $\dfrac{}{\cdot \longrightarrow \cdot} \longrightarrow\mathsf{Id}$

  This case is impossible, since by hypothesis u and v are declared in Γ.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]A = [\Delta]A'}{\Gamma, x : A \longrightarrow \Delta, x : A'} \longrightarrow\mathsf{Var}$

  Consider whether $v = x$:

  - Case $v = x$:
    It is given that u is declared to the left of v in $(\Gamma, x : A)$, so u is declared in Γ.
    By Lemma 19 (Declaration Preservation), u is declared in Δ.
    Therefore u is declared to the left of v in $(\Delta, x : A')$.
  - Case $v \neq x$:
    Here, v is declared in Γ. By i.h., u is declared to the left of v in Δ.
    Therefore u is declared to the left of v in $(\Delta, x : A')$.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \alpha : \kappa \longrightarrow \Delta, \alpha : \kappa} \longrightarrow\mathsf{Uvar}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \hat{\alpha} : \kappa \longrightarrow \Delta, \hat{\alpha} : \kappa} \longrightarrow\mathsf{Unsolved}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\Gamma, \hat{\alpha} : \kappa = t \longrightarrow \Delta, \hat{\alpha} : \kappa = t'} \longrightarrow\mathsf{Solved}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \hat{\beta} : \kappa' \longrightarrow \Delta, \hat{\beta} : \kappa' = t} \longrightarrow\mathsf{Solve}$

  Similar to the $\longrightarrow$Var case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\Gamma, \alpha = t \longrightarrow \Delta, \alpha = t'} \longrightarrow\!\mathsf{Eqn}$

  The equation $\hat{\alpha} = t$ does not declare any variables, so u and v must be declared in Γ.
  By i.h., u is declared to the left of v in Δ.
  Therefore u is declared to the left of v in $\Delta, \hat{\alpha} : \kappa = t'$.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \blacktriangleright_{\hat{\alpha}} \longrightarrow \Delta, \blacktriangleright_{\hat{\alpha}}} \longrightarrow\!\mathsf{Marker}$

  Similar to the $\longrightarrow\!\mathsf{Eqn}$ case.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \hat{\alpha} : \kappa} \longrightarrow\!\mathsf{Add}$

  By i.h., u is declared to the left of v in Δ.
  Therefore u is declared to the left of v in $(\Delta, \hat{\alpha} : \kappa)$.

- **Case** $\dfrac{\Gamma \longrightarrow \Delta}{\Gamma \longrightarrow \Delta, \hat{\alpha} : \kappa = t} \longrightarrow\!\mathsf{AddSolved}$

  Similar to the $\longrightarrow\!\mathsf{Add}$ case. □

Lemma 21 (Reverse Declaration Order Preservation)**.** *If* $\Gamma \longrightarrow \Delta$ *and* u *and* v *are both declared in* Γ *and* u *is declared to the left of* v *in* Δ*, then* u *is declared to the left of* v *in* Γ*.*

Proof. It is given that u and v are declared in Γ. Either u is declared to the left of v in Γ, or v is declared to the left of u. Suppose the latter (for a contradiction). By Lemma 20 (Declaration Order Preservation), v is declared to the left of u in Δ. But we know that u is declared to the left of v in Δ: contradiction. Therefore u is declared to the left of v in Γ. □

Lemma 22 (Extension Inversion)**.**

(i) *If* $\mathcal{D} :: \Gamma_0, \alpha : \kappa, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0 *and* Δ_1
such that $\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$*.*

Moreover, if Γ_1 *is soft, then* Δ_1 *is soft.*

(ii) *If* $\mathcal{D} :: \Gamma_0, \blacktriangleright_u, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0 *and* Δ_1
such that $\Delta = (\Delta_0, \blacktriangleright_u, \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$*.*

Moreover, if Γ_1 *is soft, then* Δ_1 *is soft.*

Moreover, if $\mathsf{dom}(\Gamma_0, \blacktriangleright_u, \Gamma_1) = \mathsf{dom}(\Delta)$ *then* $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Delta_0)$*.*

(iii) *If* $\mathcal{D} :: \Gamma_0, \alpha = \tau, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0*,* τ'*, and* Δ_1
such that $\Delta = (\Delta_0, \alpha = \tau', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *and* $[\Delta_0]\tau = [\Delta_0]\tau'$ *where* $\mathcal{D}' < \mathcal{D}$*.*

(iv) *If* $\mathcal{D} :: \Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0*,* τ'*, and* Δ_1
such that $\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *and* $[\Delta_0]\tau = [\Delta_0]\tau'$ *where* $\mathcal{D}' < \mathcal{D}$*.*

(v) *If* $\mathcal{D} :: \Gamma_0, x : A, \Gamma_1 \longrightarrow \Delta$
then there exist unique Δ_0*,* A'*, and* Δ_1
such that $\Delta = (\Delta_0, x : A', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *and* $[\Delta_0]A = [\Delta_0]A'$ *where* $\mathcal{D}' < \mathcal{D}$*.*

Moreover, if Γ_1 *is soft, then* Δ_1 *is soft.*

Moreover, if $\mathsf{dom}(\Gamma_0, x : A, \Gamma_1) = \mathsf{dom}(\Delta)$ *then* $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Delta_0)$*.*

(vi) *If* $\mathcal{D} :: \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \longrightarrow \Delta$ *then either*

- *there exist unique* Δ_0*,* τ'*, and* Δ_1
  such that $\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$*,*
  or
- *there exist unique* Δ_0 *and* Δ_1
  such that $\Delta = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1)$ *and* $\mathcal{D}' :: \Gamma_0 \longrightarrow \Delta_0$ *where* $\mathcal{D}' < \mathcal{D}$*.*

Proof. In each part, we proceed by induction on the derivation of $\Gamma_0, \dots, \Gamma_1 \longrightarrow \Delta$.

Note that in each part, the $\longrightarrow\!\mathsf{Id}$ case is impossible.

Throughout this proof, we shadow Δ so that it refers to the *largest proper prefix* of the Δ in the statement of the lemma. For example, in the $\longrightarrow\!\mathsf{Var}$ case of part (i), we really have $\Delta = (\Delta_{00}, x : A')$, but we call Δ_{00} "Δ".

(i) We have $\Gamma_0, \alpha : \kappa, \Gamma_1 \longrightarrow \Delta$.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]A = [\Delta]A'}{\underbrace{\Gamma, x : A}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, x : A'} \longrightarrow\!\mathsf{Var}$$

$$
\begin{array}{rcll}
(\Gamma, x : A) &=& (\Gamma_0, \alpha : \kappa, \Gamma_1) & \text{Given} \\
&=& (\Gamma_0, \alpha : \kappa, \Gamma_1', x : A) & \text{Since the last element must be equal} \\
(\Gamma, x : A) &=& (\Gamma_0, \alpha : \kappa, \Gamma_1', x : A) & \text{By transitivity} \\
\Gamma &=& (\Gamma_0, \alpha : \kappa, \Gamma_1') & \text{By injectivity of syntax}
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma \longrightarrow \Delta & \text{Subderivation} \\
& \Gamma_0, \alpha : \kappa, \Gamma_1' \longrightarrow \Delta & \text{By equality} \\
& \Delta = (\Delta_0, \alpha : \kappa, \Delta_1) & \text{By i.h.} \\
☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\
& \text{if } \Gamma_1' \text{ soft then } \Delta_1 \text{ soft} & '' \\
☞ & (\Delta, x : A') = (\Delta_0, \alpha : \kappa, \Delta_1, x : A') & \text{By congruence} \\
☞ & \text{if } \Gamma_1', x : A \text{ soft then } \Delta_1, x : A' \text{ soft} & \text{Since } \Gamma_1', x : A \text{ is not soft}
\end{array}
$$

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \beta : \kappa'}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \beta : \kappa'} \longrightarrow\!\mathsf{Uvar}$$

  There are two cases:

  - **Case** $\alpha : \kappa = \beta : \kappa'$:

$$
\begin{array}{lrl}
☞ & (\Gamma, \alpha : \kappa) = (\Gamma_0, \alpha : \kappa, \Gamma_1) & \text{where } \Gamma_0 = \Gamma \text{ and } \Gamma_1 = \cdot \\
☞ & (\Delta, \alpha : \kappa) = (\Delta_0, \alpha : \kappa, \Delta_1) & \text{where } \Delta_0 = \Delta \text{ and } \Delta_1 = \cdot \\
☞ & \text{if } \Gamma_1 \text{ soft then } \Delta_1 \text{ soft} & \text{since } \cdot \text{ is soft}
\end{array}
$$

- **Case** $\alpha \neq \beta$:

$$
\begin{array}{lrll}
 & (\Gamma, \beta : \kappa') = (\Gamma_0, \alpha : \kappa, \Gamma_1) & & \text{Given} \\
 & = (\Gamma_0, \alpha : \kappa, \Gamma_1', \beta : \kappa') & & \text{Since the last element must be equal} \\
 & \Gamma = (\Gamma_0, \alpha : \kappa, \Gamma_1') & & \text{By injectivity of syntax} \\
 & \Gamma \longrightarrow \Delta & & \text{Subderivation} \\
 & \Gamma_0, \alpha : \kappa, \Gamma_1' \longrightarrow \Delta & & \text{By equality} \\
 & \Delta = (\Delta_0, \alpha : \kappa, \Delta_1) & & \text{By i.h.} \\
☞ & \Gamma_0 \longrightarrow \Delta_0 & & '' \\
 & \text{if } \Gamma_1' \text{ soft then } \Delta_1 \text{ soft} & & '' \\
☞ & (\Delta, \beta : \kappa') = (\Delta_0, \alpha : \kappa, \Delta_1, \beta : \kappa') & & \text{By congruence} \\
☞ & \text{if } \Gamma_1', \beta : \kappa' \text{ soft then } \Delta_1, \beta : \kappa' \text{ soft} & & \text{Since } \Gamma_1', \beta : \kappa' \text{ is not soft}
\end{array}
$$

- **Case**

$$
\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \hat{\alpha} : \kappa'}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\alpha} : \kappa'} \longrightarrow\!\mathsf{Unsolved}
$$

$$
\begin{array}{lrl}
 & (\Gamma, \hat{\alpha} : \kappa') = (\Gamma_0, \alpha : \kappa, \Gamma_1) & \text{Given} \\
 & = (\Gamma_0, \alpha : \kappa, \Gamma_1', \hat{\alpha} : \kappa') & \text{Since the last element must be equal} \\
 & \Gamma = (\Gamma_0, \alpha : \kappa, \Gamma_1') & \text{By injectivity of syntax} \\
 & \Gamma \longrightarrow \Delta & \text{Subderivation} \\
 & \Gamma_0, \alpha : \kappa, \Gamma_1' \longrightarrow \Delta & \text{By equality} \\
 & \Delta = (\Delta_0, \alpha : \kappa, \Delta_1) & \text{By i.h.} \\
☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\
 & \text{if } \Gamma_1' \text{ soft then } \Delta_1 \text{ soft} & '' \\
☞ & (\Delta, \hat{\alpha} : \kappa') = (\Delta_0, \alpha : \kappa, \Delta_1, \hat{\alpha} : \kappa') & \text{By congruence}
\end{array}
$$

Suppose $\Gamma_1', \hat{\alpha} : \kappa'$ soft.

$$
\begin{array}{lrl}
 & \Gamma_1' \ \text{soft} & \text{By definition of softness} \\
 & \Delta_1 \ \text{soft} & \text{By induction} \\
 & \Delta_1 \ \text{soft} & \text{By definition of softness} \\
☞ & \text{if } \Gamma_1', \hat{\alpha} : \kappa' \text{ soft then } \Delta_1, \hat{\alpha} : \kappa' \text{ soft} & \text{Implication introduction}
\end{array}
$$

- **Case**

$$
\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \hat{\alpha} : \kappa = t}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\alpha} : \kappa = t'} \longrightarrow\!\mathsf{Solved}
$$

Similar to the $\longrightarrow$Unsolved case.

- **Case**

$$
\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \beta = t}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \beta = t'} \longrightarrow\!\mathsf{Eqn}
$$

	$(\Gamma, \beta = t) = (\Gamma_0, \alpha : \kappa, \Gamma_1)$	Given
	$= (\Gamma_0, \alpha : \kappa, \Gamma_1', \beta = t)$	Since the last element must be equal
	$\Gamma = (\Gamma_0, \alpha : \kappa, \Gamma_1')$	By injectivity of syntax
	$\Gamma \longrightarrow \Delta$	Subderivation
	$\Gamma_0, \alpha : \kappa, \Gamma_1' \longrightarrow \Delta$	By equality
	$\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$	By i.h.
☞	$\Gamma_0 \longrightarrow \Delta_0$	"
	if Γ_1' soft then Δ_1 soft	"
☞	$(\Delta, \beta = t') = (\Delta_0, \alpha : \kappa, \Delta_1, \beta = t')$	By congruence
☞	if $\Gamma_1', \beta = t$ soft then $\Delta_1, \beta = t'$ soft	Since $\Gamma_1', \beta = t$ is not soft

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \blacktriangleright_{\hat{\alpha}}}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \blacktriangleright_{\hat{\alpha}}} \longrightarrow\!\mathsf{Marker}$$

	$(\Gamma, \blacktriangleright_{\hat{\alpha}}) = (\Gamma_0, \alpha : \kappa, \Gamma_1)$	Given
	$= (\Gamma_0, \alpha : \kappa, \Gamma_1', \blacktriangleright_{\hat{\alpha}})$	Since the last element must be equal
	$\Gamma = (\Gamma_0, \alpha : \kappa, \Gamma_1')$	By injectivity of syntax
	$\Gamma \longrightarrow \Delta$	Subderivation
	$\Gamma_0, \alpha : \kappa, \Gamma_1' \longrightarrow \Delta$	By equality
	$\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$	By i.h.
☞	$\Gamma_0 \longrightarrow \Delta_0$	"
	if Γ_1' soft then Δ_1 soft	"
☞	$\Delta, \blacktriangleright_{\hat{\alpha}} = (\Delta_0, \alpha : \kappa, \Delta_1, \blacktriangleright_{\hat{\alpha}})$	By congruence
☞	if $\Gamma_1', \blacktriangleright_{\hat{\alpha}}$ soft then $\Delta_1, \blacktriangleright_{\hat{\alpha}}$ soft	Since $\Gamma_1', \blacktriangleright_{\hat{\alpha}}$ is not soft

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \alpha:\kappa', \Gamma_1} \longrightarrow \Delta, \hat{\alpha} : \kappa} \longrightarrow\!\mathsf{Add}$$

	$\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$	By i.h.
☞	$\Gamma_0 \longrightarrow \Delta_0$	"
	if Γ_1 soft then Δ_1 soft	"
☞	$\Delta, \hat{\alpha} : \kappa' = (\Delta_0, \alpha : \kappa, \Delta_1, \hat{\alpha} : \kappa')$	By congruence of equality

Suppose Γ_1 soft.

	Δ_1 soft	By i.h.
	$\Delta_1, \hat{\alpha} : \kappa'$ soft	By definition of softnesss
☞	if Γ_1 soft then $\Delta_1, \hat{\alpha} : \kappa'$ soft	Implication introduction

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\alpha} : \kappa' = t} \longrightarrow\!\mathsf{AddSolved}$$

	$\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$	By i.h.
☞	$\Gamma_0 \longrightarrow \Delta_0$	″
	if Γ_1 soft then Δ_1 soft	″
☞	$(\Delta, \hat{\alpha} : \kappa' = t) = (\Delta_0, \alpha : \kappa, \Delta_1, \hat{\alpha} : \kappa' = t)$	By congruence of equality

	Suppose Γ_1 soft.	
	Δ_1 soft	By i.h.
	$(\Delta_1, \hat{\alpha} : \kappa' = t)$ soft	By definition of softnesss
☞	if Γ_1 soft then $\Delta_1, \hat{\alpha} : \kappa' = t$ soft	Implication introduction

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \hat{\beta} : \kappa'}_{\Gamma_0, \alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa' = t} \longrightarrow\!\mathsf{Solve}$$

	$(\Gamma, \hat{\beta} : \kappa') = (\Gamma_0, \alpha : \kappa, \Gamma_1)$	Given
	$= (\Gamma_0, \alpha : \kappa, \Gamma_1', \hat{\beta} : \kappa')$	Since the final elements are equal
	$\Gamma = (\Gamma_0, \alpha : \kappa, \Gamma_1')$	By injectivity of context syntax

	$\Gamma \longrightarrow \Delta$	Subderivation
	$\Gamma_0, \alpha : \kappa, \Gamma_1' \longrightarrow \Delta$	By equality
	$\Delta = (\Delta_0, \alpha : \kappa, \Delta_1)$	By i.h.
☞	$\Gamma_0 \longrightarrow \Delta_0$	″
	if Γ_1' soft then Δ_1 soft	″
☞	$\Delta, \hat{\beta} : \kappa' = \Delta_0, \alpha : \kappa, \Delta_1, \hat{\beta} : \kappa'$	By congruence

	Suppose $\Gamma_1', \hat{\beta} : \kappa'$ soft.	
	Γ_1' soft	By definition of softness
	Δ_1 soft	Using i.h.
	$\Delta_1, \hat{\beta} : \kappa' = t$ soft	By definition of softness
☞	if $\Gamma_1', \hat{\beta} : \kappa'$ soft then $\Delta_1, \hat{\beta} : \kappa' = t$ soft	Implication intro

(ii) We have $\Gamma_0, \blacktriangleright_u, \Gamma_1 \longrightarrow \Delta$. This part is similar to part (i) above, except for "if $\mathsf{dom}(\Gamma_0, \blacktriangleright_u, \Gamma_1) = \mathsf{dom}(\Delta)$ then $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Delta_0)$", which follows by i.h. in most cases. In the $\longrightarrow$Marker case, either we have $\ldots, \blacktriangleright_{u'}$ where $u' = u$—in which case the i.h. gives us what we need—or we have a matching $\blacktriangleright_u$. In this latter case, we have $\Gamma_1 = \cdot$. We know that $\mathsf{dom}(\Gamma_0, \blacktriangleright_u, \Gamma_1) = \mathsf{dom}(\Delta)$ and $\Delta = (\Delta_0, \blacktriangleright_u)$. Since $\Gamma_1 = \cdot$, we have $\mathsf{dom}(\Gamma_0, \blacktriangleright_u) = \mathsf{dom}(\Delta_0, \blacktriangleright_u)$. Therefore $\mathsf{dom}(\Gamma_0) = \mathsf{dom}(\Delta_0)$.

(iii) We have $\Gamma_0, \alpha = \tau, \Gamma_1 \longrightarrow \Delta$.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \beta : \kappa'}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, \beta : \kappa'} \longrightarrow\!\mathsf{Uvar}$$

	$(\Gamma_0, \alpha = \tau, \Gamma_1) = (\Gamma, \beta : \kappa')$	Given
	$= (\Gamma_0, \alpha = \tau, \Gamma_1', \beta : \kappa')$	Since the final elements must be equal
	$\Gamma = (\Gamma_0, \alpha = \tau, \Gamma_1')$	By injectivity of context syntax

	$\Delta = (\Delta_0, \alpha = \tau', \Delta_1)$	By i.h.
☞	$[\Delta_0]\tau = [\Delta_0]\tau'$	″
☞	$\Gamma_0 \longrightarrow \Delta_0$	″
☞	$(\Delta, \beta : \kappa') = (\Delta_0, \alpha = \tau', \Delta_1, \beta : \kappa')$	By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]A = [\Delta]A'}{\underbrace{\Gamma, x : A}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, x : A'} \longrightarrow\!\mathsf{Var}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \blacktriangleright_{\hat\alpha} \longrightarrow \Delta, \blacktriangleright_{\hat\alpha}} \longrightarrow\!\mathsf{Marker}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \hat\alpha : \kappa' \longrightarrow \Delta, \hat\alpha : \kappa'} \longrightarrow\!\mathsf{Unsolved}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \hat\alpha : \kappa' = t}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, \hat\alpha : \kappa' = t'} \longrightarrow\!\mathsf{Solved}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \hat\beta : \kappa'}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, \hat\beta : \kappa' = t} \longrightarrow\!\mathsf{Solve}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \beta = t}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, \beta = t'} \longrightarrow\!\mathsf{Eqn}$$

  There are two cases:

  - **Case** $\alpha = \beta$:

	$\tau = t$ and $\Gamma_1 = \cdot$ and $\Gamma_0 = \Gamma$	By injectivity of syntax
☞	$\Gamma_0 \longrightarrow \Delta_0$	Subderivation ($\Gamma_0 = \Gamma$ and let $\Delta_0 = \Delta$)
☞	$(\Delta, \alpha = t') = (\Delta_0, \alpha = t', \Delta_1)$	where $\Delta_1 = \cdot$
☞	$[\Delta_0]t = [\Delta_0]t'$	By premise $[\Delta]t = [\Delta]t'$

  - **Case** $\alpha \neq \beta$:

	$(\Gamma_0, \alpha = \tau, \Gamma_1) = (\Gamma, \beta = t)$	Given
	$= (\Gamma_0, \alpha = \tau, \Gamma_1', \beta = t)$	Since the final elements must be equal
	$\Gamma = (\Gamma_0, \alpha = \tau, \Gamma_1')$	By injectivity of context syntax
	$\Delta = (\Delta_0, \alpha = \tau', \Delta_1)$	By i.h.
☞	$[\Delta_0]\tau = [\Delta_0]\tau'$	″
☞	$\Gamma_0 \longrightarrow \Delta_0$	″
☞	$(\Delta, \beta = t') = (\Delta_0, \alpha = \tau', \Delta_1, \beta = t')$	By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, \hat\alpha : \kappa'} \longrightarrow\!\mathsf{Add}$$

$$\Delta = (\Delta_0, \alpha = \tau', \Delta_1) \quad \text{By i.h.}$$
☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ ″
☞ $\Gamma_0 \longrightarrow \Delta_0$ ″
☞ $(\Delta, \hat{\alpha} : \kappa') = (\Delta_0, \alpha = \tau', \Delta_1, \hat{\alpha} : \kappa')$ By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \alpha = \tau, \Gamma_1} \longrightarrow \Delta, \hat{\alpha} : \kappa' = t} \longrightarrow\!\mathsf{AddSolved}$$

$$\Delta = (\Delta_0, \alpha = \tau', \Delta_1) \quad \text{By i.h.}$$
☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ ″
☞ $\Gamma_0 \longrightarrow \Delta_0$ ″
☞ $(\Delta, \hat{\alpha} : \kappa' = t) = (\Delta_0, \alpha = \tau', \Delta_1, \hat{\alpha} : \kappa' = t)$ By congruence of equality

(iv) We have $\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1 \longrightarrow \Delta$.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \beta : \kappa'}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, \beta : \kappa'} \longrightarrow\!\mathsf{Uvar}$$

$$\begin{aligned} (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1) &= (\Gamma, \beta : \kappa') && \text{Given} \\ &= (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1', \beta : \kappa') && \text{Since the final elements must be equal} \\ \Gamma &= (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1') && \text{By injectivity of context syntax} \end{aligned}$$

$$\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1) \quad \text{By i.h.}$$
☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ ″
☞ $\Gamma_0 \longrightarrow \Delta_0$ ″
☞ $(\Delta, \beta : \kappa') = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \beta : \kappa')$ By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]A = [\Delta]A'}{\underbrace{\Gamma, x : A}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, x : A'} \longrightarrow\!\mathsf{Var}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \blacktriangleright_{\hat{\beta}} \longrightarrow \Delta, \blacktriangleright_{\hat{\beta}}} \longrightarrow\!\mathsf{Marker}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \hat{\beta} : \kappa' \longrightarrow \Delta, \hat{\beta} : \kappa'} \longrightarrow\!\mathsf{Unsolved}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \hat{\beta} : \kappa' = t}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa' = t'} \longrightarrow\!\mathsf{Solved}$$

  There are two cases.

  - **Case** $\hat{\alpha} = \hat{\beta}$:

$\kappa' = \kappa$ and $t = \tau$ and $\Gamma_1 = \cdot$ and $\Gamma = \Gamma_0$ — By injectivity of syntax

☞ $(\Delta, \hat{\beta} : \kappa' = t') = (\Delta_0, \hat{\beta} : \kappa' = \tau', \Delta_1)$ where $\tau' = t'$ and $\Delta_1 = \cdot$ and $\Delta = \Delta_0$

☞ $\Gamma_0 \longrightarrow \Delta_0$ — From subderivation $\Gamma \longrightarrow \Delta$

☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ — From premise $[\Delta]t = [\Delta]t'$ and x

– **Case** $\hat{\alpha} \neq \hat{\beta}$:

$(\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1) = (\Gamma, \hat{\beta} : \kappa' = t)$ — Given

$= (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1', \hat{\beta} : \kappa' = t)$ — Since the final elements must be equal

$\Gamma = (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1')$ — By injectivity of context syntax

$\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ — By i.h.

☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ — ″

☞ $\Gamma_0 \longrightarrow \Delta_0$ — ″

☞ $(\Delta, \hat{\beta} : \kappa' = t') = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa' = t')$ — By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \beta = t}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, \beta = t'} \longrightarrow\!\mathsf{Eqn}$$

$(\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1) = (\Gamma, \beta = t)$ — Given

$= (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1', \beta = t)$ — Since the final elements must be equal

$\Gamma = (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1')$ — By injectivity of context syntax

$\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ — By i.h.

☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ — ″

☞ $\Gamma_0 \longrightarrow \Delta_0$ — ″

☞ $(\Delta, \beta = t') = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \beta = t')$ — By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa'} \longrightarrow\!\mathsf{Add}$$

$\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ — By i.h.

☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ — ″

☞ $\Gamma_0 \longrightarrow \Delta_0$ — ″

☞ $(\Delta, \hat{\beta} : \kappa') = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa')$ — By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa' = t} \longrightarrow\!\mathsf{AddSolved}$$

$\Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1)$ — By i.h.

☞ $[\Delta_0]\tau = [\Delta_0]\tau'$ — ″

☞ $\Gamma_0 \longrightarrow \Delta_0$ — ″

☞ $(\Delta, \hat{\beta} : \kappa' = t) = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa' = t)$ — By congruence of equality

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \hat{\beta} : \kappa'}_{\Gamma_0, \hat{\alpha}:\kappa = \tau, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa' = t} \longrightarrow\!\mathsf{Solve}$$

$$
\begin{array}{lrcll}
 & (\Gamma, \hat{\beta} : \kappa') & = & (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1) & \text{Given} \\
 & & = & (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1', \hat{\beta} : \kappa') & \text{Since the last elements must be equal} \\
 & \Gamma & = & (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1') & \text{By injectivity of syntax} \\
 & & & & \\
 & \Gamma & \longrightarrow & \Delta & \text{Subderivation} \\
 & \Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1' & \longrightarrow & \Delta & \text{By equality} \\
 & \Delta & = & (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1) & \text{By i.h.} \\
\text{☞} & [\Delta_0]\tau & = & [\Delta_0]\tau' & '' \\
\text{☞} & \Gamma_0 & \longrightarrow & \Delta_0 & '' \\
\text{☞} & (\Delta, \hat{\beta} : \kappa') & = & (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa') & \text{By congruence of equality}
\end{array}
$$

(v) We have $\Gamma_0, x : A, \Gamma_1 \longrightarrow \Delta$. This proof is similar to the proof of part (i), except for the domain condition, which we handle similarly to part (ii).

(vi) We have $\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \longrightarrow \Delta$.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \beta : \kappa'}_{\Gamma_0, \hat{\alpha}:\kappa, \Gamma_1} \longrightarrow \Delta, \beta : \kappa'} \longrightarrow\mathsf{Uvar}$$

  $$
  \begin{array}{rcll}
  (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1) & = & (\Gamma, \beta : \kappa') & \text{Given} \\
   & = & (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \beta : \kappa') & \text{Since the final elements must be equal} \\
  \Gamma & = & (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By injectivity of context syntax}
  \end{array}
  $$

  By induction, there are two possibilities:

  - $\hat{\alpha}$ is not solved:

    $$
    \begin{array}{lrcll}
     & \Delta & = & (\Delta_0, \hat{\alpha} : \kappa, \Delta_1) & \text{By i.h.} \\
    \text{☞} & \Gamma_0 & \longrightarrow & \Delta_0 & '' \\
    \text{☞} & (\Delta, \beta : \kappa') & = & (\Delta_0, \hat{\alpha} : \kappa, \Delta_1, \beta : \kappa') & \text{By congruence of equality}
    \end{array}
    $$

  - $\hat{\alpha}$ is solved:

    $$
    \begin{array}{lrcll}
     & \Delta & = & (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1) & \text{By i.h.} \\
    \text{☞} & \Gamma_0 & \longrightarrow & \Delta_0 & '' \\
    \text{☞} & (\Delta, \beta : \kappa') & = & (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \beta : \kappa') & \text{By congruence of equality}
    \end{array}
    $$

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]A = [\Delta]A'}{\underbrace{\Gamma, x : A}_{\Gamma_0, \hat{\alpha}:\kappa, \Gamma_1} \longrightarrow \Delta, x : A'} \longrightarrow\mathsf{Var}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\Gamma, \blacktriangleright_{\hat{\beta}} \longrightarrow \Delta, \blacktriangleright_{\hat{\beta}}} \longrightarrow\mathsf{Marker}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\Gamma, \beta = t \longrightarrow \Delta, \beta = t'} \longrightarrow\mathsf{Eqn}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta \qquad [\Delta]t = [\Delta]t'}{\underbrace{\Gamma, \hat{\beta} : \kappa' = t}_{\Gamma_0, \hat{\alpha}:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa' = t'} \longrightarrow\mathsf{Solved}$$

  Similar to the $\longrightarrow$Uvar case.

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \hat{\beta} : \kappa'}_{\Gamma_0, \hat{\alpha}:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa'} \longrightarrow\!\mathsf{Unsolved}$$

  - **Case** $\hat{\alpha} \neq \hat{\beta}$:

    $$\begin{array}{rll} (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1) & = (\Gamma, \hat{\beta} : \kappa') & \text{Given} \\ & = (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \hat{\beta} : \kappa') & \text{Since the final elements must be equal} \\ \Gamma & = (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By injectivity of context syntax} \end{array}$$

    By induction, there are two possibilities:

    * $\hat{\alpha}$ is not solved:

      $$\begin{array}{lrl} & \Delta = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1) & \text{By i.h.} \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\ ☞ & (\Delta, \hat{\beta} : \kappa') = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1, \hat{\beta} : \kappa') & \text{By congruence of equality} \end{array}$$

    * $\hat{\alpha}$ is solved:

      $$\begin{array}{lrl} & \Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1) & \text{By i.h.} \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\ ☞ & (\Delta, \hat{\beta} : \kappa') = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa') & \text{By congruence of equality} \end{array}$$

  - **Case** $\hat{\alpha} = \hat{\beta}$:

    $$\begin{array}{lrl} & \kappa' = \kappa \text{ and } \Gamma_0 = \Gamma \text{ and } \Gamma_1 = \cdot & \text{By injectivity of syntax} \\ ☞ & (\Delta, \hat{\beta} : \kappa') = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1) & \text{where } \Delta_0 = \Delta \text{ and } \Delta_1 = \cdot \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & \text{From premise } \Gamma \longrightarrow \Delta \end{array}$$

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \hat{\alpha}:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa'} \longrightarrow\!\mathsf{Add}$$

  By induction, there are two possibilities:

  - $\hat{\alpha}$ is not solved:

    $$\begin{array}{lrl} & \Delta = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1) & \text{By i.h.} \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\ ☞ & (\Delta, \hat{\beta} : \kappa') = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1, \hat{\beta} : \kappa') & \text{By congruence of equality} \end{array}$$

  - $\hat{\alpha}$ is solved:

    $$\begin{array}{lrl} & \Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1) & \text{By i.h.} \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\ ☞ & (\Delta, \hat{\beta} : \kappa') = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa') & \text{By congruence of equality} \end{array}$$

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma}_{\Gamma_0, \hat{\alpha}:\kappa, \Gamma_1} \longrightarrow \Delta, \hat{\beta} : \kappa' = t} \longrightarrow\!\mathsf{AddSolved}$$

  By induction, there are two possibilities:

  - $\hat{\alpha}$ is not solved:

    $$\begin{array}{lrl} & \Delta = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1) & \text{By i.h.} \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\ ☞ & (\Delta, \hat{\beta} : \kappa' = t) = (\Delta_0, \hat{\alpha} : \kappa, \Delta_1, \hat{\beta} : \kappa' = t) & \text{By congruence of equality} \end{array}$$

  - $\hat{\alpha}$ is solved:

    $$\begin{array}{lrl} & \Delta = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1) & \text{By i.h.} \\ ☞ & \Gamma_0 \longrightarrow \Delta_0 & '' \\ ☞ & (\Delta, \hat{\beta} : \kappa' = t) = (\Delta_0, \hat{\alpha} : \kappa = \tau', \Delta_1, \hat{\beta} : \kappa' = t) & \text{By congruence of equality} \end{array}$$

- **Case** $$\dfrac{\Gamma \longrightarrow \Delta}{\underbrace{\Gamma, \hat\beta : \kappa'}_{\Gamma_0, \hat\alpha:\kappa, \Gamma_1} \longrightarrow \Delta, \hat\beta : \kappa' = t}\ \longrightarrow\mathsf{Solve}$$

  - **Case** $\hat\alpha \neq \hat\beta$:

    $(\Gamma_0, \hat\alpha : \kappa, \Gamma_1) = (\Gamma, \hat\beta : \kappa')$ — Given

    $= (\Gamma_0, \hat\alpha : \kappa, \Gamma_1', \hat\beta : \kappa')$ — Since the final elements must be equal

    $\Gamma = (\Gamma_0, \hat\alpha : \kappa, \Gamma_1')$ — By injectivity of context syntax

    By induction, there are two possibilities:

    * $\hat\alpha$ is not solved:

      $\Delta = (\Delta_0, \hat\alpha : \kappa, \Delta_1)$ — By i.h.

      ☞ $\Gamma_0 \longrightarrow \Delta_0$ — ″

      ☞ $(\Delta, \hat\beta : \kappa' = t) = (\Delta_0, \hat\alpha : \kappa, \Delta_1, \hat\beta : \kappa' = t)$ — By congruence of equality

    * $\hat\alpha$ is solved:

      $\Delta = (\Delta_0, \hat\alpha : \kappa = \tau', \Delta_1)$ — By i.h.

      ☞ $\Gamma_0 \longrightarrow \Delta_0$ — ″

      ☞ $(\Delta, \hat\beta : \kappa' = t) = (\Delta_0, \hat\alpha : \kappa = \tau', \Delta_1, \hat\beta : \kappa' = t)$ — By congruence of equality

  - **Case** $\hat\alpha = \hat\beta$:

    $\Gamma = \Gamma_0$ and $\kappa = \kappa'$ and $\Gamma_1 = \cdot$ — By injectivity of syntax

    ☞ $(\Delta, \hat\beta : \kappa' = t) = (\Delta_0, \hat\alpha : \kappa = \tau', \Delta_1)$ — where $\Delta_0 = \Delta$ and $\tau' = t$ and $\Delta_1 = \cdot$

    ☞ $\Gamma_0 \longrightarrow \Delta_0$ — From premise $\Gamma \longrightarrow \Delta$ ☐

Lemma 23 (Deep Evar Introduction). *(i) If* Γ_0, Γ_1 *is well-formed and* $\hat\alpha$ *is not declared in* Γ_0, Γ_1 *then* $\Gamma_0, \Gamma_1 \longrightarrow \Gamma_0, \hat\alpha : \kappa, \Gamma_1$.

(ii) If $\Gamma_0, \hat\alpha : \kappa, \Gamma_1$ *is well-formed and* $\Gamma \vdash t : \kappa$ *then* $\Gamma_0, \hat\alpha : \kappa, \Gamma_1 \longrightarrow \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1$.

(iii) If Γ_0, Γ_1 *is well-formed and* $\Gamma \vdash t : \kappa$ *then* $\Gamma_0, \Gamma_1 \longrightarrow \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1$.

Proof.

(i) Assume that Γ_0, Γ_1 is well-formed. We proceed by induction on Γ_1.

- Case $\Gamma_1 = \cdot$:

  Γ_0 *ctx* — Given

  $\hat\alpha \notin \mathsf{dom}(\Gamma_0)$ — Given

  $\Gamma_0, \hat\alpha : \kappa$ *ctx* — By rule VarCtx

  $\Gamma_0 \longrightarrow \Gamma_0$ — By Lemma 32 (Extension Reflexivity)

  ☞ $\Gamma_0 \longrightarrow \Gamma_0, \hat\alpha : \kappa$ — By rule $\longrightarrow$Add

- Case $\Gamma_1 = \Gamma_1', x : A$:

  $\Gamma_0, \Gamma_1', x : A$ *ctx* — Given

  Γ_0, Γ_1' *ctx* — By inversion

  $x \notin \mathsf{dom}(\Gamma_0, \Gamma_1')$ — By inversion (1)

  $\Gamma_0, \Gamma_1' \vdash A$ *type* — By inversion

  $\hat\alpha \notin \mathsf{dom}(\Gamma_0, \Gamma_1', x : A)$ — Given

  $\hat\alpha \neq x$ — By inversion (2)

  $\Gamma_0, \hat\alpha : \kappa, \Gamma_1'$ *ctx* — By i.h.

  $\Gamma_0, \Gamma_1' \longrightarrow \Gamma_0, \hat\alpha : \kappa, \Gamma_1'$ — ″

  $\Gamma_0, \hat\alpha : \kappa, \Gamma_1' \vdash A$ *type* — By Lemma 36 (Extension Weakening (Sorts))

  $x \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa, \Gamma_1')$ — By (1) and (2)

  ☞ $\Gamma_0, \Gamma_1', x : A \longrightarrow \Gamma_0, \hat\alpha : \kappa, \Gamma_1', x : A$ — By $\longrightarrow$Var

- Case $\Gamma_1 = \Gamma_1', \beta : \kappa'$:

$$\begin{array}{rll}
& \Gamma_0, \Gamma_1', \beta : \kappa' \; \mathit{ctx} & \text{Given} \\
& \Gamma_0, \Gamma_1' \; \mathit{ctx} & \text{By inversion} \\
& \beta \notin \mathsf{dom}(\Gamma_0, \Gamma_1') & \text{By inversion (1)} \\
& \hat{\alpha} \notin \mathsf{dom}(\Gamma_0, \Gamma_1', \beta : \kappa') & \text{Given} \\
& \hat{\alpha} \neq \beta & \text{By inversion (2)} \\
& \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & \text{By i.h.} \\
\Gamma_0, \Gamma_1' \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' & '' \\
& \beta \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By (1) and (2)} \\
☞ \quad \Gamma_0, \Gamma_1', \beta : \kappa' \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \beta : \kappa' & \text{By} \longrightarrow\!\mathsf{Uvar}
\end{array}$$

- Case $\Gamma_1 = \Gamma_1', \hat{\beta} : \kappa'$:

$$\begin{array}{rll}
& \Gamma_0, \Gamma_1', \hat{\beta} : \kappa' \; \mathit{ctx} & \text{Given} \\
& \Gamma_0, \Gamma_1' \; \mathit{ctx} & \text{By inversion} \\
& \hat{\beta} \notin \mathsf{dom}(\Gamma_0, \Gamma_1') & \text{By inversion (1)} \\
& \hat{\alpha} \notin \mathsf{dom}(\Gamma_0, \Gamma_1', \hat{\beta} : \kappa') & \text{Given} \\
& \hat{\alpha} \neq \hat{\beta} & \text{By inversion (2)} \\
& \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & \text{By i.h.} \\
\Gamma_0, \Gamma_1' \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' & '' \\
& \hat{\beta} \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By (1) and (2)} \\
☞ \quad \Gamma_0, \Gamma_1', \hat{\beta} : \kappa' \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \hat{\beta} : \kappa' & \text{By} \longrightarrow\!\mathsf{Unsolved}
\end{array}$$

- Case $\Gamma_1 = (\Gamma_1', \hat{\beta} : \kappa' = t)$:

$$\begin{array}{rll}
& \Gamma_0, \Gamma_1', \hat{\beta} : \kappa' = t \; \mathit{ctx} & \text{Given} \\
& \Gamma_0, \Gamma_1' \; \mathit{ctx} & \text{By inversion} \\
& \hat{\beta} \notin \mathsf{dom}(\Gamma_0, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \Gamma_1' \vdash t : \kappa' & & \text{By inversion} \\
& \hat{\alpha} \notin \mathsf{dom}(\Gamma_0, \Gamma_1', \hat{\beta} : \kappa' = t) & \text{Given} \\
& \hat{\alpha} \neq \hat{\beta} & \text{By inversion (2)} \\
& \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & \text{By i.h.} \\
\Gamma_0, \Gamma_1' \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' & '' \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \vdash t : \kappa' & & \text{By Lemma 36 (Extension Weakening (Sorts))} \\
& \hat{\beta} \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By (1) and (2)} \\
☞ \quad \Gamma_0, \Gamma_1', \hat{\beta} : \kappa' = t \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \hat{\beta} : \kappa' = t & \text{By} \longrightarrow\!\mathsf{Solved}
\end{array}$$

- Case $\Gamma_1 = (\Gamma_1', \beta = t)$:

$$\begin{array}{rll}
& \Gamma_0, \Gamma_1', \beta = t \; \mathit{ctx} & \text{Given} \\
& \Gamma_0, \Gamma_1' \; \mathit{ctx} & \text{By inversion} \\
& \beta \notin \mathsf{dom}(\Gamma_0, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \Gamma_1' \vdash t : \mathbb{N} & & \text{By inversion} \\
& \hat{\alpha} \notin \mathsf{dom}(\Gamma_0, \Gamma_1', \beta = t) & \text{Given} \\
& \hat{\alpha} \neq \beta & \text{By inversion (2)} \\
& \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & \text{By i.h.} \\
\Gamma_0, \Gamma_1' \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' & '' \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \vdash t : \mathbb{N} & & \text{By Lemma 36 (Extension Weakening (Sorts))} \\
& \beta \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By (1) and (2)} \\
☞ \quad \Gamma_0, \Gamma_1', \beta = t \longrightarrow & \Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \beta = t & \text{By} \longrightarrow\!\mathsf{Solved}
\end{array}$$

- Case $\Gamma_1 = (\Gamma_1', \blacktriangleright_{\hat{\beta}})$:

$$
\begin{array}{rll}
& \Gamma_0, \Gamma_1', \blacktriangleright_{\hat\beta}\ \mathit{ctx} & \text{Given} \\
& \Gamma_0, \Gamma_1'\ \mathit{ctx} & \text{By inversion} \\
& \hat\beta \notin \mathsf{dom}(\Gamma_0, \Gamma_1') & \text{By inversion (1)} \\
& \hat\alpha \notin \mathsf{dom}(\Gamma_0, \Gamma_1', \blacktriangleright_{\hat\beta}) & \text{Given} \\
& \hat\alpha \neq \hat\beta & \text{By inversion (2)} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1'\ \mathit{ctx} & \text{By i.h.} \\
\Gamma_0, \Gamma_1' \longrightarrow & \Gamma_0, \hat\alpha : \kappa, \Gamma_1' & '' \\
& \hat\beta \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa, \Gamma_1') & \text{By (1) and (2)} \\
☞\quad \Gamma_0, \Gamma_1', \blacktriangleright_{\hat\beta} \longrightarrow & \Gamma_0, \hat\alpha : \kappa, \Gamma_1', \blacktriangleright_{\hat\beta} & \text{By} \longrightarrow\!\mathsf{Marker}
\end{array}
$$

(ii) Assume $\Gamma_0, \hat\alpha : \kappa, \Gamma_1\ \mathit{ctx}$. We proceed by induction on Γ_1:

- Case $\Gamma_1 = \cdot$:

$$
\begin{array}{rll}
\Gamma_0 \vdash t : \kappa & & \text{Given} \\
& \Gamma_0, \Gamma_1\ \mathit{ctx} & \text{Given} \\
& \Gamma_0\ \mathit{ctx} & \text{Since } \Gamma_1 = \cdot \\
\Gamma_0 \longrightarrow & \Gamma_0 & \text{By Lemma 32 (Extension Reflexivity)} \\
\Gamma_0, \hat\alpha : \kappa \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t & \text{By rule} \longrightarrow\!\mathsf{Solve} \\
☞\quad \Gamma_0, \hat\alpha : \kappa, \Gamma_1 \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1 & \text{Since } \Gamma_1 = \cdot
\end{array}
$$

- Case $\Gamma_1 = (\Gamma_1', x : A)$:

$$
\begin{array}{rll}
\Gamma_0 \vdash t : \kappa & & \text{Given} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1', x : A\ \mathit{ctx} & \text{Given} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1'\ \mathit{ctx} & \text{By inversion} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1' \vdash & A\ \mathit{type} & \text{By inversion} \\
& x \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1' \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1 & \text{By i.h.} \\
\Gamma_0, \hat\alpha : \kappa = t, \Gamma_1 \vdash & A\ \mathit{type} & \text{By Lemma 36 (Extension Weakening (Sorts))} \\
& x \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa = t, \Gamma_1') & \text{since this is the same domain as (1)} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1', x : A \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1, x : A & \text{By rule} \longrightarrow\!\mathsf{Var}
\end{array}
$$

- Case $\Gamma_1 = (\Gamma_1', \beta : \kappa')$:

$$
\begin{array}{rll}
\Gamma_0 \vdash t : \kappa & & \text{Given} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1', \beta : \kappa'\ \mathit{ctx} & \text{Given} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1'\ \mathit{ctx} & \text{By inversion} \\
& \beta \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1' \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1 & \text{By i.h.} \\
& \beta \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa = t, \Gamma_1') & \text{since this is the same domain as (1)} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1', \beta : \kappa' \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1, \beta : \kappa' & \text{By rule} \longrightarrow\!\mathsf{Uvar}
\end{array}
$$

- Case $\Gamma_1 = (\Gamma_1', \hat\beta : \kappa')$:

$$
\begin{array}{rll}
\Gamma_0 \vdash t : \kappa & & \text{Given} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1', \hat\beta : \kappa'\ \mathit{ctx} & \text{Given} \\
& \Gamma_0, \hat\alpha : \kappa, \Gamma_1'\ \mathit{ctx} & \text{By inversion} \\
& \hat\beta \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1' \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1 & \text{By i.h.} \\
& \hat\beta \notin \mathsf{dom}(\Gamma_0, \hat\alpha : \kappa = t, \Gamma_1') & \text{since this is the same domain as (1)} \\
\Gamma_0, \hat\alpha : \kappa, \Gamma_1', \hat\beta : \kappa' \longrightarrow & \Gamma_0, \hat\alpha : \kappa = t, \Gamma_1, \hat\beta : \kappa' & \text{By rule} \longrightarrow\!\mathsf{Unsolved}
\end{array}
$$

- Case $\Gamma_1 = (\Gamma_1', \hat{\beta} : \kappa' = t')$:

$$\begin{array}{rll}
\Gamma_0 \vdash t' : \kappa & & \text{Given} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \hat{\beta} : \kappa' = t' \; \mathit{ctx} & & \text{Given} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & & \text{By inversion} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \vdash t' : \kappa' & & \text{By inversion} \\
\hat{\beta} \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & & \text{By inversion (1)} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1 & & \text{By i.h.} \\
\hat{\beta} \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1') & & \text{since this is the same domain as (1)}
\end{array}$$

$$\Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1 \vdash t' : \kappa' \qquad \text{By Lemma 36 (Extension Weakening (Sorts))}$$

$$\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \hat{\beta} : \kappa' = t' \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t', \Gamma_1, \hat{\beta} : \kappa' = t' \qquad \text{By rule} \longrightarrow\!\mathsf{Solved}$$

- Case $\Gamma_1 = (\Gamma_1', \beta = t')$:

$$\begin{array}{rl}
\Gamma_0 \vdash t' : \kappa & \text{Given} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \beta = t' \; \mathit{ctx} & \text{Given} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & \text{By inversion} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \vdash t' : \mathbb{N} & \text{By inversion} \\
\beta \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1 & \text{By i.h.} \\
\beta \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1') & \text{since this is the same domain as (1)} \\
\Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1 \vdash t' : \mathbb{N} & \text{By Lemma 36 (Extension Weakening (Sorts))} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \beta = t' \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t', \Gamma_1, \beta = t' & \text{By rule} \longrightarrow\!\mathsf{Eqn}
\end{array}$$

- Case $\Gamma_1 = (\Gamma_1', \blacktriangleright_{\hat{\beta}})$:

$$\begin{array}{rl}
\Gamma_0 \vdash t : \kappa & \text{Given} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \blacktriangleright_{\hat{\beta}} \; \mathit{ctx} & \text{Given} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \; \mathit{ctx} & \text{By inversion} \\
\hat{\beta} \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1') & \text{By inversion (1)} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1' \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1 & \text{By i.h.} \\
\hat{\beta} \notin \mathsf{dom}(\Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1') & \text{since this is the same domain as (1)} \\
\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1', \blacktriangleright_{\hat{\beta}} \longrightarrow \Gamma_0, \hat{\alpha} : \kappa = t, \Gamma_1, \blacktriangleright_{\hat{\beta}} & \text{By rule} \longrightarrow\!\mathsf{Unsolved}
\end{array}$$

(iii) Apply parts (i) and (ii) as lemmas, then Lemma 33 (Extension Transitivity). □

Lemma 26 (Parallel Admissibility).
If $\Gamma_L \longrightarrow \Delta_L$ *and* $\Gamma_L, \Gamma_R \longrightarrow \Delta_L, \Delta_R$ *then:*

(i) $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa, \Delta_R$

(ii) If $\Delta_L \vdash \tau' : \kappa$ *then* $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$.

(iii) If $\Gamma_L \vdash \tau : \kappa$ *and* $\Delta_L \vdash \tau'$ *type and* $[\Delta_L]\tau = [\Delta_L]\tau'$, *then* $\Gamma_L, \hat{\alpha} : \kappa = \tau, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$.

Proof. By induction on Δ_R. As always, we assume that all contexts mentioned in the statement of the lemma are well-formed. Hence, $\hat{\alpha} \notin \mathsf{dom}(\Gamma_L) \cup \mathsf{dom}(\Gamma_R) \cup \mathsf{dom}(\Delta_L) \cup \mathsf{dom}(\Delta_R)$.

(i) We proceed by cases of Δ_R. Observe that in all the extension rules, the right-hand context gets smaller, so as we enter subderivations of $\Gamma_L, \Gamma_R \longrightarrow \Delta_L, \Delta_R$, the context Δ_R becomes smaller.

The only tricky part of the proof is that to apply the i.h., we need $\Gamma_L \longrightarrow \Delta_L$. So we need to make sure that as we drop items from the right of Γ_R and Δ_R, we don't go too far and start decomposing Γ_L or Δ_L! It's easy to avoid decomposing Δ_L: when $\Delta_R = \cdot$, we don't need to apply the i.h. anyway. To avoid decomposing Γ_L, we need to reason by contradiction, using Lemma 19 (Declaration Preservation).

- **Case** $\Delta_R = \cdot$:
  We have $\Gamma_L \longrightarrow \Delta_L$. Applying $\longrightarrow$Unsolved to that derivation gives the result.
- **Case** $\Delta_R = (\Delta'_R, \hat{\beta})$: We have $\hat{\beta} \neq \hat{\alpha}$ by the well-formedness assumption.
  The concluding rule of $\Gamma_L, \Gamma_R \longrightarrow \Delta_L, \Delta'_R, \hat{\beta}$ must have been $\longrightarrow$Unsolved or $\longrightarrow$Add. In both cases, the result follows by i.h. and applying $\longrightarrow$Unsolved or $\longrightarrow$Add.
  Note: In $\longrightarrow$Add, the left-hand context doesn't change, so we clearly maintain $\Gamma_L \longrightarrow \Delta_L$. In $\longrightarrow$Unsolved, we can correctly apply the i.h. because $\Gamma_R \neq \cdot$. Suppose, for a contradiction, that $\Gamma_R = \cdot$. Then $\Gamma_L = (\Gamma'_L, \hat{\beta})$. It was given that $\Gamma_L \longrightarrow \Delta_L$, that is, $\Gamma'_L, \hat{\beta} \longrightarrow \Delta_L$. By Lemma 19 (Declaration Preservation), Δ_L has a declaration of $\hat{\beta}$. But then $\Delta = (\Delta_L, \Delta'_R, \hat{\beta})$ is not well-formed: contradiction. Therefore $\Gamma_R \neq \cdot$.
- **Case** $\Delta_R = (\Delta'_R, \hat{\beta} : \kappa = t)$: We have $\hat{\beta} \neq \hat{\alpha}$ by the well-formedness assumption.
  The concluding rule must have been $\longrightarrow$Solved, $\longrightarrow$Solve or $\longrightarrow$AddSolved. In each case, apply the i.h. and then the corresponding rule. (In $\longrightarrow$Solved and $\longrightarrow$Solve, use Lemma 19 (Declaration Preservation) to show $\Gamma_R \neq \cdot$.)
- **Case** $\Delta_R = (\Delta'_R, \alpha)$: The concluding rule must have been $\longrightarrow$Uvar. The result follows by i.h. and applying $\longrightarrow$Uvar.
- **Case** $\Delta_R = (\Delta'_R, \alpha = \tau)$: The concluding rule must have been $\longrightarrow$Eqn. The result follows by i.h. and applying $\longrightarrow$Eqn.
- **Case** $\Delta_R = (\Delta'_R, \blacktriangleright_{\hat{\beta}})$: Similar to the previous case, with rule $\longrightarrow$Marker.
- **Case** $\Delta_R = (\Delta'_R, x : A)$: Similar to the previous case, with rule $\longrightarrow$Var.

(ii) Similar to part (i), except that when $\Delta_R = \cdot$, apply rule $\longrightarrow$Solve.

(iii) Similar to part (i), except that when $\Delta_R = \cdot$, apply rule $\longrightarrow$Solved, using the given equality to satisfy the second premise. □

Lemma 27 (Parallel Extension Solution)**.**
If $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$ *and* $\Gamma_L \vdash \tau : \kappa$ *and* $[\Delta_L]\tau = [\Delta_L]\tau'$
then $\Gamma_L, \hat{\alpha} : \kappa = \tau, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau', \Delta_R$.

Proof. By induction on Δ_R.
In the case where $\Delta_R = \cdot$, we know that rule $\longrightarrow$Solve must have concluded the derivation (we can use Lemma 19 (Declaration Preservation) to get a contradiction that rules out $\longrightarrow$AddSolved); then we have a subderivation $\Gamma_L \longrightarrow \Delta_L$, to which we can apply $\longrightarrow$Solved. □

Lemma 28 (Parallel Variable Update)**.**
If $\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau_0, \Delta_R$ *and* $\Gamma_L \vdash \tau_1 : \kappa$ *and* $\Delta_L \vdash \tau_2 : \kappa$ *and* $[\Delta_L]\tau_0 = [\Delta_L]\tau_1 = [\Delta_L]\tau_2$
then $\Gamma_L, \hat{\alpha} : \kappa = \tau_1, \Gamma_R \longrightarrow \Delta_L, \hat{\alpha} : \kappa = \tau_2, \Delta_R$.

Proof. By induction on Δ_R. Similar to the proof of Lemma 27 (Parallel Extension Solution), but applying $\longrightarrow$Solved at the end. □

Lemma 29 (Substitution Monotonicity)**.**

(i) If $\Gamma \longrightarrow \Delta$ *and* $\Gamma \vdash t : \kappa$ *then* $[\Delta][\Gamma]t = [\Delta]t$.

(ii) If $\Gamma \longrightarrow \Delta$ *and* $\Gamma \vdash P$ *prop then* $[\Delta][\Gamma]P = [\Delta]P$.

(iii) If $\Gamma \longrightarrow \Delta$ *and* $\Gamma \vdash A$ *type then* $[\Delta][\Gamma]A = [\Delta]A$.

Proof. We prove each part in turn; part (i) does not depend on parts (ii) or (iii), so we can use part (i) as a lemma in the proofs of parts (ii) and (iii).

- **Proof of Part (i):** By lexicographic induction on the derivation of $\mathcal{D} :: \Gamma \longrightarrow \Delta$ and $\Gamma \vdash t : \kappa$. We proceed by cases on the derivation of $\Gamma \vdash t : \kappa$.

– **Case** $\dfrac{\hat{\alpha} : \kappa \in \Gamma}{\Gamma \vdash \hat{\alpha} : \kappa}$ VarSort

$[\Gamma]\hat{\alpha} = \hat{\alpha}$ Since $\hat{\alpha}$ is not solved in Γ
$[\Delta]\hat{\alpha} = [\Delta]\hat{\alpha}$ Reflexivity
$= [\Delta][\Gamma]\hat{\alpha}$ By above equality

– **Case** $\dfrac{(\alpha : \kappa) \in \Gamma}{\Gamma \vdash \alpha : \kappa}$ VarSort

Consider whether or not there is a binding of the form $(\alpha = \tau) \in \Gamma$.

∗ **Case** $(\alpha = \tau) \in \Gamma$:

	$\Delta = (\Delta_0, \alpha = \tau', \Delta_1)$	By Lemma 22 (Extension Inversion) (i)
$\mathcal{D}'$ ::	$\Gamma_0 \longrightarrow \Delta_0$	"
	$\mathcal{D}' < \mathcal{D}$	"
(1)	$[\Delta_0]\tau' = [\Delta_0]\tau$	"
(2)	$[\Delta_0][\Gamma_0]\tau = [\Delta_0]\tau$	By i.h.
	$[\Delta][\Gamma]\alpha = [\Delta_0, \alpha = \tau', \Delta_1][\Gamma_0, \alpha = \tau, \Gamma_1]\alpha$	By definition
	$= [\Delta_0, \alpha = \tau', \Delta_1][\Gamma_0, \alpha = \tau]\alpha$	Since $\alpha \notin \mathsf{dom}(\Gamma_1)$
	$= [\Delta_0, \alpha = \tau', \Delta_1][\Gamma_0]\tau$	By definition of substitution
	$= [\Delta_0][\Gamma_0]\tau$	Since $FV([\Gamma_0]\tau) \cap \mathsf{dom}(\Delta_1) = \emptyset$
	$= [\Delta_0]\tau'$	By (2) and (1)
	$= [\Delta_0, \alpha = \tau']\alpha$	By definition of substitution
	$= [\Delta_0, \alpha = \tau', \Delta_1]\alpha$	Since $FV([\Delta_0]\tau) \cap \mathsf{dom}(\Delta_1) = \emptyset$
	$= [\Delta]\alpha$	By definition of Δ

∗ **Case** $(\alpha = \tau) \notin \Gamma$:

$[\Gamma]\alpha = \alpha$ By definition of substitution
$[\Delta][\Gamma]\alpha = [\Delta]\alpha$ Apply $[\Delta]$ to both sides

– **Case** $\dfrac{}{\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1 \vdash \hat{\alpha} : \kappa}$ SolvedVarSort

Similar to the VarSort case.

– **Case** $\dfrac{}{\Gamma \vdash 1 : \star}$ UnitSort

$[\Delta]1 = 1 = [\Delta][\Gamma]1$ Since $FV(1) = \emptyset$

– **Case** $\dfrac{\Gamma \vdash \tau_1 : \star \qquad \Gamma \vdash \tau_2 : \star}{\Gamma \vdash \tau_1 \oplus \tau_2 : \star}$ BinSort

$[\Delta][\Gamma]\tau_1 = [\Delta]\tau_1$ By i.h.
$[\Delta][\Gamma]\tau_2 = [\Delta]\tau_2$ By i.h.
$[\Delta][\Gamma]\tau_1 \oplus [\Delta][\Gamma]\tau_2 = [\Delta]\tau_1 \oplus [\Delta]\tau_2$ By congruence of equality
$[\Delta][\Gamma](\tau_1 \oplus \tau_2) = [\Delta](\tau_1 \oplus \tau_2)$ Definition of substitution

– **Case** $\dfrac{}{\Gamma \vdash \mathsf{zero} : \mathbb{N}}$ ZeroSort

$[\Delta]\mathsf{zero} = \mathsf{zero} = [\Delta][\Gamma]\mathsf{zero}$ Since $FV(\mathsf{zero}) = \emptyset$

– **Case** $\dfrac{\Gamma \vdash t : \mathbb{N}}{\Gamma \vdash \mathsf{succ}(t) : \mathbb{N}}$ SuccSort

$$\begin{array}{rcll} [\Delta][\Gamma]t &=& [\Delta]t & \text{By i.h.} \\ \mathsf{succ}([\Delta][\Gamma]t) &=& \mathsf{succ}([\Delta]t) & \text{By congruence of equality} \\ [\Delta][\Gamma]\mathsf{succ}(t) &=& [\Delta]\mathsf{succ}(t) & \text{By definition of substitution} \end{array}$$

- **Proof of Part (ii):** We have a derivation of $\Gamma \vdash P$ *prop*, and will use the previous part as a lemma.

– **Case** $\dfrac{\Gamma \vdash t : \mathbb{N} \qquad \Gamma \vdash t' : \mathbb{N}}{\Gamma \vdash t = t' \; \mathit{prop}}$ EqProp

$$\begin{array}{rcll} [\Delta][\Gamma]t &=& [\Delta]t & \text{By part (i)} \\ [\Delta][\Gamma]t' &=& [\Delta]t' & \text{By part (i)} \\ ([\Delta][\Gamma]t = [\Delta][\Gamma]t') &=& ([\Delta]t = [\Delta]t') & \text{By congruence of equality} \\ [\Delta][\Gamma](t = t') &=& [\Delta](t = t') & \text{Definition of substitution} \end{array}$$

- **Proof of Part (iii):** By induction on the derivation of $\Gamma \vdash A$ *type*, using the previous parts as lemmas.

– **Case** $\dfrac{(u : \star) \in \Gamma}{\Gamma \vdash u \; \mathit{type}}$ VarWF

$$\begin{array}{rl} \Gamma \vdash u : \star & \text{By rule VarSort} \\ [\Delta][\Gamma]u = [\Delta]u & \text{By part (i)} \end{array}$$

– **Case** $\dfrac{(\hat{\alpha} : \star = \tau) \in \Gamma}{\Gamma \vdash \hat{\alpha} \; \mathit{type}}$ SolvedVarWF

$$\begin{array}{rl} \Gamma \vdash \hat{\alpha} : \star & \text{By rule SolvedVarSort} \\ [\Delta][\Gamma]\hat{\alpha} = [\Delta]\hat{\alpha} & \text{By part (i)} \end{array}$$

– **Case** $\dfrac{}{\Gamma \vdash 1 \; \mathit{type}}$ UnitWF

$$\begin{array}{rl} \Gamma \vdash 1 : \star & \text{By rule UnitSort} \\ [\Delta][\Gamma]1 = [\Delta]1 & \text{By part (i)} \end{array}$$

– **Case** $\dfrac{\Gamma \vdash A_1 \; \mathit{type} \qquad \Gamma \vdash A_2 \; \mathit{type}}{\Gamma \vdash A_1 \oplus A_2 \; \mathit{type}}$ BinWF

$$\begin{array}{rcll} [\Delta][\Gamma]A_1 &=& [\Delta]A_1 & \text{By i.h.} \\ [\Delta][\Gamma]A_2 &=& [\Delta]A_2 & \text{By i.h.} \\ [\Delta][\Gamma]A_1 \oplus [\Delta][\Gamma]A_2 &=& [\Delta]A_1 \oplus [\Delta]A_2 & \text{By congruence of equality} \\ [\Delta][\Gamma](A_1 \oplus A_2) &=& [\Delta](A_1 \oplus A_2) & \text{Definition of substitution} \end{array}$$

– **Case** VecWF: Similar to the BinWF case.

– **Case** $\dfrac{\Gamma, \alpha : \kappa \vdash A_0 \; \mathit{type}}{\Gamma \vdash \forall \alpha : \kappa.\, A_0 \; \mathit{type}}$ ForallWF

$$
\begin{array}{rl}
\Gamma \longrightarrow \Delta & \text{Given} \\
\Gamma, \alpha : \kappa \longrightarrow \Delta, \alpha : \kappa & \text{By rule} \longrightarrow\!\mathsf{Uvar} \\
[\Delta, \alpha : \kappa][\Gamma, \alpha : \kappa]A_0 = [\Delta, \alpha : \kappa]A_0 & \text{By i.h.} \\
[\Delta][\Gamma]A_0 = [\Delta]A_0 & \text{By definition of substitution} \\
\forall \alpha : \kappa.\ [\Delta][\Gamma]A_0 = \forall \alpha : \kappa.\ [\Delta]A_0 & \text{By congruence of equality} \\
[\Delta][\Gamma](\forall \alpha : \kappa.\ A_0) = [\Delta](\forall \alpha : \kappa.\ A_0) & \text{By definition of substitution}
\end{array}
$$

– **Case** ExistsWF: Similar to the ForallWF case.

– **Case** $\dfrac{\Gamma \vdash P\ \textit{prop} \qquad \Gamma \vdash A_0\ \textit{type}}{\Gamma \vdash P \supset A_0\ \textit{type}}$ ImpliesWF

$$
\begin{array}{rl}
[\Delta][\Gamma]P = [\Delta]P & \text{By part (ii)} \\
[\Delta][\Gamma]A_0 = [\Delta]A_0 & \text{By i.h.} \\
[\Delta][\Gamma]P \supset [\Delta][\Gamma]A_0 = [\Delta]P \supset [\Delta]A_0 & \text{By congruence of equality} \\
[\Delta][\Gamma](P \supset A_0) = [\Delta](P \supset A_0) & \text{Definition of substitution}
\end{array}
$$

– **Case** $\dfrac{\Gamma \vdash P\ \textit{prop} \qquad \Gamma \vdash A_0\ \textit{type}}{\Gamma \vdash A_0 \land P\ \textit{type}}$ WithWF

Similar to the ImpliesWF case. □

Lemma 30 (Substitution Invariance).

(i) If $\Gamma \longrightarrow \Delta$ *and* $\Gamma \vdash t : \kappa$ *and* $\mathsf{FEV}([\Gamma]t) = \emptyset$ *then* $[\Delta][\Gamma]t = [\Gamma]t$.

(ii) If $\Gamma \longrightarrow \Delta$ *and* $\Gamma \vdash P$ *prop and* $\mathsf{FEV}([\Gamma]P) = \emptyset$ *then* $[\Delta][\Gamma]P = [\Gamma]P$.

(iii) If $\Gamma \longrightarrow \Delta$ *and* $\Gamma \vdash A$ *type and* $\mathsf{FEV}([\Gamma]A) = \emptyset$ *then* $[\Delta][\Gamma]A = [\Gamma]A$.

Proof. Each part is a separate induction, relying on the proofs of the earlier parts. In each part, the result follows by an induction on the derivation of $\Gamma \longrightarrow \Delta$.

The main observation is that Δ adds no equations for any variable of t, P, and A that Γ does not already contain, and as a result applying Δ as a substitution to $[\Gamma]t$ does nothing. □

Lemma 24 (Soft Extension).
If $\Gamma \longrightarrow \Delta$ *and* Γ, Θ *ctx and* Θ *is soft, then there exists* Ω *such that* $\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega)$ *and* $\Gamma, \Theta \longrightarrow \Delta, \Omega$.

Proof. By induction on Θ.

- **Case** $\Theta = \cdot$: We have $\Gamma \longrightarrow \Delta$. Let $\Omega = \cdot$. Then $\Gamma, \Theta \longrightarrow \Delta, \Omega$.
- **Case** $\Theta = (\Theta', \hat{\alpha} : \kappa = t)$:

$$
\begin{array}{rrl}
 & \Gamma, \Theta' \longrightarrow \Gamma, \Omega' & \text{By i.h.} \\
☞ & \Gamma, \underbrace{\Theta', \hat{\alpha} : \kappa = t}_{\Theta} \longrightarrow \Delta, \underbrace{\Omega', \hat{\alpha} : \kappa = t}_{\Omega} & \text{By rule} \longrightarrow\!\mathsf{Solved}
\end{array}
$$

- **Case** $\Theta = (\Theta', \hat{\alpha} : \kappa)$:

  If $\kappa = \star$, let $t = 1$; if $\kappa = \mathbb{N}$, let $t = \mathsf{zero}$.

$$
\begin{array}{rrl}
 & \Gamma, \Theta' \longrightarrow \Gamma, \Omega' & \text{By i.h.} \\
☞ & \Gamma, \underbrace{\Theta', \hat{\alpha} : \kappa}_{\Theta} \longrightarrow \Delta, \underbrace{\Omega', \hat{\alpha} : \kappa = t}_{\Omega} & \text{By rule} \longrightarrow\!\mathsf{Solve}
\end{array}
$$

□

Lemma 31 (Split Extension).
If $\Delta \longrightarrow \Omega$
and $\hat{\alpha} \in \mathsf{unsolved}(\Delta)$
and $\Omega = \Omega_1[\hat{\alpha} : \kappa = t_1]$
and Ω *is canonical (Definition 3)*
and $\Omega \vdash t_2 : \kappa$
then $\Delta \longrightarrow \Omega_1[\hat{\alpha} : \kappa = t_2]$.

Proof. By induction on the derivation of $\Delta \longrightarrow \Omega$. Use the fact that $\Omega_1[\hat{\alpha} : \kappa = t_1]$ and $\Omega_1[\hat{\alpha} : \kappa = t_2]$ agree on all solutions *except* the solution for $\hat{\alpha}$. In the $\longrightarrow$Solve case where the existential variable is $\hat{\alpha}$, use $\Omega \vdash t_2 : \kappa$. ☐

C′.1 Reflexivity and Transitivity

Lemma 32 (Extension Reflexivity).
If Γ *ctx then* $\Gamma \longrightarrow \Gamma$.

Proof. By induction on the derivation of Γ *ctx*.

- **Case** $$\frac{}{\cdot\ ctx}\ \mathsf{EmptyCtx}$$

  $\cdot \longrightarrow \cdot$ By rule $\longrightarrow$Id

- **Case** $$\frac{\Gamma\ ctx \qquad x \notin \mathsf{dom}(\Gamma) \qquad \Gamma \vdash A\ type}{\Gamma, x : A\ ctx}\ \mathsf{HypCtx}$$

$\Gamma \longrightarrow \Gamma$	By i.h.
$[\Gamma]A = [\Gamma]A$	By reflexivity
$\Gamma, x : A \longrightarrow \Gamma, x : A$	By rule $\longrightarrow$Var

- **Case** $$\frac{\Gamma\ ctx \qquad u : \kappa \notin \mathsf{dom}(\Gamma)}{\Gamma, u : \kappa\ ctx}\ \mathsf{VarCtx}$$

$\Gamma \longrightarrow \Gamma$	By i.h.
$\Gamma, u : \kappa \longrightarrow \Gamma, u : \kappa$	By rule $\longrightarrow$Uvar or $\longrightarrow$Unsolved

- **Case** $$\frac{\Gamma\ ctx \qquad \hat{\alpha} \notin \mathsf{dom}(\Gamma) \qquad \Gamma \vdash t : \kappa}{\Gamma, \hat{\alpha} : \kappa = t\ ctx}\ \mathsf{SolvedCtx}$$

$\Gamma \longrightarrow \Gamma$	By i.h.
$[\Gamma]t = [\Gamma]t$	By reflexivity
$\Gamma, \hat{\alpha} : \kappa = t \longrightarrow \Gamma, \hat{\alpha} : \kappa = t$	By rule $\longrightarrow$Solved

- **Case** $$\frac{\Gamma\ ctx \qquad \alpha : \kappa \in \Gamma \qquad (\alpha = -) \notin \Gamma \qquad \Gamma \vdash \tau : \kappa}{\Gamma, \alpha = \tau\ ctx}\ \mathsf{EqnVarCtx}$$

$\Gamma \longrightarrow \Gamma$	By i.h.
$[\Gamma]t = [\Gamma]t$	By reflexivity
$\Gamma, \alpha = t \longrightarrow \Gamma, \alpha = t$	By rule $\longrightarrow$Eqn

- **Case** $$\dfrac{\Gamma\ ctx \qquad \blacktriangleright_u \notin \Gamma}{\Gamma, \blacktriangleright_u\ ctx}\ \mathsf{MarkerCtx}$$

$$\begin{array}{rcll} \Gamma & \longrightarrow & \Gamma & \text{By i.h.} \\ \Gamma, \blacktriangleright_u & \longrightarrow & \Gamma, \blacktriangleright_u & \text{By rule } {\longrightarrow}\mathsf{Marker} \end{array}$$

□

Lemma 33 (Extension Transitivity)**.**
If $\mathcal{D} :: \Gamma \longrightarrow \Theta$ *and* $\mathcal{D}' :: \Theta \longrightarrow \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Proof. By induction on $\mathcal{D}'$.

- **Case** $$\dfrac{}{\underbrace{\cdot}_{\Theta} \longrightarrow \underbrace{\cdot}_{\Delta}}\ {\longrightarrow}\mathsf{Id}$$

$$\begin{array}{rl} \Gamma = \cdot & \text{By inversion on } \mathcal{D} \\ \cdot \longrightarrow \cdot & \text{By rule } {\longrightarrow}\mathsf{Id} \\ \Gamma \longrightarrow \Delta & \text{Since } \Gamma = \Delta = \cdot \end{array}$$

- **Case** $$\dfrac{\Theta' \longrightarrow \Delta' \qquad [\Delta']A = [\Delta']A'}{\underbrace{\Theta', x : A}_{\Theta} \longrightarrow \underbrace{\Delta', x : A'}_{\Delta}}\ {\longrightarrow}\mathsf{Var}$$

$$\begin{array}{rll} \Gamma & = (\Gamma', x : A'') & \text{By inversion on } \mathcal{D} \\ [\Theta]A'' & = [\Theta]A & \text{By inversion on } \mathcal{D} \\ \Gamma' & \longrightarrow \Theta' & \text{By inversion on } \mathcal{D} \\ \Gamma' & \longrightarrow \Delta' & \text{By i.h.} \\ [\Delta'][\Theta']A'' & = [\Delta'][\Theta']A & \text{By congruence of equality} \\ [\Delta']A'' & = [\Delta']A & \text{By Lemma 29 (Substitution Monotonicity)} \\ & = [\Delta']A' & \text{By premise } [\Delta']A = [\Delta']A' \\ \Gamma', x : A'' & \longrightarrow \Delta', x : A' & \text{By } {\longrightarrow}\mathsf{Var} \end{array}$$

- **Case** $$\dfrac{\Theta' \longrightarrow \Delta'}{\underbrace{\Theta', \alpha : \kappa}_{\Theta} \longrightarrow \underbrace{\Delta', \alpha : \kappa}_{\Delta}}\ {\longrightarrow}\mathsf{Uvar}$$

$$\begin{array}{rll} \Gamma & = (\Gamma', \alpha : \kappa) & \text{By inversion on } \mathcal{D} \\ \Gamma' & \longrightarrow \Theta' & \text{By inversion on } \mathcal{D} \\ \Gamma' & \longrightarrow \Delta' & \text{By i.h.} \\ \Gamma', \alpha : \kappa & \longrightarrow \Delta', \alpha : \kappa & \text{By } {\longrightarrow}\mathsf{Uvar} \end{array}$$

- **Case** $$\dfrac{\Theta' \longrightarrow \Delta'}{\underbrace{\Theta', \hat\alpha : \kappa}_{\Theta} \longrightarrow \underbrace{\Delta', \hat\alpha : \kappa}_{\Delta}}\ {\longrightarrow}\mathsf{Unsolved}$$

  Two rules could have concluded $\mathcal{D} :: \Gamma \longrightarrow (\Theta', \hat\alpha : \kappa)$:

  - **Case** $$\dfrac{\Gamma' \longrightarrow \Theta'}{\underbrace{\Gamma', \hat\alpha : \kappa}_{\Gamma} \longrightarrow \Theta', \hat\alpha : \kappa}\ {\longrightarrow}\mathsf{Unsolved}$$

$$\begin{array}{rl} \Gamma' \longrightarrow \Delta' & \text{By i.h.} \\ \Gamma', \hat{\alpha} : \kappa \longrightarrow \Delta', \hat{\alpha} : \kappa & \text{By rule } \longrightarrow\mathsf{Add} \end{array}$$

- **Case** $\dfrac{\Gamma \longrightarrow \Theta'}{\Gamma \longrightarrow \Theta', \hat{\alpha} : \kappa} \longrightarrow\mathsf{Add}$

$$\begin{array}{ll} \Gamma \longrightarrow \Delta' & \text{By i.h.} \\ \Gamma \longrightarrow \Delta', \hat{\alpha} : \kappa & \text{By rule } \longrightarrow\mathsf{Add} \end{array}$$

- **Case** $\dfrac{\Theta' \longrightarrow \Delta' \qquad [\Delta']t = [\Delta']t'}{\underbrace{\Theta', \hat{\alpha} : \kappa = t}_{\Theta} \longrightarrow \underbrace{\Delta', \hat{\alpha} : \kappa = t'}_{\Delta}} \longrightarrow\mathsf{Solved}$

Two rules could have concluded $\mathcal{D} :: \Gamma \longrightarrow (\Theta', \hat{\alpha} : \kappa = t)$:

- **Case** $\dfrac{\Gamma' \longrightarrow \Theta' \qquad [\Theta']t'' = [\Theta']t}{\underbrace{\Gamma', \hat{\alpha} : \kappa = t''}_{\Gamma} \longrightarrow \Theta', \hat{\alpha} : \kappa = t} \longrightarrow\mathsf{Solved}$

$$\begin{array}{rll} \Gamma' \longrightarrow \Delta' & & \text{By i.h.} \\ [\Theta']t'' &= [\Theta']t & \text{Premise} \\ [\Delta'][\Theta']t'' &= [\Delta'][\Theta']t & \text{Applying } \Delta' \text{ to both sides} \\ [\Delta']t'' &= [\Delta']t & \text{By Lemma 29 (Substitution Monotonicity)} \\ &= [\Delta']t' & \text{By premise } [\Delta']t = [\Delta']t' \\ \Gamma', \hat{\alpha} : \kappa = t'' \longrightarrow \Delta', \hat{\alpha} : \kappa = t' & & \text{By rule } \longrightarrow\mathsf{Solved} \end{array}$$

- **Case** $\dfrac{\Gamma \longrightarrow \Theta'}{\Gamma \longrightarrow \Theta', \hat{\alpha} : \kappa = t} \longrightarrow\mathsf{AddSolved}$

$$\begin{array}{ll} \Gamma \longrightarrow \Delta' & \text{By i.h.} \\ \Gamma \longrightarrow \Delta', \hat{\alpha} : \kappa = t' & \text{By rule } \longrightarrow\mathsf{AddSolved} \end{array}$$

- **Case** $\dfrac{\Theta' \longrightarrow \Delta' \qquad [\Delta']t = [\Delta']t'}{\underbrace{\Theta', \alpha = t}_{\Theta} \longrightarrow \underbrace{\Delta', \alpha = t'}_{\Delta}} \longrightarrow\mathsf{Eqn}$

$$\begin{array}{rll} \Gamma &= (\Gamma', \alpha = t'') & \text{By inversion on } \mathcal{D} \\ \Gamma' \longrightarrow \Theta' & & \text{By inversion on } \mathcal{D} \\ [\Theta']t'' &= [\Theta']t & \text{By inversion on } \mathcal{D} \\ [\Delta'][\Theta']t'' &= [\Delta'][\Theta']t & \text{Applying } \Delta' \text{ to both sides} \\ \Gamma' \longrightarrow \Delta' & & \text{By i.h.} \\ [\Delta']t'' &= [\Delta']t & \text{By Lemma 29 (Substitution Monotonicity)} \\ &= [\Delta']t' & \text{By premise } [\Delta']t = [\Delta']t' \\ \Gamma', \alpha = t'' \longrightarrow \Delta', \alpha = t' & & \text{By rule } \longrightarrow\mathsf{Eqn} \end{array}$$

- **Case** $\dfrac{\Theta \longrightarrow \Delta'}{\Theta \longrightarrow \underbrace{\Delta', \hat{\alpha} : \kappa}_{\Delta}} \longrightarrow\mathsf{Add}$

$$\begin{array}{ll} \Gamma \longrightarrow \Delta' & \text{By i.h.} \\ \Gamma \longrightarrow \Delta', \hat{\alpha} : \kappa & \text{By rule } \longrightarrow\mathsf{Add} \end{array}$$

- **Case** $$\dfrac{\Theta \longrightarrow \Delta'}{\Theta \longrightarrow \underbrace{\Delta', \hat{\alpha} : \kappa = t}_{\Delta}} \longrightarrow\!\mathsf{AddSolved}$$

$\Gamma \longrightarrow \Delta'$	By i.h.
$\Gamma \longrightarrow \Delta', \hat{\alpha} : \kappa = t$	By rule $\longrightarrow$AddSolved

- **Case** $$\dfrac{\Theta' \longrightarrow \Delta'}{\underbrace{\Theta', \blacktriangleright_u}_{\Theta} \longrightarrow \underbrace{\Delta', \blacktriangleright_u}_{\Delta}} \longrightarrow\!\mathsf{Marker}$$

$\Gamma = \Gamma', \blacktriangleright_u$	By inversion on $\mathcal{D}$
$\Gamma' \longrightarrow \Theta'$	By inversion on $\mathcal{D}$
$\Gamma' \longrightarrow \Delta'$	By i.h.
$\Gamma', \blacktriangleright_u \longrightarrow \Delta', \blacktriangleright_u$	By $\longrightarrow$Uvar

□

C′.2 Weakening

Lemma 34 (Suffix Weakening). *If* $\Gamma \vdash t : \kappa$ *then* $\Gamma, \Theta \vdash t : \kappa$.

Proof. By induction on the given derivation. All cases are straightforward. □

Lemma 35 (Suffix Weakening). *If* $\Gamma \vdash A$ *type then* $\Gamma, \Theta \vdash A$ *type.*

Proof. By induction on the given derivation. All cases are straightforward. □

Lemma 36 (Extension Weakening (Sorts)). *If* $\Gamma \vdash t : \kappa$ *and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash t : \kappa$.

Proof. By a straightforward induction on $\Gamma \vdash t : \kappa$.

In the VarSort case, use Lemma 22 (Extension Inversion) (i) or (v). In the SolvedVarSort case, use Lemma 22 (Extension Inversion) (iv). In the other cases, apply the i.h. to all subderivations, then apply the rule. □

Lemma 37 (Extension Weakening (Props)). *If* $\Gamma \vdash P$ *prop and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash P$ *prop.*

Proof. By inversion on rule EqProp, and Lemma 36 (Extension Weakening (Sorts)) twice. □

Lemma 38 (Extension Weakening (Types)). *If* $\Gamma \vdash A$ *type and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash A$ *type.*

Proof. By a straightforward induction on $\Gamma \vdash A$ *type*.

In the VarWF case, use Lemma 22 (Extension Inversion) (i) or (v). In the SolvedVarWF case, use Lemma 22 (Extension Inversion) (iv).

In the other cases, apply the i.h. and/or (for ImpliesWF and WithWF) Lemma 37 (Extension Weakening (Props)) to all subderivations, then apply the rule. □

C′.3 Principal Typing Properties

Lemma 39 (Principal Agreement).

(i) If $\Gamma \vdash A\ !$ *type and* $\Gamma \longrightarrow \Delta$ *then* $[\Delta]A = [\Gamma]A$.

(ii) If $\Gamma \vdash P$ *prop and* $\mathsf{FEV}(P) = \emptyset$ *and* $\Gamma \longrightarrow \Delta$ *then* $[\Delta]P = [\Gamma]P$.

Proof. By induction on the derivation of $\Gamma \longrightarrow \Delta$.

Part (i):

- **Case** $\dfrac{\Gamma_0 \longrightarrow \Delta_0 \qquad [\Delta_0]t = [\Delta_0]t'}{\Gamma_0, \alpha = t \longrightarrow \underbrace{\Delta_0, \alpha = t'}_{\Delta}} \longrightarrow\mathsf{Eqn}$

  If $\alpha \notin FV(A)$, then:

  $$\begin{array}{rll} [\Gamma_0, \alpha = t]A & = [\Gamma_0]A & \text{By def. of subst.} \\ & = [\Delta_0]A & \text{By i.h.} \\ & = [\Delta_0, \alpha = t']A & \text{By def. of subst.} \end{array}$$

  Otherwise, $\alpha \in FV(A)$.

  $$\begin{array}{ll} \Gamma_0 \vdash t\ \mathit{type} & \Gamma \text{ is well-formed} \\ \Gamma_0 \vdash [\Gamma_0]t\ \mathit{type} & \text{By Lemma 13 (Right-Hand Substitution for Typing)} \end{array}$$

  Suppose, for a contradiction, that $\mathsf{FEV}([\Gamma_0]t) \neq \emptyset$.
  Since $\alpha \in FV(A)$, we also have $\mathsf{FEV}([\Gamma]A) \neq \emptyset$, a contradiction.

  $$\begin{array}{rl} \mathsf{FEV}([\Gamma_0]t) \neq \emptyset & \text{Assumption (for contradiction)} \\ [\Gamma_0]t = [\Gamma]\alpha & \text{By def. of subst.} \\ \mathsf{FEV}([\Gamma]\alpha) \neq \emptyset & \text{By above equality} \\ \alpha \in FV(A) & \text{Above} \\ \mathsf{FEV}([\Gamma]A) \neq \emptyset & \text{By a property of subst.} \\ \Gamma \vdash A\ \mathbf{!}\ \mathit{type} & \text{Given} \\ \mathsf{FEV}([\Gamma]A) = \emptyset & \text{By inversion} \\ \Rightarrow\Leftarrow & \\ \mathsf{FEV}([\Gamma_0]t) = \emptyset & \text{By contradiction} \\ \Gamma_0 \vdash t\ \mathbf{!}\ \mathit{type} & \text{By PrincipalWF} \\ [\Gamma_0]t = [\Delta_0]t & \text{By i.h.} \\ & \\ \Gamma_0 \vdash [\Delta_0]t\ \mathit{type} & \text{By above equality} \\ \mathsf{FEV}([\Delta_0]t) = \emptyset & \text{By above equality} \\ \Gamma_0 \vdash \big[[\Delta_0]t/\alpha\big]A\ \mathbf{!}\ \mathit{type} & \text{By Lemma 8 (Substitution—Well-formedness) (i)} \\ [\Gamma_0]\big[[\Delta_0]t/\alpha\big]A = [\Delta_0]\big[[\Delta_0]t/\alpha\big]A & \text{By i.h. (at } \big[[\Delta_0]t/\alpha\big]A) \end{array}$$

  $$\begin{array}{rll} [\Gamma_0, \alpha = t]A & = \big[\Gamma_0\big]\big[[\Gamma_0]t/\alpha\big]A & \text{By def. of subst.} \\ & = \big[\Gamma_0\big]\big[[\Delta_0]t/\alpha\big]A & \text{By above equality} \\ & = \big[\Delta_0\big]\big[[\Delta_0]t/\alpha\big]A & \text{By above equality} \\ & = \big[\Delta_0\big]\big[[\Delta_0]t'/\alpha\big]A & \text{By } [\Delta_0]t = [\Delta_0]t' \\ & = [\Delta]A & \text{By def. of subst.} \end{array}$$

- **Case** $\longrightarrow$Solved, $\longrightarrow$Solve, $\longrightarrow$Add, $\longrightarrow$Solved: Similar to the $\longrightarrow$Eqn case.

- **Case** $\longrightarrow$Id, $\longrightarrow$Var, $\longrightarrow$Uvar, $\longrightarrow$Unsolved, $\longrightarrow$Marker:
  Straightforward, using the i.h. and the definition of substitution.

Part (ii): Similar to part (i), using part (ii) of Lemma 8 (Substitution—Well-formedness). □

Lemma 40 (Right-Hand Subst. for Principal Typing)**.** *If* $\Gamma \vdash A\ p\ \mathit{type}$ *then* $\Gamma \vdash [\Gamma]A\ p\ \mathit{type}$.

Proof. By cases of p:

- Case $p = \mathbf{!}$:

$\Gamma \vdash A\ \mathit{type}$	By inversion
$\mathsf{FEV}([\Gamma]A) = \emptyset$	By inversion
$\Gamma \vdash [\Gamma]A\ \mathit{type}$	By Lemma 13 (Right-Hand Substitution for Typing)
$\Gamma \longrightarrow \Gamma$	By Lemma 32 (Extension Reflexivity)
$[\Gamma][\Gamma]A = [\Gamma]A$	By Lemma 29 (Substitution Monotonicity)
$\mathsf{FEV}([\Gamma][\Gamma]A) = \emptyset$	By inversion
$\Gamma \vdash [\Gamma]A\ !\ \mathit{type}$	By rule PrincipalWF

- Case $p = \not{!}$:

$\Gamma \vdash A\ \mathit{type}$	By inversion
$\Gamma \vdash [\Gamma]A\ \mathit{type}$	By Lemma 13 (Right-Hand Substitution for Typing)
$\Gamma \vdash A\ \not{!}\ \mathit{type}$	By rule NonPrincipalWF

□

Lemma 41 (Extension Weakening for Principal Typing). *If* $\Gamma \vdash A\ p\ \mathit{type}$ *and* $\Gamma \longrightarrow \Delta$ *then* $\Delta \vdash A\ p\ \mathit{type}$.

Proof. By cases of p:

- Case $p = \not{!}$:

$\Gamma \vdash A\ \mathit{type}$	By inversion
$\Delta \vdash A\ \mathit{type}$	By Lemma 38 (Extension Weakening (Types))
$\Delta \vdash A\ \not{!}\ \mathit{type}$	By rule NonPrincipalWF

- Case $p = !$:

$\Gamma \vdash A\ \mathit{type}$	By inversion
$\mathsf{FEV}([\Gamma]A) = \emptyset$	By inversion
$\Delta \vdash A\ \mathit{type}$	By Lemma 38 (Extension Weakening (Types))
$\Delta \vdash [\Delta]A\ \mathit{type}$	By Lemma 13 (Right-Hand Substitution for Typing)
$[\Delta]A = [\Gamma]A$	By Lemma 30 (Substitution Invariance)
$\mathsf{FEV}([\Delta]A) = \emptyset$	By congruence of equality
$\Delta \vdash [\Delta]A\ !\ \mathit{type}$	By rule PrincipalWF

□

Lemma 42 (Inversion of Principal Typing).

(1) If $\Gamma \vdash (A \to B)\ p\ \mathit{type}$ *then* $\Gamma \vdash A\ p\ \mathit{type}$ *and* $\Gamma \vdash B\ p\ \mathit{type}$.

(2) If $\Gamma \vdash (P \supset A)\ p\ \mathit{type}$ *then* $\Gamma \vdash P\ \mathit{prop}$ *and* $\Gamma \vdash A\ p\ \mathit{type}$.

(3) If $\Gamma \vdash (A \wedge P)\ p\ \mathit{type}$ *then* $\Gamma \vdash P\ \mathit{prop}$ *and* $\Gamma \vdash A\ p\ \mathit{type}$.

Proof. Proof of part 1:
We have $\Gamma \vdash A \to B\ p\ \mathit{type}$.

- Case $p = \not{!}$:

1	$\Gamma \vdash A \to B\ \mathit{type}$	By inversion
	$\Gamma \vdash A\ \mathit{type}$	By inversion on 1
	$\Gamma \vdash B\ \mathit{type}$	By inversion on 1
	$\Gamma \vdash A\ \not{!}\ \mathit{type}$	By rule NonPrincipalWF
	$\Gamma \vdash B\ \not{!}\ \mathit{type}$	By rule NonPrincipalWF

- Case $p = \mathbf{!}$:

1	$\Gamma \vdash A \to B$ *type*	By inversion on $\Gamma \vdash A \to B\ \mathbf{!}$ *type*
	$\emptyset = \mathsf{FEV}([\Gamma](A \to B))$	$''$
	$= \mathsf{FEV}([\Gamma]A \to [\Gamma]B)$	By definition of substitution
	$= \mathsf{FEV}([\Gamma]A) \cup \mathsf{FEV}([\Gamma]B)$	By definition of $\mathsf{FEV}(-)$
	$\mathsf{FEV}([\Gamma]A) = \mathsf{FEV}([\Gamma]B) = \emptyset$	By properties of empty sets and unions
	$\Gamma \vdash A$ *type*	By inversion on 1
	$\Gamma \vdash B$ *type*	By inversion on 1
	$\Gamma \vdash A\ \mathbf{!}$ *type*	By rule PrincipalWF
	$\Gamma \vdash B\ \mathbf{!}$ *type*	By rule PrincipalWF

Part 2: We have $\Gamma \vdash P \supset A\ p$ *type*. Similar to Part 1.

Part 3: We have $\Gamma \vdash A \wedge P\ p$ *type*. Similar to Part 2. □

C′.4 Instantiation Extends

Lemma 43 (Instantiation Extension).
If $\Gamma \vdash \hat\alpha := \tau : \kappa \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Proof. By induction on the given derivation.

- **Case**

$$\frac{\Gamma_L \vdash \tau : \kappa}{\underbrace{\Gamma_L, \hat\alpha : \kappa, \Gamma_R}_{\Gamma} \vdash \hat\alpha := \tau : \kappa \dashv \Gamma_L, \hat\alpha : \kappa = \tau, \Gamma_R}\ \mathsf{InstSolve}$$

  Follows by Lemma 23 (Deep Evar Introduction) (ii).

- **Case**

$$\frac{\hat\beta \in \mathsf{unsolved}(\Gamma_0[\hat\alpha : \kappa][\hat\beta : \kappa])}{\underbrace{\Gamma_0[\hat\alpha : \kappa][\hat\beta : \kappa]}_{\Gamma} \vdash \hat\alpha := \hat\beta : \kappa \dashv \Gamma_0[\hat\alpha : \kappa][\hat\beta : \kappa = \hat\alpha]}\ \mathsf{InstReach}$$

  Follows by Lemma 23 (Deep Evar Introduction) (ii).

- **Case**

$$\frac{\Gamma_0[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \oplus \hat\alpha_2] \vdash \hat\alpha_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat\alpha_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma_0[\hat\alpha : \star] \vdash \hat\alpha := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \mathsf{InstBin}$$

$\Gamma_0[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \oplus \hat\alpha_2] \vdash \hat\alpha_1 := \tau_1 : \star \dashv \Theta$	Subderivation
$\Gamma_0[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \oplus \hat\alpha_2] \longrightarrow \Theta$	By i.h.
$\Theta \vdash \hat\alpha_2 := [\Theta]\tau_2 : \star \dashv \Delta$	Subderivation
$\Theta \longrightarrow \Delta$	By i.h.
$\Gamma_0[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \oplus \hat\alpha_2] \longrightarrow \Delta$	By Lemma 33 (Extension Transitivity)
$\Gamma_0[\hat\alpha : \star] \longrightarrow \Gamma_0[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \oplus \hat\alpha_2]$	By Lemma 23 (Deep Evar Introduction) (parts (i), (i), and (ii), using Lemma 33 (Extension Transitivity))

  $\Gamma_0[\hat\alpha : \star] \longrightarrow \Delta$ By Lemma 33 (Extension Transitivity)

- **Case**

$$\frac{}{\Gamma_0[\hat\alpha : \mathbb{N}] \vdash \hat\alpha := \mathsf{zero} : \mathbb{N} \dashv \Gamma_0[\hat\alpha : \mathbb{N} = \mathsf{zero}]}\ \mathsf{InstZero}$$

  Follows by Lemma 23 (Deep Evar Introduction) (ii).

- **Case** $$\dfrac{\Gamma[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)] \vdash \hat{\alpha}_1 := t_1 : \mathbb{N} \dashv \Delta}{\Gamma[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \mathsf{InstSucc}$$

  By reasoning similar to the InstBin case. □

C′.5 Equivalence Extends

Lemma 44 (Elimeq Extension).
If $\Gamma \mathrel{/} s \mathrel{\mathring{=}} t : \kappa \dashv \Delta$ *then there exists* Θ *such that* $\Gamma, \Theta \longrightarrow \Delta$.

Proof. By induction on the given derivation. Note that the statement restricts the output to be a (consistent) context Δ.

- **Case** $$\dfrac{}{\Gamma \mathrel{/} \alpha \mathrel{\mathring{=}} \alpha : \kappa \dashv \Gamma}\ \mathsf{ElimeqUvarRefl}$$

  Since $\Delta = \Gamma$, applying Lemma 32 (Extension Reflexivity) suffices (let $\Theta = \cdot$).

- **Case** $$\dfrac{}{\Gamma \mathrel{/} \mathsf{zero} \mathrel{\mathring{=}} \mathsf{zero} : \mathbb{N} \dashv \Gamma}\ \mathsf{ElimeqZero}$$

  Similar to the ElimeqUvarRefl case.

- **Case** $$\dfrac{\Gamma \mathrel{/} \sigma \mathrel{\mathring{=}} t : \mathbb{N} \dashv \Delta}{\Gamma \mathrel{/} \mathsf{succ}(\sigma) \mathrel{\mathring{=}} \mathsf{succ}(t) : \mathbb{N} \dashv \Delta}\ \mathsf{ElimeqSucc}$$

  Follows by i.h.

- **Case** $$\dfrac{\Gamma_0[\hat{\alpha} : \kappa] \vdash \hat{\alpha} := t : \kappa \dashv \Delta}{\underbrace{\Gamma_0[\hat{\alpha} : \kappa]}_{\Gamma} \mathrel{/} \hat{\alpha} \mathrel{\mathring{=}} t : \kappa \dashv \Delta}\ \mathsf{ElimeqInstL}$$

	$\Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta$	Subderivation
	$\Gamma \longrightarrow \Delta$	By Lemma 43 (Instantiation Extension)
	Let $\Theta = \cdot$.	
☞	$\Gamma, \Theta \longrightarrow \Delta$	By $\Theta = \cdot$

- **Case** $$\dfrac{\alpha \notin FV([\Gamma]t) \qquad (\alpha = -) \notin \Gamma}{\Gamma \mathrel{/} \alpha \mathrel{\mathring{=}} t : \kappa \dashv \Gamma, \alpha = t}\ \mathsf{ElimeqUvarL}$$

  Let Θ be $(\alpha = t)$.

  ☞ $\Gamma, \underbrace{\alpha = t}_{\Theta} \longrightarrow \Gamma, \alpha = t$ By Lemma 32 (Extension Reflexivity)

- **Cases** ElimeqInstR , ElimeqUvarR:
  Similar to the respective L cases.

- **Case** $$\dfrac{\sigma \mathrel{\#} t}{\Gamma \mathrel{/} \sigma \mathrel{\mathring{=}} t : \kappa \dashv \bot}\ \mathsf{ElimeqClash}$$

  The statement says that the output is a (consistent) context Δ, so this case is impossible. □

Lemma 45 (Elimprop Extension)**.**
If $\Gamma \;/\; P \dashv \Delta$ *then there exists* Θ *such that* $\Gamma, \Theta \longrightarrow \Delta$*.*

Proof. By induction on the given derivation. Note that the statement restricts the output to be a (consistent) context Δ.

- **Case** $\dfrac{\Gamma \;/\; \sigma \mathrel{\mathring{=}} t : \mathbb{N} \dashv \Delta}{\Gamma \;/\; \sigma = t \dashv \Delta}$ ElimpropEq

  $\Gamma \;/\; \sigma \mathrel{\mathring{=}} t : \mathbb{N} \dashv \Delta$ Subderivation
  ☞ $\Gamma, \Theta \longrightarrow \Delta$ By Lemma 44 (Elimeq Extension) □

Lemma 46 (Checkeq Extension)**.**
If $\Gamma \vdash A \equiv B \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$*.*

Proof. By induction on the given derivation.

- **Case**

  $\dfrac{}{\Gamma \vdash u \mathrel{\mathring{=}} u : \kappa \dashv \Gamma}$ CheckeqVar

  Since $\Delta = \Gamma$, applying Lemma 32 (Extension Reflexivity) suffices.

- **Cases** CheckeqUnit, CheckeqZero: Similar to the CheckeqVar case.

- **Case** $\dfrac{\Gamma \vdash \tau_1 \mathrel{\mathring{=}} \tau_1' : \star \dashv \Theta \qquad \Theta \vdash [\Theta]\tau_2 \mathrel{\mathring{=}} [\Theta]\tau_2' : \star \dashv \Delta}{\Gamma \vdash \tau_1 \oplus \tau_2 \mathrel{\mathring{=}} \tau_1' \oplus \tau_2' : \star \dashv \Delta}$ CheckeqBin

  $\Gamma \longrightarrow \Theta$ By i.h.
  $\Theta \longrightarrow \Delta$ By i.h.
  ☞ $\Gamma \longrightarrow \Delta$ By Lemma 33 (Extension Transitivity)

- **Case** $\dfrac{\Gamma \vdash \sigma \mathrel{\mathring{=}} t : \mathbb{N} \dashv \Delta}{\Gamma \vdash \mathsf{succ}(\sigma) \mathrel{\mathring{=}} \mathsf{succ}(t) : \mathbb{N} \dashv \Delta}$ CheckeqSucc

  $\Gamma \vdash \sigma \mathrel{\mathring{=}} t : \mathbb{N} \dashv \Delta$ Subderivation
  ☞ $\Gamma \longrightarrow \Delta$ By i.h.

- **Case** $\dfrac{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := t : \kappa \dashv \Delta \qquad \hat{\alpha} \notin FV([\Gamma_0[\hat{\alpha}]]t)}{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} \mathrel{\mathring{=}} t : \kappa \dashv \Delta}$ CheckeqInstL

  $\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := t : \kappa \dashv \Delta$ Subderivation
  ☞ $\underbrace{\Gamma_0[\hat{\alpha}]}_{\Gamma} \longrightarrow \Delta$ By Lemma 43 (Instantiation Extension)

- **Case** CheckeqInstR: Similar to the CheckeqInstL case. □

Lemma 47 (Checkprop Extension)**.**
If $\Gamma \vdash P$ *true* $\dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$*.*

Proof. By induction on the given derivation.

- **Case** $$\dfrac{\Gamma \vdash \sigma \stackrel{\circ}{=} t : \mathbb{N} \dashv \Delta}{\Gamma \vdash \sigma = t \; \mathit{true} \dashv \Delta} \; \mathsf{CheckpropEq}$$

	$\Gamma \vdash \sigma \stackrel{\circ}{=} t : \mathbb{N} \dashv \Delta$	Subderivation
☞	$\Gamma \longrightarrow \Delta$	By Lemma 46 (Checkeq Extension) ☐

Lemma 48 (Prop Equivalence Extension).
If $\Gamma \vdash P \equiv Q \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Proof. By induction on the given derivation.

- **Case** $$\dfrac{\Gamma \vdash \sigma_1 \stackrel{\circ}{=} \tau_1 : \mathbb{N} \dashv \Theta \qquad \Theta \vdash \sigma_2 \stackrel{\circ}{=} \tau_2 : \mathbb{N} \dashv \Delta}{\Gamma \vdash (\sigma_1 = \sigma_2) \equiv (\tau_1 = \tau_2) \dashv \Delta} \equiv\!\mathsf{PropEq}$$

	$\Gamma \vdash \sigma_1 \stackrel{\circ}{=} \tau_1 : \mathbb{N} \dashv \Theta$	Subderivation
	$\Gamma \longrightarrow \Theta$	By Lemma 46 (Checkeq Extension)
	$\Theta \vdash \sigma_2 \stackrel{\circ}{=} \tau_2 : \mathbb{N} \dashv \Delta$	Subderivation
	$\Theta \longrightarrow \Delta$	By Lemma 46 (Checkeq Extension)
☞	$\Gamma \longrightarrow \Delta$	By Lemma 33 (Extension Transitivity) ☐

Lemma 49 (Equivalence Extension).
If $\Gamma \vdash A \equiv B \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Proof. By induction on the given derivation.

- **Case** $$\dfrac{}{\Gamma \vdash \alpha \equiv \alpha \dashv \Gamma} \equiv\!\mathsf{Var}$$

  Here $\Delta = \Gamma$, so Lemma 32 (Extension Reflexivity) suffices.

- **Case** $$\dfrac{}{\Gamma \vdash \hat{\alpha} \equiv \hat{\alpha} \dashv \Gamma} \equiv\!\mathsf{Exvar}$$

  Similar to the $\equiv$Var case.

- **Case** $$\dfrac{}{\Gamma \vdash 1 \equiv 1 \dashv \Gamma} \equiv\!\mathsf{Unit}$$

  Similar to the $\equiv$Var case.

- **Case** $$\dfrac{\Gamma \vdash A_1 \equiv B_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta}{\Gamma \vdash (A_1 \oplus A_2) \equiv (B_1 \oplus B_2) \dashv \Delta} \equiv\!\oplus$$

	$\Gamma \vdash A_1 \equiv B_1 \dashv \Theta$	Subderivation
	$\Gamma \longrightarrow \Theta$	By i.h.
	$\Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta$	Subderivation
	$\Theta \longrightarrow \Delta$	By i.h.
☞	$\Gamma \longrightarrow \Delta$	By Lemma 33 (Extension Transitivity)

- **Case** $\equiv$Vec: Similar to the $\equiv\oplus$ case.

- **Cases** $\equiv\supset$, $\equiv\wedge$: Similar to the $\equiv\oplus$ case, but with Lemma 48 (Prop Equivalence Extension) on the first premise.

- **Case** $\dfrac{\Gamma, \alpha : \kappa \vdash A_0 \equiv B \dashv \Delta, \alpha : \kappa, \Delta'}{\Gamma \vdash \forall \alpha : \kappa.\, A_0 \equiv \forall \alpha : \kappa.\, B \dashv \Delta} \equiv\forall$

$$
\begin{array}{rll}
 & \Gamma, \alpha : \kappa \vdash A_0 \equiv B \dashv \Delta, \alpha : \kappa, \Delta' & \text{Subderivation} \\
 & \Gamma, \alpha : \kappa \longrightarrow \Delta, \alpha : \kappa, \Delta' & \text{By i.h.} \\
☞ & \Gamma \longrightarrow \Delta & \text{By Lemma 22 (Extension Inversion) (i)}
\end{array}
$$

- **Case** $\dfrac{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta \qquad \hat{\alpha} \notin FV([\Gamma_0[\hat{\alpha}]]\tau)}{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} \equiv \tau \dashv \Delta} \equiv\text{InstantiateL}$

$$
\begin{array}{rll}
 & \Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta & \text{Subderivation} \\
☞ & \underbrace{\Gamma_0[\hat{\alpha}]}_{\Gamma} \longrightarrow \Delta & \text{By Lemma 43 (Instantiation Extension)}
\end{array}
$$

- **Case** $\equiv$InstantiateR: Similar to the $\equiv$InstantiateL case. □

C′.6 Subtyping Extends

Lemma 50 (Subtyping Extension)**.** *If* $\Gamma \vdash A <:^{\mp} B \dashv \Delta$ *then* $\Gamma \longrightarrow \Delta$.

Proof. By induction on the given derivation.

- **Case** $\dfrac{\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa \vdash [\hat{\alpha}/\alpha]A <:^{-} B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta}{\Gamma \vdash \forall \alpha : \kappa.\, A <:^{-} B \dashv \Delta} <:\forall\text{L}$

$$
\begin{array}{rll}
 & \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa \vdash [\hat{\alpha}/\alpha]A <:^{-} B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta & \text{Subderivation} \\
 & \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa \longrightarrow \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta & \text{By i.h. (i)} \\
☞ & \Gamma \longrightarrow \Delta & \text{By Lemma 22 (Extension Inversion) (ii)}
\end{array}
$$

- **Case** $<:\exists$R: Similar to the $<:\forall$L case.

- **Case** $\dfrac{\Gamma, \alpha : \kappa \vdash A <:^{*} B \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma \vdash A <:^{*} \forall \alpha : \kappa.\, B \dashv \Delta} <:\forall\text{R}$

  Similar to the $<:\forall$L case, but using part (i) of Lemma 22 (Extension Inversion).

- **Case** $<:\exists$L: Similar to the $<:\forall$R case.

- **Case** $\dfrac{\Gamma \vdash A \equiv B \dashv \Delta}{\Gamma \vdash A <:^{*} B \dashv \Delta} <:\text{Equiv}$

$$
\begin{array}{rll}
 & \Gamma \vdash A \equiv B \dashv \Delta & \text{Subderivation} \\
☞ & \Gamma \longrightarrow \Delta & \text{By Lemma 49 (Equivalence Extension)}
\end{array}
$$

□

C′.7 Typing Extends

Lemma 51 (Typing Extension)**.**
If $\Gamma \vdash e \Leftarrow A\ p \dashv \Delta$
or $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$
or $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$
or $\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$
or $\Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$
then $\Gamma \longrightarrow \Delta$.

Proof. By induction on the given derivation.

- **Match judgments:**

  In rule MatchEmpty, $\Delta = \Gamma$, so the result follows by Lemma 32 (Extension Reflexivity).

  Rules MatchBase, Match$\times$, Match$+_k$ and MatchWild each have a single premise in which the contexts match the conclusion (input Γ and output Δ), so the result follows by i.h. For rule MatchSeq, Lemma 33 (Extension Transitivity) is also needed.

  In rule Match$\exists$, apply the i.h., then use Lemma 22 (Extension Inversion) (i).

  Match$\wedge$: Use the i.h.

  MatchNeg: Use the i.h. and Lemma 22 (Extension Inversion) (v).

  Match$\bot$: Immediate by Lemma 32 (Extension Reflexivity).

  MatchUnify:

	$\Gamma, \blacktriangleright_P, \Theta' \longrightarrow \Theta$	By Lemma 44 (Elimeq Extension)
	$\Theta \longrightarrow \Delta, \blacktriangleright_P, \Delta'$	By i.h.
	$\Gamma, \blacktriangleright_P, \Theta' \longrightarrow \Delta, \blacktriangleright_P, \Delta'$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \longrightarrow \Delta$	By Lemma 22 (Extension Inversion) (ii)

- **Synthesis, checking, and spine judgments:** In rules Var, 1I, EmptySpine, and $\supset$I$\bot$, the output context Δ is exactly Γ, so the result follows by Lemma 32 (Extension Reflexivity).

  - **Case** $\forall$I: Use the i.h. and Lemma 33 (Extension Transitivity).
  - **Case** $\forall$Spine: By $\longrightarrow$Add, $\Gamma \longrightarrow \Gamma, \hat{\alpha} : \kappa$.
    The result follows by i.h. and Lemma 33 (Extension Transitivity).
  - **Cases** $\wedge$I, $\supset$Spine: Use Lemma 47 (Checkprop Extension), the i.h., and Lemma 33 (Extension Transitivity).
  - **Cases** Nil, Cons: Using reasoning found in the $\wedge$I and $\supset$I cases.
  - **Case** $\exists$I: Use the i.h.
  - **Case** $\supset$I:

	$\Gamma, \blacktriangleright_P, \Theta' \longrightarrow \Theta$	By Lemma 45 (Elimprop Extension)
	$\Theta \longrightarrow \Delta, \blacktriangleright_P, \Delta$	By i.h.
	$\Gamma, \blacktriangleright_P, \Theta' \longrightarrow \Delta, \blacktriangleright_P, \Delta$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \longrightarrow \Delta$	By Lemma 22 (Extension Inversion)

  - **Cases** $\rightarrow$I, Rec: Use the i.h. and Lemma 22 (Extension Inversion).
  - **Cases** Sub, Anno, $\rightarrow$E, $\rightarrow$E-**!**, $\rightarrow$Spine, $+$I$_k$, $\times$I:
    Use the i.h., and Lemma 33 (Extension Transitivity) as needed.
  - **Case** 1I$\hat{\alpha}$: By Lemma 23 (Deep Evar Introduction) (ii).
  - **Case** $\hat{\alpha}$Spine, $+$I$\hat{\alpha}_k$, $\times$I$\hat{\alpha}$:
    Use Lemma 23 (Deep Evar Introduction) (i) twice, Lemma 23 (Deep Evar Introduction) (ii), the i.h., and Lemma 33 (Extension Transitivity).

- **Case** $\to\mathsf{I}\hat{\alpha}$: Use Lemma 23 (Deep Evar Introduction) (i) twice, Lemma 23 (Deep Evar Introduction) (ii), the i.h. and Lemma 22 (Extension Inversion) (v).
- **Case** Case: Use the i.h. on the synthesis premise and the match premise, and then Lemma 33 (Extension Transitivity). □

C′.8 Unfiled

Lemma 52 (Context Partitioning).
If $\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta \longrightarrow \Omega, \blacktriangleright_{\hat{\alpha}}, \Omega_Z$ *then there is a* Ψ *such that* $[\Omega, \blacktriangleright_{\hat{\alpha}}, \Omega_Z](\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta) = [\Omega]\Delta, \Psi$.

Proof. By induction on the given derivation.

- **Case** $\longrightarrow$Id: Impossible: $\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta$ cannot have the form $\cdot$.
- **Case** $\longrightarrow$Var: We have $\Omega_Z = (\Omega'_Z, x : A)$ and $\Theta = (\Theta', x : A')$. By i.h., there is Ψ' such that $[\Omega, \blacktriangleright_{\hat{\alpha}}, \Omega'_Z](\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta') = [\Omega]\Delta, \Psi'$. Then by the definition of context application, $[\Omega, \blacktriangleright_{\hat{\alpha}}, \Omega'_Z, x : A](\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta', x : A') = [\Omega]\Delta, \Psi', x : [\Omega']A$. Let $\Psi = (\Psi', x : [\Omega']A)$.
- **Case** $\longrightarrow$Uvar: Similar to the $\longrightarrow$Var case, with $\Psi = (\Psi', \alpha : \kappa)$.
- **Cases** $\longrightarrow$Eqn, $\longrightarrow$Unsolved, $\longrightarrow$Solved, $\longrightarrow$Solve, $\longrightarrow$Add, $\longrightarrow$AddSolved, $\longrightarrow$Marker:

  Broadly similar to the $\longrightarrow$Uvar case, but the rightmost context element disappears in context application, so we let $\Psi = \Psi'$. □

Lemma 54 (Completing Stability).
If $\Gamma \longrightarrow \Omega$ *then* $[\Omega]\Gamma = [\Omega]\Omega$.

Proof. By induction on the derivation of $\Gamma \longrightarrow \Omega$.

- **Case**

  $$\frac{}{\cdot \longrightarrow \cdot}\ \longrightarrow\mathsf{Id}$$

  Immediate.

- **Case** $$\frac{\Gamma_0 \longrightarrow \Omega_0 \qquad [\Omega_0]A = [\Omega_0]A'}{\Gamma_0, x : A \longrightarrow \Omega_0, x : A'}\ \longrightarrow\mathsf{Var}$$

  $$\begin{array}{rcll}
  \Gamma_0 & \longrightarrow & \Omega_0 & \text{Subderivation} \\
  [\Omega_0]\Gamma_0 & = & [\Omega_0]\Omega_0 & \text{By i.h.} \\
  [\Omega_0]A & = & [\Omega_0]A' & \text{Subderivation} \\
  [\Omega_0]\Gamma_0, x : [\Omega_0]A & = & [\Omega_0]\Omega_0, x : [\Omega_0]A' & \text{By congruence of equality} \\
  [\Omega_0, x : A'](\Gamma_0, x : A) & = & [\Omega_0, x : A'](\Omega_0, x : A') & \text{By definition of substitution}
  \end{array}$$

- **Case** $$\frac{\Gamma_0 \longrightarrow \Omega_0}{\Gamma_0, \alpha : \kappa \longrightarrow \Omega_0, \alpha : \kappa}\ \longrightarrow\mathsf{Uvar}$$

  Similar to $\longrightarrow$Var.

- **Case** $$\frac{\Gamma_0 \longrightarrow \Omega_0}{\Gamma_0, \hat{\alpha} : \kappa \longrightarrow \Omega_0, \hat{\alpha} : \kappa}\ \longrightarrow\mathsf{Unsolved}$$

  Similar to $\longrightarrow$Var.

- **Case** $\dfrac{\Gamma_0 \longrightarrow \Omega_0 \qquad [\Omega_0]t = [\Omega_0]t'}{\Gamma_0, \hat{\alpha} : \kappa = t \longrightarrow \Omega_0, \hat{\alpha} : \kappa = t'} \longrightarrow\mathsf{Solved}$

  Similar to $\longrightarrow$Var.

- **Case** $\dfrac{\Gamma_0 \longrightarrow \Omega_0}{\Gamma_0, \blacktriangleright_{\hat{\alpha}} \longrightarrow \Omega_0, \blacktriangleright_{\hat{\alpha}}} \longrightarrow\mathsf{Marker}$

  Similar to $\longrightarrow$Var.

- **Case** $\dfrac{\Gamma_0 \longrightarrow \Omega_0}{\Gamma_0, \hat{\beta} : \kappa' \longrightarrow \Omega_0, \hat{\beta} : \kappa' = t} \longrightarrow\mathsf{Solve}$

  Similar to $\longrightarrow$Var.

- **Case** $\dfrac{\Gamma_0 \longrightarrow \Omega_0 \qquad [\Omega_0]t' = [\Omega_0]t}{\Gamma_0, \alpha = t' \longrightarrow \Omega_0, \alpha = t} \longrightarrow\mathsf{Eqn}$

$$
\begin{array}{rcll}
\Gamma_0 & \longrightarrow & \Omega_0 & \text{Subderivation} \\
{[\Omega_0]t'} & = & [\Omega_0]t & \text{Subderivation} \\
{[\Omega_0]\Gamma_0} & = & [\Omega_0]\Omega_0 & \text{By i.h.} \\
{[[\Omega_0]t/\alpha]([\Omega_0]\Gamma_0)} & = & [[\Omega_0]t/\alpha]([\Omega_0]\Omega_0) & \text{By congruence of equality} \\
{[\Omega_0, \alpha = t](\Gamma_0, \alpha = t')} & = & [\Omega_0, \alpha = t](\Omega_0, \alpha = t) & \text{By definition of context substitution}
\end{array}
$$

- **Case** $\dfrac{\Gamma \longrightarrow \Omega_0}{\Gamma \longrightarrow \Omega_0, \hat{\alpha} : \kappa} \longrightarrow\mathsf{Add}$

$$
\begin{array}{rcll}
\Gamma & \longrightarrow & \Omega_0 & \text{Subderivation} \\
{[\Omega_0]\Gamma} & = & [\Omega_0]\Omega_0 & \text{By i.h.} \\
{[\Omega_0, \hat{\alpha} : \kappa]\Gamma} & = & [\Omega_0, \hat{\alpha} : \kappa](\Omega_0, \hat{\alpha} : \kappa) & \text{By definition of context substitution}
\end{array}
$$

- **Case** $\dfrac{\Gamma \longrightarrow \Omega_0}{\Gamma \longrightarrow \Omega_0, \hat{\alpha} : \kappa = t} \longrightarrow\mathsf{AddSolved}$

  Similar to the $\longrightarrow$Add case. ☐

Lemma 55 (Completing Completeness)**.**

(i) If $\Omega \longrightarrow \Omega'$ *and* $\Omega \vdash t : \kappa$ *then* $[\Omega]t = [\Omega']t$.

(ii) If $\Omega \longrightarrow \Omega'$ *and* $\Omega \vdash A$ *type then* $[\Omega]A = [\Omega']A$.

(iii) If $\Omega \longrightarrow \Omega'$ *then* $[\Omega]\Omega = [\Omega']\Omega'$.

Proof.

- **Part (i):**

  By Lemma 29 (Substitution Monotonicity) (i), $[\Omega']t = [\Omega'][\Omega]t$.

  Now we need to show $[\Omega'][\Omega]t = [\Omega]t$. Considered as a substitution, Ω' is the identity everywhere except existential variables $\hat{\alpha}$ and universal variables α. First, since Ω is complete, $[\Omega]t$ has no free existentials. Second, universal variables free in $[\Omega]t$ have no equations in Ω (if they had, their occurrences would have been replaced). But if Ω has no equation for α, it follows from $\Omega \longrightarrow \Omega'$ and the definition of context extension in Figure 13 that Ω' also lacks an equation, so applying Ω' also leaves α alone.

  Transitivity of equality gives $[\Omega']t = [\Omega]t$.

- **Part (ii):** Similar to part (i), using Lemma 29 (Substitution Monotonicity) (iii) instead of (i).
- **Part (iii):** By induction on the given derivation of $\Omega \longrightarrow \Omega'$.

  Only cases $\longrightarrow$Id, $\longrightarrow$Var, $\longrightarrow$Uvar, $\longrightarrow$Eqn, $\longrightarrow$Solved, $\longrightarrow$AddSolved and $\longrightarrow$Marker are possible. In all of these cases, we use the i.h. and the definition of context application; in cases $\longrightarrow$Var, $\longrightarrow$Eqn and $\longrightarrow$Solved, we also use the equality in the premise of the respective rule. □

Lemma 56 (Confluence of Completeness)**.**
If $\Delta_1 \longrightarrow \Omega$ *and* $\Delta_2 \longrightarrow \Omega$ *then* $[\Omega]\Delta_1 = [\Omega]\Delta_2$.

Proof.

$\Delta_1 \longrightarrow \Omega$	Given
$[\Omega]\Delta_1 = [\Omega]\Omega$	By Lemma 54 (Completing Stability)
$\Delta_2 \longrightarrow \Omega$	Given
$[\Omega]\Delta_2 = [\Omega]\Omega$	By Lemma 54 (Completing Stability)
$[\Omega]\Delta_1 = [\Omega]\Delta_2$	By transitivity of equality □

Lemma 57 (Multiple Confluence)**.**
If $\Delta \longrightarrow \Omega$ *and* $\Omega \longrightarrow \Omega'$ *and* $\Delta' \longrightarrow \Omega'$ *then* $[\Omega]\Delta = [\Omega']\Delta'$.

Proof.

$\Delta \longrightarrow \Omega$	Given
$[\Omega]\Delta = [\Omega]\Omega$	By Lemma 54 (Completing Stability)
$\Omega \longrightarrow \Omega'$	Given
$[\Omega]\Omega = [\Omega']\Omega'$	By Lemma 55 (Completing Completeness) (iii)
$= [\Omega']\Delta'$	By Lemma 54 (Completing Stability) ($\Delta' \longrightarrow \Omega'$ given) □

Lemma 59 (Canonical Completion)**.**
If $\Gamma \longrightarrow \Omega$
then there exists Ω_{canon} *such that* $\Gamma \longrightarrow \Omega_{canon}$ *and* $\Omega_{canon} \longrightarrow \Omega$ *and* $\mathsf{dom}(\Omega_{canon}) = \mathsf{dom}(\Gamma)$ *and, for all* $\hat{\alpha} : \kappa = \tau$ *and* $\alpha = \tau$ *in* Ω_{canon}*, we have* $\mathsf{FEV}(\tau) = \emptyset$.

Proof. By induction on Ω. In Ω_{canon}, make all solutions (for evars and uvars) canonical by applying Ω to them, dropping declarations of existential variables that aren't in $\mathsf{dom}(\Gamma)$. □

Lemma 60 (Split Solutions)**.**
If $\Delta \longrightarrow \Omega$ *and* $\hat{\alpha} \in \mathsf{unsolved}(\Delta)$
then there exists $\Omega_1 = \Omega_1'[\hat{\alpha} : \kappa = t_1]$ *such that* $\Omega_1 \longrightarrow \Omega$ *and* $\Omega_2 = \Omega_1'[\hat{\alpha} : \kappa = t_2]$ *where* $\Delta \longrightarrow \Omega_2$ *and* $t_2 \neq t_1$ *and* Ω_2 *is canonical.*

Proof. Use Lemma 59 (Canonical Completion) to get Ω_{canon} such that $\Delta \longrightarrow \Omega_{canon}$ and $\Omega_{canon} \longrightarrow \Omega$, where for all solutions t in Ω_{canon} we have $\mathsf{FEV}(t) = \emptyset$.

We have $\Omega_{canon} = \Omega_1'[\hat{\alpha} : \kappa = t_1]$, where $\mathsf{FEV}(t_1) = \emptyset$. Therefore ☞ $\Omega_1'[\hat{\alpha} : \kappa = t_1] \longrightarrow \Omega$.

Now choose t_2 as follows:

- If $\kappa = \star$, let $t_2 = t_1 \rightarrow t_1$.
- If $\kappa = \mathbb{N}$, let $t_2 = \mathsf{succ}(t_1)$.

Thus, ☞ $t_2 \neq t_1$. Let $\Omega_2 = \Omega_1'[\hat{\alpha} : \kappa = t_2]$.

☞ $\Delta \longrightarrow \Omega_2$ By Lemma 31 (Split Extension) □

D′ Internal Properties of the Declarative System

Lemma 61 (Interpolating With and Exists)**.**

(1) If $\mathcal{D} :: \Psi \vdash \Pi :: \vec{A} \Leftarrow C\ p$ *and* $\Psi \vdash P_0$ *true*
then $\mathcal{D}' :: \Psi \vdash \Pi :: \vec{A} \Leftarrow C \wedge P_0\ p$.

(2) If $\mathcal{D} :: \Psi \vdash \Pi :: \vec{A} \Leftarrow [\tau/\alpha]C_0\ p$ *and* $\Psi \vdash \tau : \kappa$
then $\mathcal{D}' :: \Psi \vdash \Pi :: \vec{A} \Leftarrow (\exists\alpha : \kappa.\, C_0)\ p$.

In both cases, the height of $\mathcal{D}'$ *is one greater than the height of* $\mathcal{D}$.
Moreover, similar properties hold for the eliminating judgment $\Psi \mathrel{/} P \vdash \Pi :: \vec{A} \Leftarrow C\ p$.

Proof. By induction on the given match derivation.

In the DeclMatchBase case, for part (1), apply rule $\wedge$I. For part (2), apply rule $\exists$I.

In the DeclMatchNeg case, part (1), use Lemma 2 (Declarative Weakening) (iii). In part (2), use Lemma 2 (Declarative Weakening) (i). □

Lemma 62 (Case Invertibility)**.**
If $\Psi \vdash \mathsf{case}(e_0, \Pi) \Leftarrow C\ p$
then $\Psi \vdash e_0 \Rightarrow A\ !$ *and* $\Psi \vdash \Pi :: A \Leftarrow C\ p$ *and* $\Psi \vdash \Pi$ *covers* A
where the height of each resulting derivation is strictly less than the height of the given derivation.

Proof. By induction on the given derivation.

- **Case** $$\dfrac{\Psi \vdash \mathsf{case}(e_0, \Pi) \Rightarrow A\ q \qquad \mathrm{pol}(B) \vdash \Psi \leq^{*} A B}{\Psi \vdash \mathsf{case}(e_0, \Pi) \Leftarrow B\ p}\ \mathsf{DeclSub}$$

  Impossible, because $\Psi \vdash \mathsf{case}(e_0, \Pi) \Rightarrow A\ q$ is not derivable.

- **Cases** Decl$\forall$I, Decl$\supset$I: Impossible: these rules have a value restriction, but a case expression is not a value.

- **Case** $$\dfrac{\Psi \vdash P\ \mathit{true} \qquad \Psi \vdash \mathsf{case}(e_0, \Pi) \Leftarrow C_0\ p}{\Psi \vdash \mathsf{case}(e_0, \Pi) \Leftarrow C_0 \wedge P\ p}\ \mathsf{Decl}\wedge\mathsf{I}$$

☞	$< n-1\quad \Psi \vdash e_0 \Rightarrow A\ !$	By i.h.
	$< n-1\quad \Psi \vdash \Pi :: A \Leftarrow C_0\ p$	″
☞	$< n-1\quad \Psi \vdash \Pi$ *covers* A	″
	$\leq n-1\quad \Psi \vdash P$ *true*	Subderivation
☞	$< n\quad \Psi \vdash \Pi :: A \Leftarrow C_0 \wedge P\ p$	By Lemma 61 (Interpolating With and Exists) (1)

- **Cases** Decl1I, Decl$\to$I, DeclRec, Decl$+$I$_k$, Decl$\times$I, DeclNil, DeclCons: Impossible, because in these rules e cannot have the form $\mathsf{case}(e_0, \Pi)$.

- **Case** $$\dfrac{\Psi \vdash \mathsf{case}(e_0, \Pi) \Rightarrow A\ ! \qquad \Psi \vdash \Pi :: A \Leftarrow C\ p \qquad \Psi \vdash \Pi\ \mathit{covers}\ A}{\Psi \vdash \mathsf{case}(e_0, \Pi) \Leftarrow C\ p}\ \mathsf{DeclCase}$$

  Immediate. □

E′ Miscellaneous Properties of the Algorithmic System

Lemma 63 (Well-Formed Outputs of Typing)**.**

(Spines) *If* $\Gamma \vdash s : A\ q \gg C\ p \dashv \Delta$ *or* $\Gamma \vdash s : A\ q \gg C\ \lceil p \rceil \dashv \Delta$
and $\Gamma \vdash A\ q$ *type*
then $\Delta \vdash C\ p$ *type*.

(Synthesis) *If* $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$
then $A \vdash p$ *type*.

Proof. By induction on the given derivation.

- **Case** Anno: Use Lemma 51 (Typing Extension) and Lemma 41 (Extension Weakening for Principal Typing).
- **Case** $\forall$Spine: We have $\Gamma \vdash (\forall \alpha : \kappa.\ A_0)\ q$ *type*.
  By inversion, $\Gamma, \alpha : \kappa \vdash A_0\ q$ *type*.
  By properties of substitution, $\Gamma, \hat{\alpha} : \kappa \vdash [\hat{\alpha}/\alpha]A_0\ q$ *type*.
  Now apply the i.h.
- **Case** $\supset$Spine: Use Lemma 42 (Inversion of Principal Typing) (2), Lemma 47 (Checkprop Extension), and Lemma 41 (Extension Weakening for Principal Typing).
- **Case** SpineRecover:

  By i.h., $\Delta \vdash C \not{!}$ *type*.
  We have as premise $\mathsf{FEV}(C) = \emptyset$.
  Therefore $\Delta \vdash C\ \mathbf{!}$ *type*.
- **Case** SpinePass: By i.h.
- **Case** EmptySpine: Immediate.
- **Case** $\rightarrow$Spine: Use Lemma 42 (Inversion of Principal Typing) (1), Lemma 51 (Typing Extension), and Lemma 41 (Extension Weakening for Principal Typing).
- **Case** $\hat{\alpha}$Spine: Show that $\hat{\alpha}_1 \rightarrow \hat{\alpha}_2$ is well-formed, then use the i.h. □

F′ Decidability of Instantiation

Lemma 64 (Left Unsolvedness Preservation)**.**
If $\underbrace{\Gamma_0, \hat{\alpha}, \Gamma_1}_{\Gamma} \vdash \hat{\alpha} := A : \kappa \dashv \Delta$ *and* $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0)$ *then* $\hat{\beta} \in \mathsf{unsolved}(\Delta)$.

Proof. By induction on the given derivation.

- **Case**
$$\frac{\Gamma_0 \vdash \tau : \kappa}{\underbrace{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1}_{\Gamma} \vdash \hat{\alpha} := \tau : \kappa \dashv \underbrace{\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1}_{\Delta}}\ \mathsf{InstSolve}$$

  Immediate, since to the left of $\hat{\alpha}$, the contexts Δ and Γ are the same.
- **Case**
$$\frac{\hat{\beta} \in \mathsf{unsolved}(\Gamma'[\hat{\alpha} : \kappa][\hat{\beta} : \kappa])}{\underbrace{\Gamma'[\hat{\alpha} : \kappa][\hat{\beta} : \kappa]}_{\Gamma} \vdash \hat{\alpha} := \hat{\beta} : \kappa \dashv \underbrace{\Gamma'[\hat{\alpha} : \kappa][\hat{\beta} : \kappa = \hat{\alpha}]}_{\Delta}}\ \mathsf{InstReach}$$

  Immediate, since to the left of $\hat{\alpha}$, the contexts Δ and Γ are the same.

- **Case** $$\dfrac{\Gamma_0, \hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2, \Gamma_1 \vdash \hat{\alpha}_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat{\alpha}_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma_0, \hat{\alpha} : \star, \Gamma_1 \vdash \hat{\alpha} := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \mathsf{InstBin}$$

  We have $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0)$. Therefore $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0, \hat{\alpha}_2 : \star)$.
  Clearly, $\hat{\alpha}_2 \in \mathsf{unsolved}(\Gamma_0, \hat{\alpha}_2 : \star)$.
  We have two subderivations:

  $$\Gamma_0, \hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2, \Gamma_1 \vdash \hat{\alpha}_1 := A_1 : \star \dashv \Theta \qquad (1)$$
  $$\Theta \vdash \hat{\alpha}_2 := [\Theta]A_2 : \star \dashv \Delta \qquad (2)$$

  By induction on (1), $\hat{\beta} \in \mathsf{unsolved}(\Theta)$.
  Also by induction on (1), with $\hat{\alpha}_2$ playing the role of $\hat{\beta}$, we get $\hat{\alpha}_2 \in \mathsf{unsolved}(\Theta)$.
  Since $\hat{\beta} \in \Gamma_0$, it is declared to the left of $\hat{\alpha}_2$ in $\Gamma_0, \hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2, \Gamma_1$.
  Hence by Lemma 20 (Declaration Order Preservation), $\hat{\beta}$ is declared to the left of $\hat{\alpha}_2$ in Θ. That is, $\Theta = (\Theta_0, \hat{\alpha}_2 : \star, \Theta_1)$, where $\hat{\beta} \in \mathsf{unsolved}(\Theta_0)$.
  By induction on (2), $\hat{\beta} \in \mathsf{unsolved}(\Delta)$.

- **Case**

  $$\dfrac{}{\underbrace{\Gamma'[\hat{\alpha} : \mathbb{N}]}_{\Gamma} \vdash \hat{\alpha} := \mathsf{zero} : \mathbb{N} \dashv \underbrace{\Gamma'[\hat{\alpha} : \mathbb{N} = \mathsf{zero}]}_{\Delta}}\ \mathsf{InstZero}$$

  Immediate, since to the left of $\hat{\alpha}$, the contexts Δ and Γ are the same.

- **Case** $$\dfrac{\Gamma[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)] \vdash \hat{\alpha}_1 := t_1 : \mathbb{N} \dashv \Delta}{\Gamma[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \mathsf{InstSucc}$$

  We have $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0)$. Therefore $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0, \hat{\alpha}_1 : \mathbb{N})$. By i.h., $\hat{\beta} \in \mathsf{unsolved}(\Delta)$. ∎

Lemma 65 (Left Free Variable Preservation). *If* $\overbrace{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1}^{\Gamma} \vdash \hat{\alpha} := t : \kappa \dashv \Delta$ *and* $\Gamma \vdash s : \kappa'$ *and* $\hat{\alpha} \notin FV([\Gamma]s)$ *and* $\hat{\beta} \in \mathsf{unsolved}(\Gamma_0)$ *and* $\hat{\beta} \notin FV([\Gamma]s)$*, then* $\hat{\beta} \notin FV([\Delta]s)$*.*

Proof. By induction on the given instantiation derivation.

- **Case** $$\dfrac{\Gamma_0 \vdash \tau : \kappa}{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \vdash \hat{\alpha} := \tau : \kappa \dashv \underbrace{\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1}_{\Delta}}\ \mathsf{InstSolve}$$

  We have $\hat{\alpha} \notin FV([\Gamma]\sigma)$. Since Δ differs from Γ only in $\hat{\alpha}$, it must be the case that $[\Gamma]\sigma = [\Delta]\sigma$. It is given that $\hat{\beta} \notin FV([\Gamma]\sigma)$, so $\hat{\beta} \notin FV([\Delta]\sigma)$.

- **Case** $$\dfrac{\hat{\gamma} \in \mathsf{unsolved}(\Gamma[\hat{\alpha} : \kappa][\hat{\gamma} : \kappa])}{\Gamma[\hat{\alpha} : \kappa][\hat{\gamma} : \kappa] \vdash \hat{\alpha} := \hat{\gamma} : \kappa \dashv \underbrace{\Gamma[\hat{\alpha} : \kappa][\hat{\gamma} : \kappa = \hat{\alpha}]}_{\Delta}}\ \mathsf{InstReach}$$

  Since Δ differs from Γ only in solving $\hat{\gamma}$ to $\hat{\alpha}$, applying Δ to a type will not introduce a $\hat{\beta}$. We have $\hat{\beta} \notin FV([\Gamma]\sigma)$, so $\hat{\beta} \notin FV([\Delta]\sigma)$.

- **Case**

  $$\dfrac{\overbrace{\Gamma[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2]}^{\Gamma'} \vdash \hat{\alpha}_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat{\alpha}_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma[\hat{\alpha} : \star] \vdash \hat{\alpha} := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \mathsf{InstBin}$$

We have $\Gamma \vdash \sigma$ *type* and $\hat{\alpha} \notin FV([\Gamma]\sigma)$ and $\hat{\beta} \notin FV([\Gamma]\sigma)$.
By weakening, we get $\Gamma' \vdash \sigma : \kappa'$; since $\hat{\alpha} \notin FV([\Gamma]\sigma)$ and Γ' only adds a solution for $\hat{\alpha}$, it follows that $[\Gamma']\sigma = [\Gamma]\sigma$.
Therefore $\hat{\alpha}_1 \notin FV([\Gamma']\sigma)$ and $\hat{\alpha}_2 \notin FV([\Gamma']\sigma)$ and $\hat{\beta} \notin FV([\Gamma']\sigma)$.
Since we have $\hat{\beta} \in \Gamma_0$, we also have $\hat{\beta} \in (\Gamma_0, \hat{\alpha}_2 : \star)$.
By induction on the first premise, $\hat{\beta} \notin FV([\Theta]\sigma)$.
Also by induction on the first premise, with $\hat{\alpha}_2$ playing the role of $\hat{\beta}$, we have $\hat{\alpha}_2 \notin FV([\Theta]\sigma)$.
Note that $\hat{\alpha}_2 \in \mathsf{unsolved}(\Gamma_0, \hat{\alpha}_2 : \star)$.
By Lemma 64 (Left Unsolvedness Preservation), $\hat{\alpha}_2 \in \mathsf{unsolved}(\Theta)$.
Therefore Θ has the form $(\Theta_0, \hat{\alpha}_2 : \star, \Theta_1)$.
Since $\hat{\beta} \neq \hat{\alpha}_2$, we know that $\hat{\beta}$ is declared to the left of $\hat{\alpha}_2$ in $(\Gamma_0, \hat{\alpha}_2 : \star)$, so by Lemma 20 (Declaration Order Preservation), $\hat{\beta}$ is declared to the left of $\hat{\alpha}_2$ in Θ. Hence $\hat{\beta} \in \Theta_0$.
Furthermore, by Lemma 43 (Instantiation Extension), we have $\Gamma' \longrightarrow \Theta$.
Then by Lemma 36 (Extension Weakening (Sorts)), we have $\Delta \vdash \sigma : \kappa'$.
Using induction on the second premise, $\hat{\beta} \notin FV([\Delta]\sigma)$.

- **Case**

$$\frac{}{\underbrace{\Gamma'[\hat{\alpha} : \mathbb{N}]}_{\Gamma} \vdash \hat{\alpha} := \mathsf{zero} : \mathbb{N} \dashv \underbrace{\Gamma'[\hat{\alpha} : \mathbb{N} = \mathsf{zero}]}_{\Delta}}\ \mathsf{InstZero}$$

We have $\hat{\alpha} \notin FV([\Gamma]\sigma)$. Since Δ differs from Γ only in $\hat{\alpha}$, it must be the case that $[\Gamma]\sigma = [\Delta]\sigma$. It is given that $\hat{\beta} \notin FV([\Gamma]\sigma)$, so $\hat{\beta} \notin FV([\Delta]\sigma)$.

- **Case**

$$\frac{\overbrace{\Gamma'[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)]}^{\Theta} \vdash \hat{\alpha}_1 := t_1 : \mathbb{N} \dashv \Delta}{\underbrace{\Gamma'[\hat{\alpha} : \mathbb{N}]}_{\Gamma} \vdash \hat{\alpha} := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \mathsf{InstSucc}$$

	$\Gamma \vdash \sigma : \kappa'$	Given
	$\Theta \vdash \sigma : \kappa'$	By weakening
	$\hat{\alpha} \notin FV([\Gamma]\sigma)$	Given
	$\hat{\alpha} \notin FV([\Theta]\sigma)$	$\hat{\alpha} \notin FV([\Gamma]\sigma)$ and Θ only solves $\hat{\alpha}$
	$\Theta = (\Gamma_0, \hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1), \Gamma_1)$	Given
	$\hat{\beta} \notin \mathsf{unsolved}(\Gamma_0)$	Given
	$\hat{\beta} \notin \mathsf{unsolved}(\Gamma_0, \hat{\alpha}_1 : \mathbb{N})$	$\hat{\alpha}_1$ fresh
	$\hat{\beta} \notin FV([\Gamma]\sigma)$	Given
	$\hat{\beta} \notin FV([\Theta]\sigma)$	$\hat{\alpha}_1$ fresh
☞	$\hat{\beta} \notin FV([\Delta]\sigma)$	By i.h.

□

Lemma 66 (Instantiation Size Preservation)**.** *If* $\overbrace{\Gamma_0, \hat{\alpha}, \Gamma_1}^{\Gamma} \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *and* $\Gamma \vdash s : \kappa'$ *and* $\hat{\alpha} \notin FV([\Gamma]s)$*, then* $|[\Gamma]s| = |[\Delta]s|$*, where* $|C|$ *is the plain size of the term* C*.*

Proof. By induction on the given derivation.

- **Case**

$$\frac{\Gamma_0 \vdash \tau : \kappa}{\underbrace{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1}_{\Gamma} \vdash \hat{\alpha} := \tau : \kappa \dashv \underbrace{\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1}_{\Delta}}\ \mathsf{InstSolve}$$

Since Δ differs from Γ only in solving $\hat{\alpha}$, and we know $\hat{\alpha} \notin FV([\Gamma]\sigma)$, we have $[\Delta]\sigma = [\Gamma]\sigma$; therefore $|[\Delta]\sigma = [\Gamma]\sigma|$.

- **Case**

$$\frac{}{\underbrace{\Gamma'[\hat{\alpha} : \mathbb{N}]}_{\Gamma} \vdash \hat{\alpha} := \mathsf{zero} : \mathbb{N} \dashv \underbrace{\Gamma'[\hat{\alpha} : \mathbb{N} = \mathsf{zero}]}_{\Delta}}\ \mathsf{InstZero}$$

Similar to the InstSolve case.

- **Case**

$$\frac{\hat{\beta} \in \mathsf{unsolved}(\Gamma'[\hat{\alpha} : \kappa][\hat{\beta} : \kappa])}{\underbrace{\Gamma'[\hat{\alpha} : \kappa][\hat{\beta} : \kappa]}_{\Gamma} \vdash \hat{\alpha} := \hat{\beta} : \kappa \dashv \underbrace{\Gamma'[\hat{\alpha} : \kappa][\hat{\beta} : \kappa = \hat{\alpha}]}_{\Delta}}\ \mathsf{InstReach}$$

Here, Δ differs from Γ only in solving $\hat{\beta}$ to $\hat{\alpha}$. However, $\hat{\alpha}$ has the same size as $\hat{\beta}$, so even if $\hat{\beta} \in FV([\Gamma]\sigma)$, we have $|[\Delta]\sigma = [\Gamma]\sigma|$.

- **Case**

$$\frac{\overbrace{\Gamma[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2]}^{\Gamma'} \vdash \hat{\alpha}_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat{\alpha}_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma[\hat{\alpha} : \star] \vdash \hat{\alpha} := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \mathsf{InstBin}$$

We have $\Gamma \vdash \sigma : \kappa'$ and $\hat{\alpha} \notin FV([\Gamma]\sigma)$.
Since $\hat{\alpha}_1, \hat{\alpha}_2 \notin \mathsf{dom}(\Gamma)$, we have $\hat{\alpha}, \hat{\alpha}_1, \hat{\alpha}_2 \notin FV([\Gamma]\sigma)$.
By Lemma 23 (Deep Evar Introduction), $\Gamma[\hat{\alpha} : \star] \longrightarrow \Gamma'$.
By Lemma 36 (Extension Weakening (Sorts)), $\Gamma' \vdash \sigma : \kappa'$.
Since $\hat{\alpha} \notin FV(\sigma)$, it follows that $[\Gamma']\sigma = [\Gamma]\sigma$, and so $|[\Gamma']\sigma| = |[\Gamma]\sigma|$.
By induction on the first premise, $|[\Gamma']\sigma| = |[\Theta]\sigma|$.
By Lemma 20 (Declaration Order Preservation), since $\hat{\alpha}_2$ is declared to the left of $\hat{\alpha}_1$ in Γ', we have that $\hat{\alpha}_2$ is declared to the left of $\hat{\alpha}_1$ in Θ.
By Lemma 64 (Left Unsolvedness Preservation), since $\hat{\alpha}_2 \in \mathsf{unsolved}(\Gamma')$, it is unsolved in Θ: that is, $\Theta = (\Theta_0, \hat{\alpha}_2 : \star, \Theta_1)$.
By Lemma 43 (Instantiation Extension), we have $\Gamma' \longrightarrow \Theta$.
By Lemma 36 (Extension Weakening (Sorts)), $\Theta \vdash \sigma : \kappa'$.
Since $\hat{\alpha}_2 \notin FV([\Gamma']\sigma)$, Lemma 65 (Left Free Variable Preservation) gives $\hat{\alpha}_2 \notin FV([\Theta]\sigma)$.
By induction on the second premise, $|[\Theta]\sigma| = |[\Delta]\sigma|$, and by transitivity of equality, $|[\Gamma]\sigma| = |[\Delta]\sigma|$.

- **Case**

$$\frac{\overbrace{\Gamma[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)]}^{\Gamma'} \vdash \hat{\alpha}_1 := t_1 : \mathbb{N} \dashv \Delta}{\Gamma[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \mathsf{InstSucc}$$

$\Gamma[\hat{\alpha} : \star] \vdash \sigma : \kappa'$	Given
$\hat{\alpha} \notin [\Gamma[\hat{\alpha} : \star]]\sigma$	Given
$\Gamma[\hat{\alpha} : \star] \longrightarrow \Gamma'$	By Lemma 23 (Deep Evar Introduction)
$\Gamma' \vdash \sigma : \kappa'$	By Lemma 36 (Extension Weakening (Sorts))
$[\Gamma']\sigma = [\Gamma[\hat{\alpha} : \star]]\sigma$	Since $\hat{\alpha} \notin FV([\Gamma[\hat{\alpha} : \star]]\sigma)$
$|[\Gamma']\sigma| = |[\Gamma[\hat{\alpha} : \star]]\sigma|$	By congruence of equality
$\hat{\alpha}_1 \notin [\Gamma']\sigma$	Since $[\Gamma']\sigma = [\Gamma[\hat{\alpha} : \star]]\sigma$, and $\hat{\alpha}_1 \notin \mathsf{dom}(\Gamma[\hat{\alpha} : \star])$
$|[\Gamma']\sigma| = |[\Theta]\sigma|$	By i.h.
$|[\Gamma[\hat{\alpha} : \star]]\sigma| = |[\Theta]\sigma|$	By transitivity of equality ☐

Lemma 67 (Decidability of Instantiation)**.** *If* $\Gamma = \Gamma_0[\hat{\alpha} : \kappa']$ *and* $\Gamma \vdash t : \kappa$ *such that* $[\Gamma]t = t$ *and* $\hat{\alpha} \notin FV(t)$, *then:*

(1) Either there exists Δ *such that* $\Gamma_0[\hat{\alpha} : \kappa'] \vdash \hat{\alpha} := t : \kappa \dashv \Delta$*, or not.*

Proof. By induction on the derivation of $\Gamma \vdash t : \kappa$.

- **Case** $$\frac{(u : \kappa) \in \Gamma}{\Gamma_L, \hat{\alpha} : \kappa', \Gamma_R \vdash u : \kappa}\ \mathsf{VarSort}$$

  If $\kappa \neq \kappa'$, no rule matches and no derivation exists.
  Otherwise:

  - If $(u : \kappa) \in \Gamma_L$, we can apply rule InstSolve.
  - If u is some unsolved existential variable $\hat{\beta}$ and $(\hat{\beta} : \kappa) \in \Gamma_R$, then we can apply rule InstReach.
  - Otherwise, u is declared in Γ_R and is a universal variable; no rule matches and no derivation exists.

- **Case** $$\frac{(\hat{\beta} : \kappa = \tau) \in \Gamma}{\Gamma \vdash \hat{\beta} : \kappa}\ \mathsf{SolvedVarSort}$$

  By inversion, $(\hat{\beta} : \kappa = \tau) \in \Gamma$, but $[\Gamma]\hat{\beta} = \hat{\beta}$ is given, so this case is impossible.

- **Case** UnitSort:
  If $\kappa' = \star$, then apply rule InstSolve. Otherwise, no rule matches and no derivation exists.

- **Case** $$\frac{\Gamma \vdash \tau_1 : \star \qquad \Gamma \vdash \tau_2 : \star}{\underbrace{\Gamma_L, \hat{\alpha} : \kappa', \Gamma_R}_{\Gamma} \vdash \tau_1 \oplus \tau_2 : \star}\ \mathsf{BinSort}$$

  If $\kappa' \neq \star$, then no rule matches and no derivation exists. Otherwise:
  Given, $[\Gamma](\tau_1 \oplus \tau_2) = \tau_1 \oplus \tau_2$ and $\hat{\alpha} \notin FV([\Gamma](\tau_1 \oplus \tau_2))$.
  If $\Gamma_L \vdash \tau_1 \oplus \tau_2 : \star$, then we have a derivation by InstSolve.
  If not, the only other rule whose conclusion matches $\tau_1 \oplus \tau_2$ is InstBin.
  First, consider whether $\Gamma_L, \hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2, \Gamma_R \vdash \hat{\alpha}_1 := t : \star \dashv -$ is decidable.

  By definition of substitution, $[\Gamma](\tau_1 \oplus \tau_2) = ([\Gamma]\tau_1) \oplus ([\Gamma]\tau_2)$. Since $[\Gamma](\tau_1 \oplus \tau_2) = \tau_1 \oplus \tau_2$, we have $[\Gamma]\tau_1 = \tau_1$ and $[\Gamma]\tau_2 = \tau_2$.
  By weakening, $\Gamma_L, \hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2, \Gamma_R \vdash \tau_1 \oplus \tau_2 : \star$.
  Since $\Gamma \vdash \tau_1 : \star$ and $\Gamma \vdash \tau_2 : \star$, we have $\hat{\alpha}_1, \hat{\alpha}_2 \notin FV(\tau_1) \cup FV(\tau_2)$.
  Since $\hat{\alpha} \notin FV(t) \supseteq FV(\tau_1)$, it follows that $[\Gamma']\tau_1 = \tau_1$.
  By i.h., either there exists Θ s.t. $\Gamma_L, \hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2, \Gamma_R \vdash \hat{\alpha}_1 := \tau_1 : \star \dashv \Theta$, or not.
  If not, then no derivation by InstBin exists.
  Otherwise, there exists such a Θ. By Lemma 64 (Left Unsolvedness Preservation), we have $\hat{\alpha}_2 \in \mathsf{unsolved}(\Theta)$.
  By Lemma 65 (Left Free Variable Preservation), we know that $\hat{\alpha}_2 \notin FV([\Theta]\tau_2)$.
  Substitution is idempotent, so $[\Theta][\Theta]\tau_2 = [\Theta]\tau_2$.
  By i.h., either there exists Δ such that $\Theta \vdash \hat{\alpha}_2 := [\Theta]\tau_2 : \kappa \dashv \Delta$, or not.
  If not, no derivation by InstBin exists.
  Otherwise, there exists such a Δ. By rule InstBin, we have $\Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta$.

- **Case** $$\frac{}{\Gamma \vdash \mathsf{zero} : \mathbb{N}}\ \mathsf{ZeroSort}$$

  If $\kappa' \neq \mathbb{N}$, then no rule matches and no derivation exists. Otherwise, apply rule InstSolve.

- **Case** $\dfrac{\Gamma \vdash t_0 : \mathbb{N}}{\Gamma \vdash \mathsf{succ}(t_0) : \mathbb{N}}$ SuccSort

  If $\kappa' \neq \mathbb{N}$, then no rule matches and no derivation exists. Otherwise:
  If $\Gamma_L \vdash \mathsf{succ}(t_0) : \mathbb{N}$, then we have a derivation by InstSolve.
  If not, the only other rule whose conclusion matches $\mathsf{succ}(t_0)$ is InstSucc.
  The remainder of this case is similar to the BinSort case, but shorter. □

G′ Separation

Lemma 68 (Transitivity of Separation).
If $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R)$ *and* $(\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$
then $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

Proof.

	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R)$	Given
	$(\Gamma_L, \Gamma_R) \longrightarrow (\Theta_L, \Theta_R)$	By Definition 5
	$\Gamma_L \subseteq \Theta_L$ and $\Gamma_R \subseteq \Theta_R$	″
	$(\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	Given
	$(\Theta_L, \Theta_R) \longrightarrow (\Delta_L, \Delta_R)$	By Definition 5
	$\Theta_L \subseteq \Delta_L$ and $\Theta_R \subseteq \Delta_R$	″
	$(\Gamma_L, \Gamma_R) \longrightarrow (\Delta_L, \Delta_R)$	By Lemma 33 (Extension Transitivity)
	$\Gamma_L \subseteq \Delta_L$ and $\Gamma_R \subseteq \Delta_R$	By transitivity of $\subseteq$
☞	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	By Definition 5

□

Lemma 69 (Separation Truncation).
If H *has the form* $\alpha : \kappa$ *or* $\blacktriangleright_{\hat{\alpha}}$ *or* $\blacktriangleright_P$ *or* $x : A\, p$
and $(\Gamma_L * (\Gamma_R, H)) \xrightarrow[*]{} (\Delta_L * \Delta_R)$
then $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_0)$ *where* $\Delta_R = (\Delta_0, H, \Theta)$.

Proof. By induction on Δ_R.

If $\Delta_R = (\ldots, H)$, we have $(\Gamma_L * \Gamma_R, H) \xrightarrow[*]{} (\Delta_L * (\Delta, H))$, and inversion on $\longrightarrow$Uvar (if H is $(\alpha : \kappa)$, or the corresponding rule for other forms) gives the result (with $\Theta = \cdot$).

Otherwise, proceed into the subderivation of $(\Gamma_L, \Gamma_R, \alpha : \kappa) \longrightarrow (\Delta_L, \Delta_R)$, with $\Delta_R = (\Delta_R', \Delta')$ where Δ' is a single declaration. Use the i.h. on Δ_R', producing some Θ'. Finally, let $\Theta = (\Theta', \Delta')$. □

Lemma 70 (Separation for Auxiliary Judgments).

(i) *If* $\Gamma_L * \Gamma_R \vdash \sigma \mathrel{\mathring{=}} \tau : \kappa \dashv \Delta$
and $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(\tau) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(ii) *If* $\Gamma_L * \Gamma_R \vdash P$ *true* $\dashv \Delta$
and $\mathsf{FEV}(P) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(iii) *If* $\Gamma_L * \Gamma_R \;/\; \sigma \mathrel{\mathring{=}} \tau : \kappa \dashv \Delta$
and $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(\tau) = \emptyset$
then $\Delta = (\Delta_L * (\Delta_R, \Theta))$ *and* $(\Gamma_L * (\Gamma_R, \Theta)) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(iv) *If* $\Gamma_L * \Gamma_R \;/\; P \dashv \Delta$
and $\mathsf{FEV}(P) = \emptyset$
then $\Delta = (\Delta_L * (\Delta_R, \Theta))$ *and* $(\Gamma_L * (\Gamma_R, \Theta)) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(v) *If* $\Gamma_L * \Gamma_R \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$
and $(\mathsf{FEV}(\tau) \cup \{\hat{\alpha}\}) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(vi) *If* $\Gamma_L * \Gamma_R \vdash P \equiv Q \dashv \Delta$
and $\mathsf{FEV}(P) \cup \mathsf{FEV}(Q) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(vii) *If* $\Gamma_L * \Gamma_R \vdash A \equiv B \dashv \Delta$
and $\mathsf{FEV}(A) \cup \mathsf{FEV}(B) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

Proof. Part (i): By induction on the derivation of the given checkeq judgment. Cases CheckeqVar, CheckeqUnit and CheckeqZero are immediate ($\Delta_L = \Gamma_L$ and $\Delta_R = \Gamma_R$). For case CheckeqSucc, apply the i.h. For cases CheckeqInstL and CheckeqInstR, use the i.h. (v). For case CheckeqBin, use reasoning similar to that in the $\wedge$I case of Lemma 72 (Separation—Main) (transitivity of separation, and applying Θ in the second premise).

Part (ii), checkprop: Use the i.h. (i).

Part (iii), elimeq: Cases ElimeqUvarRefl, ElimeqUnit and CheckeqZero are immediate ($\Delta_L = \Gamma_L$ and $\Delta_R = \Gamma_R$). Cases ElimeqUvarL$\bot$, ElimeqUvarR$\bot$, ElimeqBinBot and ElimeqClash are impossible (we have Δ, not $\bot$). For case ElimeqSucc, apply the i.h. The case for ElimeqBin is similar to the case CheckeqBin in part (i). For cases ElimeqUvarL and ElimeqUvarR, $\Delta = (\Gamma_L, \Gamma_R, \alpha = \tau)$ which, since $\mathsf{FEV}(\tau) \subseteq \mathsf{dom}(\Gamma_R)$, ensures that $(\Gamma_L * (\Gamma_R, \alpha = \tau)) \xrightarrow[*]{} (\Delta_L * (\Delta_R, \alpha = \tau))$.

Part (iv), elimprop: Use the i.h. (iii).

Part (v), instjudg:

- **Case** InstSolve: Here, $\Gamma = (\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1)$ and $\Delta = (\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1)$. We have $\hat{\alpha} \in \mathsf{dom}(\Gamma_R)$, so the declaration $\hat{\alpha} : \kappa$ is in Γ_R. Since $\mathsf{FEV}(\tau) \subseteq \mathsf{dom}(\Gamma_R)$, the context Δ maintains the separation.

- **Case** InstReach: Here, $\Gamma = \Gamma_0[\hat{\alpha} : \kappa][\hat{\beta} : \kappa]$ and $\Delta = \Gamma_0[\hat{\alpha} : \kappa][\hat{\beta} : \kappa = \hat{\alpha}]$. We have $\hat{\alpha} \in \mathsf{dom}(\Gamma_R)$, so the declaration $\hat{\alpha} : \kappa$ is in Γ_R. Since $\hat{\beta}$ is declared to the right of $\hat{\alpha}$, it too must be in Γ_R, which can also be shown from $\mathsf{FEV}(\hat{\beta}) \subseteq \mathsf{dom}(\Gamma_R)$. Both declarations are in Γ_R, so the context Δ maintains the separation.

- **Case** InstZero: In this rule, Δ is the same as Γ except for a solution zero, which doesn't violate separation.

- **Case** InstSucc: The result follows by i.h., taking care to keep the declaration $\hat{\alpha}_1 : \mathbb{N}$ on the right when applying the i.h., even if $\hat{\alpha} : \mathbb{N}$ is the leftmost declaration in Γ_R, ensuring that $\mathsf{succ}(\hat{\alpha}_1)$ does not violate separation.

- **Case** InstBin: As in the InstSucc case, the new declarations should be kept on the right-hand side of the separator. Otherwise the case is straightforward (using the i.h. twice and transitivity).

Part (vi), propequivjudg: Similar to the CheckeqBin case of part (i), using the i.h. (i).

Part (vii), equivjudg:

- **Cases** $\equiv$Var, $\equiv$Exvar, $\equiv$Unit: Immediate ($\Delta_L = \Gamma_L$ and $\Delta_R = \Gamma_R$).
- **Case** $\equiv\oplus$: Similar to the case CheckeqBin in part (i).
- **Case** $\equiv$Vec: Similar to the case CheckeqBin in part (i).
- **Cases** $\equiv\forall$, $\equiv\exists$: Similar to the case CheckeqBin in part (i).
- **Cases** $\equiv\supset$, $\equiv\wedge$: Similar to the case CheckeqBin in part (i), using the i.h. (vi).
- **Cases** $\equiv$InstantiateL, $\equiv$InstantiateR: Use the i.h. (v). □

Lemma 71 (Separation for Subtyping)**.** *If* $\Gamma_L * \Gamma_R \vdash A <:^{\pm} B \dashv \Delta$
and $\mathsf{FEV}(A) \subseteq \mathsf{dom}(\Gamma_R)$
and $\mathsf{FEV}(B) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

Proof. By induction on the given derivation. In the $<:$Equiv case, use Lemma 70 (Separation for Auxiliary Judgments) (vii). Otherwise, the reasoning needed follows that used in the proof of Lemma 72 (Separation—Main). ☐

Lemma 72 (Separation—Main).

(Spines) *If* $\Gamma_L * \Gamma_R \vdash s : A\ p \gg C\ q \dashv \Delta$
or $\Gamma_L * \Gamma_R \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta$
and $\Gamma_L * \Gamma_R \vdash A\ p$ *type*
and $\mathsf{FEV}(A) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$ *and* $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R)$.

(Checking) *If* $\Gamma_L * \Gamma_R \vdash e \Leftarrow C\ p \dashv \Delta$
and $\Gamma_L * \Gamma_R \vdash C\ p$ *type*
and $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(Synthesis) *If* $\Gamma_L * \Gamma_R \vdash e \Rightarrow A\ p \dashv \Delta$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(Match) *If* $\Gamma_L * \Gamma_R \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$
and $\mathsf{FEV}(\vec{A}) = \emptyset$
and $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

(Match Elim.) *If* $\Gamma_L * \Gamma_R\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$
and $\mathsf{FEV}(P) = \emptyset$
and $\mathsf{FEV}(\vec{A}) = \emptyset$
and $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$
then $\Delta = (\Delta_L * \Delta_R)$ *and* $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

Proof. By induction on the given derivation.

First, the (Match) judgment part, giving only the cases that motivate the side conditions:

- **Case** MatchBase: Here we use the i.h. (Checking), for which we need $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R)$.
- **Case** Match$\wedge$: Here we use the i.h. (Match Elim.), which requires that $\mathsf{FEV}(P) = \emptyset$, which motivates $\mathsf{FEV}(\vec{A}) = \emptyset$.
- **Case** MatchNeg: In its premise, this rule appends a type $A \in \vec{A}$ to Γ_R and claims it is principal ($z : A!$), which motivates $\mathsf{FEV}(\vec{A} = \emptyset)$.

Similarly, (Match Elim.):

- **Case** MatchUnify: Here we use Lemma 70 (Separation for Auxiliary Judgments) (iii), for which we need $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(\tau) = \emptyset$, which motivates $\mathsf{FEV}(P) = \emptyset$.

Now, we show the cases for the (Spine), (Checking), and (Synthesis) parts.

- **Cases** Var, 1I, $\supset$I$\bot$: In all of these rules, the output context is the same as the input context, so just let $\Delta_L = \Gamma_L$ and $\Delta_R = \Gamma_R$.
- **Case**

$$\frac{}{\Gamma_L * \Gamma_R \vdash \cdot : A\ p \gg \underbrace{A}_{C}\ \underbrace{p}_{q} \dashv \Gamma_L * \Gamma_R}\ \mathsf{EmptySpine}$$

Let $\Delta_L = \Gamma_L$ and $\Delta_R = \Gamma_R$.
We have $\mathsf{FEV}(A) \subseteq \mathsf{dom}(\Gamma_R)$. Since $\Delta_R = \Gamma_R$ and $C = A$, it is immediate that $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R)$.

- **Case** $$\dfrac{\Gamma_L * \Gamma_R \vdash e \Rightarrow A\ q \dashv \Theta \qquad \Theta \vdash A <:^{*} B \dashv \Delta}{\Gamma_L * \Gamma_R \vdash e \Leftarrow B\ p \dashv \Delta}\ \mathsf{Sub}$$

  By i.h., $\Theta = (\Theta_L * \Theta_R)$ and $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R)$.
  By Lemma 71 (Separation for Subtyping), $\Delta = (\Delta_L * \Delta_R)$ and $(\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.
  By Lemma 68 (Transitivity of Separation), $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$.

- **Case** $$\dfrac{\Gamma \vdash A\,!\ \mathit{type} \qquad \Gamma \vdash e \Leftarrow [\Gamma]A\ !\dashv \Delta}{\Gamma \vdash (e : A) \Rightarrow [\Delta]A\ ! \dashv \Delta}\ \mathsf{Anno}$$

  By i.h.; since $\mathsf{FEV}(A) = \emptyset$, the condition on the (Checking) part is trivial.

- **Case** $$\dfrac{}{\Gamma[\hat\alpha : \star] \vdash () \Leftarrow \hat\alpha \ \dashv \Gamma[\hat\alpha : \star = 1]}\ 1\mathsf{I}\hat\alpha$$

  Adding a solution with a ground type cannot destroy separation.

- **Case** $$\dfrac{v\ \mathit{chk\text{-}I} \qquad \Gamma_L, \Gamma_R, \alpha : \kappa \vdash v \Leftarrow A_0\ p \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma_L, \Gamma_R \vdash v \Leftarrow \forall \alpha : \kappa.\, A_0\ p \dashv \Delta}\ \forall\mathsf{I}$$

	$\mathsf{FEV}(\forall\alpha : \kappa.\, A_0) \subseteq \mathsf{dom}(\Gamma_R)$	Given
	$\mathsf{FEV}(A_0) \subseteq \mathsf{dom}(\Gamma_R, \alpha : \kappa)$	From definition of FEV
	$(\Delta, \alpha : \kappa, \Theta) = (\Delta_L * \Delta'_R)$	By i.h.
	$(\Gamma_L * (\Gamma_R, \alpha : \kappa)) \xrightarrow[*]{} (\Delta_L * \Delta'_R)$	"
☞	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	By Lemma 69 (Separation Truncation)
	$\Delta'_R = (\Delta_R, \alpha : \kappa, \Theta)$	"
	$(\Delta, \alpha : \kappa, \Theta) = (\Delta_L * \Delta'_R)$	Above
	$= (\Delta_L, \Delta'_R)$	Definition of $*$
	$= (\Delta_L, \Delta_R, \alpha : \kappa, \Theta)$	By above equation
☞	$\Delta = (\Delta_L, \Delta_R)$	α not multiply declared

- **Case** $$\dfrac{\Gamma_L, \Gamma_R, \hat\alpha : \kappa \vdash e\ s : [\hat\alpha/\alpha]A_0 \ \gg C\ q \dashv \Delta}{\Gamma_L, \Gamma_R \vdash e\ s : \forall \alpha : \kappa.\, A_0\ p \gg C\ q \dashv \Delta}\ \forall\mathsf{Spine}$$

	$\mathsf{FEV}(\forall\alpha : \kappa A_0.\,) \subseteq \mathsf{dom}(\Gamma_R)$	Given
	$\mathsf{FEV}([\hat\alpha/\alpha]A_0) \subseteq \mathsf{dom}(\Gamma_R, \hat\alpha : \kappa)$	From definition of FEV
☞	$\Delta = (\Delta_L * \Delta_R)$	By i.h.
	$(\Gamma_L * (\Gamma_R, \hat\alpha : \kappa)) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	"
☞	$\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R)$	"
	$\mathsf{dom}(\Gamma_L) \subseteq \mathsf{dom}(\Delta_L)$	By Definition 5
	$\mathsf{dom}(\Gamma_R, \hat\alpha : \kappa) \subseteq \mathsf{dom}(\Delta_R)$	By Definition 5
	$\mathsf{dom}(\Gamma_R) \cup \{\hat\alpha\} \subseteq \mathsf{dom}(\Delta_R)$	By definition of $\mathsf{dom}(-)$
	$\mathsf{dom}(\Gamma_R) \subseteq \mathsf{dom}(\Delta_R)$	Property of $\subseteq$
	$(\Gamma_L, \Gamma_R) \longrightarrow (\Delta_L, \Delta_R)$	By Lemma 51 (Typing Extension)
☞	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	By Definition 5

- **Case** $$\dfrac{e \text{ not a case} \qquad \Gamma_L * \Gamma_R \vdash P\ \mathit{true} \dashv \Theta \qquad \Theta \vdash e \Leftarrow [\Theta]A_0\ p \dashv \Delta}{\Gamma_L * \Gamma_R \vdash e \Leftarrow (A_0 \wedge P)\ p \dashv \Delta}\ \wedge\mathsf{I}$$

$$
\begin{array}{rl}
\Gamma_L * \Gamma_R \vdash (A_0 \wedge P)\ p\ \mathit{type} & \text{Given} \\
\Gamma_L * \Gamma_R \vdash P\ \mathit{prop} & \text{By inversion} \\
\Gamma_L * \Gamma_R \vdash A_0\ p\ \mathit{type} & \text{By inversion} \\
\mathsf{FEV}(A_0 \wedge P) \subseteq \mathsf{dom}(\Gamma_R) & \text{Given} \\
\mathsf{FEV}(P) \subseteq \mathsf{dom}(\Gamma_R) & \text{By def. of FEV} \\
\mathsf{FEV}(A_0) \subseteq \mathsf{dom}(\Gamma_R) & '' \\
\Theta = (\Theta_L * \Theta_R) & \text{By Lemma 70 (Separation for Auxiliary Judgments) (i)} \\
(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R) & ''
\end{array}
$$

$$
\begin{array}{rl}
\mathsf{FEV}(A_0) \subseteq \mathsf{dom}(\Gamma_R) & \text{Above} \\
\mathsf{dom}(\Gamma_R) \subseteq \mathsf{dom}(\Theta_R) & \text{By Definition 5} \\
\mathsf{FEV}(A_0) \subseteq \mathsf{dom}(\Theta_R) & \text{By previous line} \\
\mathsf{FEV}([\Theta]A_0) \subseteq \mathsf{dom}(\Theta_R) & \text{Previous line and } (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R) \\
\Gamma_L * \Gamma_R \vdash (A_0 \wedge P)\ p\ \mathit{type} & \text{Given} \\
\Gamma_L * \Gamma_R \vdash A_0\ p\ \mathit{type} & \text{By inversion} \\
\Theta \vdash A_0\ p\ \mathit{type} & \text{By Lemma 41 (Extension Weakening for Principal Typing)} \\
\Theta \vdash [\Theta]A_0\ p\ \mathit{type} & \text{By Lemma 13 (Right-Hand Substitution for Typing)} \\
☞\quad \Delta = (\Delta_L * \Delta_R) & \text{By i.h.} \\
(\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & '' \\
☞\quad (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & \text{By Lemma 68 (Transitivity of Separation)}
\end{array}
$$

- **Case** Nil: Similar to a section of the $\wedge$I case.
- **Case** Cons: Similar to the $\wedge$I case, with an extra use of the i.h. for the additional second premise.
- **Case** $$\dfrac{v\ \mathit{chk\text{-}I} \qquad \Gamma_L * (\Gamma_R, \blacktriangleright_P) \ /\ P \dashv \Theta \qquad \Theta \vdash v \Leftarrow [\Theta]A_0\ ! \dashv \Delta, \blacktriangleright_P, \Delta'}{\Gamma_L * \Gamma_R \vdash v \Leftarrow P \supset A_0\ ! \dashv \Delta} \supset\!\mathsf{I}$$

$$
\begin{array}{rl}
\Gamma_L * \Gamma_R \vdash (P \supset A_0)\ !\ \mathit{type} & \text{Given} \\
\Gamma_L * \Gamma_R \vdash P \supset A_0\ \mathit{prop} & \text{By inversion} \\
\mathsf{FEV}(P \supset A_0) = \emptyset & '' \\
\mathsf{FEV}(P) = \emptyset & \text{By def. of FEV}
\end{array}
$$

$$
\begin{array}{rl}
\Gamma_L * (\Gamma_R, \blacktriangleright_P) \ /\ P \dashv \Theta & \text{Subderivation} \\
\Theta = (\Theta_L * (\Theta_R, \Theta_Z)) & \text{By Lemma 70 (Separation for Auxiliary Judgments) (iv)} \\
(\Gamma_L * (\Gamma_R, \blacktriangleright_P, \Theta_Z)) \xrightarrow[*]{} (\Theta_L * (\Theta_R, \Theta_Z)) & ''
\end{array}
$$

$$
\begin{array}{rl}
\Gamma_L * \Gamma_R \vdash (P \supset A_0)\ !\ \mathit{type} & \text{Given} \\
\Gamma_L, \Gamma_R \vdash A_0\ !\ \mathit{type} & \text{By Lemma 42 (Inversion of Principal Typing) (2)} \\
\Gamma_L, \Gamma_R, \blacktriangleright_P, \Theta_Z \vdash A_0\ !\ \mathit{type} & \text{By Lemma 35 (Suffix Weakening)} \\
\Theta \vdash [\Theta]A_0\ !\ \mathit{type} & \text{By Lemmas 41 and 40} \\
\mathsf{FEV}(A_0) = \emptyset & \text{Above and def. of FEV} \\
\mathsf{FEV}(A_0) \subseteq \mathsf{dom}(\Theta_R, \Theta_Z) & \text{Immediate} \\
(\Delta, \blacktriangleright_P, \Delta') = (\Delta_L * \Delta_R') & \text{By i.h.} \\
(\Theta_L * (\Theta_R, \Theta_Z)) \xrightarrow[*]{} (\Delta_L * \Delta_R') & ''
\end{array}
$$

$$
\begin{array}{rl}
(\Gamma_L * (\Gamma_R, \blacktriangleright_P)) \xrightarrow[*]{} (\Delta_L * \Delta_R') & \text{By Lemma 68 (Transitivity of Separation)} \\
☞\quad (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & \text{By Lemma 69 (Separation Truncation)} \\
\Delta_R' = (\Delta_R, \blacktriangleright_P, \dots) & '' \\
☞\quad \Delta = (\Delta_L, \Delta_R) & \text{Similar to the } \forall\mathsf{I} \text{ case}
\end{array}
$$

- **Case** $\dfrac{\Gamma_L * \Gamma_R \vdash P\ \mathit{true} \dashv \Theta \qquad \Theta \vdash e\ s : [\Theta]A_0\ p \gg C\ q \dashv \Delta}{\Gamma_L * \Gamma_R \vdash e\ s : P \supset A_0\ p \gg C\ q \dashv \Delta}\ \supset\mathsf{Spine}$

$$
\begin{array}{rll}
 & \Gamma_L * \Gamma_R \vdash (P \supset A_0)\ p\ \mathit{type} & \text{Given} \\
 & \Gamma_L * \Gamma_R \vdash P\ \mathit{prop} & \text{By inversion} \\
 & \Gamma_L, \Gamma_R \vdash P\ \mathit{true} \dashv \Theta & \text{Subderivation} \\
 & \Theta = (\Theta_L * \Theta_R) & \text{By Lemma 70 (Separation for Auxiliary Judgments) (i)} \\
 & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R) & '' \\
 & \Theta \vdash e\ s : [\Theta]A_0\ p \gg C\ q \dashv \Delta & \text{Subderivation} \\
 & (\Delta, \blacktriangleright_P, \Delta') = (\Delta_L * \Delta_R') & \text{By i.h.} \\
 & (\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R') & '' \\
☞ & \mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R) & '' \\
☞ & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & \text{By Lemma 68 (Transitivity of Separation)}
\end{array}
$$

- **Case** $\dfrac{\Gamma_L, \Gamma_R, x : C\,p \vdash v \Leftarrow C\ p \dashv \Delta, x : C\,p, \Theta}{\Gamma_L, \Gamma_R \vdash \mathsf{rec}\ x.\,v \Leftarrow C\ p \dashv \Delta}\ \mathsf{Rec}$

$$
\begin{array}{rll}
 & \Gamma_L * \Gamma_R \vdash C\ p\ \mathit{type} & \text{Given} \\
 & \mathsf{FEV}(C) \subseteq \mathsf{dom}(\Gamma_R) & \text{Given} \\
 & \Gamma_L * (\Gamma_R, x : C\,p) \vdash C\ p\ \mathit{type} & \text{By weakening and Definition 4} \\
 & \Gamma_L, \Gamma_R, x : C\,p \vdash v \Leftarrow C\ p \dashv \Delta, x : C\,p, \Theta & \text{Subderivation} \\
 & (\Delta, x : C\,p, \Theta) = (\Delta_L, \Delta_R') & \text{By i.h.} \\
 & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R') & '' \\
☞ & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & \text{By Lemma 69 (Separation Truncation)} \\
 & \Delta_R' = (\Delta_R, x : C\,p, \dots) & '' \\
☞ & \Delta = (\Delta_L, \Delta_R) & \text{Similar to the } \forall\mathsf{I} \text{ case}
\end{array}
$$

- **Case** $\dfrac{\Gamma_L, \Gamma_R, x : A\,p \vdash e \Leftarrow B\ p \dashv \Delta, x : A\,p, \Theta}{\Gamma_L, \Gamma_R \vdash \lambda x.\,e \Leftarrow A \to B\ p \dashv \Delta}\ {\to}\mathsf{I}$

$$
\begin{array}{rll}
 & \Gamma_L * \Gamma_R \vdash (A \to B)\ p\ \mathit{type} & \text{Given} \\
 & \Gamma_L * \Gamma_R \vdash B\ p\ \mathit{type} & \text{By inversion} \\
 & \mathsf{FEV}(A \to B) \subseteq \mathsf{dom}(\Gamma_R) & \text{Given} \\
 & \mathsf{FEV}(A) \subseteq \mathsf{dom}(\Gamma_R) & \text{By def. of FEV} \\
 & \Gamma_L * (\Gamma_R, x : A\,p) \vdash B\ p\ \mathit{type} & \text{By weakening and Definition 4} \\
 & \Gamma_L, \Gamma_R, x : A\,p \vdash e \Leftarrow B\ p \dashv \Delta, x : A\,p, \Theta & \text{Subderivation} \\
 & (\Delta, x : A\,p, \Theta) = (\Delta_L, \Delta_R') & \text{By i.h.} \\
 & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R') & '' \\
☞ & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & \text{By Lemma 69 (Separation Truncation)} \\
 & \Delta_R' = (\Delta_R, x : A\,p, \dots) & '' \\
☞ & \Delta = (\Delta_L, \Delta_R) & \text{Similar to the } \forall\mathsf{I} \text{ case}
\end{array}
$$

- **Case** $\dfrac{\Gamma_0[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 {\to} \hat\alpha_2], x : \hat\alpha_1 \vdash e_0 \Leftarrow \hat\alpha_2 \dashv \Delta, x : \hat\alpha_1, \Delta'}{\underbrace{\Gamma_0[\hat\alpha : \star]}_{\Gamma_L * \Gamma_R} \vdash \lambda x.\,e_0 \Leftarrow \hat\alpha \dashv \Delta}\ {\to}\mathsf{I}\hat\alpha$

We have $(\Gamma_L * \Gamma_R) = \Gamma_0[\hat\alpha : \star]$. We also have $\mathsf{FEV}(\hat\alpha) \subseteq \mathsf{dom}(\Gamma_R)$. Therefore $\hat\alpha \in \mathsf{dom}(\Gamma_R)$ and

$$\Gamma_0[\hat\alpha : \star] \quad = \quad \Gamma_L, \Gamma_2, \hat\alpha : \star, \Gamma_3$$

where $\Gamma_R = (\Gamma_2, \hat\alpha : \star, \Gamma_3)$.

Then the input context in the premise has the following form:

$$\Gamma_0[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \to \hat\alpha_2], x : \hat\alpha_1 \quad = \quad \Gamma_L, \Gamma_2, \hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \to \hat\alpha_2, \Gamma_3, x : \hat\alpha_1$$

Let us separate this context at the same point as $\Gamma_0[\hat\alpha : \star]$, that is, after Γ_L and before Γ_2, and call the resulting right-hand context Γ_R'. That is,

$$\Gamma_0[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \to \hat\alpha_2], x : \hat\alpha_1 \quad = \quad \Gamma_L \ * \ \big(\underbrace{\Gamma_2, \hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \to \hat\alpha_2, \Gamma_3, x : \hat\alpha_1}_{\Gamma_R'}\big)$$

	$\mathsf{FEV}(\hat\alpha) \subseteq \mathsf{dom}(\Gamma_R)$	Given
	$\Gamma_L * \Gamma_R' \vdash e_0 \Leftarrow \hat\alpha_2 \dashv \Delta, x : \hat\alpha_1, \Delta'$	Subderivation
	$\Gamma_L * \Gamma_R' \vdash \hat\alpha_2 \not! \ \textit{type}$	$\hat\alpha_2 \in \mathsf{dom}(\Gamma_R')$
	$\mathsf{FEV}(\hat\alpha_2) \subseteq \mathsf{dom}(\Gamma_R')$	$\hat\alpha_2 \in \mathsf{dom}(\Gamma_R')$
	$(\Delta, x : \hat\alpha_1, \Delta') = (\Delta_L, \Delta_R')$	By i.h.
	$(\Gamma_L * \Gamma_R') \xrightarrow[*]{} (\Delta_L * \Delta_R')$	"
☞	$\Delta = (\Delta_L, \Delta_R)$	Similar to the $\forall$I case
☞	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	"

- **Case** $\dfrac{\Gamma \vdash e \Rightarrow A\ p \dashv \Theta \qquad \Theta \vdash s : [\Theta]A\ p \gg C\ \lceil q \rceil \dashv \Delta}{\Gamma \vdash e\,s \Rightarrow C\ q \dashv \Delta}\ \to\mathsf{E}$

  Use the i.h. and Lemma 68 (Transitivity of Separation), with Lemma 89 (Well-formedness of Algorithmic Typing) and Lemma 13 (Right-Hand Substitution for Typing).

- **Case** $\dfrac{\Gamma \vdash s : A\ ! \gg C\ \not!\ \dashv \Delta \qquad \mathsf{FEV}([\Delta]C) = \emptyset}{\Gamma \vdash s : A\ ! \gg C\ \lceil ! \rceil \dashv \Delta}\ \mathsf{SpineRecover}$

  Use the i.h.

- **Case** $\dfrac{\Gamma \vdash s : A\ p \gg C\ q \dashv \Delta \qquad \big((p = \not!) \text{ or } (q = !) \text{ or } (\mathsf{FEV}([\Delta]C) \neq \emptyset)\big)}{\Gamma \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta}\ \mathsf{SpinePass}$

  Use the i.h.

- **Case** $\dfrac{\Gamma_L * \Gamma_R \vdash e \Leftarrow A_1\ p \dashv \Theta \qquad \Theta \vdash s : [\Theta]A_2\ p \gg C\ q \dashv \Delta}{\Gamma_L * \Gamma_R \vdash e\,s : A_1 \to A_2\ p \gg C\ q \dashv \Delta}\ \to\mathsf{Spine}$

	$\Gamma \vdash (A_1 \to A_2)\ p\ \textit{type}$	Given
	$\Gamma \vdash A_1\ \textit{type}$	By inversion
	$\mathsf{FEV}(A_1 \to A_2) \subseteq \mathsf{dom}(\Gamma_R)$	Given
	$\mathsf{FEV}(A_1) \subseteq \mathsf{dom}(\Gamma_R)$	By def. of FEV
	$\Theta = (\Theta_L, \Theta_R)$	By i.h.
	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R)$	"
	$\Gamma \vdash A_2\ \textit{type}$	By inversion
	$\Gamma \vdash [\Theta]A_2\ \textit{type}$	By Lemma 13 (Right-Hand Substitution for Typing)
	$\mathsf{FEV}(A_2) \subseteq \mathsf{dom}(\Gamma_R)$	By def. of FEV
☞	$\Delta = (\Delta_L, \Delta_R)$	By i.h.
	$(\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	"
☞	$\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R)$	"
☞	$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	By Lemma 68 (Transitivity of Separation)

- **Case** $$\dfrac{\Gamma \vdash e \Leftarrow A_k \; p \dashv \Delta}{\Gamma \vdash \mathsf{inj}_k \, e \Leftarrow A_1 + A_2 \; p \dashv \Delta} {+}\mathsf{I}_k$$

  Use the i.h. (inverting $\Gamma \vdash (A_1 + A_2) \; p \; \textit{type}$).

- **Case** $$\dfrac{\Gamma \vdash e_1 \Leftarrow A_1 \; p \dashv \Theta \qquad \Theta \vdash e_2 \Leftarrow [\Theta]A_2 \; p \dashv \Delta}{\Gamma \vdash \langle e_1, e_2 \rangle \Leftarrow A_1 \times A_2 \; p \dashv \Delta} \times\mathsf{I}$$

$\Gamma \vdash (A_1 \times A_2) \; p \; \textit{type}$	Given
$\Gamma \vdash A_1 \; p \; \textit{type}$	By inversion
$\Gamma \vdash e_1 \Leftarrow A_1 \; p \dashv \Theta$	Subderivation
$\Theta = (\Theta_L, \Theta_R)$	By i.h.
$(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Theta_L * \Theta_R)$	$''$
$\Gamma \vdash A_2 \; \textit{type}$	By inversion
$\Gamma \longrightarrow \Theta$	By Lemma 51 (Typing Extension)
$\Theta \vdash A_2 \; \textit{type}$	By Lemma 36 (Extension Weakening (Sorts))
$\Theta \vdash [\Theta]A_2 \; \textit{type}$	By Lemma 13 (Right-Hand Substitution for Typing)
$\Theta \vdash e_2 \Leftarrow [\Theta]A_2 \; p \dashv \Delta$	Subderivation
$\Delta = (\Delta_L, \Delta_R)$	By i.h.
$(\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	$''$
☞ $(\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	By Lemma 68 (Transitivity of Separation)

- **Case** $$\frac{\Gamma[\hat\alpha_2{:}\star, \hat\alpha_1{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \times \hat\alpha_2] \vdash e_1 \Leftarrow \hat\alpha_1 \dashv \Theta \qquad \Theta \vdash e_2 \Leftarrow [\Theta]\hat\alpha_2 \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash \langle e_1, e_2 \rangle \Leftarrow \hat\alpha \dashv \Delta} \times\mathsf{I}\hat\alpha$$

  We have $(\Gamma_L * \Gamma_R) = \Gamma_0[\hat\alpha : \star]$. We also have $\mathsf{FEV}(\hat\alpha) \subseteq \mathsf{dom}(\Gamma_R)$. Therefore $\hat\alpha \in \mathsf{dom}(\Gamma_R)$ and

  $$\Gamma_0[\hat\alpha : \star] \quad = \quad \Gamma_L, \Gamma_2, \hat\alpha : \star, \Gamma_3$$

  where $\Gamma_R = (\Gamma_2, \hat\alpha : \star, \Gamma_3)$.

  Then the input context in the premise has the following form:

  $$\Gamma_0[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \times \hat\alpha_2] \quad = \quad (\Gamma_L, \Gamma_2, \hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \times \hat\alpha_2, \Gamma_3)$$

  Let us separate this context at the same point as $\Gamma_0[\hat\alpha : \star]$, that is, after Γ_L and before Γ_2, and call the resulting right-hand context Γ_R':

  $$\Gamma_0[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \times \hat\alpha_2] \quad = \quad \Gamma_L \ * \ \big(\underbrace{\Gamma_2, \hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \times \hat\alpha_2, \Gamma_3}_{\Gamma_R'}\big)$$

  $$\begin{array}{rll}
  & \mathsf{FEV}(\hat\alpha) \subseteq \mathsf{dom}(\Gamma_R) & \text{Given} \\
  & \Gamma_L * \Gamma_R' \vdash e_1 \Leftarrow \hat\alpha_1 \dashv \Theta & \text{Subderivation} \\
  & \mathsf{FEV}(\hat\alpha_2) \subseteq \mathsf{dom}(\Gamma_R') & \hat\alpha_2 \in \mathsf{dom}(\Gamma_R') \\
  & \Theta = (\Theta_L, \Theta_R) & \text{By i.h.} \\
  & (\Gamma_L * \Gamma_R') \xrightarrow[*]{} (\Theta_L * \Theta_R) & '' \\
  & \Theta \vdash e_2 \Leftarrow [\Theta]\hat\alpha_2 \dashv \Delta & \text{Subderivation} \\
  & \mathsf{dom}(\Gamma_R') \subseteq \mathsf{dom}(\Theta_R) & \text{By Definition 5} \\
  & \mathsf{FEV}(\hat\alpha_2) \subseteq \mathsf{dom}(\Theta_R) & \text{By above} \subseteq \\
  & \mathsf{FEV}([\Theta_R]\hat\alpha_2) \subseteq \mathsf{dom}(\Theta_R) & \text{By Definition 4} \\
  ☞ & \Delta = (\Delta_L, \Delta_R) & \text{By i.h.} \\
  & (\Theta_L * \Theta_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & '' \\
  & & \\
  & \Gamma_R = (\Gamma_2, \hat\alpha : \star, \Gamma_3) & \text{Above} \\
  & \Gamma_R' = (\Gamma_2, \hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha : \star = \hat\alpha_1 \times \hat\alpha_2, \Gamma_3) & \text{Above}
  \end{array}$$

  By Lemma 23 (Deep Evar Introduction) (i), (i), (ii) and the definition of separation, we can show

  $$(\Gamma_L * (\Gamma_2, \hat\alpha : \star, \Gamma_3)) \xrightarrow[*]{} (\Gamma_L * (\Gamma_2, \hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha{:}\star = \hat\alpha_1 \times \hat\alpha_2, \Gamma_3))$$

  $$\begin{array}{rll}
  & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Gamma_L * \Gamma_R') & \text{By above equalities} \\
  ☞ & (\Gamma_L * \Gamma_R) \xrightarrow[*]{} (\Delta_L * \Delta_R) & \text{By Lemma 68 (Transitivity of Separation) twice}
  \end{array}$$

- **Case** $$\frac{\Gamma[\hat\alpha_1 : \star, \hat\alpha_2 : \star, \hat\alpha : \star = \hat\alpha_1 + \hat\alpha_2] \vdash e \Leftarrow \hat\alpha_k \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash \mathsf{inj}_k\, e \Leftarrow \hat\alpha \dashv \Delta} +\mathsf{I}\hat\alpha_k$$

  Similar to the $\times\mathsf{I}\hat\alpha$ case, but simpler.

- **Case** $$\frac{\Gamma[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \to \hat\alpha_2] \vdash e\, s_0 : (\hat\alpha_1 \to \hat\alpha_2) \gg C \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash e\, s_0 : \hat\alpha \gg C \dashv \Delta} \hat\alpha\mathsf{Spine}$$

  Similar to the $\times\mathsf{I}\hat\alpha$ and $+\mathsf{I}\hat\alpha_k$ cases, except that (because we're in the spine part of the lemma) we have to show that $\mathsf{FEV}(C) \subseteq \mathsf{dom}(\Delta_R)$. But we have the same C in the premise and conclusion, so we get that by applying the i.h.

- **Case** $$\frac{\Gamma \vdash e \Rightarrow A\ ! \dashv \Theta \qquad \Theta \vdash \Pi :: A \Leftarrow [\Theta]C\ p \dashv \Delta \qquad \Delta \vdash \Pi\ \textit{covers}\ [\Delta]A}{\Gamma \vdash \mathsf{case}(e, \Pi) \Leftarrow C\ p \dashv \Delta} \mathsf{Case}$$

  Use the i.h. and Lemma 68 (Transitivity of Separation). □

$\mathbf{H'}$ Decidability of Algorithmic Subtyping

$\mathbf{H'.1}$ Lemmas for Decidability of Subtyping

Lemma 73 (Substitution Isn't Large).
For all contexts Θ, *we have* $\#\mathsf{large}([\Theta]A) = \#\mathsf{large}(A)$.

Proof. By induction on A, following the definition of substitution. □

Lemma 74 (Instantiation Solves).
If $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *and* $[\Gamma]\tau = \tau$ *and* $\hat{\alpha} \notin FV([\Gamma]\tau)$ *then* $|\mathsf{unsolved}(\Gamma)| = |\mathsf{unsolved}(\Delta)| + 1$.

Proof. By induction on the given derivation.

- **Case** $$\dfrac{\Gamma_L \vdash \tau : \kappa}{\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R \vdash \hat{\alpha} := \tau : \kappa \dashv \Gamma_L, \hat{\alpha} : \kappa = \tau, \Gamma_R}\ \mathsf{InstSolve}$$

  It is evident that $|\mathsf{unsolved}(\Gamma_L, \hat{\alpha} : \kappa, \Gamma_R)| = |\mathsf{unsolved}(\Gamma_L, \hat{\alpha} : \kappa = \tau, \Gamma_R)| + 1$.

- **Case** $$\dfrac{\hat{\beta} \in \mathsf{unsolved}(\Gamma[\hat{\alpha} : \kappa][\hat{\beta} : \kappa])}{\Gamma[\hat{\alpha} : \kappa][\hat{\beta} : \kappa] \vdash \hat{\alpha} := \underbrace{\hat{\beta}}_{\tau} : \kappa \dashv \Gamma[\hat{\alpha} : \kappa][\hat{\beta} : \kappa = \hat{\alpha}]}\ \mathsf{InstReach}$$

  Similar to the previous case.

- **Case** $$\dfrac{\Gamma_0[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 \oplus \hat{\alpha}_2] \vdash \hat{\alpha}_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat{\alpha}_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma_0[\hat{\alpha} : \star] \vdash \hat{\alpha} := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \mathsf{InstBin}$$

  $$\begin{array}{rrcll} & |\mathsf{unsolved}(\Gamma_0[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2])| &=& |\mathsf{unsolved}(\Gamma_0[\hat{\alpha}])| + 1 & \text{Immediate} \\ & |\mathsf{unsolved}(\Gamma_0[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2])| &=& |\mathsf{unsolved}(\Theta)| + 1 & \text{By i.h.} \\ & |\mathsf{unsolved}(\Gamma)| &=& |\mathsf{unsolved}(\Theta)| & \text{Subtracting 1} \\ ☞ & &=& |\mathsf{unsolved}(\Delta)| + 1 & \text{By i.h.} \end{array}$$

- **Case** $$\dfrac{}{\Gamma[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{zero} : \mathbb{N} \dashv \Gamma[\hat{\alpha} : \mathbb{N} = \mathsf{zero}]}\ \mathsf{InstZero}$$

  Similar to the InstSolve case.

- **Case** $$\dfrac{\Gamma_0[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)] \vdash \hat{\alpha}_1 := t_1 : \mathbb{N} \dashv \Delta}{\Gamma_0[\hat{\alpha} : \mathbb{N}] \vdash \hat{\alpha} := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \mathsf{InstSucc}$$

  $$\begin{array}{rrcll} & |\mathsf{unsolved}(\Delta)| + 1 &=& |\mathsf{unsolved}(\Gamma_0[\hat{\alpha}_1 : \mathbb{N}, \hat{\alpha} : \mathbb{N} = \mathsf{succ}(\hat{\alpha}_1)])| & \text{By i.h.} \\ ☞ & &=& |\mathsf{unsolved}(\Gamma_0[\hat{\alpha} : \mathbb{N}])| & \text{By definition of } \mathsf{unsolved}(-) \end{array}$$

□

Lemma 75 (Checkeq Solving). *If* $\Gamma \vdash s \mathrel{\dot=} t : \kappa \dashv \Delta$ *then either* $\Delta = \Gamma$ *or* $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

Proof. By induction on the given derivation.

- **Case** $$\dfrac{}{\Gamma \vdash u \mathrel{\dot=} u : \kappa \dashv \underbrace{\Gamma}_{\Delta}}\ \mathsf{CheckeqVar}$$

  Here $\Delta = \Gamma$.

- **Cases** CheckeqUnit, CheckeqZero: Similar to the CheckeqVar case.

- **Case** $$\dfrac{\Gamma \vdash \sigma \mathrel{\mathring{=}} t : \mathbb{N} \dashv \Delta}{\Gamma \vdash \mathsf{succ}(\sigma) \mathrel{\mathring{=}} \mathsf{succ}(t) : \mathbb{N} \dashv \Delta}\ \mathsf{CheckeqSucc}$$

  Follows by i.h.

- **Case** $$\dfrac{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := t : \kappa \dashv \Delta \qquad \hat{\alpha} \notin FV(t)}{\underbrace{\Gamma_0[\hat{\alpha}]}_{\Gamma} \vdash \hat{\alpha} \mathrel{\mathring{=}} t : \kappa \dashv \Delta}\ \mathsf{CheckeqInstL}$$

	$\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := t : \kappa \dashv \Delta$	Subderivation
	$\Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta$	$\Gamma = \Gamma_0[\hat{\alpha}]$
	$\Delta = \Gamma$ or $\lvert\mathsf{unsolved}(\Delta)\rvert = \lvert\mathsf{unsolved}(\Gamma)\rvert - 1$	By Lemma 74 (Instantiation Solves)
☞	$\Delta = \Gamma$ or $\lvert\mathsf{unsolved}(\Delta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$	

- **Case** $$\dfrac{\Gamma[\hat{\alpha} : \kappa] \vdash \hat{\alpha} := t : \kappa \dashv \Delta \qquad \hat{\alpha} \notin FV(t)}{\Gamma[\hat{\alpha} : \kappa] \vdash t \mathrel{\mathring{=}} \hat{\alpha} : \kappa \dashv \Delta}\ \mathsf{CheckeqInstR}$$

  Similar to the CheckeqInstL case.

- **Case** $$\dfrac{\Gamma \vdash \sigma_1 \mathrel{\mathring{=}} \tau_1 : \star \dashv \Theta \qquad \Theta \vdash [\Theta]\sigma_2 \mathrel{\mathring{=}} [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma \vdash \underbrace{\sigma_1 \oplus \sigma_2}_{\sigma} \mathrel{\mathring{=}} \underbrace{\tau_1 \oplus \tau_2}_{t} : \star \dashv \Delta}\ \mathsf{CheckeqBin}$$

$\Gamma \vdash \sigma_1 \mathrel{\mathring{=}} \tau_1 : \star \dashv \Theta$	Subderivation
$\Theta = \Gamma$ or $\lvert\mathsf{unsolved}(\Theta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$	By i.h.

  - $\Theta = \Gamma$:

	$\Theta \vdash [\Theta]\sigma_2 \mathrel{\mathring{=}} [\Theta]\tau_2 : \star \dashv \Delta$	Subderivation
	$\Gamma \vdash [\Gamma]\sigma_2 \mathrel{\mathring{=}} [\Gamma]\tau_2 : \star \dashv \Delta$	By $\Theta = \Gamma$
☞	$\Delta = \Gamma$ or $\lvert\mathsf{unsolved}(\Gamma)\rvert = \lvert\mathsf{unsolved}(\Delta)\rvert + 1$	By i.h.

  - $\lvert\mathsf{unsolved}(\Theta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$:

$\Theta \vdash [\Theta]\sigma_2 \mathrel{\mathring{=}} [\Theta]\tau_2 : \star \dashv \Delta$	Subderivation
$\Delta = \Theta$ or $\lvert\mathsf{unsolved}(\Delta)\rvert < \lvert\mathsf{unsolved}(\Theta)\rvert$	By i.h.

    If $\Delta = \Theta$ then substituting Δ for Θ in $\lvert\mathsf{unsolved}(\Theta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$ gives $\lvert\mathsf{unsolved}(\Delta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$.
    If $\lvert\mathsf{unsolved}(\Delta)\rvert < \lvert\mathsf{unsolved}(\Theta)\rvert$ then transitivity of $<$ gives $\lvert\mathsf{unsolved}(\Delta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$. ☐

Lemma 76 (Prop Equiv Solving)**.**
If $\Gamma \vdash P \equiv Q \dashv \Delta$ *then either* $\Delta = \Gamma$ *or* $\lvert\mathsf{unsolved}(\Delta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$.

Proof. Only one rule can derive the judgment:

- **Case** $$\dfrac{\Gamma \vdash \sigma_1 \mathrel{\mathring{=}} t_1 : \mathbb{N} \dashv \Theta \qquad \Theta \vdash [\Theta]\sigma 2 \mathrel{\mathring{=}} [\Theta]t_2 : \mathbb{N} \dashv \Delta}{\Gamma \vdash (\sigma_1 = \sigma 2) \equiv (t_1 = t_2) \dashv \Delta}\ \mathsf{\equiv PropEq}$$

  By Lemma 75 (Checkeq Solving) on the first premise,
  either $\Theta = \Gamma$ or $\lvert\mathsf{unsolved}(\Theta)\rvert < \lvert\mathsf{unsolved}(\Gamma)\rvert$.

  In the former case, the result follows from Lemma 75 (Checkeq Solving) on the second premise.

In the latter case, applying Lemma 75 (Checkeq Solving) to the second premise either gives $\Delta = \Theta$, and therefore

$$|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$$

or gives $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Theta)|$, which also leads to $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$. □

Lemma 77 (Equiv Solving).
If $\Gamma \vdash A \equiv B \dashv \Delta$ *then either* $\Delta = \Gamma$ *or* $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

Proof. By induction on the given derivation.

- **Case**
$$\frac{}{\Gamma \vdash \alpha \equiv \alpha \dashv \Gamma} \equiv\mathsf{Var}$$

  Here $\Delta = \Gamma$.

- **Cases** $\equiv$Exvar, $\equiv$Unit: Similar to the $\equiv$Var case.

- **Case**
$$\frac{\Gamma \vdash A_1 \equiv B_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta}{\Gamma \vdash (A_1 \oplus A_2) \equiv (B_1 \oplus B_2) \dashv \Delta} \equiv\oplus$$

  By i.h., either $\Theta = \Gamma$ or $|\mathsf{unsolved}(\Theta)| < |\mathsf{unsolved}(\Gamma)|$.

  In the former case, apply the i.h. to the second premise. Now either $\Delta = \Theta$—and therefore $\Delta = \Gamma$—or $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Theta)|$. Since $\Theta = \Gamma$, we have $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

  In the latter case, we have $|\mathsf{unsolved}(\Theta)| < |\mathsf{unsolved}(\Gamma)|$. By i.h. on the second premise, either $\Delta = \Theta$, and substituting Δ for Θ gives $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$—or $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Theta)|$, which combined with $|\mathsf{unsolved}(\Theta)| < |\mathsf{unsolved}(\Gamma)|$ gives $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

- **Case** $\equiv$Vec: Similar to the $\equiv\oplus$ case.

- **Case**
$$\frac{\Gamma, \alpha : \kappa \vdash A_0 \equiv B_0 \dashv \Delta, \alpha : \kappa, \Delta'}{\Gamma \vdash \forall\alpha : \kappa.\, A_0 \equiv \forall\alpha : \kappa.\, B_0 \dashv \Delta} \equiv\forall$$

  By i.h., either $(\Delta, \alpha : \kappa, \Delta') = (\Gamma, \alpha : \kappa)$, or $|\mathsf{unsolved}(\Delta, \alpha : \kappa, \Delta')| < |\mathsf{unsolved}(\Gamma, \alpha : \kappa)|$.

  In the former case, Lemma 22 (Extension Inversion) (i) tells us that $\Delta' = \cdot$. Thus, $(\Delta, \alpha : \kappa) = (\Gamma, \alpha : \kappa)$, and so $\Delta = \Gamma$.

  In the latter case, we have $|\mathsf{unsolved}(\Delta, \alpha : \kappa, \Delta')| < |\mathsf{unsolved}(\Gamma, \alpha : \kappa)|$, that is:

$$|\mathsf{unsolved}(\Delta)| + 0 + |\mathsf{unsolved}(\Delta')| \quad < \quad |\mathsf{unsolved}(\Gamma)| + 0$$

  Since $|\mathsf{unsolved}(\Delta')|$ cannot be negative, we have $|\mathsf{unsolved}(\Delta)| < |\mathsf{unsolved}(\Gamma)|$.

- **Case**
$$\frac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta}{\Gamma \vdash P \supset A_0 \equiv Q \supset B_0 \dashv \Delta} \equiv\supset$$

  Similar to the $\equiv\oplus$ case, but using Lemma 76 (Prop Equiv Solving) on the first premise instead of the i.h.

- **Case**
$$\frac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta}{\Gamma \vdash A_0 \wedge P \equiv B_0 \wedge Q \dashv \Delta} \equiv\wedge$$

  Similar to the $\equiv\wedge$ case.

- **Case** $$\dfrac{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta \qquad \hat{\alpha} \notin FV(\tau)}{\underbrace{\Gamma_0[\hat{\alpha}]}_{\Gamma} \vdash \hat{\alpha} \equiv \tau \dashv \Delta} \equiv\mathsf{InstantiateL}$$

  By Lemma 74 (Instantiation Solves), $|\mathsf{unsolved}(\Delta)| = |\mathsf{unsolved}(\Gamma)| - 1$.

- **Case** $$\dfrac{\Gamma_0[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta \qquad \hat{\alpha} \notin FV(\tau)}{\Gamma_0[\hat{\alpha}] \vdash \tau \equiv \hat{\alpha} \dashv \Delta} \equiv\mathsf{InstantiateR}$$

  Similar to the $\equiv\mathsf{InstantiateL}$ case. ☐

Lemma 78 (Decidability of Propositional Judgments)**.**
The following judgments are decidable, with Δ as output in (1)–(3), and $\Delta^{\perp}$ as output in (4) and (5).

We assume $\sigma = [\Gamma]\sigma$ and $t = [\Gamma]t$ in (1) and (4). Similarly, in the other parts we assume $P = [\Gamma]P$ and (in part (3)) $Q = [\Gamma]Q$.

(1) $\Gamma \vdash \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta$

(2) $\Gamma \vdash P\ \mathit{true} \dashv \Delta$

(3) $\Gamma \vdash P \equiv Q \dashv \Delta$

(4) $\Gamma \ /\ \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta^{\perp}$

(5) $\Gamma \ /\ P \dashv \Delta^{\perp}$

Proof. Since there is no mutual recursion between the judgments, we can prove their decidability in order, separately.

(1) *Decidability of* $\Gamma \vdash \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta$: By induction on the sizes of σ and t.

  - **Cases** CheckeqVar, CheckeqUnit, CheckeqZero: No premises.
  - **Case** CheckeqSucc: Both σ and t get smaller in the premise.
  - **Cases** CheckeqInstL, CheckeqInstR: Follows from Lemma 67 (Decidability of Instantiation).

(2) *Decidability of* $\Gamma \vdash P\ \mathit{true} \dashv \Delta$: By induction on σ and t. But we have only one rule deriving this judgment form, CheckpropEq, which has the judgment in (1) as a premise, so decidability follows from part (1).

(3) *Decidability of* $\Gamma \vdash P \equiv Q \dashv \Delta$: By induction on P and Q. But we have only one rule deriving this judgment form, $\equiv$PropEq, which has two premises of the form (1), so decidability follows from part (1).

(4) *Decidability of* $\Gamma \ /\ \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta^{\perp}$: By lexicographic induction, first on the number of unsolved variables (both universal and existential) in Γ, then on σ and t. We also show that the number of unsolved variables is nonincreasing in the output context (if it exists).

  - **Cases** ElimeqUvarRefl, ElimeqZero: No premises, and the output is the same as the input.
  - **Case** ElimeqClash: The only premise is the clash judgment, which is clearly decidable. There is no output.
  - **Case** ElimeqBin: In the first premise, we have the same Γ but both σ and t are smaller. By i.h., the first premise is decidable; moreover, either some variables in Θ were solved, or no additional variables were solved.

    If some variables in Θ were solved, the second premise is smaller than the conclusion according to our lexicographic measure, so by i.h., the second premise is decidable.

    If no additional variables were solved, then $\Theta = \Gamma$. Therefore $[\Theta]\tau_2 = [\Gamma]\tau_2$. It is given that $\sigma = [\Gamma]\sigma$ and $t = [\Gamma]t$, so $[\Gamma]\tau_2 = \tau_2$. Likewise, $[\Theta]\tau_2' = [\Gamma]\tau_2' = \tau_2'$, so we aremaking a recursive call on a strictly smaller subterm.

    Regardless, $\Delta^{\perp}$ is either $\perp$, or is a Δ which has no more unsolved variables than Θ, which in turn has no more unsolved variables than Γ.

- **Case** ElimeqBinBot:
  The premise is invoked on subterms, and does not yield an output context.
- **Case** ElimeqSucc: Both σ and t get smaller. By i.h., the output context has fewer unsolved variables, if it exists.
- **Cases** ElimeqInstL, ElimeqInstR: Follows from Lemma 67 (Decidability of Instantiation). Furthermore, by Lemma 74 (Instantiation Solves), instantiation solves a variable in the output.
- **Cases** ElimeqUvarL, ElimeqUvarR: These rules have no nontrivial premises, and α is solved in the output context.
- **Cases** ElimeqUvarL$\bot$, ElimeqUvarR$\bot$: These rules have no nontrivial premises, and produce the output context $\bot$.

(5) *Decidability of* $\Gamma \;/\; P \dashv \Delta^{\bot}$: By induction on P. But we have only one rule deriving this judgment form, ElimpropEq, for which decidability follows from part (4). □

Lemma 79 (Decidability of Equivalence).
Given a context Γ *and types* A, B *such that* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type and* $[\Gamma]A = A$ *and* $[\Gamma]B = B$*, it is decidable whether there exists* Δ *such that* $\Gamma \vdash A \equiv B \dashv \Delta$.

Proof. Let the judgment $\Gamma \vdash A \equiv B \dashv \Delta$ be measured lexicographically by

(E1) $\#\mathsf{large}(A) + \#\mathsf{large}(B)$;

(E2) $|\mathsf{unsolved}(\Gamma)|$, the number of unsolved existential variables in Γ;

(E3) $|A| + |B|$.

- **Cases** $\equiv$Var, $\equiv$Exvar, $\equiv$Unit: No premises.

- **Case** $$\dfrac{\Gamma \vdash A_1 \equiv B_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta}{\Gamma \vdash A_1 \oplus A_2 \equiv B_1 \oplus B_2 \dashv \Delta}\equiv\oplus$$

  In the first premise, part (E1) either gets smaller (if A_2 or B_2 have large connectives) or stays the same. Since the first premise has the same input context, part (E2) remains the same. However, part (E3) gets smaller.

  In the second premise, part (E1) either gets smaller (if A_1 or B_1 have large connectives) or stays the same.

- **Case** $\equiv$Vec: Similar to a special case of $\equiv\oplus$, where two of the types are monotypes.

- **Case** $$\dfrac{\Gamma, \alpha : \kappa \vdash A_0 \equiv B_0 \dashv \Delta, \alpha : \kappa, \Delta'}{\Gamma \vdash \underbrace{\forall \alpha : \kappa.\, A_0}_{A} \equiv \underbrace{\forall \alpha : \kappa.\, B_0}_{B} \dashv \Delta}\equiv\forall$$

  Since $\#\mathsf{large}(A_0) + \#\mathsf{large}(B_0) = \#\mathsf{large}(A) + \#\mathsf{large}(B) - 2$, the first part of the measure gets smaller.

- **Case** $$\dfrac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta}{\Gamma \vdash \underbrace{P \supset A_0}_{A} \equiv \underbrace{Q \supset B_0}_{B} \dashv \Delta}\equiv\supset$$

  The first premise is decidable by Lemma 78 (Decidability of Propositional Judgments) (3).

  For the second premise, by Lemma 73 (Substitution Isn't Large), $\#\mathsf{large}([\Theta]A_0) = \#\mathsf{large}(A_0)$ and $\#\mathsf{large}([\Theta]B_0) = \#\mathsf{large}(B_0)$. Since $\#\mathsf{large}(A) = \#\mathsf{large}(A_0) + 1$ and $\#\mathsf{large}(B) = \#\mathsf{large}(B_0) + 1$, we have
  $$\#\mathsf{large}([\Theta]A_0) + \#\mathsf{large}([\Theta]B_0) \;<\; \#\mathsf{large}(A) + \#\mathsf{large}(B)$$
  which makes the first part of the measure smaller.

- **Case** $\dfrac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta}{\Gamma \vdash A_0 \wedge P \equiv B_0 \wedge Q \dashv \Delta} \equiv\wedge$

  Similar to the $\equiv\supset$ case.

- **Case** $\dfrac{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} := \tau : \star \dashv \Delta \qquad \hat{\alpha} \notin FV(\tau)}{\Gamma[\hat{\alpha}] \vdash \hat{\alpha} \equiv \tau \dashv \Delta} \equiv\mathsf{InstantiateL}$

  Follows from Lemma 67 (Decidability of Instantiation).

- **Case** $\equiv\mathsf{InstantiateR}$: Similar to the $\equiv\mathsf{InstantiateL}$ case. ☐

H′.2 Decidability of Subtyping

Theorem 1 (Decidability of Subtyping).
Given a context Γ and types A, B such that $\Gamma \vdash A$ type and $\Gamma \vdash B$ type and $[\Gamma]A = A$ and $[\Gamma]B = B$, it is decidable whether there exists Δ such that $\Gamma \vdash A <:^{\pm} B \dashv \Delta$.

Proof. Let the judgments be measured lexicographically by $\#\mathsf{large}(A) + \#\mathsf{large}(B)$.

For each subtyping rule, we show that every premise is smaller than the conclusion, or already known to be decidable. The condition that $[\Gamma]A = A$ and $[\Gamma]B = B$ is easily satisfied at each inductive step, using the definition of substitution.

Now, we consider the rules deriving $\Gamma \vdash A <:^{\pm} B \dashv \Delta$.

- **Case** $\dfrac{\begin{array}{l} A \text{ not headed by } \forall/\exists \\ B \text{ not headed by } \forall/\exists \qquad \Gamma \vdash A \equiv B \dashv \Delta \end{array}}{\Gamma \vdash A <:^{\pm} B \dashv \Delta} <:\mathsf{Equiv}$

  In this case, we appeal to Lemma 79 (Decidability of Equivalence).

- **Case** $\dfrac{\begin{array}{l} B \text{ not headed by } \forall \\ \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa \vdash [\hat{\alpha}/\alpha]A <:^{-} B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta \end{array}}{\Gamma \vdash \forall\alpha : \kappa.\, A <:^{-} B \dashv \Delta} <:\forall\mathsf{L}$

  The premise has one fewer quantifier.

- **Case** $\dfrac{\Gamma, \beta : \kappa \vdash A <:^{-} B \dashv \Delta, \beta : \kappa, \Theta}{\Gamma \vdash A <:^{-} \forall\beta : \kappa.\, B \dashv \Delta} <:\forall\mathsf{R}$

  The premise has one fewer quantifier.

- **Case** $\dfrac{\Gamma, \alpha : \kappa \vdash A <:^{+} B \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma \vdash \exists\alpha : \kappa.\, A <:^{+} B \dashv \Delta} <:\exists\mathsf{L}$

  The premise has one fewer quantifier.

- **Case** $\dfrac{\begin{array}{l} A \text{ not headed by } \exists \\ \Gamma, \blacktriangleright_{\hat{\beta}}, \hat{\beta} : \kappa \vdash A <:^{+} [\hat{\beta}/\beta]B \dashv \Delta, \blacktriangleright_{\hat{\beta}}, \Theta \end{array}}{\Gamma \vdash A <:^{+} \exists\beta : \kappa.\, B \dashv \Delta} <:\exists\mathsf{R}$

  The premise has one fewer quantifier.

- **Case**

$$\dfrac{\Gamma \vdash A <:^{-} B \dashv \Delta \qquad \begin{array}{l} \mathit{neg}(A) \\ \mathit{nonpos}(B) \end{array}}{\Gamma \vdash A <:^{+} B \dashv \Delta} <:^{-}_{+}\mathsf{L}$$

  Consider whether B is negative.

  - Case $\mathit{neg}(B)$:

$$\begin{array}{rl} B = \forall\beta : \kappa.\, B' & \text{Definition of } \mathit{neg}(B) \\ \Gamma, \beta : \kappa \vdash A <:^{-} B' \dashv \Delta, \beta : \kappa, \Theta & \text{Inversion on the premise} \end{array}$$

    There is one fewer quantifier in the subderivation.

  - Case $\mathit{nonneg}(B)$:
    In this case, B is not headed by a $\forall$.

$$\begin{array}{rl} A = \forall\alpha : \kappa.\, A' & \text{Definition of } \mathit{neg}(A) \\ \Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \vdash [\hat\alpha/\alpha]A' <:^{-}\ ' \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Theta & \text{Inversion on the premise} \end{array}$$

  There is one fewer quantifier in the subderivation.

- **Case**

$$\dfrac{\Gamma \vdash A <:^{-} B \dashv \Delta \qquad \begin{array}{l} \mathit{nonpos}(A) \\ \mathit{neg}(B) \end{array}}{\Gamma \vdash A <:^{+} B \dashv \Delta} <:^{-}_{+}\mathsf{R}$$

$$\begin{array}{rl} B = \forall\beta : \kappa.\, B' & \text{Definition of } \mathit{neg}(B) \\ \Gamma, \beta : \kappa \vdash A <:^{-} B' \dashv \Delta, \beta : \kappa, \Theta & \text{Inversion on the premise} \end{array}$$

  There is one fewer quantifier in the subderivation.

- **Case**

$$\dfrac{\Gamma \vdash A <:^{+} B \dashv \Delta \qquad \begin{array}{l} \mathit{pos}(A) \\ \mathit{nonneg}(B) \end{array}}{\Gamma \vdash A <:^{-} B \dashv \Delta} <:^{+}_{-}\mathsf{L}$$

  This case is similar to the $<:^{-}_{+}\mathsf{R}$ case.

- **Case**

$$\dfrac{\Gamma \vdash A <:^{+} B \dashv \Delta \qquad \begin{array}{l} \mathit{nonneg}(A) \\ \mathit{pos}(B) \end{array}}{\Gamma \vdash A <:^{-} B \dashv \Delta} <:^{+}_{-}\mathsf{R}$$

  This case is similar to the $<:^{-}_{+}\mathsf{L}$ case.

□

H′.3 Decidability of Matching and Coverage

Lemma 80 (Decidability of Expansion Judgments).
Given branches Π*, it is decidable whether:*

(1) there exists Π' *such that* $\Pi \overset{\times}{\rightsquigarrow} \Pi'$*;*

(2) there exist Π_L *and* Π_R *such that* $\Pi \overset{+}{\rightsquigarrow} \Pi_L \parallel \Pi_R$*;*

(3) there exists Π' *such that* $\Pi \overset{\text{var}}{\rightsquigarrow} \Pi'$*;*

(4) there exists Π' *such that* $\Pi \overset{1}{\rightsquigarrow} \Pi'$*.*

Proof. In each part, by induction on Π: Every rule either has no premises, or breaks down Π in its nontrivial premise. □

Theorem 2 (Decidability of Coverage).
Given a context Γ, branches Π and types $\vec{A}$, it is decidable whether $\Gamma \vdash \Pi$ covers $\vec{A}$ is derivable.

Proof. By induction on, lexicographically, (1) the number of $\wedge$ connectives appearing in $\vec{A}$, and then (2) the size of $\vec{A}$, considered to be the sum of the sizes $|A|$ of each type A in $\vec{A}$.

(For CoversVar, Covers×, and Covers+, we also use the appropriate part of Lemma 80 (Decidability of Expansion Judgments).)

- **Case** CoversEmpty: No premises.
- **Case** CoversVar: The number of $\wedge$ connectives does not grow, and $\vec{A}$ gets smaller.
- **Case** Covers1: The number of $\wedge$ connectives does not grow, and $\vec{A}$ gets smaller.
- **Case** Covers×: The number of $\wedge$ connectives does not grow, and $\vec{A}$ gets smaller, since $|A_1| + |A_2| < |A_1 \times A_2|$.
- **Case** Covers+: Here we have $\vec{A} = (A_1 + A_2, \vec{B})$. In the first premise, we have $(A_1, \vec{B})$, which is smaller than $\vec{A}$, and in the second premise we have $(A_2, \vec{B})$, which is likewise smaller. (In both premises, the number of $\wedge$ connectives does not grow.)
- **Case** Covers∃: The number of $\wedge$ connectives does not grow, and $\vec{A}$ gets smaller.
- **Case** CoversEq: The first premise is decidable by Lemma 78 (Decidability of Propositional Judgments) (4). The number of $\wedge$ connectives in $\vec{A}$ gets smaller (note that applying Δ as a substitution cannot add $\wedge$ connectives).
- **Case** CoversEqBot: Decidable by Lemma 78 (Decidability of Propositional Judgments) (4). □

H′.4 Decidability of Typing

Theorem 3 (Decidability of Typing).

(i) Synthesis: *Given a context Γ, a principality p, and a term e,*
it is decidable whether there exist a type A and a context Δ such that
$\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$.

(ii) Spines: *Given a context Γ, a spine s, a principality p, and a type A such that $\Gamma \vdash A$ type,*
it is decidable whether there exist a type B, a principality q and a context Δ such that
$\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$.

(iii) Checking: *Given a context Γ, a principality p, a term e, and a type B such that $\Gamma \vdash B$ type,*
it is decidable whether there is a context Δ such that
$\Gamma \vdash e \Leftarrow B\ p \dashv \Delta$.

(iv) Matching: *Given a context Γ, branches Π, a list of types $\vec{A}$, a type C, and a principality p, it is decidable whether there exists Δ such that $\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$.*

Also, if given a proposition P as well, it is decidable whether there exists Δ such that $\Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$.

Proof. For rules deriving judgments of the form

$$
\begin{array}{l}
\Gamma \vdash e \Rightarrow -\ - \dashv - \\
\Gamma \vdash e \Leftarrow B\ p \dashv - \\
\Gamma \vdash s : B\ p \gg -\ - \dashv - \\
\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv -
\end{array}
$$

(where we write "–" for parts of the judgments that are outputs), the following induction measure on such judgments is adequate to prove decidability:

$$\left\langle e/s/\Pi, \quad \begin{array}{l} \Rightarrow \\ \Leftarrow / \gg, \\ \mathsf{Match}, \end{array} \quad \begin{array}{l} \\ \#\mathsf{large}(B), \\ \vec{A}, \end{array} \quad \begin{array}{l} \\ B \\ \text{match judgment form} \end{array} \right\rangle$$

where $\langle \dots \rangle$ denotes lexicographic order, and where (when comparing two judgments typing terms of the same size) the synthesis judgment (top line) is considered smaller than the checking judgment (second line). That is,

$$\Rightarrow \quad \prec \quad \Leftarrow / \gg / \mathsf{Match}$$

Two match judgments are compared according to, first, the list of branches Π (which is a subterm of the containing case expression, allowing us to invoke the i.h. for the Case rule), then the size of the list of types $\vec{A}$ (considered to be the sum of the sizes $|A|$ of each type A in $\vec{A}$), and then, finally, whether the judgment is $\Gamma / P \vdash \dots$ or $\Gamma \vdash \dots$, considering the former judgment ($\Gamma / P \vdash \dots$) to be larger.

Note that this measure only uses the input parts of the judgments, leading to a straightforward decidability argument.

We will show that in each rule deriving a synthesis, checking, spine or match judgment, every premise is smaller than the conclusion.

- **Case** EmptySpine: No premises.
- **Case** $\rightarrow$Spine: In each premise, the expression/spine gets smaller (we have $e\ s$ in the conclusion, e in the first premise, and s in the second premise).
- **Case** Var: No nontrivial premises.
- **Case** Sub: The first premise has the same subject term e as the conclusion, but the judgment is smaller because our measure considers synthesis to be smaller than checking.

  The second premise is a subtyping judgment, which by Theorem 1 is decidable.
- **Case** Anno: It is easy to show that the judgment $\Gamma \vdash A\ !\ \mathit{type}$ is decidable. The second premise types e, but the conclusion types $(e : A)$, so the first part of the measure gets smaller.
- **Cases** 1I, 1I$\hat{\alpha}$: No premises.
- **Case** $\forall$I: Both the premise and conclusion type e, and both are checking; however, $\#\mathsf{large}(A_0) < \#\mathsf{large}(\forall \alpha : \kappa.\ A_0)$, so the premise is smaller.
- **Case** $\forall$Spine: Both the premise and conclusion type $e\ s$, and both are spine judgments; however, $\#\mathsf{large}(-)$ decreases.
- **Case** $\wedge$I: By Lemma 78 (Decidability of Propositional Judgments) (2), the first premise is decidable. For the second premise, $\#\mathsf{large}([\Theta]A_0) = \#\mathsf{large}(A_0) < \#\mathsf{large}(A_0 \wedge P)$.
- **Case** $\supset$I: For the first premise, use Lemma 78 (Decidability of Propositional Judgments) (5). In the second premise, $\#\mathsf{large}(-)$ gets smaller (similar to the $\wedge$I case).
- **Case** $\supset$I$\bot$: The premise is decidable by Lemma 78 (Decidability of Propositional Judgments) (5).
- **Case** $\supset$Spine: Similar to the $\wedge$I case.
- **Cases** $\rightarrow$I, $\rightarrow$I$\hat{\alpha}$: In the premise, the term is smaller.
- **Cases** $\rightarrow$E, $\rightarrow$E-**!**: In all premises, the term is smaller.
- **Cases** $+\mathsf{I}_k$, $+\mathsf{I}\hat{\alpha}_k$, $\times$I, $\times$I$\hat{\alpha}$: In all premises, the term is smaller.
- **Case** Case: In the first premise, the term is smaller. In the second premise, we have a list of branches that is a proper subterm of the case expression. The third premise is decidable by Theorem 2.

We now consider the match rules:

- **Case** MatchEmpty: No premises.
- **Case** MatchSeq: In each premise, the list of branches is properly contained in Π, making each premise smaller by the first part ("$e/s/\Pi$") of the measure.
- **Case** MatchBase: The term e in the premise is properly contained in Π.
- **Cases** Match$\exists$, Match$\times$, Match$+_k$, MatchNeg, MatchWild: Smaller by part (2) of the measure.
- **Case** Match$\wedge$: The premise has a smaller $\vec{A}$, so it is smaller by the $\vec{A}$ part of the measure. (The premise is the other judgment form, so it is *larger* by the "match judgment form" part, but $\vec{A}$ lexicographically dominates.)
- **Case** Match$\bot$: For the premise, use Lemma 78 (Decidability of Propositional Judgments) (4).
- **Case** MatchUnify:

  Lemma 78 (Decidability of Propositional Judgments) (4) shows that the first premise is decidable. The second premise has the same (single) branch and list of types, but is smaller by the "match judgment form" part of the measure. ☐

I′ Determinacy

Lemma 81 (Determinacy of Auxiliary Judgments)**.**

(1) Elimeq: *Given* Γ, σ, t, κ *such that* $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(t) = \emptyset$ *and* $\mathcal{D}_1 :: \Gamma \mathrel{/} \sigma \mathrel{\dot{=}} t : \kappa \dashv \Delta_1^{\bot}$ *and* $\mathcal{D}_2 :: \Gamma \mathrel{/} \sigma \mathrel{\dot{=}} t : \kappa \dashv \Delta_2^{\bot}$,
it is the case that $\Delta_1^{\bot} = \Delta_2^{\bot}$.

(2) Instantiation: *Given* Γ, $\hat{\alpha}$, t, κ *such that* $\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$ *and* $\Gamma \vdash t : \kappa$ *and* $\hat{\alpha} \notin FV(t)$
and $\mathcal{D}_1 :: \Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \hat{\alpha} := t : \kappa \dashv \Delta_2$
it is the case that $\Delta_1 = \Delta_2$.

(3) Symmetric instantiation:
Given Γ, $\hat{\alpha}$, $\hat{\beta}$, κ *such that* $\hat{\alpha}, \hat{\beta} \in \mathsf{unsolved}(\Gamma)$ *and* $\hat{\alpha} \neq \hat{\beta}$
and $\mathcal{D}_1 :: \Gamma \vdash \hat{\alpha} := \hat{\beta} : \kappa \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \hat{\beta} := \hat{\alpha} : \kappa \dashv \Delta_2$
it is the case that $\Delta_1 = \Delta_2$.

(4) Checkeq: *Given* Γ, σ, t, κ *such that* $\mathcal{D}_1 :: \Gamma \vdash \sigma \mathrel{\dot{=}} t : \kappa \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \sigma \mathrel{\dot{=}} t : \kappa \dashv \Delta_2$
it is the case that $\Delta_1 = \Delta_2$.

(5) Elimprop: *Given* Γ, P *such that* $\mathcal{D}_1 :: \Gamma \mathrel{/} P \dashv \Delta_1^{\bot}$ *and* $\mathcal{D}_2 :: \Gamma \mathrel{/} P \dashv \Delta_2^{\bot}$
it is the case that $\Delta_1 = \Delta_2$.

(6) Checkprop: *Given* Γ, P *such that* $\mathcal{D}_1 :: \Gamma \vdash P\ \mathit{true} \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash P\ \mathit{true} \dashv \Delta_2$,
it is the case that $\Delta_1 = \Delta_2$.

Proof.
Proof of Part (1) (Elimeq).
Rule ElimeqZero applies if and only if $\sigma = t = \mathsf{zero}$.
Rule ElimeqSucc applies if and only if σ and t are headed by succ.
Now suppose $\sigma = \alpha$.

- Rule ElimeqUvarRefl applies if and only if $t = \alpha$. (Rule ElimeqClash cannot apply; rules ElimeqUvarL and ElimeqUvarR have a free variable condition; rules ElimeqUvarL$\bot$ and ElimeqUvarR$\bot$ have a condition that $\sigma \neq t$.)

  In the remainder, assume $t \neq alpha$.

- If $\alpha \in FV(t)$, then rule ElimeqUvarL⊥ applies, and no other rule applies (including ElimeqUvarR⊥ and ElimeqClash).

  In the remainder, assume $\alpha \notin FV(t)$.

- Consider whether ElimeqUvarR⊥ applies. The conclusion matches if we have $t = \beta$ for some $\beta \neq \alpha$ (that is, $\sigma = \alpha$ and $t = \beta$). But ElimeqUvarR⊥ has a condition that $\beta \in FV(\sigma)$, and $\sigma = \alpha$, so the condition is not satisfied.

In the symmetric case, use the reasoning above, exchanging L's and R's in the rule names.

Proof of Part (2) (Instantiation).
Rule InstBin applies if and only if t has the form $t_1 \oplus t_2$.
Rule InstZero applies if and only if t has the form zero.
Rule InstSucc applies if and only if t has the form $\mathsf{succ}(t_0)$.
If t has the form $\hat{\beta}$, then consider whether $\hat{\beta}$ is declared to the left of $\hat{\alpha}$ in the given context:

- If $\hat{\beta}$ is declared to the left of $\hat{\alpha}$, then rule InstReach cannot be used, which leaves only InstSolve.
- If $\hat{\beta}$ is declared to the right of $\hat{\alpha}$, then InstSolve cannot be used because $\hat{\beta}$ is not well-formed under Γ_0 (the context to the left of $\hat{\alpha}$ in InstSolve). That leaves only InstReach.
- $\hat{\alpha}$ cannot be $\hat{\beta}$, because it is given that $\hat{\alpha} \notin FV(t) = FV(\hat{\beta}) = \{\hat{\beta}\}$.

Proof of Part (3) (Symmetric instantiation).
InstBin, InstZero and InstSucc cannot have been used in either derivation.

Suppose that InstSolve concluded $\mathcal{D}_1$. Then Δ_1 is the same as Γ with $\hat{\alpha}$ solved to $\hat{\beta}$. Moreover, $\hat{\beta}$ is declared to the left of $\hat{\alpha}$ in Γ. Thus, InstSolve cannot conclude $\mathcal{D}_2$. However, InstReach can conclude $\mathcal{D}_2$, but produces a context Δ_2 which is the same as Γ but with $\hat{\alpha}$ solved to $\hat{\beta}$. Therefore $\Delta_1 = \Delta_2$.

The other possibility is that InstReach concluded $\mathcal{D}_1$. Then Δ_1 is the same as Γ with $\hat{\beta}$ solved to $\hat{\alpha}$, with $\hat{\alpha}$ declared to the left of $\hat{\beta}$ in Γ. Thus, InstReach cannot conclude $\mathcal{D}_2$. However, InstSolve can conclude $\mathcal{D}_2$, producing a context Δ_2 which is the same as Γ but with $\hat{\beta}$ solved to $\hat{\alpha}$. Therefore $\Delta_1 = \Delta_2$.

Proof of Part (4) (Checkeq).
Rule CheckeqVar applies if and only if $\sigma = t = \hat{\alpha}$ or $\sigma = t = \alpha$ (note the free variable conditions in CheckeqInstL and CheckeqInstR).

Rule CheckeqUnit applies if and only if $\sigma = t = 1$.
Rule CheckeqBin applies if and only if σ and t are both headed by the same binary connective.
Rule CheckeqZero applies if and only if $\sigma = t = \mathsf{zero}$.
Rule CheckeqSucc applies if and only if σ and t are headed by succ.

Now suppose $\sigma = \hat{\alpha}$. If t is not an existential variable, then CheckeqInstR cannot be used, which leaves only CheckeqInstL. If t is an existential variable, that is, some $\hat{\beta}$ (distinct from $\hat{\alpha}$), and is unsolved, then both CheckeqInstL and CheckeqInstR apply, but by part (3), we get the same output context from each.

The $t = \hat{\alpha}$ subcase is similar.

Proof of Part (5) (Elimprop). There is only one rule deriving this judgment; the result follows by part (1).

Proof of Part (6) (Checkprop). There is only one rule deriving this judgment; the result follows by part (4). □

Lemma 82 (Determinacy of Equivalence).

(1) Propositional equivalence: *Given* Γ, P, Q *such that* $\mathcal{D}_1 :: \Gamma \vdash P \equiv Q \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash P \equiv Q \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

(2) Type equivalence: *Given* Γ*,* A*,* B *such that* $\mathcal{D}_1 :: \Gamma \vdash A \equiv B \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash A \equiv B \dashv \Delta_2$*,* *it is the case that* $\Delta_1 = \Delta_2$*.*

Proof.
Proof of Part (1) (propositional equivalence). Only one rule derives judgments of this form; the result follows from Lemma 81 (Determinacy of Auxiliary Judgments) (4).

Proof of Part (2) (type equivalence). If neither A nor B is an existential variable, they must have the same head connectives, and the same rule must conclude both derivations.

If A and B are the same existential variable, then only $\equiv$Exvar applies (due to the free variable conditions in $\equiv$InstantiateL and $\equiv$InstantiateR).

If A and B are different unsolved existential variables, the judgment matches the conclusion of both $\equiv$InstantiateL and $\equiv$InstantiateR, but by part (3) of Lemma 81 (Determinacy of Auxiliary Judgments), we get the same output context regardless of which rule we choose. □

Theorem 4 (Determinacy of Subtyping)**.**

(1) Subtyping: *Given* Γ*, e,* A*,* B *such that* $\mathcal{D}_1 :: \Gamma \vdash A <:^{\pm} B \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash A <:^{\pm} B \dashv \Delta_2$*,* *it is the case that* $\Delta_1 = \Delta_2$*.*

Proof. First, we consider whether we are looking at positive or negative subtyping, and then consider the outermost connective of A and B:

- If $\Gamma \vdash A <:^{+} B \dashv \Delta_1$ and $\Gamma \vdash A <:^{+} B \dashv \Delta_2$, then we know the last rule ending the derivation of $\mathcal{D}_1$ and $\mathcal{D}_2$ must be:

		B		
		$\forall$	$\exists$	other
	$\forall$	$<:^{-}_{+}$R, $<:^{-}_{+}$L	$<:\exists$R	$<:^{-}_{+}$L
A	$\exists$	$<:\exists$L	$<:\exists$L	$<:\exists$L
	other	$<:^{-}_{+}$R	$<:\exists$R	$<:$Equiv

  The only case in which there are two possible final rules is in the $\forall/\forall$ case. In this case, regardless of the choice of rule, by inversion we get subderivations $\Gamma \vdash A <:^{-} B \dashv \Delta_1$ and $\Gamma \vdash A <:^{-} B \dashv \Delta_2$.

- If $\Gamma \vdash A <:^{-} B \dashv \Delta_1$ and $\Gamma \vdash A <:^{-} B \dashv \Delta_2$, then we know the last rule ending the derivation of $\mathcal{D}_1$ and $\mathcal{D}_2$ must be:

		B		
		$\forall$	$\exists$	other
	$\forall$	$<:\forall$R	$<:\forall$L	$<:\forall$L
A	$\exists$	$<:\forall$R	$<:^{+}_{-}$L, $<:^{+}_{-}$R	$<:^{+}_{-}$L
	other	$<:\forall$R	$<:^{+}_{-}$R	$<:$Equiv

  The only case in which there are two possible final rules is in the $\forall/\forall$ case. In this case, regardless of the choice of rule, by inversion we get subderivations $\Gamma \vdash A <:^{+} B \dashv \Delta_1$ and $\Gamma \vdash A <:^{+} B \dashv \Delta_2$.

As a result, the result follows by a routine induction. □

Theorem 5 (Determinacy of Typing)**.**

(1) Checking: *Given* Γ*, e,* A*, p such that* $\mathcal{D}_1 :: \Gamma \vdash e \Leftarrow A\ p \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash e \Leftarrow A\ p \dashv \Delta_2$*,* *it is the case that* $\Delta_1 = \Delta_2$*.*

(2) Synthesis: *Given* Γ, e *such that* $\mathcal{D}_1 :: \Gamma \vdash e \Rightarrow B_1\ p_1 \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash e \Rightarrow B_2\ p_2 \dashv \Delta_2$, *it is the case that* $B_1 = B_2$ *and* $p_1 = p_2$ *and* $\Delta_1 = \Delta_2$.

(3) Spine judgments:

Given Γ, e, A, p *such that* $\mathcal{D}_1 :: \Gamma \vdash e : A\ p \gg C_1\ q_1 \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash e : A\ p \gg C_2\ q_2 \dashv \Delta_2$, *it is the case that* $C_1 = C_2$ *and* $q_1 = q_2$ *and* $\Delta_1 = \Delta_2$.

The same applies for derivations of the principality-recovering judgments $\Gamma \vdash e : A\ p \gg C_k\ \lceil q_k \rceil \dashv \Delta_k$.

(4) Match judgments:

Given Γ, Π, $\vec{A}$, p, C *such that* $\mathcal{D}_1 :: \Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

Given Γ, P, Π, $\vec{A}$, p, C *such that* $\mathcal{D}_1 :: \Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_1$ *and* $\mathcal{D}_2 :: \Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta_2$, *it is the case that* $\Delta_1 = \Delta_2$.

Proof.

Proof of Part (1) (checking).

The rules with a checking judgment in the conclusion are: $1\mathsf{I}$, $1\mathsf{I}\hat{\alpha}$, $\forall\mathsf{I}$, $\wedge\mathsf{I}$, $\supset\mathsf{I}$, $\supset\mathsf{I}\bot$, $\to\mathsf{I}$, $\to\mathsf{I}\hat{\alpha}$, Rec, $+\mathsf{I}_k$, $+\mathsf{I}\hat{\alpha}_k$, $\times\mathsf{I}$, $\times\mathsf{I}\hat{\alpha}$, Case, Nil, Cons.

The table below shows which rules apply for given e and A. The extra "*chk-I*?" column highlights the role of the "*chk-I*" ("check-intro") category of syntactic forms: we restrict the introduction rules for $\forall$ and $\supset$ to type only these forms. For example, given $e = x$ and $A = (\forall\alpha : \kappa.\ A_0)$, we need not choose between Sub and $\forall\mathsf{I}$: the latter is ruled out by its *chk-I* premise.

			A									
		chk-I?	$\forall$	*Note 1* $\supset$	$\exists$	$\wedge$	$\to$	$+$	$\times$	1	$\hat{\alpha}$	α	Vec
	$\lambda x.\ e_0$	*chk-I*	$\forall\mathsf{I}$	$\supset\mathsf{I}/\supset\mathsf{I}\bot$	Sub	$\wedge\mathsf{I}$	$\to\mathsf{I}$	$\emptyset$	$\emptyset$	$\emptyset$	$\to\mathsf{I}\hat{\alpha}$	$\emptyset$	$\emptyset$
	$\mathsf{rec}\ x.\ v$	*Note 2*	Rec	Rec	Rec	Rec	Rec	Rec	Rec	Rec	Rec	Rec	$\emptyset$
	$\mathsf{inj}_k\ e_0$	*chk-I*	$\forall\mathsf{I}$	$\supset\mathsf{I}/\supset\mathsf{I}\bot$	Sub	$\wedge\mathsf{I}$	$\emptyset$	$+\mathsf{I}_k$	$\emptyset$	$\emptyset$	$+\mathsf{I}\hat{\alpha}_k$	$\emptyset$	$\emptyset$
	$\langle e_1, e_2 \rangle$	*chk-I*	$\forall\mathsf{I}$	$\supset\mathsf{I}/\supset\mathsf{I}\bot$	Sub	$\wedge\mathsf{I}$	$\emptyset$	$\emptyset$	$\times\mathsf{I}$	$\emptyset$	$\times\mathsf{I}\hat{\alpha}$	$\emptyset$	$\emptyset$
	$[]$	*chk-I*	$\forall\mathsf{I}$	$\supset\mathsf{I}/\supset\mathsf{I}\bot$	Sub	$\wedge\mathsf{I}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	Nil
e	$e_1 :: e_2$	*chk-I*	$\forall\mathsf{I}$	$\supset\mathsf{I}/\supset\mathsf{I}\bot$	Sub	$\wedge\mathsf{I}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	Cons
	$()$	*chk-I*	$\forall\mathsf{I}$	$\supset\mathsf{I}/\supset\mathsf{I}\bot$	Sub	$\wedge\mathsf{I}$	$\emptyset$	$\emptyset$	$\emptyset$	$1\mathsf{I}$	$1\mathsf{I}\hat{\alpha}$	$\emptyset$	$\emptyset$
	$\mathsf{case}(e_0, \Pi)$	*Note 3*	Case	Case	Case	Case	Case	Case	Case	Case	Case	Case	Case
	x		Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub
	$(e_0 : A)$		Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub
	$e_1\ s$		Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub

Notes:

- *Note 1:* The choice between $\supset\mathsf{I}$ and $\supset\mathsf{I}\bot$ is resolved by Lemma 81 (Determinacy of Auxiliary Judgments) (5).
- *Note 2:* Fixed points are a checking form, but not an introduction form. So if e is $\mathsf{rec}\ x.\ v$, we need not choose between an introduction rule for a large connective and the Rec rule: only the Rec rule is viable. Large connectives must, therefore, be introduced *inside* the typing of the body v.
- *Note 3:* Case expressions are a checking form, but not an introduction form. So if e is a case expression, we need not choose between an introduction rule for a large connective and the Case rule: only the Case rule is viable. Large connectives must, therefore, be introduced *inside* the branches.

Proof of Part (2) (synthesis). Only four rules have a synthesis judgment in the conclusion: Var, Anno, $\to$E, and $\to$E-**!**. Rule Var applies if and only if e has the form x. Rule Anno applies if and only if e has the form $(e_0 : A)$.

Otherwise, the judgment can be derived only if e has the form $e_1\, e_2$, by $\to$E or $\to$E-**!**. If $\mathcal{D}_1$ and $\mathcal{D}_2$ both end in $\to$E or $\to$E-**!**, we are done. Suppose $\mathcal{D}_1$ ends in $\to$E and $\mathcal{D}_2$ ends in $\to$E-**!**. By i.h., the p in the first subderivation of $\to$E must be equal to the one in the first subderivation of $\to$E-**!**, that is, $p = \mathbf{!}$. Thus the inputs to the respective second subderivations match, so by i.h. their outputs match; in particular, $q = \not{!}$. However, from the condition in $\to$E, it must be the case that $\mathsf{FEV}([\Delta]C) \neq \emptyset$, which contradicts the condition $\mathsf{FEV}([\Delta]C) = \emptyset$ in $\to$E-**!**.

Proof of Part (3) (spine judgments). For the ordinary spine judgment, rule EmptySpine applies if and only if the given spine is empty. Otherwise, the choice of rule is determined by the head constructor of the input type: $\to$/$\to$Spine; $\forall$/$\forall$Spine; $\supset$/$\supset$Spine; $\hat{\alpha}$/$\hat{\alpha}$Spine.

For the principality-recovering spine judgment: If $p = \not{!}$, only rule SpinePass applies. If $p = \mathbf{!}$ and $q = \mathbf{!}$, only rule SpinePass applies. If $p = \mathbf{!}$ and $q = \not{!}$, then the rule is determined by $\mathsf{FEV}(C)$: if $\mathsf{FEV}(C) = \emptyset$ then only SpineRecover applies; otherwise, $\mathsf{FEV}(C) \neq \emptyset$ and only SpinePass applies.

Proof of Part (4) (matching). First, the elimination judgment form $\Gamma \,/\, P \vdash \ldots$: It cannot be the case that both $\Gamma \,/\, \sigma \mathrel{\dot=} t : \kappa \dashv \bot$ and $\Gamma \,/\, \sigma \mathrel{\dot=} t : \kappa \dashv \Theta$, so either Match$\bot$ concludes both $\mathcal{D}_1$ and $\mathcal{D}_2$ (and the result follows), or MatchUnify concludes both $\mathcal{D}_1$ and $\mathcal{D}_2$ (in which case, apply the i.h.).

Now the main judgment form, without "/ P": either Π is empty, or has length one, or has length greater than one. MatchEmpty applies if and only if Π is empty, and MatchSeq applies if and only if Π has length greater than one. So in the rest of this part, we assume Π has length one.

Moreover, MatchBase applies if and only if $\vec{A}$ has length zero. So in the rest of this part, we assume the length of $\vec{A}$ is at least one.

Let A be the first type in $\vec{A}$. Inspection of the rules shows that given particular A and ρ, where ρ is the first pattern, only a single rule can apply, or no rule ("$\emptyset$") can apply, as shown in the following table:

ρ \ A	$\exists$	$\wedge$	$+$	$\times$	Vec	other
$\mathsf{inj}_k\,\rho_0$	Match$\exists$	Match$\wedge$	Match$+_k$	$\emptyset$	$\emptyset$	$\emptyset$
$\langle\rho_1, \rho_2\rangle$	Match$\exists$	Match$\wedge$	$\emptyset$	Match$\times$	$\emptyset$	$\emptyset$
z	Match$\exists$	Match$\wedge$	MatchNeg	MatchNeg	MatchNeg	MatchNeg
_	Match$\exists$	Match$\wedge$	MatchWild	MatchWild	MatchWild	MatchWild
$[]$	Match$\exists$	Match$\wedge$	$\emptyset$	$\emptyset$	MatchNil	$\emptyset$
$\rho_1 :: \rho_2$	Match$\exists$	Match$\wedge$	$\emptyset$	$\emptyset$	MatchCons	$\emptyset$

□

J′ Soundness

J′.1 Instantiation

Lemma 83 (Soundness of Instantiation).
If $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$ *and* $\hat{\alpha} \notin FV([\Gamma]\tau)$ *and* $[\Gamma]\tau = \tau$ *and* $\Delta \longrightarrow \Omega$ *then* $[\Omega]\hat{\alpha} = [\Omega]\tau$*.*

Proof. By induction on the derivation of $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$.

- **Case**
$$\dfrac{\Gamma_0 \vdash \tau : \kappa}{\Gamma_0, \hat{\alpha} : \kappa, \Gamma_1 \vdash \hat{\alpha} := \tau : \kappa \dashv \underbrace{\Gamma_0, \hat{\alpha} : \kappa = \tau, \Gamma_1}_{\Delta}}\ \mathsf{InstSolve}$$

  $[\Delta]\hat{\alpha} = [\Delta]\tau$ By definition
  ☞ $[\Omega]\hat{\alpha} = [\Omega]\tau$ By Lemma 29 (Substitution Monotonicity) to each side

- **Case**

$$\frac{\hat\beta \in \mathsf{unsolved}(\Gamma[\hat\alpha:\kappa][\hat\beta:\kappa])}{\Gamma[\hat\alpha:\kappa][\hat\beta:\kappa] \vdash \hat\alpha := \underbrace{\hat\beta}_{\tau} : \kappa \dashv \underbrace{\Gamma[\hat\alpha:\kappa][\hat\beta:\kappa=\hat\alpha]}_{\Delta}}\ \mathsf{InstReach}$$

$$\begin{array}{rll}
 & [\Delta]\hat\beta = [\Delta]\hat\alpha & \text{By definition} \\
 & [\Omega][\Delta]\hat\beta = [\Omega][\Delta]\hat\alpha & \text{Applying } \Omega \text{ to each side} \\
☞ & [\Omega]\underbrace{\hat\beta}_{\tau} = [\Omega]\hat\alpha & \text{By Lemma 29 (Substitution Monotonicity) to each side}
\end{array}$$

- **Case**

$$\frac{\overbrace{\Gamma_0[\hat\alpha_2:\star,\hat\alpha_1:\star,\hat\alpha:\star=\hat\alpha_1\oplus\hat\alpha_2]}^{\Gamma'} \vdash \hat\alpha_1 := \tau_1 : \star \dashv \Theta \qquad \Theta \vdash \hat\alpha_2 := [\Theta]\tau_2 : \star \dashv \Delta}{\Gamma_0[\hat\alpha:\star] \vdash \hat\alpha := \tau_1 \oplus \tau_2 : \star \dashv \Delta}\ \mathsf{InstBin}$$

$$\begin{array}{rrll}
 & \Delta \longrightarrow \Omega & & \text{Given} \\
 & \Gamma' \vdash \hat\alpha_1 := \tau_1 : \star \dashv \Theta & & \text{Subderivation} \\
 & \Theta \longrightarrow \Delta & & \text{By Lemma 43 (Instantiation Extension)} \\
 & \Theta \longrightarrow \Omega & & \text{By Lemma 33 (Extension Transitivity)} \\
 & [\Omega]\hat\alpha_1 & = [\Omega]\tau_1 & \text{By i.h.} \\
 & \Theta \vdash \hat\alpha_2 := [\Theta]\tau_2 : \star \dashv \Delta & & \text{Subderivation} \\
 & [\Omega]\hat\alpha_2 & = [\Omega][\Theta]\tau_2 & \text{By i.h.} \\
 & & = [\Omega]\tau_2 & \text{By Lemma 29 (Substitution Monotonicity)} \\
 & ([\Omega]\tau_1) \oplus ([\Omega]\tau_2) & = ([\Omega]\hat\alpha_1) \oplus ([\Omega]\hat\alpha_2) & \text{By above equalities} \\
 & & = [\Omega](\hat\alpha_1 \oplus \hat\alpha_2) & \text{By definition of substitution} \\
 & & = [\Omega]([\Gamma']\hat\alpha) & \text{By definition of substitution} \\
 & & = [\Omega]\hat\alpha & \text{By Lemma 29 (Substitution Monotonicity)} \\
☞ & [\Omega]\underbrace{(\tau_1 \oplus \tau_2)}_{\tau} & = [\Omega]\hat\alpha & \text{By definition of substitution}
\end{array}$$

- **Case**

$$\frac{}{\Gamma_0[\hat\alpha:\mathbb{N}] \vdash \hat\alpha := \mathsf{zero} : \mathbb{N} \dashv \Gamma_0[\hat\alpha:\mathbb{N}=\mathsf{zero}]}\ \mathsf{InstZero}$$

Similar to the InstSolve case.

- **Case**

$$\frac{\Gamma_0[\hat\alpha_1:\mathbb{N},\hat\alpha:\mathbb{N}=\mathsf{succ}(\hat\alpha_1)] \vdash \hat\alpha_1 := t_1 : \mathbb{N} \dashv \Delta}{\Gamma_0[\hat\alpha:\mathbb{N}] \vdash \hat\alpha := \mathsf{succ}(t_1) : \mathbb{N} \dashv \Delta}\ \mathsf{InstSucc}$$

Similar to the InstBin case, but simpler. □

Lemma 84 (Soundness of Checkeq)**.**
If $\Gamma \vdash \sigma \mathrel{\mathring{=}} t : \kappa \dashv \Delta$ *where* $\Delta \longrightarrow \Omega$ *then* $[\Omega]\sigma = [\Omega]t$.

Proof. By induction on the given derivation.

- **Case**

$$\frac{}{\Gamma \vdash u \mathrel{\mathring{=}} u : \kappa \dashv \Gamma}\ \mathsf{CheckeqVar}$$

☞ $[\Omega]u = [\Omega]u$ By reflexivity of equality

- **Cases** CheckeqUnit, CheckeqZero: Similar to the CheckeqVar case.

- **Case**
$$\dfrac{\Gamma \vdash \sigma_0 \doteq t_0 : \mathbb{N} \dashv \Delta}{\Gamma \vdash \mathsf{succ}(\sigma_0) \doteq \mathsf{succ}(t_0) : \mathbb{N} \dashv \Delta}\ \mathsf{CheckeqSucc}$$

$$\begin{array}{rll}
& \Gamma \vdash \sigma_0 \doteq t_0 : \mathbb{N} \dashv \Delta & \text{Subderivation} \\
& [\Omega]\sigma_0 = [\Omega]t_0 & \text{By i.h.} \\
& \mathsf{succ}([\Omega]\sigma_0) = \mathsf{succ}([\Omega]t_0) & \text{By congruence} \\
☞ & [\Omega](\mathsf{succ}(\sigma_0)) = [\Omega](\mathsf{succ}(t_0)) & \text{By definition of substitution}
\end{array}$$

- **Case**
$$\dfrac{\Gamma \vdash \sigma_0 \doteq t_0 : \star \dashv \Theta \qquad \Theta \vdash [\Theta]\sigma_1 \doteq [\Theta]t_1 : \star \dashv \Delta}{\Gamma \vdash \sigma_0 \oplus \sigma_1 \doteq t_0 \oplus t_1 : \star \dashv \Delta}\ \mathsf{CheckeqBin}$$

$$\begin{array}{rll}
& \Gamma \vdash \sigma_0 \doteq t_0 : \mathbb{N} \dashv \Delta & \text{Subderivation} \\
& \Theta \vdash [\Theta]\sigma_1 \doteq [\Theta]t_1 : \star \dashv \Delta & \text{Subderivation} \\
& \Delta \longrightarrow \Omega & \text{Given} \\
& \Theta \longrightarrow \Delta & \text{By Lemma 46 (Checkeq Extension)} \\
& \Theta \longrightarrow \Omega & \text{By Lemma 33 (Extension Transitivity)} \\
& [\Omega]\sigma_0 = [\Omega]t_0 & \text{By i.h. on first subderivation} \\
& [\Omega][\Theta]\sigma_1 = [\Omega][\Theta]t_1 & \text{By i.h. on second subderivation} \\
& [\Omega][\Theta]\sigma_1 = [\Omega]\sigma_1 & \text{By Lemma 29 (Substitution Monotonicity)} \\
& [\Omega][\Theta]t_1 = [\Omega]t_1 & \text{By Lemma 29 (Substitution Monotonicity)} \\
& [\Omega]\sigma_1 = [\Omega]t_1 & \text{By transitivity of equality} \\
& [\Omega]\sigma_0 \oplus [\Omega]\sigma_1 = [\Omega]t_0 \oplus [\Omega]t_1 & \text{By congruence of equality} \\
☞ & [\Omega](\sigma_0 \oplus \sigma_1) = [\Omega](t_0 \oplus t_1) & \text{By definition of substitution}
\end{array}$$

- **Case**
$$\dfrac{\Gamma[\hat\alpha] \vdash \hat\alpha := t : \kappa \dashv \Delta \qquad \hat\alpha \notin FV(t)}{\Gamma[\hat\alpha] \vdash \hat\alpha \doteq t : \kappa \dashv \Delta}\ \mathsf{CheckeqInstL}$$

$$\begin{array}{rll}
& \Gamma[\hat\alpha] \vdash \hat\alpha := t : \kappa \dashv \Delta & \text{Subderivation} \\
& \hat\alpha \notin FV(t) & \text{Premise} \\
☞ & [\Omega]\hat\alpha = [\Omega]t & \text{By Lemma 83 (Soundness of Instantiation)}
\end{array}$$

- **Case**
$$\dfrac{\Gamma[\hat\alpha : \kappa] \vdash \hat\alpha := \sigma : \kappa \dashv \Delta \qquad \hat\alpha \notin FV(t)}{\Gamma[\hat\alpha : \kappa] \vdash \sigma \doteq \hat\alpha : \kappa \dashv \Delta}\ \mathsf{CheckeqInstR}$$

  Similar to the CheckeqInstL case. □

Lemma 85 (Soundness of Propositional Equivalence)**.**
If $\Gamma \vdash P \equiv Q \dashv \Delta$ *where* $\Delta \longrightarrow \Omega$ *then* $[\Omega]P = [\Omega]Q$.

Proof. By induction on the given derivation.

- **Case**
$$\dfrac{\Gamma \vdash \sigma_1 \doteq t_1 : \mathbb{N} \dashv \Theta \qquad \Theta \vdash [\Theta]\sigma_2 \doteq [\Theta]t_2 : \mathbb{N} \dashv \Delta}{\Gamma \vdash (\sigma_1 = \sigma_2) \equiv (t_1 = t_2) \dashv \Delta}\ \equiv\!\mathsf{PropEq}$$

$\Delta \longrightarrow \Omega$	Given
$\Theta \longrightarrow \Delta$	By Lemma 46 (Checkeq Extension) (on 2nd premise)
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Gamma \vdash \sigma_1 \doteq t_1 : \mathbb{N} \dashv \Theta$	Given
$[\Omega]\sigma_1 = [\Omega]t_1$	By Lemma 84 (Soundness of Checkeq)
$\Theta \vdash [\Theta]\sigma_2 \doteq [\Theta]t_2 : \mathbb{N} \dashv \Delta$	Given
$[\Omega][\Theta]\sigma_2 = [\Omega][\Theta]t_2$	By Lemma 84 (Soundness of Checkeq)
$[\Omega][\Theta]\sigma_2 = [\Omega]\sigma_2$	By Lemma 29 (Substitution Monotonicity)
$[\Omega][\Theta]t_2 = [\Omega]t_2$	By Lemma 29 (Substitution Monotonicity)
$[\Omega]\sigma_2 = [\Omega]t_2$	By transitivity of equality
$([\Omega]\sigma_1 = [\Omega]\sigma_2) = ([\Omega]t_1 = [\Omega]t_2)$	By congruence of equality
☞ $[\Omega](\sigma_1 = \sigma_2) = [\Omega](t_1 = t_2)$	By definition of substitution □

Lemma 86 (Soundness of Algorithmic Equivalence).
If $\Gamma \vdash A \equiv B \dashv \Delta$ *where* $\Delta \longrightarrow \Omega$ *then* $[\Omega]A = [\Omega]B$.

Proof. By induction on the given derivation.

- **Case**

$$\frac{}{\Gamma \vdash \alpha \equiv \alpha \dashv \Gamma}\ \equiv\mathsf{Var}$$

☞ $[\Omega]\alpha = [\Omega]\alpha$ By reflexivity of equality

- **Cases** $\equiv$Exvar, $\equiv$Unit: Similar to the $\equiv$Var case.

- **Case**

$$\frac{\Gamma \vdash A_1 \equiv B_1 \dashv \Theta \qquad \Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta}{\Gamma \vdash A_1 \oplus A_2 \equiv B_1 \oplus B_2 \dashv \Delta}\ \equiv\oplus$$

$\Delta \longrightarrow \Omega$	Given
$\Theta \vdash [\Theta]A_2 \equiv [\Theta]B_2 \dashv \Delta$	Subderivation
$\Theta \longrightarrow \Delta$	By Lemma 49 (Equivalence Extension)
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Gamma \vdash A_1 \equiv B_1 \dashv \Theta$	Subderivation
$[\Omega]A_1 = [\Omega]B_1$	By i.h.
$\Delta \longrightarrow \Omega$	Given
$[\Omega][\Theta]A_2 = [\Omega][\Theta]B_2$	By i.h.
$[\Omega]A_2 = [\Omega]B_2$	By Lemma 29 (Substitution Monotonicity)

☞ $([\Omega]A_1) \oplus ([\Omega]A_2) = ([\Omega]B_1) \oplus ([\Omega]B_2)$ By above equations

- **Case**

$$\frac{\Gamma, \alpha : \kappa \vdash A_0 \equiv B_0 \dashv \Delta, \alpha : \kappa, \Delta'}{\Gamma \vdash \forall\alpha : \kappa.\, A_0 \equiv \forall\alpha : \kappa.\, B_0 \dashv \Delta}\ \equiv\forall$$

$\Gamma, \alpha : \kappa \vdash A_0 \equiv B_0 \dashv \Delta, \alpha : \kappa, \Delta'$	Subderivation
$\Delta \longrightarrow \Omega$	Given
$\Gamma, \alpha : \kappa, \cdot \longrightarrow \Delta, \alpha : \kappa, \Delta'$	By Lemma 49 (Equivalence Extension)

Δ' soft	Since $\cdot$ is soft
$\Delta, \alpha : \kappa, \Delta' \longrightarrow \Omega, \alpha : \kappa, \Omega_Z$	By Lemma 24 (Soft Extension)
$\Gamma, \alpha : \kappa \vdash A_0$ *type*	By validity on subderivation
$\Gamma, \alpha : \kappa \vdash B_0$ *type*	By validity on subderivation
$FV(A_0) \subseteq \mathsf{dom}(\Gamma, \alpha : \kappa)$	By well-typing of A_0
$FV(B_0) \subseteq \mathsf{dom}(\Gamma, \alpha : \kappa)$	By well-typing of B_0
$\Gamma, \alpha : \kappa \longrightarrow \Omega, \alpha : \kappa$	By $\longrightarrow$Uvar
$FV(A_0) \subseteq \mathsf{dom}(\Omega, \alpha : \kappa)$	By Lemma 20 (Declaration Order Preservation)
$FV(B_0) \subseteq \mathsf{dom}(\Omega, \alpha : \kappa)$	By Lemma 20 (Declaration Order Preservation)
$[\Omega, \alpha : \kappa, \Omega_Z]A_0 = [\Omega, \alpha : \kappa]A_0$	By definition of substitution, since $FV(A_0) \cap \mathsf{dom}(\Omega_Z) = \emptyset$
$[\Omega, \alpha : \kappa, \Omega_Z]B_0 = [\Omega, \alpha : \kappa]B_0$	By definition of substitution, since $FV(B_0) \cap \mathsf{dom}(\Omega_Z) = \emptyset$
$[\Omega, \alpha : \kappa]A_0 = [\Omega, \alpha : \kappa]B_0$	By transitivity of equality
$[\Omega]A_0 = [\Omega]B_0$	From definition of substitution
$\forall\alpha : \kappa.\, [\Omega]A_0 = \forall\alpha : \kappa.\, [\Omega]B_0$	Adding quantifier to each side
$[\Omega](\forall\alpha : \kappa.\, A_0) = [\Omega](\forall\alpha : \kappa.\, B_0)$	By definition of subsitution

- **Case** $\dfrac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta}{\Gamma \vdash P \supset A_0 \equiv Q \supset B_0 \dashv \Delta} \equiv\!\supset$

$\Delta \longrightarrow \Omega$	Given
$\Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta$	Subderivation
$\Theta \longrightarrow \Delta$	By Lemma 49 (Equivalence Extension)
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Gamma \vdash P \equiv Q \dashv \Theta$	Subderivation
$[\Omega]P = [\Omega]Q$	By Lemma 85 (Soundness of Propositional Equivalence)
$\Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta$	Subderivation
$[\Omega][\Theta]A_0 = [\Omega][\Theta]B_0$	By i.h.
$[\Omega]A_0 = [\Omega]B_0$	By Lemma 29 (Substitution Monotonicity)

- **Case** $\dfrac{\Gamma \vdash P \equiv Q \dashv \Theta \qquad \Theta \vdash [\Theta]A_0 \equiv [\Theta]B_0 \dashv \Delta}{\Gamma \vdash A_0 \wedge P \equiv B_0 \wedge Q \dashv \Delta} \equiv\!\wedge$

  Similar to the $\equiv\!\supset$ case.

- **Case** $\dfrac{\Gamma[\hat\alpha] \vdash \hat\alpha := \tau : \star \dashv \Delta \qquad \hat\alpha \notin FV(\tau)}{\Gamma[\hat\alpha] \vdash \hat\alpha \equiv \underbrace{\tau}_{A} \dashv \Delta} \equiv\!\mathsf{InstantiateL}$

$\Gamma[\hat\alpha] \vdash \hat\alpha := \tau : \star \dashv \Delta$	Subderivation
☞ $[\Omega]\hat\alpha = [\Omega]\tau$	By Lemma 83 (Soundness of Instantiation)

- **Case** $\equiv$InstantiateR: Similar to the $\equiv$InstantiateL case. □

J′.2 Soundness of Checkprop

Lemma 87 (Soundness of Checkprop).
If $\Gamma \vdash P$ *true* $\dashv \Delta$ *and* $\Delta \longrightarrow \Omega$ *then* $\Psi \vdash [\Omega]P$ *true.*

Proof. By induction on the derivation of $\Gamma \vdash P$ *true* $\dashv \Delta$.

- **Case** $$\dfrac{\Gamma \vdash \sigma \stackrel{\circ}{=} t : \mathbb{N} \dashv \Delta}{\Gamma \vdash \underbrace{\sigma = t}_{P}\ \mathit{true} \dashv \Delta}\ \textsf{CheckpropEq}$$

	$\Gamma \vdash \sigma \stackrel{\circ}{=} t : \mathbb{N} \dashv \Delta$	Subderivation
	$[\Omega]\sigma = [\Omega]t$	By Lemma 84 (Soundness of Checkeq)
	$\Psi \vdash [\Omega]\sigma = [\Omega]t\ \mathit{true}$	By DeclCheckpropEq
	$\Psi \vdash [\Omega](\sigma = t)\ \mathit{true}$	By def. of subst.
☞	$\Psi \vdash [\Omega]P\ \mathit{true}$	By $P = (\sigma = t)$ ☐

J′.3 Soundness of Eliminations (Equality and Proposition)

Lemma 88 (Soundness of Equality Elimination)**.**
If $[\Gamma]\sigma = \sigma$ *and* $[\Gamma]t = t$ *and* $\Gamma \vdash \sigma : \kappa$ *and* $\Gamma \vdash t : \kappa$ *and* $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(t) = \emptyset$*, then:*

(1) If $\Gamma \,/\, \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta$
then $\Delta = (\Gamma, \Theta)$ *where* $\Theta = (\alpha_1 = t_1, \ldots, \alpha_n = t_n)$ *and*
for all Ω *such that* $\Gamma \longrightarrow \Omega$
and all t' *such that* $\Omega \vdash t' : \kappa'$*,*
it is the case that $[\Omega, \Theta]t' = [\theta][\Omega]t'$*, where* $\theta = \mathsf{mgu}(\sigma, t)$*.*

(2) If $\Gamma \,/\, \sigma \stackrel{\circ}{=} t : \kappa \dashv \bot$ *then* $\mathsf{mgu}(\sigma, t) = \bot$ *(that is, no most general unifier exists).*

Proof. First, we need to recall a few properties of term unification.

(i) If σ is a term, then $\mathsf{mgu}(\sigma, \sigma) = \mathit{id}$.

(ii) If f is a unary constructor, then $\mathsf{mgu}(f(\sigma), f(t)) = \mathsf{mgu}(\sigma, t)$, supposing that $\mathsf{mgu}(\sigma, t)$ exists.

(iii) If f is a binary constructor, and $\sigma = \mathsf{mgu}(f(\sigma_1, \sigma_2), f(t_1, t_2))$ and $\sigma_1 = \mathsf{mgu}(\sigma_1, t_1)$ and $\sigma_2 = \mathsf{mgu}([\sigma_1]\sigma_2, [\sigma_1]t_2)$, then $\sigma = \sigma_2 \circ \sigma_1 = \sigma_1 \circ \sigma_2$.

(iv) If $\alpha \notin FV(t)$, then $\mathsf{mgu}(\alpha, t) = (\alpha = t)$.

(v) If f is an n-ary constructor, and σ_i and t_i (for $i \le n$) have no unifier, then $f(\sigma_1, \ldots, \sigma_n)$ and $f(t_1, \ldots, t_n)$ have no unifier.

We proceed by induction on the derivation of $\Gamma \,/\, \sigma \stackrel{\circ}{=} t : \kappa \dashv \Delta^{\bot}$, proving both parts with a single induction.

- **Case** $$\dfrac{}{\Gamma \,/\, \alpha \stackrel{\circ}{=} \alpha : \kappa \dashv \Gamma}\ \textsf{ElimeqUvarRefl}$$

  Here we have $\Delta = \Gamma$, so we are in part (1).
  Let $\theta = \mathit{id}$ (which is $\mathsf{mgu}(\sigma, \sigma)$).
  We can easily show $[\mathit{id}][\Omega]\alpha = [\Omega, \alpha] = [\Omega, \cdot]\alpha$.

- **Case** $$\dfrac{}{\Gamma \,/\, \mathsf{zero} \stackrel{\circ}{=} \mathsf{zero} : \mathbb{N} \dashv \Gamma}\ \textsf{ElimeqZero}$$
  Similar to the ElimeqUvarRefl case.

- **Case** $$\dfrac{\Gamma \,/\, t_1 \stackrel{\circ}{=} t_2 : \mathbb{N} \dashv \Delta^{\bot}}{\Gamma \,/\, \mathsf{succ}(t_1) \stackrel{\circ}{=} \mathsf{succ}(t_2) : \mathbb{N} \dashv \Delta^{\bot}}\ \textsf{ElimeqSucc}$$

  We distinguish two subcases:

- **Case** $\Delta^\perp = \Delta$:
  Since we have the same output context in the conclusion and premise, the "for all $t' \ldots$" part follows immediately from the i.h. (1).
  The i.h. also gives us $\theta_0 = \mathsf{mgu}(t_1, t_2)$.
  Let $\theta = \theta_0$. By property (ii), $\mathsf{mgu}(t_1, t_2) = \mathsf{mgu}(\mathsf{succ}(t_1), \mathsf{succ}(t_2)) = \theta$.
- **Case** $\Delta^\perp = \perp$:

	$\Gamma \;/\; t_1 \doteq t_2 : \mathbb{N} \dashv \perp$	Subderivation
	$\mathsf{mgu}(t_1, t_2) = \perp$	By i.h. (2)
☞	$\mathsf{mgu}(\mathsf{succ}(t_1), \mathsf{succ}(t_2)) = \perp$	By contrapositive of property (ii)

- **Case** $$\dfrac{\alpha \notin FV(t) \qquad (\alpha = -) \notin \Gamma}{\Gamma \;/\; \alpha \doteq t : \kappa \dashv \Gamma, \alpha = t}\;\textsf{ElimeqUvarL}$$

  Here $\Delta \neq \perp$, so we are in part (1).

	$[\Omega, \alpha = t]t' = \big[[\Omega]t/\alpha\big][\Omega]t'$	By a property of substitution
	$= [\Omega][t/\alpha][\Omega]t'$	By a property of substitution
	$= [\Omega][\theta][\Omega]t'$	By $\mathsf{mgu}(\alpha, t) = (\alpha/t)$
☞	$= [\theta][\Omega]t'$	By a property of substitution (θ creates no evars)

- **Case** $$\dfrac{\alpha \notin FV(t) \qquad (\alpha = -) \notin \Gamma}{\Gamma \;/\; t \doteq \alpha : \kappa \dashv \Gamma, \alpha = t}\;\textsf{ElimeqUvarR}$$

  Similar to the ElimeqUvarL case.

- **Case**
  $$\dfrac{}{\Gamma \;/\; 1 \doteq 1 : \star \dashv \Gamma}\;\textsf{ElimeqUnit}$$

  Similar to the ElimeqUvarRefl case.

- **Case** $$\dfrac{\Gamma \;/\; \tau_1 \doteq \tau_1' : \star \dashv \Theta \qquad \Theta \;/\; [\Theta]\tau_1 \doteq [\Theta]\tau_2' : \star \dashv \Delta^\perp}{\Gamma \;/\; \tau_1 \oplus \tau_2 \doteq \tau_1' \oplus \tau_2' : \star \dashv \Delta^\perp}\;\textsf{ElimeqBin}$$

  Either $\Delta^\perp$ is some Δ, or it is $\perp$.

  - **Case** $\Delta^\perp = \Delta$:

	$\Gamma \;/\; \tau_1 \doteq \tau_1' : \star \dashv \Theta$	Subderivation
	$\Theta = (\Gamma, \Delta_1)$	By i.h. (1)
(IH-1st)	$[\Omega, \Delta_1]u_1 = [\theta_1][\Omega]u_1$	$''$ for all $\Omega \vdash u_1 : \kappa'$
	$\theta_1 = \mathsf{mgu}(\tau_1, \tau_1')$	$''$
	$\Theta \;/\; [\Theta]\tau_1 \doteq [\Theta]\tau_2' : \star \dashv \Delta$	Subderivation
	$\Delta = (\Theta, \Delta_2)$	By i.h. (1)
(IH-2nd)	$[\Omega, \Delta_1, \Delta_2]u_2 = [\theta_2][\Omega, \Delta_1]u_2$	$''$ for all $\Omega \vdash u_2 : \kappa'$
	$\theta_2 = \mathsf{mgu}(\tau_2, \tau_2')$	$''$

  Suppose $\Omega \vdash u : \kappa'$.

	$[\Omega, \Delta_1, \Delta_2]u = [\theta_2][\Omega, \Delta_1]u$	By (IH-2nd), with $u_2 = u$
	$= [\theta_2][\theta_1][\Omega]u$	By (IH-1st), with $u_1 = u$
☞	$= [\Omega][\theta_2 \circ \theta_1]u$	By a property of substitution
☞	$\theta_2 \circ \theta_1 = \mathsf{mgu}((\tau_1 \oplus \tau_2), (\tau_1' \oplus \tau_2'))$	By property (iii) of substitution

- **Case** $\Delta^\perp = \perp$:
  Use the i.h. (2) on the second premise to show $\mathsf{mgu}(\tau_2, \tau_2') = \perp$, then use property (v) of unification to show $\mathsf{mgu}((\tau_1 \oplus \tau_2), (\tau_1' \oplus \tau_2')) = \perp$.

- **Case** $\dfrac{\Gamma \mathbin{/} \tau_1 \mathrel{\mathring{=}} \tau_1' : \star \dashv \perp}{\Gamma \mathbin{/} \tau_1 \oplus \tau_2 \mathrel{\mathring{=}} \tau_1' \oplus \tau_2' : \star \dashv \perp}$ ElimeqBinBot

  Similar to the $\perp$ subcase for ElimeqSucc, but using property (v) instead of property (ii).

- **Case** $\dfrac{\sigma \# t}{\Gamma \mathbin{/} \sigma \mathrel{\mathring{=}} t : \kappa \dashv \perp}$ ElimeqClash

  Since $\sigma \# t$, we know σ and t have different head constructors, and thus no unifier. ☐

Theorem 6 (Soundness of Algorithmic Subtyping)**.**
If $[\Gamma]A = A$ *and* $[\Gamma]B = B$ *and* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type and* $\Delta \longrightarrow \Omega$ *and* $\Gamma \vdash A <:^{\pm} B \dashv \Delta$ *then* $[\Omega]\Delta \vdash [\Omega]A \leq^{\pm} [\Omega]B$.

Proof. By induction on the given derivation.

- **Case** $\dfrac{B \text{ not headed by } \forall \qquad \Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \vdash [\hat\alpha/\alpha]A_0 <:^- B \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Theta}{\Gamma \vdash \forall\alpha : \kappa.\, A_0 <:^- B \dashv \Delta} <:\forall\mathsf{L}$

  Let $\Omega' = (\Omega, |\blacktriangleright_{\hat\alpha}, \Theta|)$.

$$
\begin{array}{rl}
\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \vdash [\hat\alpha/\alpha]A_0 <:^- B \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Theta & \text{Subderivation} \\
\\
\Delta \longrightarrow \Omega & \text{Given} \\
(\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \longrightarrow \Omega' & \text{By Lemma 25 (Filling Completes)} \\
\\
\Gamma \vdash \forall\alpha : \kappa.\, A_0 \ \textit{type} & \text{Given} \\
\Gamma, \alpha : \kappa \vdash A_0 \ \textit{type} & \text{By inversion (ForallWF)} \\
\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \vdash [\hat\alpha/\alpha]A_0 \ \textit{type} & \text{By a property of substitution} \\
\Gamma \vdash B \ \textit{type} & \text{Given} \\
\\
[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \vdash [\Omega'][\hat\alpha/\alpha]A_0 \leq^- [\Omega']B & \text{By i.h.} \\
\Omega \vdash B \ \textit{type} & \text{By Lemma 36 (Extension Weakening (Sorts))} \\
[\Omega']B = [\Omega]B & \text{By Lemma 17 (Substitution Stability)} \\
[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \vdash [\Omega'][\hat\alpha/\alpha]A_0 \leq^- [\Omega]B & \text{By above equality} \\
[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \vdash \big[[\Omega']\hat\alpha/\alpha\big][\Omega']A_0 \leq^- [\Omega]B & \text{By distributivity of substitution} \\
\\
\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \vdash \hat\alpha : \kappa & \text{By VarSort} \\
\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \longrightarrow \Delta, \blacktriangleright_{\hat\alpha}, \Theta & \text{By Lemma 50 (Subtyping Extension)} \\
\Theta \text{ is soft} & \text{By Lemma 22 (Extension Inversion) (ii)} \\
\Delta, \blacktriangleright_{\hat\alpha}, \Theta \vdash \hat\alpha : \kappa & \text{By Lemma 36 (Extension Weakening (Sorts))} \\
(\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \longrightarrow \Omega' & \text{Above} \\
[\Omega']\Omega' \vdash [\Omega']\hat\alpha : \kappa & \text{By Lemma 14 (Substitution for Sorting)} \\
[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \vdash [\Omega']\hat\alpha : \kappa & \text{By Lemma 54 (Completing Stability)} \\
\\
[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \vdash \forall\alpha : \kappa.\, [\Omega']A_0 \leq^- [\Omega]B & \text{By } \leq\forall\mathsf{L} \\
[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Theta) \vdash \forall\alpha : \kappa.\, [\Omega, \alpha : \kappa]A_0 \leq^- [\Omega]B & \text{By Lemma 17 (Substitution Stability)} \\
[\Omega]\Delta \vdash \forall\alpha : \kappa.\, [\Omega, \alpha : \kappa]A_0 \leq^- [\Omega]B & \text{By Lemma 52 (Context Partitioning) + thinning} \\
[\Omega]\Delta \vdash \forall\alpha : \kappa.\, [\Omega]A_0 \leq^- [\Omega]B & \text{By def. of substitution} \\
[\Omega]\Delta \vdash [\Omega](\forall\alpha : \kappa.\, A_0) \leq^- [\Omega]B & \text{By def. of substitution}
\end{array}
$$

- **Case** $<:\exists\mathsf{R}$: Similar to the $<:\forall\mathsf{L}$ case.

- **Case** $\dfrac{\Gamma, \beta : \kappa \vdash A <:^- B_0 \dashv \Delta, \beta : \kappa, \Theta}{\Gamma \vdash A <:^- \forall\beta : \kappa.\, B_0 \dashv \Delta} <:\forall\mathsf{R}$

$\Gamma, \beta : \kappa \vdash A <:^- B_0 \dashv \Delta, \beta : \kappa, \Theta$	Subderivation
Let $\Omega_Z = \lvert\Theta\rvert$.	
Let $\Omega' = (\Omega, \beta : \kappa, \Omega_Z)$.	
$(\Delta, \beta : \kappa, \Theta) \longrightarrow \Omega'$	By Lemma 25 (Filling Completes)
$\Gamma \vdash A$ *type*	Given
$\Gamma, \beta : \kappa \vdash A$ *type*	By Lemma 35 (Suffix Weakening)
$\Gamma \vdash \forall \beta : \kappa.\, B_0$ *type*	Given
$\Gamma, \beta : \kappa \vdash B_0$ *type*	By inversion (ForallWF)
$[\Omega'](\Delta, \beta : \kappa, \Theta) \vdash [\Omega']A \leq^- [\Omega']B_0$	By i.h.
$\Gamma, \beta : \kappa \longrightarrow \Delta, \beta : \kappa, \Theta$	By Lemma 50 (Subtyping Extension)
Θ is soft	By Lemma 22 (Extension Inversion) (i)
$[\Omega, \beta : \kappa](\Delta, \beta : \kappa) \vdash [\Omega, \beta : \kappa]A \leq^- [\Omega, \beta : \kappa]B_0$	By Lemma 17 (Substitution Stability)
$[\Omega, \beta : \kappa](\Delta, \beta : \kappa) \vdash [\Omega]A \leq^- [\Omega]B_0$	By def. of substitution
$[\Omega]\Delta \vdash [\Omega]A \leq^- \forall \beta : \kappa.\, [\Omega]B_0$	By $\leq\forall$R
$[\Omega]\Delta \vdash [\Omega]A \leq^- [\Omega](\forall \beta : \kappa.\, B_0)$	By def. of substitution

- **Case** $<:\exists$L: Similar to the $<:\forall$R case.

- **Case** $$\dfrac{\Gamma \vdash A \equiv B \dashv \Delta}{\Gamma \vdash A <:^{\pm} B \dashv \Delta}\ <:\mathsf{Equiv}$$

	$\Gamma \vdash A \equiv B \dashv \Delta$	Subderivation
	$\Delta \longrightarrow \Omega$	Given
	$[\Omega]A = [\Omega]B$	By Lemma 86 (Soundness of Algorithmic Equivalence)
	$\Gamma \longrightarrow \Delta$	By Lemma 49 (Equivalence Extension)
	$\Gamma \vdash A$ *type*	Given
	$[\Omega]\Omega \vdash [\Omega]A$ *type*	By Lemma 16 (Substitution for Type Well-Formedness)
	$[\Omega]\Delta \vdash [\Omega]A$ *type*	By Lemma 54 (Completing Stability)
☞	$[\Omega]\Delta \vdash [\Omega]A \leq^{\pm} [\Omega]B$	By $\leq$Refl$\pm$

- **Case** $$\dfrac{\Gamma \vdash A <:^- B \dashv \Delta \qquad \begin{array}{l} \mathit{neg}(A) \\ \mathit{nonpos}(B) \end{array}}{\Gamma \vdash A <:^+ B \dashv \Delta}\ <:^-_+\mathsf{L}$$

	$\Gamma \vdash A <:^- B \dashv \Delta$	By inversion
	$\mathit{neg}(A)$	By inversion
	$\mathit{nonpos}(B)$	By inversion
	$\mathit{nonpos}(A)$	since $\mathit{neg}(A)$
	$[\Omega]\Gamma \vdash [\Omega]A \leq^- [\Omega]B$	By induction
☞	$[\Omega]\Gamma \vdash [\Omega]A \leq^+ [\Omega]B$	By $\leq^-_+$

- **Case** $$\dfrac{\Gamma \vdash A <:^- B \dashv \Delta \qquad \begin{array}{l} \mathit{nonpos}(A) \\ \mathit{neg}(B) \end{array}}{\Gamma \vdash A <:^+ B \dashv \Delta}\ <:^-_+\mathsf{R}$$

  Similar to the $<:^-_+$L case.

- **Case**

$$\frac{\Gamma \vdash A <:^{+} B \dashv \Delta \qquad \begin{array}{l} \mathit{pos}(A) \\ \mathit{nonneg}(B) \end{array}}{\Gamma \vdash A <:^{-} B \dashv \Delta} <:^{+}_{-}\mathsf{L}$$

Similar to the $<:^{-}_{+}\mathsf{L}$ case.

- **Case**

$$\frac{\Gamma \vdash A <:^{+} B \dashv \Delta \qquad \begin{array}{l} \mathit{nonneg}(A) \\ \mathit{pos}(B) \end{array}}{\Gamma \vdash A <:^{-} B \dashv \Delta} <:^{+}_{-}\mathsf{R}$$

Similar to the $<:^{-}_{+}\mathsf{L}$ case.

□

J′.4 Soundness of Typing

Theorem 7 (Soundness of Match Coverage)**.**

1. *If* $\Gamma \vdash \Pi$ *covers* $\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *then* $[\Omega]\Gamma \vdash \Pi$ *covers* $\vec{A}$.
2. *If* $\Gamma \;/\; P \vdash \Pi$ *covers* $\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *and* $[\Gamma]P = P$ *then* $[\Omega]\Gamma \;/\; P \vdash \Pi$ *covers* $\vec{A}$.

Proof. By mutual induction on the given algorithmic coverage derivation.

1. - **Case**

$$\frac{}{\Gamma \vdash \cdot \Rightarrow e_1 \mid \ldots \textit{ covers } \cdot}\ \mathsf{CoversEmpty}$$

$[\Omega]\Gamma \vdash \cdot \Rightarrow e_1 \mid \ldots$ *covers* $\cdot$ By DeclCoversEmpty

   - **Cases** CoversVar, Covers1, Covers$\times$, Covers$+$, Covers$\exists$, Covers$\wedge$, CoversVec:
     Use the i.h. and apply the corresponding declarative rule.

2. - **Case**

$$\frac{\Gamma \;/\; [\Gamma]t_1 \mathrel{\mathring{=}} [\Gamma]t_2 : \kappa \dashv \Delta \qquad \Delta \vdash [\Delta]\Pi \textit{ covers } [\Delta]\vec{A}}{\Gamma \;/\; t_1 = t_2 \vdash \Pi \textit{ covers } \vec{A}}\ \mathsf{CoversEq}$$

$\Gamma \;/\; [\Gamma]t_1 \mathrel{\mathring{=}} [\Gamma]t_2 : \kappa \dashv \Delta$	Subderivation
$\Delta \vdash [\Delta]\Pi$ *covers* $[\Delta]\vec{A}$	Subderivation
$[\Omega]\Delta \vdash [\Delta]\Pi$ *covers* $[\Delta]A_0, [\Delta]\vec{A})$	By i.h.
$\Delta = (\Gamma, \Theta)$	By Lemma 88 (Soundness of Equality Elimination) (1)
$\mathsf{mgu}(t_1, t_2) = \theta$	″
$\ldots$	″
$[\Omega]\Delta = [\theta][\Omega]\Gamma$	By Lemma 93 (Substitution Upgrade) (iii)
$[\Delta]\Pi = [\theta]\Pi$	By Lemma 93 (Substitution Upgrade) (iv)
$([\Delta]\vec{A}) = ([\theta]A_0, [\theta]\vec{A})$	By Lemma 93 (Substitution Upgrade) (i)
$[\theta][\Omega]\Gamma \vdash [\theta]\Pi$ *covers* $[\theta]\vec{A}$	By above equalities
☞ $[\Omega]\Gamma \;/\; t_1 = t_2 \vdash \Pi$ *covers* $\vec{A}$	By DeclCoversEq

- **Case** $$\dfrac{\Gamma \mathrel{/} [\Gamma]t_1 \stackrel{\circ}{=} [\Gamma]t_2 : \kappa \dashv \bot}{\Gamma \mathrel{/} t_1 = t_2 \vdash \Pi \; \textit{covers} \; \vec{A}} \; \mathsf{CoversEqBot}$$

$\Gamma \mathrel{/} [\Gamma]t_1 \stackrel{\circ}{=} [\Gamma]t_2 : \kappa \dashv \bot$ Subderivation
$\mathsf{mgu}([\Gamma]t_1, [\Gamma]t_2) = \bot$ By Lemma 88 (Soundness of Equality Elimination) (2)
$\mathsf{mgu}(t_1, t_2) = \bot$ By given equality
☞ $[\Omega]\Gamma \mathrel{/} t_1 = t_2 \vdash \Pi \; \textit{covers} \; \vec{A}$ By DeclCoversEqBot □

Lemma 89 (Well-formedness of Algorithmic Typing)**.**
Given Γ *ctx:*

(i) If $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$ *then* $\Delta \vdash A\ p$ *type.*

(ii) If $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $\Delta \vdash B\ q$ *type.*

Proof. 1. Suppose $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$:

- **Case** $$\dfrac{(x : A\ p) \in \Gamma}{\Gamma \vdash x \Rightarrow [\Gamma]A\ p \dashv \Gamma} \; \mathsf{Var}$$

$\Gamma = (\Gamma_0, x : A\ p, \Gamma_1)$ $(x : A\ p) \in \Gamma$
$\Gamma \vdash A\ p$ *type* Follows from Γ *ctx*

- **Case** $$\dfrac{\Gamma \vdash A! \; \textit{type} \qquad \Gamma \vdash e \Leftarrow [\Gamma]A\ ! \dashv \Delta}{\Gamma \vdash (e : A) \Rightarrow [\Delta]A\ ! \dashv \Delta} \; \mathsf{Anno}$$

$\Gamma \vdash A\ !$ *type* By inversion
$\Gamma \longrightarrow \Delta$ By Lemma 51 (Typing Extension)
$\Delta \vdash A\ !$ *type* By Lemma 41 (Extension Weakening for Principal Typing)
☞ $\Delta \vdash [\Delta]A\ !$ *type* By Lemma 39 (Principal Agreement) (i)

- **Case** $$\dfrac{\Gamma \vdash e \Rightarrow A\ p \dashv \Theta \qquad \Theta \vdash s : [\Theta]A\ p \gg C\ q \dashv \Delta \qquad \begin{array}{l} p = \not{!} \text{ or } q = ! \\ \text{or } \mathsf{FEV}([\Delta]C) \neq \emptyset \end{array}}{\Gamma \vdash e\, s \Rightarrow C\ q \dashv \Delta} \; {\to}\mathsf{E}$$

$\Gamma \vdash e \Rightarrow A\ p \dashv \Theta$ By inversion
$\Theta \vdash A\ p$ *type* By induction
$\Theta \vdash [\Theta]A\ p$ *type* By Lemma 40 (Right-Hand Subst. for Principal Typing)
Θ *ctx* By implicit assumption
$\Theta \vdash s : [\Theta]A\ p \gg C\ q \dashv \Delta$ By inversion
☞ $\Delta \vdash C\ q$ *type* By mutual induction

- **Case** $$\dfrac{\Gamma \vdash e \Rightarrow A\ ! \dashv \Theta \qquad \Theta \vdash s : [\Theta]A\ ! \gg C \ \dashv \Delta \qquad \mathsf{FEV}([\Delta]C) = \emptyset}{\Gamma \vdash e\, s \Rightarrow C\ ! \dashv \Delta} \; {\to}\mathsf{E}\text{-}!$$

$\Gamma \vdash e \Rightarrow A\ p \dashv \Theta$ By inversion
$\Theta \vdash A\ p$ *type* By induction
$\Theta \vdash [\Theta]A\ p$ *type* By Lemma 40 (Right-Hand Subst. for Principal Typing)
Θ *ctx* By implicit assumption
$\Theta \vdash s : [\Theta]A\ p \gg C \ \dashv \Delta$ By inversion
$\Delta \vdash C$ *type* By mutual induction
$\mathsf{FEV}([\Delta]C) = \emptyset$ By inversion
$\Delta \vdash C!$ *type* By PrincipalWF

2. Suppose $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$ and $\Gamma \vdash A\ p\ \mathit{type}$:

- **Case**

$$\frac{}{\Gamma \vdash \cdot : A\ p \gg A\ p \dashv \Gamma}\ \mathsf{EmptySpine}$$

$\Gamma \vdash A\ p\ \mathit{type}$ Given

- **Case**

$$\frac{\Gamma \vdash e \Leftarrow A\ p \dashv \Theta \qquad \Theta \vdash s : [\Theta]B\ p \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s : A \to B\ p \gg C\ q \dashv \Delta}\ \to\!\mathsf{Spine}$$

$\Gamma \vdash A \to B\ p\ \mathit{type}$	Given
$\Gamma \vdash B\ p\ \mathit{type}$	By Lemma 42 (Inversion of Principal Typing)
$\Theta \vdash B\ p\ \mathit{type}$	By Lemma 41 (Extension Weakening for Principal Typing)
$\Theta \vdash [\Theta]B\ p\ \mathit{type}$	By Lemma 40 (Right-Hand Subst. for Principal Typing)
$\Delta \vdash C\ q\ \mathit{type}$	By induction

- **Case**

$$\frac{\Gamma, \hat\alpha : \kappa \vdash e\ s : [\hat\alpha/\alpha]A \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s : \forall\alpha : \kappa.\ A\ p \gg C\ q \dashv \Delta}\ \forall\mathsf{Spine}$$

	$\Gamma \vdash \forall\alpha : \kappa.\ A\ p\ \mathit{type}$	Given
	$\Gamma \vdash \forall\alpha : \kappa.\ A\ \mathit{type}$	By inversion
	$\Gamma, \alpha : \kappa \vdash A\ \mathit{type}$	By inversion
	$\Gamma, \hat\alpha : \kappa, \alpha : \kappa \vdash A\ \mathit{type}$	By weakening
	$\Gamma, \hat\alpha : \kappa \vdash [\hat\alpha/\alpha]A\ \mathit{type}$	By substitution
☞	$\Delta \vdash C\ q\ \mathit{type}$	By induction

- **Case**

$$\frac{\Gamma \vdash P\ \mathit{true} \dashv \Theta \qquad \Theta \vdash e\ s : [\Theta]A\ p \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s : P \supset A\ p \gg C\ q \dashv \Delta}\ \supset\!\mathsf{Spine}$$

	$\Gamma \vdash P \supset A\ p\ \mathit{type}$	Given
	$\Gamma \vdash P\ \mathit{prop}$	By Lemma 42 (Inversion of Principal Typing)
	$\Gamma \vdash A\ p\ \mathit{type}$	"
	$\Gamma \longrightarrow \Theta$	By Lemma 47 (Checkprop Extension)
	$\Theta \vdash A\ p\ \mathit{type}$	By Lemma 41 (Extension Weakening for Principal Typing)
	$\Theta \vdash [\Theta]A\ p\ \mathit{type}$	By Lemma 40 (Right-Hand Subst. for Principal Typing)
☞	$\Delta \vdash C\ q\ \mathit{type}$	By induction

- **Case**

$$\frac{\overbrace{\Gamma[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \to \hat\alpha_2]}^{\Theta} \vdash e\ s : (\hat\alpha_1 \to \hat\alpha_2) \gg C \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash e\ s : \hat\alpha \gg C \dashv \Delta}\ \hat\alpha\mathsf{Spine}$$

	$\Theta \vdash \hat\alpha_1 \to \hat\alpha_2\ \mathit{type}$	By rules
☞	$\Delta \vdash C\ q\ \mathit{type}$	By induction

□

Theorem 8 (Soundness of Algorithmic Typing).
Given $\Delta \longrightarrow \Omega$*:*

(i) If $\Gamma \vdash e \Leftarrow A\ p \dashv \Delta$ *and* $\Gamma \vdash A\ p\ \mathit{type}$ *then* $[\Omega]\Delta \vdash [\Omega]e \Leftarrow [\Omega]A\ p$.

(ii) If $\Gamma \vdash e \Rightarrow A\ p \dashv \Delta$ *then* $[\Omega]\Delta \vdash [\Omega]e \Rightarrow [\Omega]A\ p$.

(iii) *If* $\Gamma \vdash s : A\ p \gg B\ q \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $[\Omega]\Delta \vdash [\Omega]s : [\Omega]A\ p \gg [\Omega]B\ q$.

(iv) *If* $\Gamma \vdash s : A\ p \gg B\ \lceil q \rceil \dashv \Delta$ *and* $\Gamma \vdash A\ p$ *type then* $[\Omega]\Delta \vdash [\Omega]s : [\Omega]A\ p \gg [\Omega]B\ \lceil q \rceil$.

(v) *If* $\Gamma \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$ *and* $\Gamma \vdash \vec{A}\ \mathbf{!}$ *types and* $[\Gamma]\vec{A} = \vec{A}$ *and* $\Gamma \vdash C\ p$ *type*
then $[\Omega]\Delta \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$.

(vi) *If* $\Gamma\ /\ P \vdash \Pi :: \vec{A} \Leftarrow C\ p \dashv \Delta$ *and* $\Gamma \vdash P$ *prop and* $\mathsf{FEV}(P) = \emptyset$ *and* $[\Gamma]P = P$
and $\Gamma \vdash \vec{A}\ \mathbf{!}$ *types and* $\Gamma \vdash C\ p$ *type*
then $[\Omega]\Delta\ /\ [\Omega]P \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$.

Proof. By induction, using the measure in Definition 7.

- **Case** $$\dfrac{(x : A\ p) \in \Gamma}{\Gamma \vdash x \Rightarrow [\Gamma]A\ p \dashv \Gamma}\ \mathsf{Var}$$

	$(x : A\ p) \in \Gamma$	Premise
	$(x : A\ p) \in \Delta$	$\Gamma = \Delta$
	$\Delta \longrightarrow \Omega$	Given
	$(x : [\Omega]A\ p) \in [\Omega]\Gamma$	By Lemma 9 (Uvar Preservation) (ii)
	$[\Omega]\Gamma \vdash [\Omega]x \Rightarrow [\Omega]A\ p$	By DeclVar
	$\Delta \longrightarrow \Omega$	Given
	$\Gamma \longrightarrow \Omega$	$\Gamma = \Delta$
	$[\Omega]A = [\Omega][\Gamma]A$	By Lemma 29 (Substitution Monotonicity) (iii)
☞	$[\Omega]\Gamma \vdash [\Omega]x \Rightarrow [\Omega][\Gamma]A\ p$	By above equality

- **Case** $$\dfrac{\Gamma \vdash e \Rightarrow A\ q \dashv \Theta \qquad \Theta \vdash A <:^{\pm} B \dashv \Delta}{\Gamma \vdash e \Leftarrow B\ p \dashv \Delta}\ \mathsf{Sub}$$

	$\Gamma \vdash e \Rightarrow A\ q \dashv \Theta$	Subderivation
	$\Theta \vdash A <:^{\pm} B \dashv \Delta$	Subderivation
	$\Theta \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
	$\Delta \longrightarrow \Omega$	Given
	$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
	$[\Omega]\Theta \vdash [\Omega]e \Rightarrow [\Omega]A\ q$	By i.h.
	$[\Omega]\Theta = [\Omega]\Delta$	By Lemma 56 (Confluence of Completeness)
	$[\Omega]\Delta \vdash [\Omega]e \Rightarrow [\Omega]A\ q$	By above equality
	$\Theta \vdash A <:^{\pm} B \dashv \Delta$	Subderivation
	$[\Omega]\Delta \vdash [\Omega]A \leq^{\pm} [\Omega]B$	By Theorem 6
☞	$[\Omega]\Delta \vdash [\Omega]e \Leftarrow [\Omega]B\ p$	By DeclSub

- **Case** $$\dfrac{\Gamma \vdash A_0\mathbf{!}\ \mathit{type} \qquad \Gamma \vdash e_0 \Leftarrow [\Gamma]A_0\ \mathbf{!} \dashv \Delta}{\Gamma \vdash (e_0 : A_0) \Rightarrow [\Delta]A_0\ \mathbf{!} \dashv \Delta}\ \mathsf{Anno}$$

$\Gamma \vdash e_0 \Leftarrow [\Gamma]A_0\ \mathbf{!} \dashv \Delta$	Subderivation
$[\Omega]\Delta \vdash [\Omega]e_0 \Leftarrow [\Omega][\Gamma]A_0\ \mathbf{!}$	By i.h.
$\Gamma \vdash A_0\mathbf{!}\ \mathit{type}$	Subderivation
$\Gamma \vdash A_0\ \mathit{type}$	By inversion
$\mathsf{FEV}(A_0) = \emptyset$	"

$\Gamma \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
$\Delta \longrightarrow \Omega$	Given
$\Gamma \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Omega \vdash A_0$ *type*	By Lemma 36 (Extension Weakening (Sorts))
$[\Omega]\Omega \vdash [\Omega]A_0$ *type*	By Lemma 16 (Substitution for Type Well-Formedness)
$[\Omega]\Omega = [\Omega]\Delta$	By Lemma 54 (Completing Stability)
$[\Omega]\Delta \vdash [\Omega]A_0$ *type*	By above equality
$[\Omega][\Gamma]A_0 = [\Omega]A_0$	By Lemma 29 (Substitution Monotonicity) (iii)
$[\Omega]\Delta \vdash [\Omega]e_0 \Leftarrow [\Omega]A_0\ !$	By above equality
$[\Omega]\Delta \vdash ([\Omega]e_0 : [\Omega]A_0) \Rightarrow [\Omega]A_0\ !$	By DeclAnno
$[\Omega]A_0 = A_0$	From definition of substitution
☞ $[\Omega]\Delta \vdash [\Omega](e_0 : A_0) \Rightarrow [\Omega]A_0\ !$	By above equality

- **Case**

$$\frac{}{\Gamma \vdash () \Leftarrow 1\ p \dashv \underbrace{\Gamma}_{\Delta}}\ 1\mathsf{I}$$

$[\Omega]\Delta \vdash () \Leftarrow 1\ p$	By Decl1I
☞ $[\Omega]\Delta \vdash [\Omega]() \Leftarrow [\Omega]1\ p$	By definition of substitution

- **Case**

$$\frac{}{\Gamma_0[\hat{\alpha} : \star] \vdash () \Leftarrow \hat{\alpha}\ \not{!} \dashv \underbrace{\Gamma_0[\hat{\alpha} : \star = 1]}_{\Delta}}\ 1\mathsf{I}\hat{\alpha}$$

$\Gamma_0[\hat{\alpha} : \star = 1] \longrightarrow \Omega$	Given
$[\Omega]\hat{\alpha} = [\Omega][\Delta]\hat{\alpha}$	By Lemma 29 (Substitution Monotonicity) (i)
$= [\Omega]1$	By definition of context application
$= 1$	By definition of context application
$[\Omega]\Delta \vdash () \Leftarrow 1\ \not{!}$	By Decl1I
☞ $[\Omega]\Delta \vdash [\Omega]() \Leftarrow [\Omega]\hat{\alpha}\ \not{!}$	By above equality

- **Case**

$$\frac{v\ \mathit{chk\text{-}I} \qquad \Gamma, \alpha : \kappa \vdash v \Leftarrow A_0\ p \dashv \Delta, \alpha : \kappa, \Theta}{\Gamma \vdash v \Leftarrow \forall \alpha : \kappa.\ A_0\ p \dashv \Delta}\ \forall\mathsf{I}$$

$\Delta \longrightarrow \Omega$	Given
$\Delta, \alpha \longrightarrow \Omega, \alpha$	By $\longrightarrow$Uvar
$\Gamma, \alpha \longrightarrow \Delta, \alpha, \Theta$	By Lemma 51 (Typing Extension)
Θ soft	By Lemma 22 (Extension Inversion) (i) (with $\Gamma_R = \cdot$, which is soft)
$\underbrace{\Delta, \alpha, \Theta}_{\Delta'} \longrightarrow \underbrace{\Omega, \alpha, \lvert\Theta\rvert}_{\Omega'}$	By Lemma 25 (Filling Completes)
$\Gamma, \alpha \vdash v \Leftarrow A_0\ p \dashv \Delta'$	Subderivation
$[\Omega']\Delta' \vdash [\Omega]v \Leftarrow [\Omega']A_0\ p$	By i.h.
$[\Omega']A_0 = [\Omega]A_0$	By Lemma 17 (Substitution Stability)
$[\Omega']\Delta' \vdash [\Omega]v \Leftarrow [\Omega]A_0\ p$	By above equality

$\underbrace{\Delta, \alpha, \Theta}_{\Delta'} \longrightarrow \underbrace{\Omega, \alpha, \lvert\Theta\rvert}_{\Omega'}$	Above
Θ is soft	Above
$[\Omega']\Delta' = ([\Omega]\Delta, \alpha)$	By Lemma 53 (Softness Goes Away)
$[\Omega]\Delta, \alpha \vdash [\Omega]v \Leftarrow [\Omega]A_0\ p$	By above equality
$[\Omega]\Delta \vdash [\Omega]v \Leftarrow \forall\alpha.\ [\Omega]A_0\ p$	By Decl∀I
☞ $[\Omega]\Delta \vdash [\Omega]v \Leftarrow [\Omega](\forall\alpha.\ A_0)\ p$	By definition of substitution

- **Case** $\dfrac{\Gamma, \hat{\alpha} : \kappa \vdash e\ s_0 : [\hat{\alpha}/\alpha]A_0\ \not{p} \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s_0 : \forall\alpha : \kappa.\ A_0\ p \gg C\ q \dashv \Delta}$ ∀Spine

$\Gamma, \hat{\alpha} : \kappa \vdash e\ s_0 : [\hat{\alpha}/\alpha]A_0\ \not{p} \gg C\ q \dashv \Delta$	Subderivation
$[\Omega]\Delta \vdash [\Omega](e\ s_0) : [\Omega][\hat{\alpha}/\alpha]A_0\ \not{p} \gg [\Omega]C\ q$	By i.h.
$[\Omega]\Delta \vdash [\Omega](e\ s_0) : \big[[\Omega]\hat{\alpha}/\alpha\big][\Omega]A_0\ \not{p} \gg [\Omega]C\ q$	By a property of substitution

$\Gamma, \hat{\alpha} : \kappa \vdash \hat{\alpha} : \kappa$	By VarSort
$\Gamma, \hat{\alpha} : \kappa \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
$\Delta \vdash \hat{\alpha} : \kappa$	By Lemma 36 (Extension Weakening (Sorts))
$\Delta \longrightarrow \Omega$	Given
$[\Omega]\Delta \vdash [\Omega]\hat{\alpha} : \kappa$	By Lemma 58 (Bundled Substitution for Sorting)

$[\Omega]\Delta \vdash [\Omega](e\ s_0) : \forall\alpha : \kappa.\ [\Omega]A_0\ p \gg [\Omega]C\ q$	By Decl∀Spine
☞ $[\Omega]\Delta \vdash [\Omega](e\ s_0) : [\Omega](\forall\alpha : \kappa.\ A_0)\ p \gg [\Omega]C\ q$	By def. of subst.

- **Case** $$\dfrac{e\ \textit{chk-I} \qquad \Gamma \vdash P\ \textit{true} \dashv \Theta \qquad \Theta \vdash e \Leftarrow [\Theta]A_0\ p \dashv \Delta}{\Gamma \vdash e \Leftarrow A_0 \wedge P\ p \dashv \Delta}\ \wedge\mathsf{I}$$

$\Gamma \vdash P\ \textit{true} \dashv \Theta$	Subderivation
$\Delta \longrightarrow \Omega$	Given
$\Theta \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$[\Omega]\Theta \vdash [\Omega]P\ \textit{true}$	By Lemma 87 (Soundness of Checkprop)
$[\Omega]\Delta \vdash [\Omega]P\ \textit{true}$	By Lemma 56 (Confluence of Completeness)
$\Theta \vdash e \Leftarrow [\Theta]A_0\ p \dashv \Delta$	Subderivation
$[\Omega]\Delta \vdash [\Omega]e \Leftarrow ([\Omega][\Theta]A_0)\ p$	By i.h.
$[\Omega]\Delta \vdash [\Omega]e \Leftarrow ([\Omega][\Theta]A_0) \wedge [\Omega]P\ p$	By Decl$\wedge$I
$[\Omega][\Theta]A_0 = [\Omega]A_0$	By Lemma 29 (Substitution Monotonicity) (iii)
$[\Omega]\Delta \vdash [\Omega]e \Leftarrow ([\Omega]A_0) \wedge [\Omega]P\ p$	By above equality
☞ $[\Omega]\Delta \vdash [\Omega]e \Leftarrow [\Omega](A_0 \wedge P)\ p$	By def. of substitution

- **Case** $$\dfrac{\Gamma \vdash t = \mathsf{zero}\ \textit{true} \dashv \Delta}{\Gamma \vdash [] \Leftarrow (\mathsf{Vec}\ t\ A)\ p \dashv \Delta}\ \mathsf{Nil}$$

$\Gamma \vdash t = \mathsf{zero}\ \textit{true} \dashv \Delta$	Subderivation
$\Delta \longrightarrow \Omega$	Given
$[\Omega]\Delta \vdash [\Omega](t = \mathsf{zero})\ \textit{true}$	By Lemma 87 (Soundness of Checkprop)
$[\Omega]\Delta \vdash [\Omega]t = \mathsf{zero}\ \textit{true}$	By def. of substitution
☞ $[\Omega]\Delta \vdash [\Omega]\,[] \Leftarrow (\mathsf{Vec}\ [\Omega]t\ [\Omega]A)\ p$	By DeclNil

- **Case** $$\dfrac{\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} \vdash t = \mathsf{succ}(\hat\alpha)\ \textit{true} \dashv \Gamma' \qquad \begin{array}{l}\Gamma' \vdash e_1 \Leftarrow [\Gamma']A_0\ p \dashv \Theta \\ \Theta \vdash e_2 \Leftarrow [\Theta](\mathsf{Vec}\ \hat\alpha\ A_0)\ \not{p} \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Delta'\end{array}}{\Gamma \vdash e_1 :: e_2 \Leftarrow (\mathsf{Vec}\ t\ A_0)\ p \dashv \Delta}\ \mathsf{Cons}$$

	$\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} \vdash t = \mathsf{succ}(\hat\alpha)\ \mathit{true} \dashv \Gamma'$	Subderivation
	$\Delta \longrightarrow \Omega$	Given
	$\Gamma' \longrightarrow \Theta$	By Lemma 51 (Typing Extension)
	$\Theta \longrightarrow \Delta, \blacktriangleright_{\hat\alpha}, \Delta'$	By Lemma 51 (Typing Extension)
	$\Delta, \blacktriangleright_{\hat\alpha}, \Delta' \longrightarrow \Omega'$	By Lemma 25 (Filling Completes)
	$\Gamma' \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
	$[\Omega']\Gamma' \vdash [\Omega'](t = \mathsf{succ}(\hat\alpha))\ \mathit{true}$	By Lemma 87 (Soundness of Checkprop)
	$[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Delta') \vdash [\Omega'](t = \mathsf{succ}(\hat\alpha))\ \mathit{true}$	By Lemma 56 (Confluence of Completeness)
	$[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Delta') \vdash [\Omega](t = \mathsf{succ}(\hat\alpha))\ \mathit{true}$	By Lemma 17 (Substitution Stability)
	$[\Omega]\Delta \vdash [\Omega](t = \mathsf{succ}(\hat\alpha))\ \mathit{true}$	By Lemma 52 (Context Partitioning) + thinning
1	$[\Omega]\Delta \vdash ([\Omega]t) = \mathsf{succ}([\Omega]\hat\alpha)\ \mathit{true}$	By def. of substitution
	$\Gamma' \vdash e_1 \Leftarrow [\Gamma']A_0\ p \dashv \Theta$	Subderivation
	$[\Omega']\Theta \vdash [\Omega']e_1 \Leftarrow ([\Omega'][\Gamma']A_0)\ p$	By i.h.
	$[\Omega'][\Gamma']A_0 = [\Omega']A_0$	By Lemma 29 (Substitution Monotonicity) (iii)
	$[\Omega']\Theta \vdash [\Omega']e_1 \Leftarrow [\Omega']A_0\ p$	By above equality
2	$[\Omega]\Delta \vdash [\Omega]e_1 \Leftarrow [\Omega]A_0\ p$	Similar to above
	$\Theta \vdash e_2 \Leftarrow [\Theta](\mathsf{Vec}\ \hat\alpha\ A_0)\ \not{p} \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Delta'$	Subderivation
	$[\Omega'](\Delta, \blacktriangleright_{\hat\alpha}, \Delta') \vdash [\Omega']e_2 \Leftarrow [\Omega'][\Theta](\mathsf{Vec}\ \hat\alpha\ A_0)\ \not{p}$	By i.h.
	$[\Omega]\Delta \vdash [\Omega]e_2 \Leftarrow [\Omega](\mathsf{Vec}\ \hat\alpha\ A_0)\ \not{p}$	Similar to above
3	$[\Omega]\Delta \vdash [\Omega]e_2 \Leftarrow (\mathsf{Vec}\ ([\Omega]\hat\alpha)\ [\Omega]A_0)\ p$	By def. of substitution
	$[\Omega]\Delta \vdash ([\Omega]e_1) :: [\Omega]e_2 \Leftarrow \mathsf{Vec}\ ([\Omega]t)\ [\Omega]A_0\ p$	By DeclCons (premises: 1, 2, 3)
☞	$[\Omega]\Delta \vdash [\Omega](e_1 :: e_2) \Leftarrow [\Omega](\mathsf{Vec}\ t\ A_0)\ p$	By def. of substitution

- **Case** $$\dfrac{v\ \textit{chk-I} \qquad \Gamma, \blacktriangleright_P \ /\ P \dashv \Theta^+ \qquad \Theta^+ \vdash v \Leftarrow [\Theta^+]A_0\ ! \dashv \Delta, \blacktriangleright_P, \Delta'}{\Gamma \vdash v \Leftarrow P \supset A_0\ ! \dashv \Delta}\supset\mathsf{I}$$

$$
\begin{array}{rll}
\Gamma \vdash A\ !\ \textit{type} & & \text{Given} \\
\mathsf{FEV}([\Gamma]A) = \emptyset & & \text{By inversion on rule PrincipalWF} \\
\mathsf{FEV}([\Gamma]P) = \emptyset & & A = (P \supset A_0) \\[1ex]
\Gamma, \blacktriangleright_P \ /\ P \dashv \Theta^+ & & \text{Subderivation} \\
\Gamma, \blacktriangleright_P \ /\ \sigma \mathrel{\hat{=}} t : \kappa \dashv \Theta^+ & & \text{By inversion} \\
\mathsf{FEV}([\Gamma]\sigma) \cup \mathsf{FEV}([\Gamma]t) = \emptyset & & \text{By } \mathsf{FEV}([\Gamma]P) = \emptyset \text{ above} \\[1ex]
\Theta^+ = (\Gamma, \blacktriangleright_P, \Theta) & & \text{By Lemma 88 (Soundness of Equality Elimination)} \\
[\Omega', \Theta]t' = [\theta][\Gamma, \blacktriangleright_P]t' & & {''}\quad \text{(for all } \Omega' \text{ extending } (\Gamma, \blacktriangleright_P) \text{ and } t' \text{ s.t. } \Omega' \vdash t' : \kappa') \\
\theta = \mathsf{mgu}(\sigma, t) & & {''} \\[1ex]
\Delta \longrightarrow \Omega & & \text{Given} \\
\Theta^+ \longrightarrow \Delta, \blacktriangleright_P, \Delta' & & \text{By Lemma 51 (Typing Extension)} \\
\Gamma, \blacktriangleright_P, \Theta \longrightarrow \Delta, \blacktriangleright_P, \Delta' & & \text{By above equalities} \\
\text{Let } \Omega^+ = (\Omega, \blacktriangleright_P, \Delta'). & & \\
\Delta, \blacktriangleright_P, \Theta \longrightarrow \Omega, \blacktriangleright_P, \Delta' & & \text{By repeated } {\longrightarrow}\mathsf{Eqn} \\
\Theta^+ \longrightarrow \Omega^+ & & \text{By Lemma 33 (Extension Transitivity)} \\[1ex]
[\Omega', \Theta]B = [\theta][\Gamma, \blacktriangleright_P]B & & \text{By Lemma 93 (Substitution Upgrade) (i)} \\
& & \quad \text{(for all } \Omega' \text{ extending } (\Gamma, \blacktriangleright_P \text{ and } B \text{ s.t. } \Omega' \vdash B : \kappa')
\end{array}
$$

$$
\begin{array}{rl}
\Theta^+ \vdash v \Leftarrow [\Theta^+]A_0\ ! \dashv \Delta, \blacktriangleright_P, \Delta' & \text{Subderivation} \\
[\Omega^+](\Delta, \blacktriangleright_P, \Delta') \vdash [\Omega]v \Leftarrow [\Omega^+][\Theta^+]A_0\ ! & \text{By i.h.} \\[1ex]
\Gamma, \blacktriangleright_P, \Theta \longrightarrow \Omega, \blacktriangleright_P, \Delta' & \text{By Lemma 33 (Extension Transitivity)} \\
\Gamma \longrightarrow \Omega & \text{By Lemma 22 (Extension Inversion)} \\
[\Omega^+][\Theta^+]A_0 = [\Omega^+]A_0 & \text{By Lemma 29 (Substitution Monotonicity)} \\
= [\theta][\Omega, \blacktriangleright_P]A_0 & \text{Above, with } (\Omega, \blacktriangleright_P) \text{ as } \Omega' \text{ and } A_0 \text{ as } B \\
= [\theta][\Omega]A_0 & \text{By def. of substitution} \\[1ex]
[\Omega, \blacktriangleright_P, \Theta](\Delta, \blacktriangleright_P, \Delta') = [\theta][\Omega]\Delta & \text{By Lemma 93 (Substitution Upgrade) (iii)}
\end{array}
$$

$$
\begin{array}{rl}
[\theta][\Omega]\Delta \vdash [\Omega][\theta]v \Leftarrow [\theta][\Omega]A_0\ ! & \text{By above equalities} \\[1ex]
[\Omega^+](\Delta, \blacktriangleright_P, \Delta') \ /\ (\sigma = t) \vdash [\Omega]v \Leftarrow [\Omega]A_0\ ! & \text{By DeclCheckUnify} \\
[\Omega^+](\Delta, \blacktriangleright_P, \Delta') = [\Omega]\Delta & \text{From def. of context application} \\
[\Omega]\Delta \ /\ (\sigma = t) \vdash [\Omega]v \Leftarrow [\Omega]A_0\ ! & \text{By above equality} \\
[\Omega]\Delta \vdash [\Omega]v \Leftarrow (\sigma = t) \supset [\Omega]A_0\ ! & \text{By Decl}{\supset}\mathsf{I} \\
[\Omega]\Delta \vdash [\Omega]v \Leftarrow ([\Omega]\sigma = [\Omega]t) \supset [\Omega]A_0\ ! & \text{By FEV condition above}
\end{array}
$$

- **Case** $$\dfrac{v\ \textit{chk-I} \qquad \Gamma, \blacktriangleright_P \ /\ P \dashv \bot}{\Gamma \vdash v \Leftarrow P \supset A_0\ ! \dashv \underbrace{\Gamma}_{\Delta}}\supset\mathsf{I}\bot$$

$$
\begin{array}{rl}
\Gamma, \blacktriangleright_P \ /\ P \dashv \bot & \text{Subderivation} \\
\Gamma, \blacktriangleright_P \ /\ \sigma \mathrel{\hat{=}} t : \kappa \dashv \bot & \text{By inversion} \\
P = (\sigma = t) & {''} \\
\mathsf{FEV}([\Gamma]\sigma) \cup \mathsf{FEV}([\Gamma]t) = \emptyset & \text{As in } {\supset}\mathsf{I} \text{ case (above)} \\
\mathsf{mgu}(\sigma, t) = \bot & \text{By Lemma 88 (Soundness of Equality Elimination)}
\end{array}
$$

	$[\Omega]\Delta \;/\; (\sigma = t) \vdash [\Omega]v \Leftarrow [\Omega]A_0\ !$	By DeclCheck⊥
	$[\Omega]\Delta \vdash [\Omega]v \Leftarrow (\sigma = t) \supset [\Omega]A_0\ !$	By Decl⊃I
	$[\Omega]\Delta \vdash [\Omega]v \Leftarrow \big([\Omega](\sigma = t)\big) \supset [\Omega]A_0\ !$	By above FEV condition
☞	$[\Omega]\Delta \vdash [\Omega]v \Leftarrow [\Omega]\big(P \supset A_0\big)\ !$	By def. of subst.
	Let $\Omega' = \Omega$.	
☞	$\Omega \longrightarrow \Omega'$	By Lemma 32 (Extension Reflexivity)
☞	$\Delta \longrightarrow \Omega'$	Given

- **Case** $\dfrac{\Gamma \vdash P\ \mathit{true} \dashv \Theta \qquad \Theta \vdash e\ s_0 : [\Theta]A_0\ p \gg C\ q \dashv \Delta}{\Gamma \vdash e\ s_0 : P \supset A_0\ p \gg C\ q \dashv \Delta}\ \supset\!\mathsf{Spine}$

	$\Theta \vdash e\ s_0 : [\Theta]A_0\ p \gg C\ q \dashv \Delta$	Subderivation
	$\Theta \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
	$\Delta \longrightarrow \Omega$	Given
	$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
	$[\Omega]\Delta \vdash [\Omega](e\ s_0) : [\Omega][\Theta]A_0\ p \gg [\Omega]C\ q$	By i.h.
	$[\Omega][\Theta]A_0 = [\Omega]A_0$	By Lemma 29 (Substitution Monotonicity) (iii)
	$[\Omega]\Delta \vdash [\Omega](e\ s_0) : [\Omega]A_0\ p \gg [\Omega]C\ q$	By above equality
	$\Gamma \vdash P\ \mathit{true} \dashv \Theta$	Subderivation
	$[\Omega]\Theta \vdash [\Omega]P\ \mathit{true}$	By Lemma 95 (Completeness of Checkprop)
	$[\Omega]\Theta = [\Omega]\Delta$	By Lemma 56 (Confluence of Completeness)
	$[\Omega]\Delta \vdash [\Omega]P\ \mathit{true}$	By above equality
	$[\Omega]\Delta \vdash [\Omega](e\ s_0) : ([\Omega]P) \supset [\Omega]A_0\ p \gg [\Omega]C\ q$	By Decl⊃Spine
☞	$[\Omega]\Delta \vdash [\Omega](e\ s_0) : [\Omega](P \supset A_0)\ p \gg [\Omega]C\ q$	By def. of subst.

- **Case** $\dfrac{\Gamma, x : A_1\ p \vdash e_0 \Leftarrow A_2\ p \dashv \Delta, x : A_1\ p, \Theta}{\Gamma \vdash \lambda x.\, e_0 \Leftarrow A_1 \to A_2\ p \dashv \Delta}\ \to\!\mathsf{I}$

$\Delta \longrightarrow \Omega$	Given
$\Delta, x : A_1\ p \longrightarrow \Omega, x : [\Omega]A_1\ p$	By $\longrightarrow$Var
$\Gamma, x : A_1\ p \longrightarrow \Delta, x : A_1\ p, \Theta$	By Lemma 51 (Typing Extension)
Θ soft	By Lemma 22 (Extension Inversion) (v)
	(with $\Gamma_R = \cdot$, which is soft)
$\underbrace{\Delta, x : A_1\ p, \Theta}_{\Delta'} \longrightarrow \underbrace{\Omega, x : [\Omega]A_1\ p, \lvert\Theta\rvert}_{\Omega'}$	By Lemma 25 (Filling Completes)
$\Gamma, x : A_1\ p \vdash e_0 \Leftarrow A_2\ p \dashv \Delta'$	Subderivation
$[\Omega']\Delta' \vdash [\Omega]e_0 \Leftarrow [\Omega']A_2\ p$	By i.h.
$[\Omega']A_2 = [\Omega]A_2$	By Lemma 17 (Substitution Stability)
$[\Omega']\Delta' \vdash [\Omega]e_0 \Leftarrow [\Omega]A_2\ p$	By above equality
$\underbrace{\Delta, x : A_1\ p, \Theta}_{\Delta'} \longrightarrow \underbrace{\Omega, x : [\Omega]A_1\ p, \lvert\Theta\rvert}_{\Omega'}$	Above
Θ soft	Above
$[\Omega']\Delta' = ([\Omega]\Delta, x : [\Omega]A_1\ p)$	By Lemma 53 (Softness Goes Away)
$[\Omega]\Delta, x : [\Omega]A_1\ p \vdash [\Omega]e_0 \Leftarrow [\Omega]A_2\ p$	By above equality

$$[\Omega]\Delta \vdash \lambda x.\,[\Omega]e_0 \Leftarrow ([\Omega]A_1) \rightarrow ([\Omega]A_2)\ p \quad \text{By Decl}{\rightarrow}\text{I}$$

☞ $[\Omega]\Delta \vdash [\Omega](\lambda x.\,e_0) \Leftarrow [\Omega](A_1 \rightarrow A_2)\ p$ — By definition of substitution

- **Case** $\dfrac{v\ \mathit{chk\text{-}I} \qquad \Gamma, x : A\ p \vdash v \Leftarrow A\ p \dashv \Delta, x : A\ p, \Theta}{\Gamma \vdash \mathsf{rec}\ x.\,v \Leftarrow A\ p \dashv \Delta}$ Rec

  Similar to the $\rightarrow$I case, applying DeclRec instead of Decl$\rightarrow$I.

- **Case** $\dfrac{\Gamma[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha : \star = \hat\alpha_1{\rightarrow}\hat\alpha_2], x : \hat\alpha_1\ \not{p} \vdash e_0 \Leftarrow \hat\alpha_2\ \not{p} \dashv \Delta, x : \hat\alpha_1\ \not{p}, \Theta}{\Gamma[\hat\alpha : \star] \vdash \lambda x.\,e_0 \Leftarrow \hat\alpha\ \not{p} \dashv \Delta}$ $\rightarrow$I$\hat\alpha$

$$\begin{array}{rll}
\Gamma[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha : \star = \hat\alpha_1{\rightarrow}\hat\alpha_2], x : \hat\alpha\ \not{p} \longrightarrow \Delta, x : \hat\alpha\ \not{p}, \Theta & & \text{By Lemma 51 (Typing Extension)} \\
\Theta\ \ \text{soft} & & \text{By Lemma 22 (Extension Inversion) (v)} \\
& & \quad \text{(with } \Gamma_R = \cdot\text{, which is soft)} \\
\Gamma[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha : \star = \hat\alpha_1{\rightarrow}\hat\alpha_2] \longrightarrow \Delta & & '' \\
\Delta \longrightarrow \Omega & & \text{Given} \\
\Delta, x : \hat\alpha_1\ \not{p} \longrightarrow \Omega, x : [\Omega]\hat\alpha_1\ \not{p} & & \text{By} \longrightarrow\text{Var} \\
\underbrace{\Delta, x : \hat\alpha_1\ \not{p}, \Theta}_{\Delta'} \longrightarrow \underbrace{\Omega, x : [\Omega]\hat\alpha_1\ \not{p}, |\Theta|}_{\Omega'} & & \text{By Lemma 25 (Filling Completes)}
\end{array}$$

$\Gamma[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha : \star = \hat\alpha_1{\rightarrow}\hat\alpha_2], x : \hat\alpha_1\ \not{p} \vdash e_0 \Leftarrow \hat\alpha_2\ \not{p} \dashv \Delta, x : \hat\alpha_1\ \not{p}, \Theta$ — Subderivation

$$\begin{array}{rcll}
[\Omega']\Delta' \vdash [\Omega']e_0 \Leftarrow [\Omega']\hat\alpha_2\ \not{p} & & & \text{By i.h.} \\
[\Omega']\hat\alpha_2 &=& \big[\Omega, x : [\Omega]\hat\alpha_1\ \not{p}\big]\hat\alpha_2 & \text{By Lemma 17 (Substitution Stability)} \\
&=& [\Omega]\hat\alpha_2 & \text{By definition of substitution} \\
[\Omega']\Delta' &=& \big[\Omega, x : [\Omega]\hat\alpha_1\ \not{p}\big]\big(\Delta, x : \hat\alpha_1\ \not{p}\big) & \text{By Lemma 53 (Softness Goes Away)} \\
&=& [\Omega]\Delta, x : [\Omega]\hat\alpha_1\ \not{p} & \text{By definition of context substitution} \\
[\Omega]\Delta, x : [\Omega]\hat\alpha_1\ \not{p} \vdash [\Omega]e_0 \Leftarrow [\Omega]\hat\alpha_2\ \not{p} & & & \text{By above equalities} \\
[\Omega]\Delta \vdash \lambda x.\,[\Omega]e_0 \Leftarrow ([\Omega]\hat\alpha_1) \rightarrow [\Omega]\hat\alpha_2\ \not{p} & & & \text{By Decl}{\rightarrow}\text{I}
\end{array}$$

$\Gamma[\hat\alpha_1{:}\star, \hat\alpha_2{:}\star, \hat\alpha : \star = \hat\alpha_1{\rightarrow}\hat\alpha_2] \longrightarrow \Omega$ — Above and Lemma 33 (Extension Transitivity)

$$\begin{array}{rcll}
[\Omega]\hat\alpha &=& [\Omega][\Gamma]\hat\alpha & \text{By Lemma 29 (Substitution Monotonicity) (i)} \\
&=& [\Omega]\big(([\Gamma]\hat\alpha_1) \rightarrow [\Gamma]\hat\alpha_2\big) & \text{By definition of substitution} \\
&=& ([\Omega][\Gamma]\hat\alpha_1) \rightarrow ([\Omega][\Gamma]\hat\alpha_2) & \text{By definition of substitution} \\
&=& ([\Omega]\hat\alpha_1) \rightarrow ([\Omega]\hat\alpha_2) & \text{By Lemma 29 (Substitution Monotonicity) (i)}
\end{array}$$

☞ $[\Omega]\Delta \vdash [\Omega](\lambda x.\,e_0) \Leftarrow [\Omega]\hat\alpha\ \not{p}$ — By above equality

- **Case** $\dfrac{\Gamma \vdash e_0 \Rightarrow A\ q \dashv \Theta \qquad \Theta \vdash s_0 : A\ q \gg C\ \lceil p \rceil \dashv \Delta}{\Gamma \vdash e_0\ s_0 \Rightarrow C\ p \dashv \Delta}$ $\rightarrow$E

$\Gamma \vdash e_0 \Rightarrow A\ q \dashv \Theta$	Subderivation
$\Theta \vdash s_0 : A\ q \gg C\ \lceil p \rceil \dashv \Delta$	Subderivation
$\Gamma \longrightarrow \Theta$ and $\Theta \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
$\Delta \longrightarrow \Omega$	Given
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Gamma \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$[\Omega]\Gamma = [\Omega]\Theta = [\Omega]\Delta$	By Lemma 56 (Confluence of Completeness)
$[\Omega]\Gamma \vdash [\Omega]e_0 \Rightarrow [\Omega]A\ q$	By i.h.
$[\Omega]\Delta \vdash [\Omega]e_0 \Rightarrow [\Omega]A\ q$	By above equality
$[\Omega]\Theta \vdash [\Omega]s_0 : [\Omega]A\ q \gg [\Omega]C\ \lceil p \rceil$	By i.h.
☞ $[\Omega]\Delta \vdash [\Omega](e_0\ s_0) \Rightarrow [\Omega]C\ p$	By rule Decl$\rightarrow$E

- **Case** $\dfrac{\Gamma \vdash s : A\ ! \gg C\ \not! \dashv \Delta \qquad \mathsf{FEV}(C) = \emptyset}{\Gamma \vdash s : A\ ! \gg C\ \lceil ! \rceil \dashv \Delta}$ SpineRecover

$\Gamma \vdash s : A\ ! \gg C\ \not! \dashv \Delta$	Subderivation
$[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ ! \gg [\Omega]C\ q$	By i.h.

We show the quantified premise of DeclSpineRecover, namely,

$$\text{for all } C'.\ \text{if } [\Omega]\Theta \vdash s : [\Omega]A\ ! \gg C'\ \not!\ \text{ then } C' = [\Omega]C$$

Suppose we have C' such that $[\Omega]\Gamma \vdash s : [\Omega]A\ ! \gg C'\ \not!$. To apply DeclSpineRecover, we need to show $C' = [\Omega]C$.

$[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ ! \gg C'\ \not!$	Assumption
$\Omega_{canon} \longrightarrow \Omega$	By Lemma 59 (Canonical Completion)
$\mathsf{dom}(\Omega_{canon}) = \mathsf{dom}(\Gamma)$	"
$\Gamma \longrightarrow \Omega_{canon}$	"
$[\Omega]\Gamma = [\Omega_{canon}]\Gamma$	By Lemma 57 (Multiple Confluence)
$[\Omega]A = [\Omega_{canon}]A$	By Lemma 55 (Completing Completeness) (ii)
$[\Omega_{canon}]\Gamma \vdash [\Omega]s : [\Omega_{canon}]A\ ! \gg C'\ \not!$	By above equalities
$\Gamma \vdash s : [\Gamma]A\ ! \gg C''\ q \dashv \Delta''$	By Theorem 11
$\Omega_{canon} \longrightarrow \Omega''$	"
$\Delta'' \longrightarrow \Omega''$	"
$C' = [\Omega'']C''$	"
$C'' = C$ and $q = \not!$ and $\Delta'' = \Delta$	By Theorem 5
$C' = [\Omega'']C''$	Above
$= [\Omega'']C$	By above equality
$= [\Omega_{canon}]C$	By Lemma 55 (Completing Completeness) (ii)
$= [\Omega]C$	By Lemma 55 (Completing Completeness) (ii)

We have thus shown the above "for all C'. ..." statement.

☞ $[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ ! \gg [\Omega]C\ \lceil ! \rceil$ — By DeclSpineRecover

- **Case** $\dfrac{\Gamma \vdash s : A\ p \gg C\ q \dashv \Delta \qquad ((p = \not!) \text{ or } (q = !) \text{ or } (\mathsf{FEV}(C) \neq \emptyset))}{\Gamma \vdash s : A\ p \gg C\ \lceil q \rceil \dashv \Delta}$ SpinePass

$$
\begin{array}{lll}
 & \Gamma \vdash s : A\ p \gg C\ q \dashv \Delta & \text{Subderivation} \\
 & [\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg [\Omega]C\ q & \text{By i.h.} \\
☞ & [\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg [\Omega]C\ \lceil q \rceil & \text{By DeclSpinePass}
\end{array}
$$

- **Case**

$$
\frac{}{\Gamma \vdash \cdot : A\ p \gg A\ p \dashv \Gamma}\ \textsf{EmptySpine}
$$

☞ $[\Omega]\Gamma \vdash \cdot : [\Omega]A\ p \gg [\Omega]A\ p$ By DeclEmptySpine

- **Case** $\dfrac{\Gamma \vdash e_0 \Leftarrow A_1\ p \dashv \Theta \qquad \Theta \vdash s_0 : [\Theta]A_2\ p \gg C\ q \dashv \Delta}{\Gamma \vdash e_0\ s_0 : A_1 \to A_2\ p \gg C\ q \dashv \Delta}\ {\to}\textsf{Spine}$

$$
\begin{array}{ll}
\Delta \longrightarrow \Omega & \text{Given} \\
\Theta \longrightarrow \Delta & \text{By Lemma 51 (Typing Extension)} \\
\Theta \longrightarrow \Omega & \text{By Lemma 33 (Extension Transitivity)}
\end{array}
$$

$$
\begin{array}{ll}
\Gamma \vdash e_0 \Leftarrow A_1\ p \dashv \Theta & \text{Subderivation} \\
[\Omega]\Theta \vdash [\Omega]e_0 \Leftarrow [\Omega]A_1\ p & \text{By i.h.} \\
[\Omega]\Theta = [\Omega]\Delta & \text{By Lemma 56 (Confluence of Completeness)} \\
[\Omega]\Delta \vdash [\Omega]e_0 \Leftarrow [\Omega]A_1\ p & \text{By above equality}
\end{array}
$$

$$
\begin{array}{ll}
\Theta \vdash s_0 : [\Theta]A_2\ p \gg C\ q \dashv \Delta & \text{Subderivation} \\
[\Omega]\Delta \vdash [\Omega]s_0 : [\Omega][\Theta]A_2\ p \gg [\Omega]C\ q & \text{By i.h.} \\
[\Omega][\Theta]A_2 = [\Omega]A_2 & \text{By Lemma 29 (Substitution Monotonicity)} \\
[\Omega]\Delta \vdash [\Omega]s_0 : [\Omega]A_2\ p \gg [\Omega]C\ q & \text{By above equality}
\end{array}
$$

$$
\begin{array}{lll}
 & [\Omega]\Delta \vdash [\Omega](e_0\ s_0) : ([\Omega]A_1) \to [\Omega]A_2\ p \gg [\Omega]C\ q & \text{By Decl}{\to}\text{Spine} \\
☞ & [\Omega]\Delta \vdash [\Omega](e_0\ s_0) : [\Omega](A_1 \to A_2)\ p \gg [\Omega]C\ q & \text{By def. of subst.}
\end{array}
$$

- **Case** $\dfrac{\Gamma \vdash e_0 \Leftarrow A_k\ p \dashv \Delta}{\Gamma \vdash \textsf{inj}_k\ e_0 \Leftarrow A_1 + A_2\ p \dashv \Delta}\ {+}\textsf{I}_k$

$$
\begin{array}{lll}
 & \Gamma \vdash e_0 \Leftarrow A_k\ p \dashv \Delta & \text{Subderivation} \\
 & [\Omega]\Delta \vdash [\Omega]e_0 \Leftarrow [\Omega]A_k\ p & \text{By i.h.} \\
 & [\Omega]\Delta \vdash \textsf{inj}_k\ [\Omega]e_0 \Leftarrow ([\Omega]A_1) + ([\Omega]A_2)\ p & \text{By Decl}{+}\text{I}_k \\
☞ & [\Omega]\Delta \vdash [\Omega](\textsf{inj}_k\ e_0) \Leftarrow [\Omega](A_1 + A_2)\ p & \text{By def. of substitution}
\end{array}
$$

- **Case** $\dfrac{\Gamma[\hat\alpha_1 : \star, \hat\alpha_2 : \star, \hat\alpha : \star = \hat\alpha_1 {+} \hat\alpha_2] \vdash e_0 \Leftarrow \hat\alpha_k\ \not{p} \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash \textsf{inj}_k\ e_0 \Leftarrow \hat\alpha\ \not{p} \dashv \Delta}\ {+}\textsf{I}\hat\alpha_k$

$$
\begin{array}{lll}
 & \Gamma[\ldots, \hat\alpha : \star = \hat\alpha_1 {+} \hat\alpha_2] \vdash e_0 \Leftarrow \hat\alpha_k\ \not{p} \dashv \Delta & \text{Subderivation} \\
 & [\Omega]\Delta \vdash [\Omega]e_0 \Leftarrow [\Omega]\hat\alpha_k\ \not{p} & \text{By i.h.} \\
 & [\Omega]\Delta \vdash \textsf{inj}_k\ [\Omega]e_0 \Rightarrow ([\Omega]\hat\alpha_1) + ([\Omega]\hat\alpha_2)\ \not{p} & \text{By Decl}{+}\text{I}_k \\
 & ([\Omega]\hat\alpha_1) + ([\Omega]\hat\alpha_2) = [\Omega]\hat\alpha & \text{Similar to the } {\to}\text{I}\hat\alpha \text{ case (above)} \\
☞ & [\Omega]\Delta \vdash [\Omega](\textsf{inj}_k\ e_0) \Rightarrow [\Omega]\hat\alpha\ \not{p} & \text{By above equality / def. of subst.}
\end{array}
$$

- **Case** $\dfrac{\Gamma \vdash e_1 \Leftarrow A_1\ p \dashv \Theta \qquad \Theta \vdash e_2 \Leftarrow [\Theta]A_2\ p \dashv \Delta}{\Gamma \vdash \langle e_1, e_2 \rangle \Leftarrow A_1 \times A_2\ p \dashv \Delta}\ {\times}\textsf{I}$

$\Theta \vdash e_2 \Leftarrow [\Theta]A_2\ p \dashv \Delta$	Subderivation
$\Theta \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Gamma \vdash e_1 \Leftarrow A_1\ p \dashv \Theta$	Subderivation
$[\Omega]\Theta \vdash [\Omega]e_1 \Leftarrow [\Omega]A_1\ p$	By i.h.
$[\Omega]\Delta \vdash [\Omega]e_1 \Leftarrow [\Omega]A_1\ p$	By Lemma 56 (Confluence of Completeness)
$\Theta \vdash e_2 \Leftarrow [\Theta]A_2\ p \dashv \Delta$	Subderivation
$[\Omega]\Delta \vdash [\Omega]e_2 \Leftarrow [\Omega][\Theta]A_2\ p$	By i.h.
$\Gamma \vdash A_1 \times A_2\ \mathit{type}$	Given
$\Gamma \vdash A_2\ \mathit{type}$	By inversion
$\Gamma \longrightarrow \Theta$	By Lemma 51 (Typing Extension)
$\Theta \vdash A_2\ \mathit{type}$	By Lemma 38 (Extension Weakening (Types))
$[\Omega]\Delta \vdash [\Omega]e_2 \Leftarrow [\Omega]A_2\ p$	By Lemma 29 (Substitution Monotonicity)
$[\Omega]\Delta \vdash \langle [\Omega]e_1, [\Omega]e_2 \rangle \Leftarrow ([\Omega]A_1) \times [\Omega]A_2\ p$	By Decl×I
☞ $[\Omega]\Delta \vdash [\Omega]\langle e_1, e_2 \rangle \Leftarrow [\Omega](A_1 \times A_2)\ p$	By def. of substitution

- **Case** $$\dfrac{\Gamma[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \times \hat\alpha_2] \vdash e_1 \Leftarrow \hat\alpha_1\ \not{p} \dashv \Theta \qquad \Theta \vdash e_2 \Leftarrow [\Theta]\hat\alpha_2\ \not{p} \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash \langle e_1, e_2 \rangle \Leftarrow \hat\alpha\ \not{p} \dashv \Delta}\ \times\mathsf{I}\hat\alpha$$

$\Delta \longrightarrow \Omega$	Given
$\Theta \longrightarrow \Delta$	By Lemma 51 (Typing Extension)
$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
$\Gamma[\ldots, \hat\alpha : \star = \hat\alpha_1 \times \hat\alpha_2] \vdash e_1 \Leftarrow \hat\alpha_1\ \not{p} \dashv \Theta$	Subderivation
$[\Omega]\Theta \vdash [\Omega]e_1 \Leftarrow [\Omega]\hat\alpha_1\ \not{p}$	By i.h.
$[\Omega]\Theta = [\Omega]\Delta$	By Lemma 56 (Confluence of Completeness)
$[\Omega]\Delta \vdash [\Omega]e_1 \Leftarrow [\Omega]\hat\alpha_1\ \not{p}$	By above equality
$\Theta \vdash e_2 \Leftarrow [\Theta]\hat\alpha_2\ \not{p} \dashv \Delta$	Subderivation
$[\Omega]\Delta \vdash [\Omega]e_2 \Leftarrow [\Omega][\Theta]\hat\alpha_2\ \not{p}$	By i.h.
$[\Omega][\Theta]\hat\alpha_2 = [\Omega]\hat\alpha_2$	By Lemma 29 (Substitution Monotonicity)
$[\Omega]\Delta \vdash [\Omega]e_2 \Leftarrow [\Omega]\hat\alpha_2\ \not{p}$	By above equality
$[\Omega]\Delta \vdash \langle [\Omega]e_1, [\Omega]e_2 \rangle \Leftarrow ([\Omega]\hat\alpha_1) \times [\Omega]\hat\alpha_2\ \not{p}$	By Decl×I
$([\Omega]\hat\alpha_1) \times [\Omega]\hat\alpha_2 = [\Omega]\hat\alpha$	Similar to the $\to\mathsf{I}\hat\alpha$ case (above)
☞ $[\Omega]\Delta \vdash [\Omega]\langle e_1, e_2 \rangle \Leftarrow [\Omega]\hat\alpha\ \not{p}$	By above equality

- **Case** $$\dfrac{\Gamma[\hat\alpha_2 : \star, \hat\alpha_1 : \star, \hat\alpha : \star = \hat\alpha_1 \to \hat\alpha_2] \vdash e_0\ s_0 : (\hat\alpha_1 \to \hat\alpha_2)\ \not{p} \gg C\ \not{q} \dashv \Delta}{\Gamma[\hat\alpha : \star] \vdash e_0\ s_0 : \hat\alpha\ \not{p} \gg C\ \not{q} \dashv \Delta}\ \hat\alpha\mathsf{Spine}$$

$\Gamma[\ldots, \hat\alpha : \star = \hat\alpha_1 \to \hat\alpha_2] \vdash e_0\ s_0 : (\hat\alpha_1 \to \hat\alpha_2)\ \not{p} \gg C\ \not{q} \dashv \Delta$	Subderivation
$[\Omega]\Delta \vdash [\Omega](e_0\ s_0) : [\Omega](\hat\alpha_1 \to \hat\alpha_2)\ \not{p} \gg [\Omega]C\ \not{q}$	By i.h.
$[\Omega](\hat\alpha_1 \to \hat\alpha_2) = [\Omega]\hat\alpha$	Similar to the $\to\mathsf{I}\hat\alpha$ case
☞ $[\Omega]\Delta \vdash [\Omega](e_0\ s_0) : [\Omega]\hat\alpha\ \not{p} \gg [\Omega]C\ \not{q}$	By above equality

- **Case** $$\dfrac{\Gamma \vdash e_0 \Rightarrow B\ ! \dashv \Theta \qquad \Theta \vdash \Pi :: [\Theta]B \Leftarrow [\Theta]C\ p \dashv \Delta \qquad \Delta \vdash \Pi\ \mathit{covers}\ [\Delta]B}{\Gamma \vdash \mathsf{case}(e_0, \Pi) \Leftarrow C\ p \dashv \Delta}\ \mathsf{Case}$$

$$\begin{array}{ll}
\Gamma \vdash e_0 \Rightarrow B\ \mathbf{!} \dashv \Theta & \text{Subderivation} \\
\Theta \longrightarrow \Delta & \text{By Lemma 51 (Typing Extension)} \\
\Theta \longrightarrow \Omega & \text{By Lemma 33 (Extension Transitivity)} \\
[\Omega]\Theta \vdash [\Omega]e_0 \Rightarrow [\Omega]B\ \mathbf{!} & \text{By i.h.} \\
[\Omega]\Delta \vdash [\Omega]e_0 \Rightarrow [\Omega]B\ \mathbf{!} & \text{By Lemma 56 (Confluence of Completeness)}
\end{array}$$

$$\begin{array}{ll}
\Theta \vdash \Pi :: [\Theta]B \Leftarrow [\Theta]C\ p \dashv \Delta & \text{Subderivation}
\end{array}$$

$$\begin{array}{ll}
\Gamma \vdash e_0 \Rightarrow B\ \mathbf{!} \dashv \Theta & \text{Subderivation} \\
\Theta \vdash B\ \mathbf{!}\ \mathit{type} & \text{By Lemma 63 (Well-Formed Outputs of Typing) (Synthesis)}
\end{array}$$

$$\begin{array}{ll}
\Gamma \vdash C\ p\ \mathit{type} & \text{Given} \\
\Gamma \longrightarrow \Theta & \text{By Lemma 51 (Typing Extension)} \\
\Theta \vdash C\ p\ \mathit{type} & \text{By Lemma 41 (Extension Weakening for Principal Typing)} \\
\Theta \vdash [\Theta]C\ p\ \mathit{type} & \text{By Lemma 40 (Right-Hand Subst. for Principal Typing)}
\end{array}$$

$$\begin{array}{ll}
[\Omega]\Delta \vdash [\Omega]\Pi :: [\Omega]B \Leftarrow [\Omega][\Theta]C\ p & \text{By i.h. (v)} \\
[\Omega][\Theta]C = [\Omega]C & \text{By Lemma 29 (Substitution Monotonicity)} \\
[\Omega]\Delta \vdash [\Omega]\Pi :: [\Omega]B \Leftarrow [\Omega]C\ p & \text{By above equalities}
\end{array}$$

$$\begin{array}{ll}
\Delta \vdash \Pi\ \mathit{covers}\ [\Delta]B & \text{Subderivation} \\
[\Delta][\Delta]B = [\Delta]B & \text{By idempotence of substitution} \\
\Theta \vdash B\ \mathbf{!}\ \mathit{type} & \text{By Lemma 63 (Well-Formed Outputs of Typing)} \\
\Delta \vdash B\ \mathbf{!}\ \mathit{type} & \text{By Lemma 41 (Extension Weakening for Principal Typing)} \\
\Delta \vdash [\Delta]B\ \mathbf{!}\ \mathit{type} & \text{By Lemma 40 (Right-Hand Subst. for Principal Typing)} \\
[\Omega]\Delta \vdash [\Omega]\Pi\ \mathit{covers}\ [\Delta]B & \text{By Theorem 7} \\
[\Delta]B = [\Omega]B & \text{By Lemma 39 (Principal Agreement) (i)} \\
[\Omega]\Delta \vdash [\Omega]\Pi\ \mathit{covers}\ [\Omega]B & \text{By above equality}
\end{array}$$

☞ $[\Omega]\Delta \vdash [\Omega]\mathsf{case}(e_0, \Pi) \Leftarrow [\Omega]C\ p$ By DeclCase

Part (v):

- **Case** MatchEmpty: Apply rule DeclMatchEmpty.

- **Case** $\dfrac{\Gamma \vdash e \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash (\cdot \Rightarrow e) :: \cdot \Leftarrow C\ p \dashv \Delta}$ MatchBase

  Apply the i.h. and DeclMatchBase.

- **Case** MatchUnit: Apply the i.h. and DeclMatchUnit.

- **Case** $\dfrac{\Gamma \vdash \pi :: \vec{A} \Leftarrow C\ p \dashv \Theta \qquad \Theta \vdash \Pi' :: \vec{A} \Leftarrow C\ p \dashv \Delta}{\Gamma \vdash \pi \mid \Pi' :: \vec{A} \Leftarrow C\ p \dashv \Delta}$ MatchSeq

  Apply the i.h. to each premise, using lemmas for well-formedness under Θ; then apply DeclMatchSeq.

- **Cases** Match$\exists$, Match$\wedge$, MatchWild, MatchNil, MatchCons:
  Apply the i.h. and the corresponding declarative match rule.

- **Cases** Match$\times$, Match$+_k$:
  We have $\Gamma \vdash \vec{A}\ \mathbf{!}\ \mathit{types}$, so the first type in $\vec{A}$ has no free existential variables.
  Apply the i.h. and the corresponding declarative match rule.

- **Case** $$\dfrac{A \text{ not headed by } \wedge \text{ or } \exists \qquad \Gamma, z : A\,! \vdash \vec{\rho} \Rightarrow e' :: \vec{A} \Leftarrow C\ p \dashv \Delta, z : A\,!, \Delta'}{\Gamma \vdash z, \vec{\rho} \Rightarrow e :: A, \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchNeg}$$

  Construct Ω' and show $\Delta, z : A\,!, \Delta' \longrightarrow \Omega'$ as in the $\to$I case.
  Use the i.h., then apply rule DeclMatchNeg.

Part (vi):

- **Case** $$\dfrac{\Gamma \ /\ \sigma \stackrel{\circ}{=} \tau : \kappa \dashv \bot}{\Gamma \ /\ \sigma = \tau \vdash \vec{\rho} p e :: \vec{A} \Leftarrow C\ p \dashv \Gamma}\ \textsf{Match}\bot$$

$$\begin{array}{rll}
\Gamma \ /\ \sigma \stackrel{\circ}{=} \tau : \kappa \dashv \bot & & \text{Subderivation} \\
[\Gamma](\sigma = \tau) = (\sigma = \tau) & & \text{Given} \\
(\sigma = \tau) = [\Gamma](\sigma = \tau) & & \text{Given} \\
= [\Omega](\sigma = \tau) & & \text{By Lemma 29 (Substitution Monotonicity) (i)} \\
\mathsf{mgu}(\sigma, \tau) = \bot & & \text{By Lemma 88 (Soundness of Equality Elimination)} \\
\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \bot & & \text{By above equality}
\end{array}$$

☞ $[\Omega]\Gamma \ /\ [\Omega](\sigma = \tau) \vdash [\Omega](\vec{\rho} p e) :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$ By DeclMatch$\bot$

- **Case** $$\dfrac{\Gamma, \blacktriangleright_P \ /\ \sigma \stackrel{\circ}{=} \tau : \kappa \dashv \Gamma' \qquad \Gamma' \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta, \blacktriangleright_P, \Delta'}{\Gamma \ /\ \sigma = \tau \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta}\ \textsf{MatchUnify}$$

$$\begin{array}{rll}
\Gamma, \blacktriangleright_P \ /\ \sigma \stackrel{\circ}{=} \tau : \kappa \dashv \Gamma' & & \text{Subderivation} \\
(\sigma = \tau) = [\Gamma](\sigma = \tau) & & \text{Given} \\
= [\Omega](\sigma = \tau) & & \text{By Lemma 29 (Substitution Monotonicity) (i)} \\
\Gamma' = (\Gamma, \blacktriangleright_P, \Theta) & & \text{By Lemma 88 (Soundness of Equality Elimination)} \\
\Theta = ((\alpha_1 = t_1), \dots, (\alpha_n = t_n)) & & '' \\
\theta = \mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) & & '' \\
[\Omega, \blacktriangleright_P, \Theta]t' = [\theta][\Omega, \blacktriangleright_P]t' & & ''\ \text{for all } \Omega, \blacktriangleright_P \vdash t' : \kappa'
\end{array}$$

$$\Gamma, \blacktriangleright_P, \Theta \vdash \vec{\rho} \Rightarrow e :: \vec{A} \Leftarrow C\ p \dashv \Delta, \blacktriangleright_P, \Delta' \qquad \text{Subderivation}$$

$$[\Omega, \blacktriangleright_P, \Theta](\Delta, \blacktriangleright_P, \Delta') \vdash [\Omega, \blacktriangleright_P, \Theta](\vec{\rho} \Rightarrow e) :: [\Omega, \blacktriangleright_P, \Theta]\vec{A} \Leftarrow [\Omega, \blacktriangleright_P, \Theta]C\ p \qquad \text{By i.h.}$$

$$\begin{array}{rll}
(\Omega, \blacktriangleright_P, \Theta) = [\theta](\Omega, \blacktriangleright_P) & & \text{By Lemma 93 (Substitution Upgrade) (iii)} \\
[\Omega, \blacktriangleright_P, \Theta]\vec{A} = [\theta][\Omega, \blacktriangleright_P]\vec{A} & & \text{By Lemma 93 (Substitution Upgrade) (i)} \\
[\Omega, \blacktriangleright_P, \Theta]C = [\theta][\Omega, \blacktriangleright_P]C & & \text{By Lemma 93 (Substitution Upgrade) (i)} \\
[\Omega, \blacktriangleright_P, \Theta](\vec{\rho} \Rightarrow e) = [\theta][\Omega](\vec{\rho} \Rightarrow e) & & \text{By Lemma 93 (Substitution Upgrade) (iv)}
\end{array}$$

$$\begin{array}{rl}
\theta([\Omega, \blacktriangleright_P]\Gamma) \vdash [\theta][\Omega](\vec{\rho} \Rightarrow e) :: \theta([\Omega, \blacktriangleright_P]\vec{A}) \Leftarrow \theta([\Omega, \blacktriangleright_P]C)\ p & \text{By above equalities} \\
\theta([\Omega]\Gamma) \vdash [\theta][\Omega](\vec{\rho} \Rightarrow e) :: \theta([\Omega]\vec{A}) \Leftarrow \theta([\Omega]C)\ p & \text{Subst. not affected by } \blacktriangleright_P
\end{array}$$

☞ $[\Omega]\Gamma \ /\ [\Omega](\sigma = \tau) \vdash [\Omega](\vec{\rho} \Rightarrow e) :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$ By DeclMatchUnify ☐

K′ Completeness

K′.1 Completeness of Auxiliary Judgments

Lemma 90 (Completeness of Instantiation)**.**
Given $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$ *and* $\Gamma \vdash \tau : \kappa$ *and* $\tau = [\Gamma]\tau$ *and* $\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$ *and* $\hat{\alpha} \notin FV(\tau)$*:*
If $[\Omega]\hat{\alpha} = [\Omega]\tau$
then there are Δ, Ω' *such that* $\Omega \longrightarrow \Omega'$ *and* $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$.

Proof. By induction on τ.

We have $[\Omega]\Gamma \vdash [\Omega]\hat{\alpha} \leq^{*} [\Omega]A$. We now case-analyze the shape of τ.

- **Case** $\tau = \hat{\beta}$:

$$\begin{array}{lll} & \hat{\alpha} \notin FV(\hat{\beta}) & \text{Given} \\ & \hat{\alpha} \neq \hat{\beta} & \text{From definition of } FV(-) \\ & \hat{\beta} \in \mathsf{unsolved}(\Gamma) & \text{From } [\Gamma]\hat{\beta} = \hat{\beta} \\ & \text{Let } \Omega' = \Omega. & \\ ☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 32 (Extension Reflexivity)} \end{array}$$

  Now consider whether $\hat{\alpha}$ is declared to the left of $\hat{\beta}$, or vice versa.

  - **Case** $\Gamma = \Gamma_0[\hat{\alpha} : \kappa][\hat{\beta} : \kappa]$:

$$\begin{array}{lll} & \text{Let } \Delta = \Gamma_0[\hat{\alpha} : \kappa][\hat{\beta} : \kappa = \hat{\alpha}]. & \\ & \Gamma \vdash \hat{\alpha} := \hat{\beta} : \kappa \dashv \Delta & \text{By } \mathsf{InstReach} \\ & [\Omega]\hat{\alpha} = [\Omega]\hat{\beta} & \text{Given} \\ & \Gamma \longrightarrow \Omega & \text{Given} \\ ☞ & \Delta \longrightarrow \Omega & \text{By Lemma 27 (Parallel Extension Solution)} \\ ☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & \mathsf{dom}(\Delta) = \mathsf{dom}(\Gamma) \text{ and } \mathsf{dom}(\Omega') = \mathsf{dom}(\Omega) \end{array}$$

  - **Case** $(\Gamma = \Gamma_0[\hat{\beta} : \kappa][\hat{\alpha} : \kappa]$:
    Similar, but using $\mathsf{InstSolve}$ instead of $\mathsf{InstReach}$.

- **Case** $\tau = \alpha$:

  We have $[\Omega]\hat{\alpha} = \alpha$, so (since Ω is well-formed), α is declared to the left of $\hat{\alpha}$ in Ω.
  We have $\Gamma \longrightarrow \Omega$.
  By Lemma 21 (Reverse Declaration Order Preservation), we know that α is declared to the left of $\hat{\alpha}$ in Γ; that is, $\Gamma = \Gamma_L[\alpha : \kappa][\hat{\alpha} : \kappa]$.
  Let $\Delta = \Gamma_L[\alpha : \kappa][\hat{\alpha} : \kappa = \alpha]$ and $\Omega' = \Omega$.
  By $\mathsf{InstSolve}$, $\Gamma_L[\alpha : \kappa][\hat{\alpha} : \kappa] \vdash \hat{\alpha} := \alpha : \kappa \dashv \Delta$.
  By Lemma 27 (Parallel Extension Solution), $\Gamma_L[\alpha : \kappa][\hat{\alpha} : \kappa = \alpha] \longrightarrow \Omega$.
  We have $\mathsf{dom}(\Delta) = \mathsf{dom}(\Gamma)$ and $\mathsf{dom}(\Omega') = \mathsf{dom}(\Omega)$; therefore, $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$.

- **Case** $\tau = 1$:

  Similar to the $\tau = \alpha$ case, but without having to reason about where α is declared.

- **Case** $\tau = \mathsf{zero}$:

  Similar to the $\tau = 1$ case.

- **Case** $\tau = \tau_1 \oplus \tau_2$:

$$\begin{array}{rcll}
[\Omega]\hat{\alpha} &=& [\Omega](\tau_1 \oplus \tau_2) & \text{Given} \\
&=& ([\Omega]\tau_1) \oplus ([\Omega]\tau_2) & \text{By definition of substitution}
\end{array}$$

$$\begin{array}{rcll}
\tau_1 \oplus \tau_2 &=& [\Gamma](\tau_1 \oplus \tau_2) & \text{Given} \\
\tau_1 &=& [\Gamma]\tau_1 & \text{By definition of substitution and congruence} \\
\tau_2 &=& [\Gamma]\tau_2 & \text{Similarly}
\end{array}$$

$$\begin{array}{rl}
\hat{\alpha} \notin \mathit{FV}(\tau_1 \oplus \tau_2) & \text{Given} \\
\hat{\alpha} \notin \mathit{FV}(\tau_1) & \text{From definition of } \mathit{FV}(-) \\
\hat{\alpha} \notin \mathit{FV}(\tau_2) & \text{Similarly}
\end{array}$$

$$\begin{array}{rl}
\Gamma = \Gamma_0[\hat{\alpha} : \star] & \text{By } \hat{\alpha} \in \mathsf{unsolved}(\Gamma) \\
\Gamma \longrightarrow \Omega & \text{Given} \\
\Gamma_0[\hat{\alpha} : \star] \longrightarrow \Gamma_0[\hat{\alpha}_2 : \star, \hat{\alpha}_1 : \star, \hat{\alpha} : \star] & \text{By Lemma 23 (Deep Evar Introduction) (i) twice} \\
\ldots, \hat{\alpha}_2, \hat{\alpha}_1 \vdash \hat{\alpha}_1 \oplus \hat{\alpha}_2 : \star & \text{Straightforward} \\
\Gamma_0[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha}] \longrightarrow \Gamma_0[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2] & \text{By Lemma 23 (Deep Evar Introduction) (ii)} \\
\Gamma_0[\hat{\alpha}] \longrightarrow \Gamma_0[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2] & \text{By Lemma 33 (Extension Transitivity)}
\end{array}$$

(In the last few lines above, and the rest of this case, we omit the "$: \star$" annotations in contexts.)

Since $\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$ and $\Gamma \longrightarrow \Omega$, we know that Ω has the form $\Omega_0[\hat{\alpha} = \tau_0]$.

To show that we can extend this context, we apply Lemma 23 (Deep Evar Introduction) (iii) twice to introduce $\hat{\alpha}_2 = \tau_2$ and $\hat{\alpha}_1 = \tau_1$, and then Lemma 28 (Parallel Variable Update) to overwrite τ_0:

$$\underbrace{\Omega_0[\hat{\alpha} = \tau_0]}_{\Omega} \longrightarrow \Omega_0[\hat{\alpha}_2 = \tau_2, \hat{\alpha}_1 = \tau_1, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2]$$

We have $\Gamma \longrightarrow \Omega$, that is,

$$\Gamma_0[\hat{\alpha}] \longrightarrow \Omega_0[\hat{\alpha} = \tau_0]$$

By Lemma 26 (Parallel Admissibility) (i) twice, inserting unsolved variables $\hat{\alpha}_2$ and $\hat{\alpha}_1$ on both contexts in the above extension preserves extension:

$$\begin{array}{rcll}
\Gamma_0[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha}] &\longrightarrow& \Omega_0[\hat{\alpha}_2 = \tau_2, \hat{\alpha}_1 = \tau_1, \hat{\alpha} = \tau_0] & \text{By Lemma 26 (Parallel Admissibility) (ii) twice} \\
\underbrace{\Gamma_0[\hat{\alpha}_2, \hat{\alpha}_1, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2]}_{\Gamma_1} &\longrightarrow& \underbrace{\Omega_0[\hat{\alpha}_2 = \tau_2, \hat{\alpha}_1 = \tau_1, \hat{\alpha} = \hat{\alpha}_1 \oplus \hat{\alpha}_2]}_{\Omega_1} & \text{By Lemma 28 (Parallel Variable Update)}
\end{array}$$

Since $\hat{\alpha} \notin \mathit{FV}(\tau)$, it follows that $[\Gamma_1]\tau = [\Gamma]\tau = \tau$.
Therefore $\hat{\alpha}_1 \notin \mathit{FV}(\tau_1)$ and $\hat{\alpha}_1, \hat{\alpha}_2 \notin \mathit{FV}(\tau_2)$.
By Lemma 55 (Completing Completeness) (i) and (iii), $[\Omega_1]\Gamma_1 = [\Omega]\Gamma$ and $[\Omega_1]\hat{\alpha}_1 = \tau_1$.
By i.h., there are Δ_2 and Ω_2 such that $\Gamma_1 \vdash \hat{\alpha}_1 := \tau_1 : \kappa \dashv \Delta_2$ and $\Delta_2 \longrightarrow \Omega_2$ and $\Omega_1 \longrightarrow \Omega_2$.

Next, note that $[\Delta_2][\Delta_2]\tau_2 = [\Delta_2]\tau_2$.
By Lemma 64 (Left Unsolvedness Preservation), we know that $\hat{\alpha}_2 \in \mathsf{unsolved}(\Delta_2)$.
By Lemma 65 (Left Free Variable Preservation), we know that $\hat{\alpha}_2 \notin \mathit{FV}([\Delta_2]\tau_2)$.
By Lemma 33 (Extension Transitivity), $\Omega \longrightarrow \Omega_2$.
We know $[\Omega_2]\Delta_2 = [\Omega]\Gamma$ because:

$$\begin{array}{rcll}
[\Omega_2]\Delta_2 &=& [\Omega_2]\Omega_2 & \text{By Lemma 54 (Completing Stability)} \\
&=& [\Omega]\Omega & \text{By Lemma 55 (Completing Completeness) (iii)} \\
&=& [\Omega]\Gamma & \text{By Lemma 54 (Completing Stability)}
\end{array}$$

By Lemma 55 (Completing Completeness) (i), we know that $[\Omega_2]\hat{\alpha}_2 = [\Omega_1]\hat{\alpha}_2 = \tau_2$.
By Lemma 55 (Completing Completeness) (i), we know that $[\Omega_2]\tau_2 = [\Omega]\tau_2$.
Hence we know that $[\Omega_2]\Delta_2 \vdash [\Omega_2]\hat{\alpha}_2 \leq^* [\Omega_2]\tau_2$.
By i.h., we have Δ and Ω' such that $\Delta_2 \vdash \hat{\alpha}_2 := [\Delta_2]\tau_2 : \kappa \dashv \Delta$ and $\Omega_2 \longrightarrow \Omega'$ and $\Delta \longrightarrow \Omega'$.
By rule InstBin, $\Gamma \vdash \hat{\alpha} := \tau : \kappa \dashv \Delta$.
By Lemma 33 (Extension Transitivity), $\Omega \longrightarrow \Omega'$.

- **Case** $\tau = \mathsf{succ}(\tau_0)$:

  Similar to the $\tau = \tau_1 \oplus \tau_2$ case, but simpler. □

Lemma 91 (Completeness of Checkeq).
Given $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$
and $\Gamma \vdash \sigma : \kappa$ *and* $\Gamma \vdash \tau : \kappa$
and $[\Omega]\sigma = [\Omega]\tau$
then $\Gamma \vdash [\Gamma]\sigma \mathrel{\mathring{=}} [\Gamma]\tau : \kappa \dashv \Delta$
where $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$.

Proof. By mutual induction on the sizes of $[\Gamma]\sigma$ and $[\Gamma]\tau$.
We distinguish cases of $[\Gamma]\sigma$ and $[\Gamma]\tau$.

- **Case** $[\Gamma]\sigma = [\Gamma]\tau = 1$:

☞	$\Gamma \vdash 1 \mathrel{\mathring{=}} 1 : \star \dashv \underbrace{\Gamma}_{\Delta}$	By CheckeqUnit
	Let $\Omega' = \Omega$.	
	$\Gamma \longrightarrow \Omega$	Given
☞	$\Delta \longrightarrow \Omega'$	$\Delta = \Gamma$ and $\Omega' = \Omega$
☞	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
☞	$\Omega \longrightarrow \Omega'$	By Lemma 32 (Extension Reflexivity)

- **Case** $[\Gamma]\sigma = [\Gamma]t = \mathsf{zero}$:

  Similar to the case for 1, applying CheckeqZero instead of CheckeqUnit.

- **Case** $[\Gamma]\sigma = [\Gamma]t = \alpha$:

  Similar to the case for 1, applying CheckeqVar instead of CheckeqUnit.

- **Case** $[\Gamma]\sigma = \hat{\alpha}$ and $[\Gamma]t = \hat{\beta}$:

  - If $\hat{\alpha} = \hat{\beta}$: Similar to the case for 1, applying CheckeqVar instead of CheckeqUnit.
  - If $\hat{\alpha} \neq \hat{\beta}$:

	$\Gamma \longrightarrow \Omega$	Given
	$\hat{\alpha} \notin FV(\underbrace{\hat{\beta}}_{[\Gamma]t})$	By definition of $FV(-)$
	$[\Omega]\sigma = [\Omega]t$	Given
	$[\Omega][\Gamma]\sigma = [\Omega][\Gamma]t$	By Lemma 29 (Substitution Monotonicity) (i) twice
	$[\Omega]\hat{\alpha} = [\Omega][\Gamma]t$	$[\Gamma]\sigma = \hat{\alpha}$
	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
	$\Gamma \vdash \hat{\alpha} := [\Gamma]t : \kappa \dashv \Delta$	By Lemma 90 (Completeness of Instantiation)
☞	$\Omega \longrightarrow \Omega'$	"
☞	$\Delta \longrightarrow \Omega$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
☞	$\Gamma \vdash \hat{\alpha} \mathrel{\mathring{=}} [\Gamma]t : \kappa \dashv \Delta$	By CheckeqInstL

- **Case** $[\Gamma]\sigma = \hat{\alpha}$ and $[\Gamma]t = 1$ or zero or α:

  Similar to the previous case, except:

  $\hat{\alpha} \notin FV(\underbrace{1}_{[\Gamma]t})$ By definition of $FV(-)$

  and similarly for 1 and α.

- **Case** $[\Gamma]t = \hat{\alpha}$ and $[\Gamma]\sigma = 1$ or zero or α: Symmetric to the previous case.
- **Case** $[\Gamma]\sigma = \hat{\alpha}$ and $[\Gamma]t = \mathsf{succ}([\Gamma]t_0)$:

  If $\hat{\alpha} \notin FV([\Gamma]t_0)$, then $\hat{\alpha} \notin FV([\Gamma]t)$. Proceed as in the previous several cases.

  The other case, $\hat{\alpha} \in FV([\Gamma]t_0)$, is impossible:

  We have $\hat{\alpha} \preceq [\Gamma]t_0$.
  Therefore $\hat{\alpha} \prec \mathsf{succ}([\Gamma]t_0)$, that is, $\hat{\alpha} \prec [\Gamma]t$.
  By a property of substitutions, $[\Omega]\hat{\alpha} \prec [\Omega][\Gamma]t$.
  Since $\Gamma \longrightarrow \Omega$, by Lemma 29 (Substitution Monotonicity) (i), $[\Omega][\Gamma]t = [\Omega]t$, so $[\Omega]\hat{\alpha} \prec [\Omega]t$.
  But it is given that $[\Omega]\hat{\alpha} = [\Omega]t$, a contradiction.

- **Case** $[\Gamma]t = \hat{\alpha}$ and $[\Gamma]\sigma = \mathsf{succ}([\Gamma]\sigma_0)$: Symmetric to the previous case.
- **Case** $[\Gamma]\sigma = [\Gamma]\sigma_1 \oplus [\Gamma]\sigma_2$ and $[\Gamma]t = [\Gamma]t_1 \oplus [\Gamma]t_2$:

	$\Gamma \longrightarrow \Omega$	Given
	$\Gamma \vdash [\Gamma]\sigma_1 \stackrel{\circ}{=} [\Gamma]t_1 : \star \dashv \Theta$	By i.h.
	$\Theta \longrightarrow \Omega_0$	″
	$\Omega \longrightarrow \Omega_0$	″
	$\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_0)$	″
	$\Theta \vdash [\Theta][\Gamma]\sigma_2 \stackrel{\circ}{=} [\Theta][\Gamma]t_2 : \star \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega'$	″
	$\Omega_0 \longrightarrow \Omega'$	″
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	″
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \vdash [\Gamma]\sigma_1 \oplus [\Gamma]\sigma_2 \stackrel{\circ}{=} [\Gamma]t_1 \oplus [\Gamma]t_2) : \star \dashv \Delta$	By CheckeqBin

- **Case** $[\Gamma]\sigma = \hat{\alpha}$ and $[\Gamma]t = t_1 \oplus t_2$: Similar to the $\hat{\alpha}/\mathsf{succ}(-)$ case, showing the impossibility of $\hat{\alpha} \in FV([\Gamma]t_k)$ for $k = 1$ and $k = 2$.
- **Case** $[\Gamma]t = \hat{\alpha}$ and $[\Gamma]\sigma = \sigma_1 \oplus \sigma_2$: Symmetric to the previous case. □

Lemma 92 (Completeness of Elimeq)**.**
If $[\Gamma]\sigma = \sigma$ *and* $[\Gamma]t = t$ *and* $\Gamma \vdash \sigma : \kappa$ *and* $\Gamma \vdash t : \kappa$ *and* $\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(t) = \emptyset$ *then:*

(1) If $\mathsf{mgu}(\sigma, t) = \theta$
then $\Gamma \;/\; \sigma \stackrel{\circ}{=} t : \kappa \dashv (\Gamma, \Delta)$
where Δ *has the form* $\alpha_1 = t_1, \ldots, \alpha_n = t_n$
and for all u *such that* $\Gamma \vdash u : \kappa$*, it is the case that* $[\Gamma, \Delta]u = \theta([\Gamma]u)$.

(2) If $\mathsf{mgu}(\sigma, t) = \bot$ *(that is, no most general unifier exists) then* $\Gamma \;/\; \sigma \stackrel{\circ}{=} t : \kappa \dashv \bot$.

Proof. By induction on the structure of $[\Gamma]\sigma$ and $[\Gamma]t$.

- Case $[\Omega]\sigma = t = \mathsf{zero}$:

	$\mathsf{mgu}(\mathsf{zero}, \mathsf{zero}) = \cdot$	By properties of unification
	$\Gamma \;/\; \mathsf{zero} \stackrel{\circ}{=} \mathsf{zero} : \mathbb{N} \dashv \Gamma$	By rule ElimeqZero
☞	$\Gamma \;/\; \mathsf{zero} \stackrel{\circ}{=} \mathsf{zero} : \mathbb{N} \dashv \Gamma, \Delta$	where $\Delta = \cdot$
☞	Suppose $\Gamma \vdash u : \kappa'$.	
	$[\Gamma, \Delta]u = [\Gamma]u$	where $\Delta = \cdot$
	$= \theta([\Gamma]u)$	where θ is the identity

- Case $\sigma = \mathsf{succ}(\sigma')$ and $t = \mathsf{succ}(t')$:

- Case $\mathsf{mgu}(\mathsf{succ}(\sigma'), \mathsf{succ}(t')) = \theta$:

$$
\begin{array}{rcll}
& \mathsf{mgu}(\sigma', t') = \mathsf{mgu}(\mathsf{succ}(\sigma'), \mathsf{succ}(t')) = \theta & & \text{By properties of unification} \\
& \mathsf{succ}(\sigma') = [\Gamma]\mathsf{succ}(\sigma') & & \text{Given} \\
& = \mathsf{succ}([\Gamma]\sigma') & & \text{By definition of substitution} \\
& \sigma' = [\Gamma]\sigma' & & \text{By injectivity of successor} \\
& \mathsf{succ}(t') = [\Gamma]\mathsf{succ}(t') & & \text{Given} \\
& = \mathsf{succ}([\Gamma]t') & & \text{By definition of substitution} \\
& t' = [\Gamma]t' & & \text{By injectivity of successor} \\
& \Gamma \;/\; \sigma' \stackrel{\circ}{=} t' : \mathbb{N} \dashv \Gamma, \Delta & & \text{By i.h.} \\
☞ & [\Gamma, \Delta]u = \theta([\Gamma]u) \text{ for all } u \text{ such that} \ldots & & '' \\
☞ & \Gamma \;/\; \mathsf{succ}(\sigma') \stackrel{\circ}{=} \mathsf{succ}(t') : \mathbb{N} \dashv \Gamma, \Delta & & \text{By rule ElimeqSucc}
\end{array}
$$

- Case $\mathsf{mgu}(\mathsf{succ}(\sigma'), \mathsf{succ}(t')) = \bot$:

$$
\begin{array}{rcll}
& \mathsf{mgu}(\sigma', t') = \mathsf{mgu}(\mathsf{succ}(\sigma'), \mathsf{succ}(t')) = \bot & & \text{By properties of unification} \\
& \mathsf{succ}(\sigma') = [\Gamma]\mathsf{succ}(\sigma') & & \text{Given} \\
& = \mathsf{succ}([\Gamma]\sigma') & & \text{By definition of substitution} \\
& \sigma' = [\Gamma]\sigma' & & \text{By injectivity of successor} \\
& \mathsf{succ}(t') = [\Gamma]\mathsf{succ}(t') & & \text{Given} \\
& = \mathsf{succ}([\Gamma]t') & & \text{By definition of substitution} \\
& t' = [\Gamma]t' & & \text{By injectivity of successor} \\
& \Gamma \;/\; \sigma' \stackrel{\circ}{=} t' : \mathbb{N} \dashv \bot & & \text{By i.h.} \\
☞ & \Gamma \;/\; \mathsf{succ}(\sigma') \stackrel{\circ}{=} \mathsf{succ}(t') : \mathbb{N} \dashv \bot & & \text{By rule ElimeqSucc}
\end{array}
$$

- Case $\sigma = \sigma_1 \oplus \sigma_2$ and $t = t_1 \oplus t_2$:

  First we establish some properties of the subterms:

$$
\begin{array}{rrcll}
& \sigma_1 \oplus \sigma_2 & = & [\Gamma](\sigma_1 \oplus \sigma_2) & \text{Given} \\
& & = & [\Gamma]\sigma_1 \oplus [\Gamma]\sigma_2 & \text{By definition of substitution} \\
☞ & [\Gamma]\sigma_1 & = & \sigma_1 & \text{By injectivity of } \oplus \\
☞ & [\Gamma]\sigma_2 & = & \sigma_2 & \text{By injectivity of } \oplus \\
& t_1 \oplus t_2 & = & [\Gamma](t_1 \oplus t_2) & \text{Given} \\
& & = & [\Gamma]t_1 \oplus [\Gamma]t_2 & \text{By definition of substitution} \\
☞ & [\Gamma]t_1 & = & t_1 & \text{By injectivity of } \oplus \\
☞ & [\Gamma]t_2 & = & t_2 & \text{By injectivity of } \oplus
\end{array}
$$

  - Subcase $\mathsf{mgu}(\sigma, t) = \bot$:
    - Subcase $\mathsf{mgu}(\sigma_1, t_1) = \bot$:

$$
\begin{array}{ll}
\Gamma \;/\; \sigma_1 \stackrel{\circ}{=} t_1 : \kappa \dashv \bot & \text{By i.h.} \\
\Gamma \;/\; \sigma_1 \oplus \sigma_2 \stackrel{\circ}{=} t_1 \oplus t_2 : \kappa \dashv \bot & \text{By rule ElimeqBinBot}
\end{array}
$$

    - Subcase $\mathsf{mgu}(\sigma_1, t_1) = \theta_1$ and $\mathsf{mgu}(\theta_1(\sigma_2), \theta_1(t_2)) = \bot$:

$$
\begin{array}{rl}
\Gamma \;/\; \sigma_1 \stackrel{\circ}{=} t_1 : \kappa \dashv \Gamma, \Delta_1 & \text{By i.h.} \\
[\Gamma, \Delta_1]u = \theta_1([\Gamma]u) \text{ for all } u \text{ such that} \ldots & ''
\end{array}
$$

$$
\begin{array}{rcll}
[\Gamma, \Delta_1]\sigma_2 & = & \theta_1([\Gamma]\sigma_2) & \text{Above line with } \sigma_2 \text{ as } u \\
& = & \theta_1(\sigma_2) & [\Gamma]\sigma_2 = \sigma_2 \\
[\Gamma, \Delta_1]t_2 & = & \theta_1([\Gamma]t_2) & \text{Above line with } t_2 \text{ as } u \\
& = & \theta_1(t_2) & \text{Since } [\Gamma]\sigma_2 = \sigma_2 \\
\mathsf{mgu}([\Gamma, \Delta_1]\sigma_2, [\Gamma, \Delta_1]t_2) & = & \theta_2 & \text{By transitivity of equality}
\end{array}
$$

$$
\begin{array}{rcll}
[\Gamma, \Delta_1][\Gamma, \Delta_1]\sigma_2 & = & [\Gamma, \Delta_1]\sigma_2 & \text{By Lemma 29 (Substitution Monotonicity)} \\
[\Gamma, \Delta_1][\Gamma, \Delta_1]t_2 & = & [\Gamma, \Delta_1]t_2 & \text{By Lemma 29 (Substitution Monotonicity)}
\end{array}
$$

$$\begin{array}{rll}
& \Gamma, \Delta_1 \ / \ [\Gamma, \Delta_1]\sigma_2 \doteq [\Gamma, \Delta_1]t_2 : \kappa \dashv \bot & \text{By i.h.} \\
☞ & \Gamma \ / \ \sigma_1 \oplus \sigma_2 \doteq t_1 \oplus t_2 : \kappa \dashv \bot & \text{By rule ElimeqBin}
\end{array}$$

– Subcase $\mathsf{mgu}(\sigma, t) = \theta$:

$$\begin{array}{rrcll}
& \mathsf{mgu}(\sigma_1 \oplus \sigma_2, t_1 \oplus t_2) & = & \theta = \theta_2 \circ \theta_1 & \text{By properties of unifiers} \\
& \mathsf{mgu}(\sigma_1, t_1) & = & \theta_1 & '' \\
& \mathsf{mgu}(\theta_1(\sigma_2), \theta_1(t_2)) & = & \theta_2 & '' \\
& & & \Gamma \ / \ \sigma_1 \doteq t_1 : \kappa \dashv \Gamma, \Delta_1 & \text{By i.h.} \\
* & & & [\Gamma, \Delta_1]u = \theta_1([\Gamma]u) \text{ for all } u \text{ such that} \ldots & '' \\
& [\Gamma, \Delta_1]\sigma_2 & = & \theta_1([\Gamma]\sigma_2) & \text{Above line with } \sigma_2 \text{ as } u \\
& & = & \theta_1(\sigma_2) & [\Gamma]\sigma_2 = \sigma_2 \\
& [\Gamma, \Delta_1]t_2 & = & \theta_1([\Gamma]t_2) & \text{Above line with } t_2 \text{ as } u \\
& & = & \theta_1(t_2) & [\Gamma]\sigma_2 = \sigma_2 \\
& \mathsf{mgu}([\Gamma, \Delta_1]\sigma_2, [\Gamma, \Delta_1]t_2) & = & \theta_2 & \text{By transitivity of equality}
\end{array}$$

$$\begin{array}{rcll}
[\Gamma, \Delta_1][\Gamma, \Delta_1]\sigma_2 & = & [\Gamma, \Delta_1]\sigma_2 & \text{By Lemma 29 (Substitution Monotonicity)} \\
[\Gamma, \Delta_1][\Gamma, \Delta_1]t_2 & = & [\Gamma, \Delta_1]t_2 & \text{By Lemma 29 (Substitution Monotonicity)}
\end{array}$$

$$\begin{array}{rll}
& \Gamma, \Delta_1 \ / \ [\Gamma, \Delta_1]\sigma_2 \doteq [\Gamma, \Delta_1]t_2 : \kappa \dashv \Gamma, \Delta_1, \Delta_2 & \text{By i.h.} \\
** & [\Gamma, \Delta_1, \Delta_2]u' = \theta_2([\Gamma, \Delta_1]u') \text{ for all } u' \text{ such that} \ldots & '' \\
☞ & \Gamma \ / \ \sigma_1 \oplus \sigma_2 \doteq t_1 \oplus t_2 : \kappa \dashv \Gamma, \Delta_1, \Delta_2 & \text{By rule ElimeqBin}
\end{array}$$

☞ Suppose $\Gamma \vdash u : \kappa'$.

$$\begin{array}{rcll}
[\Gamma, \Delta_1, \Delta_2]u & = & \theta_2([\Gamma, \Delta_1]u) & \text{By } ** \\
& = & \theta_2(\theta_1([\Gamma]u)) & \text{By } * \\
& = & \theta([\Gamma]u) & \theta = \theta_2 \circ \theta_1
\end{array}$$

- Case $\sigma = \alpha$:

  – Subcase $\alpha \in FV(t)$:

$$\begin{array}{rll}
& \mathsf{mgu}(\alpha, t) = \bot & \text{By properties of unification} \\
☞ & \Gamma \ / \ \alpha \doteq t : \kappa \dashv \bot & \text{By rule ElimeqUvarL}\bot
\end{array}$$

  – Subcase $\alpha \notin FV(t)$:

$$\begin{array}{rll}
& \mathsf{mgu}(\alpha, t) = [t/\alpha] & \text{By properties of unification} \\
& (\alpha = t') \notin \Gamma & [\Gamma]\alpha = \alpha \\
☞ & \Gamma \ / \ \alpha \doteq t : \kappa \dashv \Gamma, \alpha = t & \text{By rule ElimeqUvarL}
\end{array}$$

☞ Suppose $\Gamma \vdash u : \kappa'$.

$$\begin{array}{rcll}
[\Gamma, \alpha = t]u & = & [\Gamma]([t/\alpha]u) & \text{By definition of substitution} \\
& = & \big[[\Gamma]t/\alpha\big][\Gamma]u & \text{By properties of substitution} \\
& = & [t/\alpha][\Gamma]u & [\Gamma]t = t
\end{array}$$

- Case $t = \alpha$: Similar to previous case. □

Lemma 93 (Substitution Upgrade)**.**
If Δ *has the form* $\alpha_1 = t_1, \ldots, \alpha_n = t_n$
and, for all u *such that* $\Gamma \vdash u : \kappa$*, it is the case that* $[\Gamma, \Delta]u = \theta([\Gamma]u)$*,*
then:

(i) *If* $\Gamma \vdash A$ *type then* $[\Gamma, \Delta]A = \theta([\Gamma]A)$.

(ii) *If* $\Gamma \longrightarrow \Omega$ *then* $[\Omega]\Gamma = \theta([\Omega]\Gamma)$.

(iii) *If* $\Gamma \longrightarrow \Omega$ *then* $[\Omega, \Delta](\Gamma, \Delta) = \theta([\Omega]\Gamma)$.

(iv) *If* $\Gamma \longrightarrow \Omega$ *then* $[\Omega, \Delta]e = \theta([\Omega]e)$.

Proof. Part (i): By induction on the given derivation, using the given "for all" at the leaves.

Part (ii): By induction on the given derivation, using part (i) in the $\longrightarrow$Var case.

Part (iii): By induction on Δ. In the base case ($\Delta = \cdot$), use part (ii). Otherwise, use the i.h. and the definition of context substitution.

Part (iv): By induction on e, using part (i) in the $e = (e_0 : A)$ case. □

Lemma 94 (Completeness of Propequiv).
Given $\Gamma \longrightarrow \Omega$
and $\Gamma \vdash P$ *prop* *and* $\Gamma \vdash Q$ *prop*
and $[\Omega]P = [\Omega]Q$
then $\Gamma \vdash [\Gamma]P \equiv [\Gamma]Q \dashv \Delta$
where $\Delta \longrightarrow \Omega'$ *and* $\Omega \longrightarrow \Omega'$.

Proof. By induction on the given derivations. There is only one possible case:

- **Case** $\dfrac{\Gamma \vdash \sigma_1 : \mathbb{N} \qquad \Gamma \vdash \sigma_2 : \mathbb{N}}{\Gamma \vdash \sigma_1 = \sigma_2\ \textit{prop}}\ \textsf{EqProp} \qquad \dfrac{\Gamma \vdash \tau_1 : \mathbb{N} \qquad \Gamma \vdash \tau_2 : \mathbb{N}}{\Gamma \vdash \tau_1 = \tau_2\ \textit{prop}}\ \textsf{EqProp}$

	$[\Omega](\sigma_1 = \sigma_2) = [\Omega](\tau_1 = \tau_2)$	Given
	$[\Omega]\sigma_1 = [\Omega]\tau_1$	Definition of substitution
	$[\Omega]\sigma_2 = [\Omega]\tau_2$	$''$
	$\Gamma \vdash \sigma_1 : \mathbb{N}$	Subderivation
	$\Gamma \vdash \tau_1 : \mathbb{N}$	Subderivation
	$\Gamma \vdash [\Gamma]\sigma_1 \doteq [\Gamma]\sigma_2 : \mathbb{N} \dashv \Theta$	By Lemma 91 (Completeness of Checkeq)
	$\Theta \longrightarrow \Omega_0$	$''$
	$\Omega \longrightarrow \Omega_0$	$''$
	$\Gamma \vdash \sigma_2 : \mathbb{N}$	Subderivation
	$\Theta \vdash \sigma_2 : \mathbb{N}$	By Lemma 36 (Extension Weakening (Sorts))
	$\Theta \vdash \tau_2 : \mathbb{N}$	Similarly
	$\Theta \vdash [\Theta]\tau_1 \doteq [\Theta]\tau_2 : \mathbb{N} \dashv \Delta$	By Lemma 91 (Completeness of Checkeq)
☞	$\Delta \longrightarrow \Omega_0$	$''$
	$\Omega_0 \longrightarrow \Omega'$	$''$
	$[\Theta]\tau_1 = [\Theta][\Gamma]\tau_1$	By Lemma 29 (Substitution Monotonicity) (i)
	$[\Theta]\tau_2 = [\Theta][\Gamma]\tau_2$	$''$
	$\Theta \vdash [\Theta][\Gamma]\tau_1 \doteq [\Theta][\Gamma]\tau_2 : \mathbb{N} \dashv \Delta$	By above equalities
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
	$\Gamma \vdash ([\Gamma]\sigma_1 = [\Theta]\sigma_2) \equiv ([\Gamma]\tau_1 = [\Theta]\tau_2) \dashv \Gamma$	By $\equiv$PropEq
☞	$\Gamma \vdash ([\Gamma]\sigma_1 = [\Gamma]\sigma_2) \equiv ([\Gamma]\tau_1 = [\Gamma]\tau_2) \dashv \Gamma$	By above equalities

□

Lemma 95 (Completeness of Checkprop).
If $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$
and $\Gamma \vdash P$ *prop*
and $[\Gamma]P = P$
and $[\Omega]\Gamma \vdash [\Omega]P$ *true*
then $\Gamma \vdash P$ *true* $\dashv \Delta$
where $\Delta \longrightarrow \Omega'$ *and* $\Omega \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$.

Proof. Only one rule, DeclCheckpropEq, can derive $[\Omega]\Gamma \vdash [\Omega]P$ *true*, so by inversion, P has the form $(t_1 = t_2)$ where $[\Omega]t_1 = [\Omega]t_2$.
By inversion on $\Gamma \vdash (t_1 = t_2)$ *prop*, we have $\Gamma \vdash t_1 : \mathbb{N}$ and $\Gamma \vdash t_2 : \mathbb{N}$.
Then by Lemma 91 (Completeness of Checkeq), $\Gamma \vdash [\Gamma]t_1 \stackrel{\circ}{=} [\Gamma]t_2 : \mathbb{N} \dashv \Delta$ where $\Delta \longrightarrow \Omega'$ and $\Omega \longrightarrow \Omega'$.
By CheckpropEq, $\Gamma \vdash (t_1 = t_2)$ *true* $\dashv \Delta$. □

K′.2 Completeness of Equivalence and Subtyping

Lemma 96 (Completeness of Equiv)**.**
If $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type*
and $[\Omega]A = [\Omega]B$
then there exist Δ *and* Ω' *such that* $\Delta \longrightarrow \Omega'$ *and* $\Omega \longrightarrow \Omega'$ *and* $\Gamma \vdash [\Gamma]A \equiv [\Gamma]B \dashv \Delta$.

Proof. By induction on the derivations of $\Gamma \vdash A$ *type* and $\Gamma \vdash B$ *type*.

We distinguish cases of the rule concluding the first derivation. In the first four cases (ImpliesWF, WithWF, ForallWF, ExistsWF), it follows from $[\Omega]A = [\Omega]B$ and the syntactic invariant that Ω substitutes terms t (rather than types A) that the second derivation is concluded by the *same* rule. Moreover, if none of these three rules concluded the first derivation, the rule concluding the second derivation must *not* be ImpliesWF, WithWF, ForallWF or ExistsWF either.

Because Ω is predicative, the head connective of $[\Gamma]A$ must be the same as the head connective of $[\Omega]A$.

We distinguish cases that are *imposs.* (impossible), **fully written out**, and similar to fully-written-out cases. For the lower-right case, where both $[\Gamma]A$ and $[\Gamma]B$ have a binary connective $\oplus$, it must be the same connective.

The Vec type is omitted from the table, but can be treated similarly to $\supset$ and $\wedge$.

$[\Gamma]A$ \ $[\Gamma]B$	$\supset$	$\wedge$	$\forall\beta.\,B'$	$\exists\beta.\,B'$	1	α	$\hat{\beta}$	$B_1 \oplus B_2$
$\supset$	**Implies**	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*
$\wedge$	*imposs.*	With	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*
$\forall\alpha.\,A'$	*imposs.*	*imposs.*	**Forall**	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*
$\exists\alpha.\,A'$	*imposs.*	*imposs.*	*imposs.*	Exists	*imposs.*	*imposs.*	*imposs.*	*imposs.*
1	*imposs.*	*imposs.*	*imposs.*	*imposs.*	**2.Units**	*imposs.*	2.BEx.Unit	*imposs.*
α	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*	**2.Uvars**	2.BEx.Uvar	*imposs.*
$\hat{\alpha}$	*imposs.*	*imposs.*	*imposs.*	*imposs.*	**2.AEx.Unit**	2.AEx.Uvar	**2.AEx.SameEx** **2.AEx.OtherEx**	**2.AEx.Bin**
$A_1 \oplus A_2$	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*	*imposs.*	2.BEx.Bin	2.Bins

- **Case** $$\dfrac{\Gamma \vdash P\ \textit{prop} \qquad \Gamma \vdash A_0\ \textit{type}}{\Gamma \vdash P \supset A_0\ \textit{type}}\ \textsf{ImpliesWF}$$

  This case of the rule concluding the first derivation coincides with the **Implies** entry in the table.

  We have $[\Omega]A = [\Omega]B$, that is, $[\Omega](P \supset A_0) = [\Omega]B$.
  Because Ω is predicative, B must have the form $Q \supset B_0$, where $[\Omega]P = [\Omega]Q$ and $[\Omega]A_0 = [\Omega]B_0$.

$\Gamma \vdash P\ prop$	Subderivation
$\Gamma \vdash A_0\ type$	Subderivation
$\Gamma \vdash Q \supset B_0\ type$	Given
$\Gamma \vdash Q\ prop$	By inversion on rule ImpliesWF
$\Gamma \vdash B_0\ type$	"
$\Gamma \vdash [\Gamma]P \equiv [\Gamma]Q \dashv \Theta$	By Lemma 94 (Completeness of Propequiv)
$\Theta \longrightarrow \Omega_0$	"
$\Omega \longrightarrow \Omega_0$	"
$\Gamma \longrightarrow \Theta$	By Lemma 48 (Prop Equivalence Extension)
$\Gamma \vdash A_0\ type$	Above
$\Gamma \vdash B_0\ type$	Above
$[\Omega]A_0 = [\Omega]B_0$	Above
$[\Omega_0]A_0 = [\Omega_0]B_0$	By Lemma 55 (Completing Completeness) (ii) twice
$\Gamma \vdash [\Gamma]A_0 \equiv [\Gamma]B_0 \dashv \Delta$	By i.h.
☞ $\Delta \longrightarrow \Omega'$	"
$\Omega_0 \longrightarrow \Omega'$	"
☞ $\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
$\Gamma \vdash ([\Gamma]P \supset [\Gamma]A_0) \equiv ([\Gamma]Q \supset [\Gamma]B_0) \dashv \Delta$	By $\equiv\supset$
☞ $\Gamma \vdash [\Gamma](P \supset A_0) \equiv [\Gamma](Q \supset B_0) \dashv \Delta$	By definition of substitution

- **Case** WithWF: Similar to the ImpliesWF case, coinciding with the With entry in the table.

- **Case** $\dfrac{\Gamma, \alpha : \kappa \vdash A_0\ type}{\Gamma \vdash \forall \alpha : \kappa.\ A_0\ type}$ ForallWF

  This case coincides with the **Forall** entry in the table.

$\Gamma \longrightarrow \Omega$	Given
$\Gamma, \alpha : \kappa \longrightarrow \Omega, \alpha : \kappa$	By $\longrightarrow$Uvar
$\Gamma, \alpha : \kappa \vdash A_0\ type$	Subderivation
$B = \forall \alpha : \kappa.\ B_0$	Ω predicative
$[\Omega]A_0 = [\Omega]B_0$	From definition of substitution
$\Gamma, \alpha : \kappa \vdash [\Gamma]A_0 \equiv [\Gamma]B_0 \dashv \Delta_0$	By i.h.
$\Delta_0 \longrightarrow \Omega_0$	"
$\Omega, \alpha : \kappa \longrightarrow \Omega_0$	"
☞ $\Omega \longrightarrow \Omega'$ and $\Omega_0 = (\Omega', \alpha : \kappa, \dots)$	By Lemma 22 (Extension Inversion) (i)
$\Delta_0 = (\Delta, \alpha : \kappa, \Delta')$	By Lemma 22 (Extension Inversion) (i)
☞ $\Delta \longrightarrow \Omega'$	"
$\Gamma \vdash \forall \alpha : \kappa.\ [\Gamma]A_0 \equiv \forall \alpha : \kappa.\ [\Gamma]B_0 \dashv \Delta$	By $\equiv\forall$
☞ $\Gamma \vdash [\Gamma](\forall \alpha : \kappa.\ A_0) \equiv [\Gamma](\forall \alpha : \kappa.\ B_0) \dashv \Delta$	By definition of substitution

- **Case** ExistsWF: Similar to the ForallWF case. (This is the Exists entry in the table.)

- **Case** BinWF: If BinWF also concluded the second derivation, then the proof is similar to the ImpliesWF case, but on the first premise, using the i.h. instead of Lemma 94 (Completeness of Propequiv). This is the 2.Bins entry in the lower right corner of the table.

  If BinWF did not conclude the second derivation, we are in the **2.AEx.Bin** or 2.BEx.Bin entries; see below.

In the remainder, we cover the 4×4 region in the lower right corner, starting from **2.Units**. We already handled the 2.Bins entry in the extreme lower right corner. At this point, we split on the forms of $[\Gamma]A$ and $[\Gamma]B$ instead; in the remaining cases, one or both types is atomic (e.g. **2.Uvars**, **2.AEx.Bin**) and we will not need to use the induction hypothesis.

- **Case 2.Units:** $[\Gamma]A = [\Gamma]B = 1$

☞	$\Gamma \vdash 1 \equiv 1 \dashv \Gamma$	By $\equiv$Unit
	$\Gamma \longrightarrow \Omega$	Given
	Let $\Omega' = \Omega'$.	
☞	$\Delta \longrightarrow \Omega$	$\Delta = \Gamma$
☞	$\Omega \longrightarrow \Omega'$	By Lemma 32 (Extension Reflexivity) and $\Omega' = \Omega$

- **Case 2.Uvars:** $[\Gamma]A = [\Gamma]B = \alpha$

	$\Gamma \longrightarrow \Omega$	Given
	Let $\Omega' = \Omega'$.	
☞	$\Gamma \vdash \alpha \equiv \alpha \dashv \Gamma$	By $\equiv$Var
☞	$\Delta \longrightarrow \Omega$	$\Delta = \Gamma$
☞	$\Omega \longrightarrow \Omega'$	By Lemma 32 (Extension Reflexivity) and $\Omega' = \Omega$

- **Case 2.AExUnit:** $[\Gamma]A = \hat{\alpha}$ and $[\Gamma]B = 1$

	$\Gamma \longrightarrow \Omega$	Given
	$1 = [\Omega]1$	By definition of substitution
	$\hat{\alpha} \notin FV(1)$	By definition of $FV(-)$
	$[\Omega]\Gamma \vdash [\Omega]\hat{\alpha} \leq^{\pm} [\Omega]1$	Given
	$\Gamma \vdash \hat{\alpha} := 1 : \star \dashv \Delta$	By Lemma 90 (Completeness of Instantiation) (1)
☞	$\Omega \longrightarrow \Omega'$	$''$
☞	$\Delta \longrightarrow \Omega'$	$''$
	$1 = [\Gamma]1$	By definition of substitution
	$\hat{\alpha} \notin FV(1)$	By definition of $FV(-)$
☞	$\Gamma \vdash \hat{\alpha} \equiv 1 \dashv \Delta$	By $\equiv$InstantiateL

- **Case 2.BExUnit:** $[\Gamma]A = 1$ and $[\Gamma]B = \hat{\alpha}$

  Symmetric to the **2.AExUnit** case.

- **Case 2.AEx.Uvar:** $[\Gamma]A = \hat{\alpha}$ and $[\Gamma]B = \alpha$

  Similar to the **2.AEx.Unit** case, using $\beta = [\Omega]\beta = [\Gamma]\beta$ and $\hat{\alpha} \notin FV(\beta)$.

- **Case 2.BExUvar:** $[\Gamma]A = 1$ and $[\Gamma]B = \hat{\alpha}$

  Symmetric to the **2.AExUvar** case.

- **Case 2.AEx.SameEx:** $[\Gamma]A = \hat{\alpha} = \hat{\beta} = [\Gamma]B$

	$\Gamma \vdash \hat{\alpha} \equiv \hat{\alpha} \dashv \Gamma$	By $\equiv$Exvar ($\hat{\alpha} = \hat{\beta}$)
	$[\Gamma]\hat{\alpha} = \hat{\alpha}$	$\hat{\alpha}$ unsolved in Γ
☞	$\Gamma \vdash [\Gamma]\hat{\alpha} \equiv [\Gamma]\hat{\beta} \dashv \Gamma$	By above equality + $\hat{\alpha} = \hat{\beta}$
	$\Gamma \longrightarrow \Omega$	Given
☞	$\Delta \longrightarrow \Omega$	$\Delta = \Gamma$
	Let $\Omega' = \Omega$.	
☞	$\Omega \longrightarrow \Omega'$	By Lemma 32 (Extension Reflexivity) and $\Omega' = \Omega$

- **Case 2.AEx.OtherEx:** $[\Gamma]A = \hat{\alpha}$ and $[\Gamma]B = \hat{\beta}$ and $\hat{\alpha} \neq \hat{\beta}$

  Either $\hat{\alpha} \in FV([\Gamma]\hat{\beta})$, or $\hat{\alpha} \notin FV([\Gamma]\hat{\beta})$.

  - $\hat{\alpha} \in FV([\Gamma]\hat{\beta})$:
    We have $\hat{\alpha} \preceq [\Gamma]\hat{\beta}$.
    Therefore $\hat{\alpha} = [\Gamma]\hat{\beta}$, or $\hat{\alpha} \prec [\Gamma]\hat{\beta}$.
    But we are in Case 2.AEx.*Other*Ex, so the former is impossible.
    Therefore, $\hat{\alpha} \prec [\Gamma]\hat{\beta}$.
    By a property of substitutions, $[\Omega]\hat{\alpha} \prec [\Omega][\Gamma]\hat{\beta}$.
    Since $\Gamma \longrightarrow \Omega$, by Lemma 29 (Substitution Monotonicity) (iii), $[\Omega][\Gamma]\hat{\beta} = [\Omega]\hat{\beta}$, so $[\Omega]\hat{\alpha} \prec [\Omega]\hat{\beta}$.
    But it is given that $[\Omega]\hat{\alpha} = [\Omega]\hat{\beta}$, a contradiction.
  - $\hat{\alpha} \notin FV([\Gamma]\hat{\beta})$:

	$\Gamma \vdash \hat{\alpha} := [\Gamma]\hat{\beta} : \star \dashv \Delta$	By Lemma 90 (Completeness of Instantiation)
☞	$\Gamma \vdash \hat{\alpha} \equiv [\Gamma]\hat{\beta} \dashv \Delta$	By $\equiv$InstantiateL
☞	$\Delta \longrightarrow \Omega'$	''
☞	$\Omega \longrightarrow \Omega'$	''

- **Case 2.AEx.Bin:** $[\Gamma]A = \hat{\alpha}$ and $[\Gamma]B = B_1 \oplus B_2$

  Since $[\Gamma]B$ is an arrow, it cannot be exactly $\hat{\alpha}$. By the same reasoning as in the previous case (2.AEx.OtherEx), $\hat{\alpha} \notin FV([\Gamma]\hat{\beta})$.

	$\Gamma \vdash \hat{\alpha} := [\Gamma]B : \star \dashv \Delta$	By Lemma 90 (Completeness of Instantiation)
☞	$\Delta \longrightarrow \Omega'$	''
☞	$\Omega \longrightarrow \Omega'$	''
☞	$\Gamma \vdash \underbrace{[\Gamma]A}_{\hat{\alpha}} \equiv \underbrace{[\Gamma]B}_{B_1 \oplus B_2} \dashv \Delta$	By $\equiv$InstantiateL

- **Case 2.BEx.Bin:** $[\Gamma]A = A_1 \oplus A_2$ and $[\Gamma]B = \hat{\beta}$

  Symmetric to the **2.AEx.Bin** case, applying $\equiv$InstantiateR instead of $\equiv$InstantiateL. □

Theorem 9 (Completeness of Subtyping)**.**
If $\Gamma \longrightarrow \Omega$ *and* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$ *and* $\Gamma \vdash A$ *type and* $\Gamma \vdash B$ *type*
and $[\Omega]\Gamma \vdash [\Omega]A \leq^{\pm} [\Omega]B$
then there exist Δ *and* Ω' *such that* $\Delta \longrightarrow \Omega'$
and $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$
and $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash [\Gamma]A <:^{\pm} [\Gamma]B \dashv \Delta$.

Proof. By induction on the number of $\forall/\exists$ quantifiers in $[\Omega]A$ and $[\Omega]B$.

It is straightforward to show $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$; for examples of the necessary reasoning, see the proof of Theorem 11.

We have $[\Omega]\Gamma \vdash [\Omega]A \leq^{\pm} [\Omega]B$.

- **Case** $$\dfrac{[\Omega]\Gamma \vdash [\Omega]A \; type \qquad nonpos([\Omega]A)}{[\Omega]\Gamma \vdash [\Omega]A \leq^{-} \underbrace{[\Omega]A}_{[\Omega]B}}\ \leq\mathsf{Refl}-$$

  First, we observe that, since applying Ω as a substitution leaves quantifiers alone, the quantifiers that head A must also head B. For convenience, we alpha-vary B to quantify over the same variables as A.

  - If A is headed by $\forall$, then $[\Omega]A = (\forall\alpha : \kappa.\ [\Omega]A_0) = (\forall\alpha : \kappa.\ [\Omega]B_0) = [\Omega]B$.
    Let $\Gamma_0 = (\Gamma, \alpha : \kappa, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa)$.
    Let $\Omega_0 = (\Omega, \alpha : \kappa, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa = \alpha)$.

- * If $\mathrm{pol}(A_0) \in \{-, 0\}$, then:
  (We elide the straightforward use of lemmas about context extension.)

	$[\Omega_0]\Gamma_0 \vdash [\Omega]A_0 \leq^- [\Omega]A_0$	By $\leq$Refl$-$
	$[\Omega_0]\Gamma_0 \vdash [\Omega_0][\hat{\alpha}/\alpha]A_0 \leq^- A_0$	By def. of subst.
	$\Delta_0 \longrightarrow \Omega_0'$	By i.h. (fewer quantifiers)
	$\Omega_0 \longrightarrow \Omega_0'$	"
	$\Gamma_0 \vdash [\Gamma_0][\hat{\alpha}/\alpha]A_0 <:^- [\Gamma]B_0 \dashv \Delta_0$	"
	$\Gamma_0 \vdash [\hat{\alpha}/\alpha][\Gamma_0]A_0 <:^- [\Gamma]B_0 \dashv \Delta_0$	$\hat{\alpha}$ unsolved in Γ_0
	$\Gamma_0 \vdash [\hat{\alpha}/\alpha][\Gamma]A_0 <:^- [\Gamma]B_0 \dashv \Delta_0$	Γ_0 substitutes as Γ
	$\Gamma, \alpha : \kappa \vdash \forall\alpha : \kappa.\, [\Gamma]A_0 <:^- [\Gamma]B_0 \dashv \Delta, \alpha : \kappa, \Theta$	By $<:\forall$L
	$\Gamma \vdash \forall\alpha : \kappa.\, [\Gamma]A_0 <:^- \forall\alpha : \kappa.\, [\Gamma]B_0 \dashv \Delta$	By $<:\forall$R
☞	$\Gamma \vdash [\Gamma](\forall\alpha : \kappa.\, A_0) <:^- [\Gamma](\forall\alpha : \kappa.\, B_0) \dashv \Delta$	By def. of subst.
☞	$\Delta \longrightarrow \Omega$	By lemma
☞	$\Omega \longrightarrow \Omega_0'$	By lemma

  * If $\mathrm{pol}(A_0) = +$, then proceed as above, but apply $\leq$Refl$+$ instead of $\leq$Refl$-$, and apply $<:^+_-$L after applying the i.h. (Rule $<:^+_-$R also works.)
- – If A is not headed by $\forall$:
  We have *nonneg*$([\Omega]A)$. Therefore *nonneg*(A), and thus A is not headed by $\exists$. Since the same quantifiers must also head B, the conditions in rule $<:$Equiv are satisfied.

	$\Gamma \longrightarrow \Omega$	Given
	$\Gamma \vdash [\Gamma]A \equiv [\Gamma]B \dashv \Delta$	By Lemma 96 (Completeness of Equiv)
☞	$\Delta \longrightarrow \Omega'$	"
☞	$\Omega \longrightarrow \Omega'$	"
☞	$\Gamma \vdash [\Gamma]A <:^- [\Gamma]B \dashv \Delta$	By $<:$Equiv

- **Case** $\leq$Refl$+$: Symmetric to the $\leq$Refl$-$ case, using $<:^-_+$L (or $<:^-_+$R), and $<:\exists$R/$<:\exists$L instead of $<:\forall$L/$<:\forall$R.

- **Case** $$\dfrac{[\Omega]\Gamma \vdash \tau : \kappa \qquad [\Omega]\Gamma \vdash [\tau/\alpha][\Omega]A_0 \leq^- [\Omega]B}{[\Omega]\Gamma \vdash \underbrace{\forall\alpha : \kappa.\, [\Omega]A_0}_{[\Omega]A} \leq^- [\Omega]B}\ \leq\forall\mathsf{L}$$

  We begin by considering whether or not $[\Omega]B$ is headed by a universal quantifier.

  – $[\Omega]B = (\forall\beta : \kappa'.\, B')$:

  $[\Omega]\Gamma, \beta : \kappa' \vdash [\Omega]A \leq^- B'$ By Lemma 5 (Subtyping Inversion)

  The remaining steps are similar to the $\leq\forall$R case.

  – $[\Omega]B$ not headed by $\forall$:

$[\Omega]\Gamma \vdash \tau : \kappa$	Subderivation
$\Gamma \longrightarrow \Omega$	Given
$\Gamma, \blacktriangleright_{\hat{\alpha}} \longrightarrow \Omega, \blacktriangleright_{\hat{\alpha}}$	By $\longrightarrow$Marker
$\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa \longrightarrow \underbrace{\Omega, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa = \tau}_{\Omega_0}$	By $\longrightarrow$Solve
$[\Omega]\Gamma = [\Omega_0](\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} : \kappa)$	By definition of context application (lines 16, 13)

$$
\begin{array}{rll}
& [\Omega]\Gamma \vdash [\tau/\alpha][\Omega]A_0 \leq^- [\Omega]B & \text{Subderivation} \\
& [\Omega_0](\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa) \vdash [\tau/\alpha][\Omega]A_0 \leq^- [\Omega]B & \text{By above equality} \\
& [\Omega_0](\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa) \vdash \big[[\Omega_0]\hat{\alpha}/\alpha\big][\Omega]A_0 \leq^- [\Omega]B & \text{By definition of substitution} \\
& [\Omega_0](\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa) \vdash \big[[\Omega_0]\hat{\alpha}/\alpha\big][\Omega_0]A_0 \leq^- [\Omega_0]B & \text{By definition of substitution} \\
& [\Omega_0](\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa) \vdash [\Omega_0][\hat{\alpha}/\alpha]A_0 \leq^- [\Omega_0]B & \text{By distributivity of substitution}
\end{array}
$$

$$
\begin{array}{ll}
\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha} \vdash [\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa][\hat{\alpha}/\alpha]A_0 <:^- [\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa]B \dashv \Delta_0 & \text{By i.h. (A lost a quantifier)} \\
\quad \Delta_0 \longrightarrow \Omega'' & '' \\
\quad \Omega_0 \longrightarrow \Omega'' & ''
\end{array}
$$

$$
\begin{array}{ll}
\Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa \vdash [\Gamma][\hat{\alpha}/\alpha]A_0 <:^- [\Gamma]B \dashv \Delta_0 & \text{By definition of substitution}
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa \longrightarrow \Delta_0 & \text{By Lemma 50 (Subtyping Extension)} \\
& \Delta_0 = (\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta) & \text{By Lemma 22 (Extension Inversion) (ii)} \\
& \Gamma \longrightarrow \Delta & '' \\
& \Omega'' = (\Omega', \blacktriangleright_{\hat{\alpha}}, \Omega_Z) & \text{By Lemma 22 (Extension Inversion) (ii)} \\
☞ & \Delta \longrightarrow \Omega' & '' \\
& \Omega_0 \longrightarrow \Omega'' & \text{Above} \\
& \Omega, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa{=}\tau \longrightarrow \Omega', \blacktriangleright_{\hat{\alpha}}, \Omega_Z & \text{By above equalities} \\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 22 (Extension Inversion) (ii)}
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa \vdash [\Gamma][\hat{\alpha}/\alpha]A_0 <:^- [\Gamma]B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta & \text{By above equality } \Delta_0 = (\Delta, \blacktriangleright_{\hat{\alpha}}, \Theta) \\
& \Gamma, \blacktriangleright_{\hat{\alpha}}, \hat{\alpha}:\kappa \vdash [\hat{\alpha}/\alpha][\Gamma]A_0 <:^- [\Gamma]B \dashv \Delta, \blacktriangleright_{\hat{\alpha}}, \Theta & \text{By def. of subst. ($[\Gamma]\hat{\alpha} = \hat{\alpha}$ and $[\Gamma]\alpha = \alpha$)} \\
& [\Gamma]B \text{ not headed by } \forall & \text{From the case assumption} \\
& \Gamma \vdash \forall\alpha:\kappa.\, [\Gamma]A_0 <:^- [\Gamma]B \dashv \Delta & \text{By } {<:}\forall\mathsf{L} \\
☞ & \Gamma \vdash [\Gamma](\forall\alpha:\kappa.\, A_0) <:^- [\Gamma]B \dashv \Delta & \text{By definition of substitution}
\end{array}
$$

- **Case** $\dfrac{[\Omega]\Gamma, \beta:\kappa \vdash [\Omega]A \leq^- [\Omega]B_0}{[\Omega]\Gamma \vdash [\Omega]A \leq^- \underbrace{\forall\beta:\kappa.\, [\Omega]B_0}_{[\Omega]B}} \leq\forall\mathsf{R}$

$$
\begin{array}{rl}
B = \forall\beta:\kappa.\, B_0 & \Omega \text{ predicative} \\
[\Omega]\Gamma \vdash [\Omega]A \leq^- [\Omega]B & \text{Given} \\
[\Omega]\Gamma \vdash [\Omega]A \leq^- \forall\beta.\, [\Omega]B_0 & \text{By above equality} \\
[\Omega]\Gamma, \beta:\kappa \vdash [\Omega]A \leq^- [\Omega]B_0 & \text{Subderivation} \\
[\Omega, \beta:\kappa](\Gamma, \beta:\kappa) \vdash [\Omega, \beta:\kappa]A \leq^- [\Omega, \beta:\kappa]B_0 & \text{By definitions of substitution} \\
\Gamma, \beta:\kappa \vdash [\Gamma, \beta{:}\kappa]A <:^- [\Gamma, \beta{:}\kappa]B_0 \dashv \Delta' & \text{By i.h. (B lost a quantifier)} \\
\Delta' \longrightarrow \Omega'_0 & '' \\
\Omega, \beta:\kappa \longrightarrow \Omega'_0 & '' \\
\Gamma, \beta:\kappa \vdash [\Gamma]A <:^- [\Gamma]B_0 \dashv \Delta' & \text{By definition of substitution}
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \beta:\kappa \longrightarrow \Delta' & \text{By Lemma 43 (Instantiation Extension)} \\
& \Delta' = (\Delta, \beta:\kappa, \Theta) & \text{By Lemma 22 (Extension Inversion) (i)} \\
& \Gamma \longrightarrow \Delta & '' \\
& \Delta, \beta:\kappa, \Theta \longrightarrow \Omega'_0 & \text{By } \Delta' \longrightarrow \Omega'_0 \text{ and above equality} \\
& \Omega'_0 = (\Omega', \beta:\kappa, \Omega_R) & \text{By Lemma 22 (Extension Inversion) (i)} \\
☞ & \Delta \longrightarrow \Omega' & ''
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \beta : \kappa \vdash [\Gamma]A \mathrel{<:^-} [\Gamma]B_0 \dashv \Delta, \beta : \kappa, \Theta & \text{By above equality} \\
& \Omega, \beta : \kappa \longrightarrow \Omega', \beta : \kappa, \Omega_R & \text{By above equality} \\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 33 (Extension Transitivity)} \\
& \Gamma \vdash [\Gamma]A \mathrel{<:^-} \forall\beta : \kappa.\, [\Gamma]B_0 \dashv \Delta & \text{By } {<:}\forall\mathsf{R} \\
☞ & \Gamma \vdash [\Gamma]A \mathrel{<:^-} [\Gamma](\forall\beta : \kappa.\, B_0) \dashv \Delta & \text{By definition of substitution}
\end{array}
$$

- **Case** $\dfrac{[\Omega]\Gamma, \alpha : \kappa \vdash [\Omega]A_0 \leq^+ [\Omega]B}{[\Omega]\Gamma \vdash \underbrace{\exists\alpha : \kappa.\, [\Omega]A_0}_{[\Omega]A} \leq^+ [\Omega]B}\ {\leq}\exists\mathsf{L}$

$$
\begin{array}{rl}
A = \exists\alpha : \kappa.\, A_0 & \Omega \text{ predicative} \\
[\Omega]\Gamma \vdash [\Omega]A) \leq^+ [\Omega]B & \text{Given} \\
[\Omega]\Gamma \vdash [\Omega]\exists\alpha : \kappa.\, A_0 \leq^+ [\Omega]B & \text{By above equality} \\
[\Omega]\Gamma, \alpha : \kappa \vdash [\Omega]A_0 \leq^+ [\Omega]B & \text{Subderivation} \\
[\Omega, \alpha : \kappa](\Gamma, \alpha : \kappa) \vdash [\Omega, \alpha : \kappa]A_0 \leq^+ [\Omega, \alpha : \kappa]B & \text{By definitions of substitution} \\
\Gamma, \alpha : \kappa \vdash [\Gamma, \beta{:}\kappa]A_0 \mathrel{<:^+} [\Gamma, \beta{:}\kappa]B \dashv \Delta' & \text{By i.h. (A lost a quantifier)} \\
\Delta' \longrightarrow \Omega'_0 & '' \\
\Omega, \alpha : \kappa \longrightarrow \Omega'_0 & '' \\
\Gamma, \alpha : \kappa \vdash [\Gamma]A \mathrel{<:^+} [\Gamma]B_0 \dashv \Delta' & \text{By definition of substitution}
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \alpha : \kappa \longrightarrow \Delta' & \text{By Lemma 43 (Instantiation Extension)} \\
& \Delta' = (\Delta, \alpha : \kappa, \Theta) & \text{By Lemma 22 (Extension Inversion) (i)} \\
& \Gamma \longrightarrow \Delta & '' \\
& \Delta, \alpha : \kappa, \Theta \longrightarrow \Omega'_0 & \text{By } \Delta' \longrightarrow \Omega'_0 \text{ and above equality} \\
& \Omega'_0 = (\Omega', \alpha : \kappa, \Omega_R) & \text{By Lemma 22 (Extension Inversion) (i)} \\
☞ & \Delta \longrightarrow \Omega' & ''
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \alpha : \kappa \vdash [\Gamma]A_0 \mathrel{<:^+} [\Gamma]B \dashv \Delta, \alpha : \kappa, \Theta & \text{By above equality} \\
& \Omega, \alpha : \kappa \longrightarrow \Omega', \alpha : \kappa, \Omega_R & \text{By above equality} \\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 33 (Extension Transitivity)} \\
& \Gamma \vdash \exists\alpha : \kappa.\, [\Gamma]A_0 \mathrel{<:^+} [\Gamma]B \dashv \Delta & \text{By } {<:}\forall\mathsf{R} \\
☞ & \Gamma \vdash [\Gamma](\exists\alpha : \kappa.\, A_0) \mathrel{<:^+} [\Gamma]B \dashv \Delta & \text{By definition of substitution}
\end{array}
$$

- **Case** $\dfrac{\Psi \vdash \tau : \kappa \qquad \Psi \vdash [\Omega]A \leq^+ [\tau/\beta]B_0}{\Psi \vdash [\Omega]A \leq^+ \underbrace{\exists\beta : \kappa.\, B_0}_{[\Omega]B}}\ {\leq}\exists\mathsf{R}$

  We consider whether $[\Omega]A$ is headed by an existential.

  If $[\Omega]A = \exists\alpha : \kappa'.\, A'$:

  $[\Omega]\Gamma, \alpha : \kappa' \vdash A' \leq^+ [\Omega]B$ By Lemma 5 (Subtyping Inversion)

  The remaining steps are similar to the $\leq\exists\mathsf{L}$ case.

  If $[\Omega]A$ not headed by $\exists$:

$$
\begin{array}{rl}
[\Omega]\Gamma \vdash \tau : \kappa & \text{Subderivation} \\
\Gamma \longrightarrow \Omega & \text{Given} \\
\Gamma, \blacktriangleright_{\hat\alpha} \longrightarrow \Omega, \blacktriangleright_{\hat\alpha} & \text{By } {\longrightarrow}\mathsf{Marker} \\
\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa \longrightarrow \underbrace{\Omega, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa = \tau}_{\Omega_0} & \text{By } {\longrightarrow}\mathsf{Solve} \\
[\Omega]\Gamma = [\Omega_0](\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \kappa) & \text{By definition of context application (lines 16, 13)}
\end{array}
$$

$$
\begin{array}{rl}
[\Omega]\Gamma \vdash [\Omega]A \leq^+ [\tau/\beta][\Omega]B_0 & \text{Subderivation}\\
[\Omega_0](\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa) \vdash [\Omega]A \leq^+ [\tau/\beta][\Omega]B_0 & \text{By above equality}\\
[\Omega_0](\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa) \vdash [\Omega]A \leq^+ \big[[\Omega_0]\hat\alpha/\beta\big][\Omega]B_0 & \text{By definition of substitution}\\
[\Omega_0](\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa) \vdash [\Omega_0]A \leq^+ \big[[\Omega_0]\hat\alpha/\beta\big][\Omega_0]B_0 & \text{By definition of substitution}\\
[\Omega_0](\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa) \vdash [\Omega_0]A \leq^+ [\Omega_0][\hat\alpha/\beta]B_0 & \text{By distributivity of substitution}
\end{array}
$$

$$
\begin{array}{ll}
\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha \vdash [\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa]A <:^+ [\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa][\hat\alpha/\beta]B_0 \dashv \Delta_0 & \text{By i.h. (B lost a quantifier)}\\
\quad \Delta_0 \longrightarrow \Omega'' & ''\\
\quad \Omega_0 \longrightarrow \Omega'' & ''
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa \vdash [\Gamma][\hat\alpha/\beta]B_0 <:^+ [\Gamma]B \dashv \Delta_0 & \text{By definition of substitution}\\
& \Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa \longrightarrow \Delta_0 & \text{By Lemma 50 (Subtyping Extension)}\\
& \Delta_0 = (\Delta, \blacktriangleright_{\hat\alpha}, \Theta) & \text{By Lemma 22 (Extension Inversion) (ii)}\\
& \Gamma \longrightarrow \Delta & ''\\
& \Omega'' = (\Omega', \blacktriangleright_{\hat\alpha}, \Omega_Z) & \text{By Lemma 22 (Extension Inversion) (ii)}\\
☞ & \Delta \longrightarrow \Omega' & ''\\
& \Omega_0 \longrightarrow \Omega'' & \text{Above}\\
& \Omega, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa = \tau \longrightarrow \Omega', \blacktriangleright_{\hat\alpha}, \Omega_Z & \text{By above equalities}\\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 22 (Extension Inversion) (ii)}
\end{array}
$$

$$
\begin{array}{lrl}
& \Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa \vdash [\Gamma]A <:^+ [\Gamma][\hat\alpha/\beta]B_0 \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Theta & \text{By above equality } \Delta_0 = (\Delta, \blacktriangleright_{\hat\alpha}, \Theta)\\
& \Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha:\kappa \vdash [\Gamma]A <:^+ [\hat\alpha/\beta][\Gamma]B_0 \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Theta & \text{By def. of subst. } ([\Gamma]\hat\alpha = \hat\alpha \text{ and } [\Gamma]\beta = \beta)\\
& [\Gamma]A \text{ not headed by } \exists & \text{From the case hypothesis}\\
& \Gamma \vdash [\Gamma]A <:^+ \exists\beta:\kappa.\, [\Gamma]B_0 \dashv \Delta & \text{By } {<:}\exists\mathsf{R}\\
☞ & \Gamma \vdash [\Gamma]A <:^+ [\Gamma](\exists\beta:\kappa.\, B_0) \dashv \Delta & \text{By definition of substitution}
\end{array}
$$

□

K′.3 Completeness of Typing

Theorem 10 (Completeness of Match Coverage).

1. *If* $[\Omega]\Gamma \vdash [\Omega]\Pi$ *covers* $[\Omega]\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *then* $\Gamma \vdash \Pi$ *covers* $\vec{A}$.

2. *If* $[\Omega]\Gamma \;/\; [\Omega]P \vdash [\Omega]\Pi$ *covers* $[\Omega]\vec{A}$ *and* $\Gamma \longrightarrow \Omega$ *and* $\Gamma \vdash \vec{A}$ **!** *types and* $[\Gamma]\vec{A} = \vec{A}$ *and* $[\Gamma]P = P$ *then* $\Gamma \;/\; P \vdash \Pi$ *covers* $\vec{A}$.

Proof. By mutual induction on the derivation of the given coverage rule.

1.
   - **Case**
     $$\frac{}{[\Omega]\Gamma \vdash \cdot \Rightarrow e_1 \mid \ldots \textit{ covers } \cdot}\ \mathsf{DeclCoversEmpty}$$
     Apply CoversEmpty.
   - **Cases** DeclCoversVar, DeclCovers1, DeclCovers$\times$, DeclCovers$+$, DeclCovers$\exists$, DeclCovers$\wedge$, DeclCoversVec:
     Use the i.h. and apply the corresponding algorithmic coverage rule.

2.
   - **Case**
     $$\frac{\theta = \mathsf{mgu}(t_1, t_2) \qquad [\theta][\Omega]\Gamma \vdash [\theta][\Omega]\Pi \textit{ covers } [\theta]\vec{A}}{[\Omega]\Gamma \;/\; [\Omega](t_1 = t_2) \vdash [\Omega]\Pi \textit{ covers } \vec{A}}\ \mathsf{DeclCoversEq}$$

     $$
     \begin{array}{rl}
     \mathsf{mgu}(t_1, t_2) = \theta & \text{Premise}\\
     \Gamma \;/\; t_1 \doteq t_2 : \kappa \dashv \Gamma, \Theta & \text{By Lemma 92 (Completeness of Elimeq) (1)}\\
     \Gamma \;/\; [\Gamma]t_1 \doteq [\Gamma]t_2 : \kappa \dashv \Gamma, \Theta & \text{Follows from given assumption}
     \end{array}
     $$

$[\theta][\Omega]\Gamma \vdash [\theta][\Omega]\Pi \; covers \; [\theta]A_0, [\theta]\vec{A}$	Subderivation
$[\theta][\Omega]\Gamma = [\Omega,\Theta](\Gamma,\Theta)$	By Lemma 93 (Substitution Upgrade) (iii)
$[\theta][\Omega]\Pi = [\Omega,\Theta]\Pi$	By Lemma 93 (Substitution Upgrade) (iv)
$([\theta]A_0, [\theta]\vec{A}) = ([\Gamma,\Theta]\vec{A})$	By Lemma 93 (Substitution Upgrade) (i)
$[\Omega,\Theta](\Gamma,\Theta) \vdash [\Omega,\Theta]\Pi \; covers \; [\Gamma,\Theta]\vec{A}$	By above equalities
$\Gamma,\Theta \vdash [\Gamma,\Theta]\Pi \; covers \; [\Gamma,\Theta]\vec{A}$	By i.h.
☞ $\Gamma \;/\; t_1 = t_2 \vdash \Pi \; covers \; \vec{A}$	By CoversEq

- **Case** $$\dfrac{mgu(t_1, t_2) = \bot}{[\Omega]\Gamma \;/\; [\Omega](t_1 = t_2) \vdash [\Omega]\Pi \; covers \; \vec{A}} \; \mathsf{DeclCoversEqBot}$$

$mgu(t_1, t_2) = \bot$	Premise
$\Gamma \;/\; t_1 \triangleq t_2 : \kappa \dashv \bot$	By Lemma 92 (Completeness of Elimeq) (2)
$\Gamma \;/\; [\Gamma]t_1 \triangleq [\Gamma]t_2 : \kappa \dashv \bot$	Follows from given assumption
☞ $\Gamma \;/\; t_1 = t_2 \vdash \Pi \; covers \; \vec{A}$	By CoversEqBot

□

Theorem 11 (Completeness of Algorithmic Typing)**.** *Given* $\Gamma \longrightarrow \Omega$ *such that* $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$*:*

(i) *If* $\Gamma \vdash A\ p\ type$ *and* $[\Omega]\Gamma \vdash [\Omega]e \Leftarrow [\Omega]A\ p$ *and* $p' \sqsubseteq p$
then there exist Δ *and* Ω'
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash e \Leftarrow [\Gamma]A\ p' \dashv \Delta$.

(ii) *If* $\Gamma \vdash A\ p\ type$ *and* $[\Omega]\Gamma \vdash [\Omega]e \Rightarrow A\ p$
then there exist Δ, Ω', A', *and* $p' \sqsubseteq p$
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash e \Rightarrow A'\ p' \dashv \Delta$ *and* $A' = [\Delta]A'$ *and* $A = [\Omega']A'$.

(iii) *If* $\Gamma \vdash A\ p\ type$ *and* $[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg B\ q$ *and* $p' \sqsubseteq p$
then there exist Δ, Ω', B' *and* $q' \sqsubseteq q$
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash s : [\Gamma]A\ p' \gg B'\ q' \dashv \Delta$ *and* $B' = [\Delta]B'$ *and* $B = [\Omega']B'$.

(iv) *If* $\Gamma \vdash A\ p\ type$ *and* $[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg B\ \lceil q \rceil$ *and* $p' \sqsubseteq p$
then there exist Δ, Ω', B', *and* $q' \sqsubseteq q$
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash s : [\Gamma]A\ p' \gg B'\ \lceil q' \rceil \dashv \Delta$ *and* $B' = [\Delta]B'$ *and* $B = [\Omega']B'$.

(v) *If* $\Gamma \vdash \vec{A}\ \mathbf{!}\ types$ *and* $\Gamma \vdash C\ p\ type$ *and* $[\Omega]\Gamma \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$ *and* $p' \sqsubseteq p$
then there exist Δ, Ω', *and* C
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \vdash \Pi :: [\Gamma]\vec{A} \Leftarrow [\Gamma]C\ p' \dashv \Delta$.

(vi) *If* $\Gamma \vdash \vec{A}\ \mathbf{!}\ types$ *and* $\Gamma \vdash P\ prop$ *and* $\mathsf{FEV}(P) = \emptyset$ *and* $\Gamma \vdash C\ p\ type$
and $[\Omega]\Gamma \;/\; [\Omega]P \vdash [\Omega]\Pi :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$
and $p' \sqsubseteq p$
then there exist Δ, Ω', *and* C
such that $\Delta \longrightarrow \Omega'$ *and* $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ *and* $\Omega \longrightarrow \Omega'$
and $\Gamma \;/\; [\Gamma]P \vdash \Pi :: [\Gamma]\vec{A} \Leftarrow [\Gamma]C\ p' \dashv \Delta$.

Proof. By induction, using the measure in Definition 7.

- **Case** $\dfrac{(x : A\ p) \in [\Omega]\Gamma}{[\Omega]\Gamma \vdash x \Rightarrow A\ p}$ DeclVar

	$(x : A\ p) \in [\Omega]\Gamma$	Premise
	$\Gamma \longrightarrow \Omega$	Given
	$(x : A'\ p) \in \Gamma$ where $[\Omega]A' = A$	From definition of context application
	Let $\Delta = \Gamma$.	
	Let $\Omega' = \Omega$.	
☞	$\Gamma \longrightarrow \Omega$	Given
☞	$\Omega \longrightarrow \Omega$	By Lemma 32 (Extension Reflexivity)
☞	$\Gamma \vdash x \Rightarrow [\Gamma]A'\ p \dashv \Gamma$	By Var
☞	$[\Gamma]A' = [\Gamma][\Gamma]A'$	By idempotence of substitution
☞	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
	$\Gamma \longrightarrow \Omega$	Given
	$[\Omega][\Gamma]A' = [\Omega]A'$	By Lemma 29 (Substitution Monotonicity) (iii)
☞	$= A$	By above equality

- **Case** $\dfrac{[\Omega]\Gamma \vdash [\Omega]e \Rightarrow B\ q \qquad [\Omega]\Gamma \vdash B \leq^{\pm} [\Omega]A}{[\Omega]\Gamma \vdash [\Omega]e \Leftarrow [\Omega]A\ p}$ DeclSub

	$[\Omega]\Gamma \vdash [\Omega]e \Rightarrow B\ q$	Subderivation
	$\Gamma \vdash e \Rightarrow B'\ q \dashv \Theta$	By i.h.
	$B = [\Omega]B'$	"
	$\Theta \longrightarrow \Omega_0$	"
	$\Omega \longrightarrow \Omega_0$	"
	$\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_0)$	"
	$\Gamma \longrightarrow \Omega$	Given
	$\Gamma \longrightarrow \Omega_0$	By Lemma 33 (Extension Transitivity)
	$[\Omega]\Gamma \vdash B \leq^{\pm} [\Omega]A$	Subderivation
	$[\Omega]\Gamma = [\Omega]\Theta$	By Lemma 56 (Confluence of Completeness)
	$[\Omega]\Theta \vdash B \leq^{\pm} [\Omega]A$	By above equalities
	$\Theta \longrightarrow \Omega_0$	Above
	$\Theta \vdash B' <:^{\pm} A \dashv \Delta$	By Theorem 9
	$\Omega_0 \longrightarrow \Omega'$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
☞	$\Delta \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \vdash e \Leftarrow A\ p \dashv \Delta$	By Sub

- **Case** $\dfrac{[\Omega]\Gamma \vdash [\Omega]A\ \mathit{type} \qquad [\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega]A\ !}{[\Omega]\Gamma \vdash [\Omega](e_0 : A) \Rightarrow A\ !}$ DeclAnno

	$[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega]A\ !$	Subderivation
	$[\Omega]A = [\Omega][\Gamma]A$	By Lemma 29 (Substitution Monotonicity)
	$[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega][\Gamma]A\ !$	By above equality
	$\Gamma \vdash e_0 \Leftarrow [\Gamma]A\ ! \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega$	//
☞	$\Omega \longrightarrow \Omega'$	//
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	//
	$\Delta \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
	$\Gamma \vdash A\ !\ \mathit{type}$	Given
☞	$\Gamma \vdash (e_0 : A) \Rightarrow [\Delta]A\ ! \dashv \Delta$	By Anno
☞	$[\Delta]A = [\Delta][\Delta]A$	By idempotence of substitution
	$A = [\Omega]A$	Above
	$= [\Omega']A$	By Lemma 55 (Completing Completeness) (ii)
☞	$= [\Omega'][\Delta]A$	By Lemma 29 (Substitution Monotonicity)

- **Case**
$$\frac{}{[\Omega]\Gamma \vdash () \Leftarrow 1\ p}\ \mathsf{Decl1I}$$

We have $[\Omega]A = 1$. Either $[\Gamma]A = 1$, or $[\Gamma]A = \hat{\alpha}$ where $\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$.

In the former case:

	Let $\Delta = \Gamma$.	
	Let $\Omega' = \Omega$.	
☞	$\Gamma \longrightarrow \Omega$	Given
☞	$\Omega \longrightarrow \Omega'$	By Lemma 32 (Extension Reflexivity)
☞	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
	$\Gamma \vdash () \Leftarrow 1\ p \dashv \Gamma$	By 1I
☞	$\Gamma \vdash () \Leftarrow [\Gamma]1\ p \dashv \Gamma$	$1 = [\Gamma]1$

In the latter case, since $A = \hat{\alpha}$ and $\Gamma \vdash \hat{\alpha}\ p\ \mathit{type}$ is given, it must be the case that $p = \not{!}$.

	$\Gamma_0[\hat{\alpha} : \star] \vdash () \Leftarrow \hat{\alpha}\ \not{!} \dashv \Gamma_0[\hat{\alpha} : \star = 1]$	By 1I$\hat{\alpha}$
☞	$\Gamma_0[\hat{\alpha} : \star] \vdash () \Leftarrow \big[\Gamma_0[\hat{\alpha} : \star]\big]\hat{\alpha}\ \not{!} \dashv \Gamma_0[\hat{\alpha} : \star = 1]$	By def. of subst.
	$\Gamma_0[\hat{\alpha} : \star] \longrightarrow \Omega$	Given
☞	$\Gamma_0[\hat{\alpha} : \star = 1] \longrightarrow \Omega$	By Lemma 27 (Parallel Extension Solution)
☞	$\Omega \longrightarrow \Omega$	By Lemma 32 (Extension Reflexivity)

- **Case**
$$\frac{v\ \mathit{chk\text{-}I} \qquad [\Omega]\Gamma, \alpha : \kappa \vdash [\Omega]v \Leftarrow A_0\ p}{[\Omega]\Gamma \vdash [\Omega]v \Leftarrow \forall \alpha : \kappa.\ A_0\ p}\ \mathsf{Decl}\forall\mathsf{I}$$

$[\Omega]A = \forall \alpha : \kappa.\ A_0$	Given
$= \forall \alpha : \kappa.\ [\Omega]A'$	By def. of subst. and predicativity of Ω
$A_0 = [\Omega]A'$	Follows from above equality
$[\Omega]\Gamma, \alpha : \kappa \vdash [\Omega]v \Leftarrow [\Omega]A'\ p$	Subderivation and above equality
$\Gamma \longrightarrow \Omega$	Given
$\Gamma, \alpha : \kappa \longrightarrow \Omega, \alpha : \kappa$	By $\longrightarrow$Uvar
$[\Omega]\Gamma, \alpha : \kappa = [\Omega, \alpha : \kappa](\Gamma, \alpha : \kappa)$	By definition of context substitution
$[\Omega, \alpha : \kappa](\Gamma, \alpha : \kappa) \vdash [\Omega]v \Leftarrow [\Omega]A'\ p$	By above equality
$[\Omega, \alpha : \kappa](\Gamma, \alpha : \kappa) \vdash [\Omega]v \Leftarrow [\Omega, \alpha : \kappa]A'\ p$	By definition of substitution

$$
\begin{array}{lrl}
 & \Gamma, \alpha : \kappa \vdash v \Leftarrow [\Gamma, \alpha : \kappa]A' \; p \dashv \Delta' & \text{By i.h.} \\
 & \Delta' \longrightarrow \Omega_0' & '' \\
 & \Omega, \alpha : \kappa \longrightarrow \Omega_0' & '' \\
 & \mathsf{dom}(\Delta') = \mathsf{dom}(\Omega_0') & '' \\
 & \Gamma, \alpha : \kappa \longrightarrow \Delta' & \text{By Lemma 51 (Typing Extension)} \\
 & \Delta' = (\Delta, \alpha : \kappa, \Theta) & \text{By Lemma 22 (Extension Inversion) (i)} \\
 & \Delta, \alpha : \kappa, \Theta \longrightarrow \Omega_0' & \text{By above equality} \\
 & \Omega_0' = (\Omega', \alpha : \kappa, \Omega_Z) & \text{By Lemma 22 (Extension Inversion) (i)} \\
☞ & \Delta \longrightarrow \Omega' & '' \\
☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & '' \\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 22 (Extension Inversion) on } \Omega, \alpha : \kappa \longrightarrow \Omega_0'
\end{array}
$$

$$
\begin{array}{lrl}
 & \Gamma, \alpha : \kappa \vdash v \Leftarrow [\Gamma, \alpha : \kappa]A' \; p \dashv \Delta, \alpha : \kappa, \Theta & \text{By above equality} \\
 & \Gamma, \alpha : \kappa \vdash v \Leftarrow [\Gamma]A' \; p \dashv \Delta, \alpha : \kappa, \Theta & \text{By definition of substitution} \\
 & \Gamma \vdash v \Leftarrow \forall \alpha : \kappa. \, [\Gamma]A' \; p \dashv \Delta & \text{By } \forall\mathsf{I} \\
☞ & \Gamma \vdash v \Leftarrow [\Gamma](\forall \alpha : \kappa. \, A') \; p \dashv \Delta & \text{By definition of substitution}
\end{array}
$$

- **Case** $\dfrac{[\Omega]\Gamma \vdash \tau : \kappa \qquad [\Omega]\Gamma \vdash [\Omega](e\ s_0) : [\tau/\alpha][\Omega]A_0 \not{p} \gg B\ q}{[\Omega]\Gamma \vdash [\Omega](e\ s_0) : \forall \alpha : \kappa. \, [\Omega]A_0 \; p \gg B\ q}\ \mathsf{Decl}\forall\mathsf{Spine}$

$$
\begin{array}{rl}
[\Omega]\Gamma \vdash \tau : \kappa & \text{Subderivation}
\end{array}
$$

$$
\begin{array}{rl}
\Gamma \longrightarrow \Omega & \text{Given} \\
\Gamma, \hat{\alpha} : \kappa \longrightarrow \Omega, \hat{\alpha} : \kappa = \tau & \text{By } \longrightarrow\mathsf{Solve}
\end{array}
$$

$$
\begin{array}{rcll}
\multicolumn{3}{l}{[\Omega]\Gamma \vdash [\Omega](e\ s_0) : [\tau/\alpha][\Omega]A_0 \not{p} \gg B\ q} & \text{Subderivation} \\
\tau & = & [\Omega]\tau & \mathsf{FEV}(\tau) = \emptyset \\
[\tau/\alpha][\Omega]A_0 & = & [\tau/\alpha][\Omega, \hat{\alpha} : \kappa = \tau]A_0 & \text{By def. of subst.} \\
 & = & \big[[\Omega]\tau/\alpha\big]\big[\Omega, \hat{\alpha} : \kappa = \tau\big]A_0 & \text{By above equality} \\
 & = & \big[\Omega, \hat{\alpha} : \kappa = \tau\big][\hat{\alpha}/\alpha]A_0 & \text{By distributivity of substitution} \\
[\Omega]\Gamma & = & [\Omega, \hat{\alpha} : \kappa = \tau](\Gamma, \hat{\alpha} : \kappa) & \text{By definition of context application}
\end{array}
$$

$$
\begin{array}{lrl}
 & [\Omega, \hat{\alpha} : \kappa = \tau](\Gamma, \hat{\alpha} : \kappa) \vdash [\Omega](e\ s_0) : \big[\Omega, \hat{\alpha} : \kappa = \tau\big][\hat{\alpha}/\alpha]A_0 \not{p} \gg B\ q & \text{By above equalities} \\
 & \Gamma, \hat{\alpha} : \kappa \vdash e\ s_0 : [\Gamma, \hat{\alpha} : \kappa][\hat{\alpha}/\alpha]A_0 \not{p} \gg B'\ q \dashv \Delta & \text{By i.h.} \\
 & B = [\Omega, \hat{\alpha} : \kappa = \tau]B' & '' \\
☞ & \Delta \longrightarrow \Omega' & '' \\
☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & '' \\
☞ & \Omega \longrightarrow \Omega' & '' \\
☞ & B' \longrightarrow [\Delta]B' & '' \\
☞ & B \longrightarrow [\Omega']B' & ''
\end{array}
$$

$$
\begin{array}{lrcll}
 & [\Gamma, \hat{\alpha} : \kappa][\hat{\alpha}/\alpha]A_0 & = & [\Gamma][\hat{\alpha}/\alpha]A_0 & \text{By def. of context application} \\
 & & = & [\hat{\alpha}/\alpha][\Gamma]A_0 & \Gamma \text{ does not subst. for } \alpha \\
 & \multicolumn{3}{r}{\Gamma, \hat{\alpha} : \kappa \vdash e\ s_0 : [\hat{\alpha}/\alpha][\Gamma]A_0 \not{p} \gg B'\ q \dashv \Delta} & \text{By above equality} \\
 & \multicolumn{3}{r}{\Gamma \vdash e\ s_0 : \forall \alpha : \kappa. \, [\Gamma]A_0 \; p \gg B'\ q \dashv \Delta} & \text{By } \forall\mathsf{Spine} \\
☞ & \multicolumn{3}{r}{\Gamma \vdash e\ s_0 : [\Gamma](\forall \alpha : \kappa. \, A_0) \; p \gg B'\ q \dashv \Delta} & \text{By def. of subst.}
\end{array}
$$

- **Case** $\dfrac{v \ \mathit{chk\text{-}I} \qquad [\Omega]\Gamma \ / \ [\Omega]P \vdash [\Omega]v \Leftarrow [\Omega]A_0 \ !}{[\Omega]\Gamma \vdash [\Omega]v \Leftarrow ([\Omega]P) \supset [\Omega]A_0 \ !}\ \mathsf{Decl}{\supset}\mathsf{I}$

$$
\begin{array}{rl}
[\Omega]\Gamma \ / \ [\Omega]P \vdash [\Omega]v \Leftarrow [\Omega]A_0 \ ! & \text{Subderivation}
\end{array}
$$

The concluding rule in this subderivation must be DeclCheck⊥ or DeclCheckUnify. In either case, $[\Omega]P$ has the form $(\sigma' = \tau')$ where $\sigma' = [\Omega]\sigma$ and $\tau' = [\Omega]\tau$.

- **Case** $$\dfrac{\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \bot}{[\Omega]\Gamma \;/\; [\Omega](\sigma = \tau) \vdash [\Omega]v \Leftarrow [\Omega]A_0\;!}\;\mathsf{DeclCheck}\bot$$

  We have $\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \bot$. To apply Lemma 92 (Completeness of Elimeq) (2), we need to show conditions 1–5.

***	$\Gamma \vdash (\sigma = \tau) \supset A_0\;!\;\mathit{type}$	Given
	$[\Omega]\big((\sigma{=}\tau) \supset A_0\big) = [\Gamma]\big((\sigma{=}\tau) \supset A_0\big)$	By Lemma 39 (Principal Agreement) (i)
	$[\Omega]\sigma = [\Gamma]\sigma$	By a property of subst.
	$[\Omega]\tau = [\Gamma]\tau$	Similar
	$\Gamma \vdash \sigma : \kappa$	By inversion
3	$\Gamma \vdash [\Gamma]\sigma : \kappa$	By Lemma 11 (Right-Hand Substitution for Sorting)
4	$\Gamma \vdash [\Gamma]\tau : \kappa$	Similar
	$\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \bot$	Given
	$\mathsf{mgu}([\Gamma]\sigma, [\Gamma]\tau) = \bot$	By above equalities
	$\mathsf{FEV}(\sigma) \cup \mathsf{FEV}(\tau) = \emptyset$	By inversion on ***
	$\mathsf{FEV}([\Omega]\sigma) \cup \mathsf{FEV}([\Omega]\tau) = \emptyset$	By a property of complete contexts
5	$\mathsf{FEV}([\Gamma]\sigma) \cup \mathsf{FEV}([\Gamma]\tau) = \emptyset$	By above equalities
1	$[\Gamma][\Gamma]\sigma = [\Gamma]\sigma$	By idempotence of subst.
2	$[\Gamma][\Gamma]\tau = [\Gamma]\tau$	By idempotence of subst.
	$\Gamma \;/\; [\Gamma]\sigma \doteq [\Gamma]\tau : \kappa \dashv \bot$	By Lemma 92 (Completeness of Elimeq) (2)
	$\Gamma, \blacktriangleright_P \;/\; [\Gamma]\sigma = [\Gamma]\tau \dashv \bot$	By ElimpropEq
	$\Gamma \vdash v \Leftarrow ([\Gamma]\sigma = [\Gamma]\tau) \supset [\Gamma]A_0\;! \dashv \Gamma$	By $\supset$I⊥
☞	$\Gamma \vdash v \Leftarrow [\Gamma]\big((\sigma = \tau) \supset A_0\big)\;! \dashv \Gamma$	By def. of subst.
☞	$\Gamma \longrightarrow \Omega$	Given
☞	$\Omega \longrightarrow \Omega$	By Lemma 32 (Extension Reflexivity)
☞	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given

- **Case** $$\dfrac{\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \theta \qquad \theta([\Omega]\Gamma) \vdash \theta([\Omega]e) \Leftarrow \theta([\Omega]A_0)\;!}{[\Omega]\Gamma \;/\; (([\Omega]\sigma) = [\Omega]\tau) \vdash [\Omega]e \Leftarrow [\Omega]A_0\;!}\;\mathsf{DeclCheckUnify}$$

  We have $\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \theta$, and will need to apply Lemma 92 (Completeness of Elimeq) (1). That lemma has five side conditions, which can be shown exactly as in the DeclCheck⊥ case above.

	$\mathsf{mgu}(\sigma, \tau) = \theta$	Premise
	Let $\Omega_0 = (\Omega, \blacktriangleright_P)$.	
	$\Gamma \longrightarrow \Omega$	Given
	$\Gamma, \blacktriangleright_P \longrightarrow \Omega_0$	By $\longrightarrow$Marker
	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
	$\mathsf{dom}(\Gamma, \blacktriangleright_P) = \mathsf{dom}(\Omega_0)$	By def. of $\mathsf{dom}(-)$
	$\Gamma, \blacktriangleright_P \;/\; [\Gamma]\sigma \doteq [\Gamma]\tau : \kappa \dashv \Gamma, \blacktriangleright_P, \Theta$	By Lemma 92 (Completeness of Elimeq) (1)
	$\Gamma, \blacktriangleright_P \;/\; [\Gamma]\sigma = [\Gamma]\tau \dashv \Gamma, \blacktriangleright_P, \Theta$	By ElimpropEq
EQ0	for all $\Gamma, \blacktriangleright_P \vdash u : \kappa.\;\; [\Gamma, \blacktriangleright_P, \Theta]u = \theta([\Gamma, \blacktriangleright_P]u)$	″

$\Gamma \vdash P \supset A_0 \ \mathbf{!}\ \mathit{type}$ Given
$\Gamma \vdash A_0 \ \mathbf{!}\ \mathit{type}$ By inversion
$\Gamma \longrightarrow \Omega$ Given
EQa $[\Gamma]A_0 = [\Omega]A_0$ By Lemma 39 (Principal Agreement) (i)

Let $\Omega_1 = (\Omega, \blacktriangleright_P, \Theta)$.
$\theta([\Omega]\Gamma) \vdash \theta(e) \Leftarrow \theta([\Omega]A_0)\ \mathbf{!}$ Subderivation

$\Gamma, \blacktriangleright_P, \Theta \longrightarrow \Omega_1$ By induction on Θ

$\theta([\Omega]A_0) = \theta([\Gamma]A_0)$ By above equality EQa
$= [\Gamma, \blacktriangleright_P, \Theta]A_0$ By Lemma 93 (Substitution Upgrade) (i) (with EQ0)
$= [\Omega_1]A_0$ By Lemma 39 (Principal Agreement) (i)
$= [\Omega_1][\Gamma, \blacktriangleright_P, \Theta]A_0$ By Lemma 29 (Substitution Monotonicity) (iii)

$\theta([\Omega]\Gamma) = [\Omega_1](\Gamma, \blacktriangleright_P, \Theta)$ By Lemma 93 (Substitution Upgrade) (iii)
$\theta([\Omega]e) = [\Omega_1]e$ By Lemma 93 (Substitution Upgrade) (iv)

$[\Omega_1](\Gamma, \blacktriangleright_P, \Theta) \vdash [\Omega_1]e \Leftarrow [\Omega_1][\Gamma, \blacktriangleright_P, \Theta]A_0\ \mathbf{!}$ By above equalities

$\mathsf{dom}(\Gamma, \blacktriangleright_P, \Theta) = \mathsf{dom}(\Omega_1)$ $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$

$\Gamma, \blacktriangleright_P, \Theta \vdash e \Leftarrow [\Gamma, \blacktriangleright_P, \Theta]A_0\ \mathbf{!} \dashv \Delta'$ By i.h.
$\Delta' \longrightarrow \Omega_2'$ "
$\Omega_1 \longrightarrow \Omega_2'$ "
$\mathsf{dom}(\Delta') = \mathsf{dom}(\Omega_2')$ "
$\Delta' = (\Delta, \blacktriangleright_P, \Delta'')$ By Lemma 22 (Extension Inversion) (ii)
$\Omega_2' = (\Omega', \blacktriangleright_P, \Omega_Z)$ By Lemma 22 (Extension Inversion) (ii)
☞ $\Delta \longrightarrow \Omega'$ "
$\Omega_0 \longrightarrow \Omega_2'$ By Lemma 33 (Extension Transitivity)
$\Omega, \blacktriangleright_P \longrightarrow \Omega', \blacktriangleright_P, \Omega_Z$ By above equalities
☞ $\Omega \longrightarrow \Omega'$ By Lemma 22 (Extension Inversion) (ii)
☞ $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ "

$\Gamma, \blacktriangleright_P, \Theta \vdash e \Leftarrow [\Gamma, \blacktriangleright_P, \Theta]A_0\ \mathbf{!} \dashv \Delta, \blacktriangleright_P, \Delta''$ By above equality
$\Gamma \vdash e \Leftarrow ([\Gamma]\sigma = [\Gamma]\tau) \supset [\Gamma]A_0\ \mathbf{!} \dashv \Delta$ By $\supset$I
☞ $\Gamma \vdash e \Leftarrow [\Gamma](P \supset A_0)\ \mathbf{!} \dashv \Delta$ By def. of subst.

- **Case** $$\dfrac{[\Omega]\Gamma \vdash [\Omega]P\ \mathit{true} \qquad [\Omega]\Gamma \vdash [\Omega](e\ s_0) : [\Omega]A_0\ p \gg B\ q}{[\Omega]\Gamma \vdash [\Omega](e\ s_0) : ([\Omega]P) \supset [\Omega]A_0\ p \gg B\ q}\ \mathsf{Decl{\supset}Spine}$$

$[\Omega]\Gamma \vdash [\Omega]P\ \mathit{true}$ Subderivation
$[\Omega]\Gamma \vdash [\Omega][\Gamma]P\ \mathit{true}$ By Lemma 29 (Substitution Monotonicity) (ii)
$\Gamma \vdash [\Gamma]P\ \mathit{true} \dashv \Theta$ By Lemma 95 (Completeness of Checkprop)
$\Theta \longrightarrow \Omega_1$ "
$\Omega \longrightarrow \Omega_1$ "
$\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_1)$ "

$\Gamma \longrightarrow \Omega$ Given
$[\Omega]\Gamma = [\Omega_1]\Theta$ By Lemma 57 (Multiple Confluence)
$[\Omega]A_0 = [\Omega_1]A_0$ By Lemma 55 (Completing Completeness) (ii)

	$[\Omega]\Gamma \vdash [\Omega](e\ s_0) : [\Omega]A_0\ p \gg B\ q$	Subderivation
	$[\Omega_1]\Theta \vdash [\Omega](e\ s_0) : [\Omega_1]A_0\ p \gg B\ q$	By above equalities
	$\Theta \vdash e\ s_0 : [\Theta]A_0\ p \gg B'\ q \dashv \Delta$	By i.h.
☞	$B' = [\Delta]B'$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
☞	$B = [\Omega']B'$	"
☞	$\Delta \longrightarrow \Omega'$	"
	$\Omega_1 \longrightarrow \Omega'$	"
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
	$[\Theta]A_0 = [\Theta][\Gamma]A_0$	By Lemma 29 (Substitution Monotonicity) (iii)

	$\Theta \vdash e\ s_0 : [\Theta][\Gamma]A_0\ p \gg B'\ q \dashv \Delta$	By above equality
	$\Gamma \vdash e\ s_0 : ([\Gamma]P) \supset [\Gamma]A_0\ p \gg B'\ q \dashv \Delta$	By $\supset$Spine
☞	$\Gamma \vdash e\ s_0 : [\Gamma](P \supset A_0)\ p \gg B'\ q \dashv \Delta$	By def. of subst.

- **Case** $$\frac{[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow A'_k\ p}{[\Omega]\Gamma \vdash \mathsf{inj}_k\ [\Omega]e_0 \Leftarrow \underbrace{A'_1 + A'_2}_{[\Omega]A}\ p}\ \mathsf{Decl}{+}\mathsf{I}_k$$

  Either $[\Gamma]A = A_1 + A_2$ (where $[\Omega]A_k = A'_k$) or $[\Gamma]A = \hat{\alpha} \in \mathsf{unsolved}(\Gamma)$.

  In the former case:

	$[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow A'_k\ p$	Subderivation
	$[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega]A_k\ p$	$[\Omega]A_k = A'_k$
	$\Gamma \vdash e_0 \Leftarrow [\Gamma]A_k\ p \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
☞	$\Omega \longrightarrow \Omega'$	"
	$\Gamma \vdash \mathsf{inj}_k\ e_0 \Leftarrow ([\Gamma]A_1) + ([\Gamma]A_2)\ p \dashv \Delta$	By $+\mathsf{I}_k$
☞	$\Gamma \vdash \mathsf{inj}_k\ e_0 \Leftarrow [\Gamma](A_1 + A_2)\ p \dashv \Delta$	By def. of subst.

  In the latter case, $A = \hat{\alpha}$ and $[\Omega]A = [\Omega]\hat{\alpha} = A'_1 + A'_2 = \tau'_1 + \tau'_2$.
  By inversion on $\Gamma \vdash \hat{\alpha}\ p$ *type*, it must be the case that $p = \not{!}$.

$\Gamma \longrightarrow \Omega$	Given
$\Gamma = \Gamma_0[\hat{\alpha} : \star]$	$\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$
$\Omega = \Omega_0[\hat{\alpha} : \star = \tau_0]$	By Lemma 22 (Extension Inversion) (vi)

  Let $\Omega_2 = \Omega_0[\hat{\alpha}_1 : \star = \tau'_1, \hat{\alpha}_1 : \star = \tau'_2, \hat{\alpha} : \star = \hat{\alpha}_1 + \hat{\alpha}_2]$.
  Let $\Gamma_2 = \Gamma_0[\hat{\alpha}_1 : \star, \hat{\alpha}_2 : \star, \hat{\alpha} : \star = \hat{\alpha}_1 + \hat{\alpha}_2]$.

$\Gamma \longrightarrow \Gamma_2$	By Lemma 23 (Deep Evar Introduction) (iii) twice and Lemma 26 (Parallel Admissibility) (ii)
$\Omega \longrightarrow \Omega_2$	By Lemma 23 (Deep Evar Introduction) (iii) twice and Lemma 26 (Parallel Admissibility) (iii)
$\Gamma_2 \longrightarrow \Omega_2$	By Lemma 26 (Parallel Admissibility) (ii), (ii), (iii)

	$[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega_2]\hat{\alpha}_k \not\!\!\!\,p$	Subd. and $A'_k = \tau'_k = [\Omega_2]\hat{\alpha}_k$
	$[\Omega]\Gamma = [\Omega_2]\Gamma_2$	By Lemma 57 (Multiple Confluence)
	$[\Omega_2]\Gamma_2 \vdash e_0 \Leftarrow [\Omega_2]\hat{\alpha}_k \not\!\!\!\,p$	By above equality
	$\Gamma_2 \vdash e_0 \Leftarrow [\Gamma_2]\hat{\alpha}_k \not\!\!\!\,p \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega'$	″
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	″
	$\Omega_2 \longrightarrow \Omega'$	″
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
	$\Gamma \vdash \mathsf{inj}_k\, e_0 \Rightarrow \hat{\alpha} \not\!\!\!\,p \dashv \Delta$	By $+\mathsf{I}\hat{\alpha}_k$
☞	$\Gamma \vdash \mathsf{inj}_k\, e_0 \Rightarrow [\Gamma]\hat{\alpha} \not\!\!\!\,p \dashv \Delta$	$\hat{\alpha} \in \mathsf{unsolved}(\Gamma)$

- **Case** $\dfrac{[\Omega]\Gamma, x : A'_1\ p \vdash [\Omega]e_0 \Leftarrow A'_2\ p}{[\Omega]\Gamma \vdash \lambda x.\, [\Omega]e_0 \Leftarrow A'_1 \to A'_2\ p}\ \mathsf{Decl}{\to}\mathsf{I}$

  We have $[\Omega]A = A'_1 \to A'_2$. Either $[\Gamma]A = A_1 \to A_2$ where $A'_1 = [\Omega]A_1$ and $A'_2 = [\Omega]A_2$—or $[\Gamma]A = \hat{\alpha}$ and $[\Omega]\hat{\alpha} = A'_1 \to A'_2$.

  In the former case:

	$[\Omega]\Gamma, x : A'_1\ p \vdash [\Omega]e_0 \Leftarrow A'_2\ p$	Subderivation
	$A'_1 = [\Omega]A_1$	Known in this subcase
	$= [\Omega][\Gamma]A_1$	By Lemma 30 (Substitution Invariance)
	$[\Omega]A'_1 = [\Omega][\Omega][\Gamma]A_1$	Applying Ω on both sides
	$= [\Omega][\Gamma]A_1$	By idempotence of substitution
	$[\Omega]\Gamma, x : A'_1\ p = [\Omega, x : A'_1\ p](\Gamma, x : [\Gamma]A_1\ p)$	By definition of context application
	$[\Omega, x : A'_1\ p](\Gamma, x : [\Gamma]A_1\ p) \vdash [\Omega]e_0 \Leftarrow A'_2\ p$	By above equality
	$\Gamma \longrightarrow \Omega$	Given
	$\Gamma, x : [\Gamma]A_1\ p \longrightarrow \Omega, x : A'_1\ p$	By $\longrightarrow$Var
	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
	$\mathsf{dom}(\Gamma, x : [\Gamma]A_1\ p) = \Omega, x : A'_1\ p$	By def. of $\mathsf{dom}(-)$
	$\Gamma, x : [\Gamma]A_1\ p \vdash e_0 \Leftarrow A_2\ p \dashv \Delta'$	By i.h.
	$\Delta' \longrightarrow \Omega'_0$	″
	$\mathsf{dom}(\Delta') = \mathsf{dom}(\Omega'_0)$	″
	$\Omega, x : A'_1\ p \longrightarrow \Omega'_0$	″
	$\Omega'_0 = (\Omega', x : A'_1\ p, \Omega_Z)$	By Lemma 22 (Extension Inversion) (v)
☞	$\Omega \longrightarrow \Omega'$	″
	$\Gamma, x : [\Gamma]A_1\ p \longrightarrow \Delta'$	By Lemma 51 (Typing Extension)
	$\Delta' = (\Delta, x : \cdots, \Theta)$	By Lemma 22 (Extension Inversion) (v)
	$\Delta, x : \cdots, \Theta \longrightarrow \Omega', x : A'_1\ p, \Omega_Z$	By above equalities
☞	$\Delta \longrightarrow \Omega'$	By Lemma 22 (Extension Inversion) (v)
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	″
	$\Gamma, x : [\Gamma]A_1\ p \vdash e_0 \Leftarrow [\Gamma]A_2\ p \dashv \Delta, x : \cdots p, \Theta$	By above equality
	$\Gamma \vdash \lambda x.\, e_0 \Leftarrow ([\Gamma]A_1) \to ([\Gamma]A_2)\ p \dashv \Delta$	By $\to$I
☞	$\Gamma \vdash \lambda x.\, e_0 \Leftarrow [\Gamma](A_1 \to A_2)\ p \dashv \Delta$	By definition of substitution

  In the latter case ($[\Gamma]A = \hat{\alpha} \in \mathsf{unsolved}(\Gamma)$ and $[\Omega]\hat{\alpha} = A'_1 \to A'_2 = \tau'_1 \to \tau'_2$):

By inversion on $\Gamma \vdash \hat\alpha\ p\ \mathit{type}$, it must be the case that $p = \not{!}$.
Since $\hat\alpha \in \mathsf{unsolved}(\Gamma)$, the context Γ must have the form $\Gamma_0[\hat\alpha : \star]$.
Let $\Gamma_2 = \Gamma_0[\hat\alpha_1 : \star, \hat\alpha_2 : \star, \hat\alpha : \star = \hat\alpha_1 \to \hat\alpha_2]$.

$\Gamma \longrightarrow \Gamma_2$ — By Lemma 23 (Deep Evar Introduction) (iii) twice and Lemma 26 (Parallel Admissibility) (ii)
$[\Omega]\hat\alpha = \tau_1' \to \tau_2'$ — Known in this subcase

$\Gamma \longrightarrow \Omega$ — Given
$\Omega = \Omega_0[\hat\alpha : \star = \tau_0]$ — By Lemma 22 (Extension Inversion) (vi)

Let $\Omega_2 = \Omega_0[\hat\alpha_1 : \star = \tau_1', \hat\alpha_1 : \star = \tau_2', \hat\alpha : \star = \hat\alpha_1 \to \hat\alpha_2]$.

$\Gamma \longrightarrow \Gamma_2$ — By Lemma 23 (Deep Evar Introduction) (iii) twice and Lemma 26 (Parallel Admissibility) (ii)
$\Omega \longrightarrow \Omega_2$ — By Lemma 23 (Deep Evar Introduction) (iii) twice and Lemma 26 (Parallel Admissibility) (iii)
$\Gamma_2 \longrightarrow \Omega_2$ — By Lemma 26 (Parallel Admissibility) (ii), (ii), (iii)

$[\Omega]\Gamma, x : \tau_1'\ \not{!} \vdash [\Omega]e_0 \Leftarrow \tau_2'\ \not{!}$ — Subderivation
$[\Omega]\Gamma = [\Omega_2]\Gamma_2$ — By Lemma 57 (Multiple Confluence)
$\tau_2' = [\Omega]\hat\alpha_2$ — From above equality
$\quad = [\Omega_2]\hat\alpha_2$ — By Lemma 55 (Completing Completeness) (i)
$\tau_1' = [\Omega_2]\hat\alpha_1$ — Similar

$[\Omega_2]\Gamma_2, x : \tau_1'\ \not{!} = [\Omega_2, x : \tau_1'\ \not{!}](\Gamma_2, x : \hat\alpha_1\ \not{!})$ — By def. of context application
$[\Omega_2, x : \tau_1'\ \not{!}](\Gamma_2, x : \hat\alpha_1\ \not{!}) \vdash [\Omega]e_0 \Leftarrow [\Omega_2]\hat\alpha_2\ \not{!}$ — By above equalities

$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$ — Given
$\mathsf{dom}(\Gamma_2, x : \hat\alpha_1\ \not{!}) = \mathsf{dom}(\Omega_2, x : \tau_1'\ \not{!})$ — By def. of Γ_2 and Ω_2
$\Gamma_2, x : \hat\alpha_1\ \not{!} \vdash e_0 \Leftarrow [\Gamma_2, x : \hat\alpha_1\ \not{!}]\hat\alpha_2\ \not{!} \dashv \Delta^+$ — By i.h.
$\Delta^+ \longrightarrow \Omega^+$ — ″
$\mathsf{dom}(\Delta^+) = \mathsf{dom}(\Omega^+)$ — ″
$\Omega_2 \longrightarrow \Omega^+$ — ″

$\Gamma_2, x : \hat\alpha_1\ \not{!} \longrightarrow \Delta^+$ — By Lemma 51 (Typing Extension)
$\Delta^+ = (\Delta, x : \hat\alpha_1\ \not{!}, \Delta_Z)$ — By Lemma 22 (Extension Inversion) (v)
$\Omega^+ = (\Omega', x : \ldots\ \not{!}, \Omega_Z)$ — By Lemma 22 (Extension Inversion) (v)
☞ $\Delta \longrightarrow \Omega'$ — ″
☞ $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ — ″
$\Omega \longrightarrow \Omega_2$ — Above
$\Omega \longrightarrow \Omega^+$ — By Lemma 33 (Extension Transitivity)
☞ $\Omega \longrightarrow \Omega'$ — By Lemma 22 (Extension Inversion) (v)
$\Gamma \vdash \lambda x.\, e_0 \Leftarrow \hat\alpha\ \not{!} \dashv \Delta$ — By $\to\mathsf{I}\hat\alpha$
$\hat\alpha = [\Gamma]\hat\alpha$ — $\hat\alpha \in \mathsf{unsolved}(\Gamma)$
☞ $\Gamma \vdash \lambda x.\, e_0 \Leftarrow [\Gamma]\hat\alpha\ \not{!} \dashv \Delta$ — By above equality

- **Case** $\dfrac{[\Omega]\Gamma, x : [\Omega]A\ p \vdash [\Omega]v \Leftarrow [\Omega]A\ p}{[\Omega]\Gamma \vdash \mathsf{rec}\ x.\ [\Omega]v \Leftarrow [\Omega]A\ p}$ DeclRec

$$
\begin{array}{ll}
[\Omega]\Gamma, x : [\Omega]A\ p \vdash [\Omega]v \Leftarrow [\Omega]A\ p & \text{Subderivation} \\
[\Omega]\Gamma, x : [\Omega]A\ p = [\Omega, x : [\Omega]A\ p](\Gamma, x : [\Gamma]A\ p) & \text{By definition of context application} \\
[\Omega, x : [\Omega]A\ p](\Gamma, x : [\Gamma]A\ p) \vdash [\Omega]v \Leftarrow [\Omega]A\ p & \text{By above equality}
\end{array}
$$

$$
\begin{array}{rll}
& \Gamma \longrightarrow \Omega & \text{Given} \\
& \Gamma, x : [\Gamma]A\ p \longrightarrow \Omega, x : [\Omega]A\ p & \text{By } \longrightarrow\mathsf{Var} \\
& \mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega) & \text{Given} \\
& \mathsf{dom}(\Gamma, x : [\Gamma]A\ p) = \Omega, x : [\Omega]A\ p & \text{By def. of } \mathsf{dom}(-) \\
& \Gamma, x : [\Gamma]A\ p \vdash v \Leftarrow [\Gamma]A\ p \dashv \Delta' & \text{By i.h.} \\
& \Delta' \longrightarrow \Omega'_0 & '' \\
& \mathsf{dom}(\Delta') = \mathsf{dom}(\Omega'_0) & '' \\
& \Omega, x : [\Omega]A\ p \longrightarrow \Omega'_0 & '' \\
& \Omega'_0 = (\Omega', x : [\Omega]A\ p, \Theta) & \text{By Lemma 22 (Extension Inversion) (v)} \\
☞ & \Omega \longrightarrow \Omega' & '' \\
& \Gamma, x : [\Gamma]A\ p \longrightarrow \Delta' & \text{By Lemma 51 (Typing Extension)} \\
& \Delta' = (\Delta, x : \cdots, \Theta) & \text{By Lemma 22 (Extension Inversion) (v)} \\
& \Delta, x : \cdots, \Theta \longrightarrow \Omega', x : [\Omega]A\ p, \Theta & \text{By above equalities} \\
☞ & \Delta \longrightarrow \Omega' & \text{By Lemma 22 (Extension Inversion) (v)} \\
☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & '' \\
& \Gamma, x : [\Gamma]A\ p \vdash v \Leftarrow [\Gamma]A\ p \dashv \Delta, x : [\Gamma]A\ p, \Theta & \text{By above equality} \\
☞ & \Gamma \vdash \mathsf{rec}\ x.\ v \Leftarrow [\Gamma]A\ p \dashv \Delta & \text{By Rec}
\end{array}
$$

- **Case** $\dfrac{[\Omega]\Gamma \vdash [\Omega]e_0 \Rightarrow A\ q \qquad [\Omega]\Gamma \vdash [\Omega]s_0 : A\ q \gg C\ \lceil p \rceil}{[\Omega]\Gamma \vdash [\Omega](e_0\ s_0) \Rightarrow C\ p}$ Decl$\to$E

$$
\begin{array}{rl}
[\Omega]\Gamma \vdash [\Omega]e_0 \Rightarrow A\ q & \text{Subderivation} \\
\Gamma \vdash e_0 \Rightarrow A'\ q \dashv \Theta & \text{By i.h.} \\
\Theta \longrightarrow \Omega_\Theta & '' \\
\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_\Theta) & '' \\
\Omega \longrightarrow \Omega_\Theta & '' \\
A = [\Omega_\Theta]A' & '' \\
A' = [\Theta]A' & ''
\end{array}
$$

	$\Gamma \longrightarrow \Omega$	Given
	$[\Omega]\Gamma = [\Omega_\Theta]\Theta$	By Lemma 57 (Multiple Confluence)
	$[\Omega]\Gamma \vdash [\Omega]s_0 : A\ q \gg C\ \lceil p \rceil$	Subderivation
	$[\Omega_\Theta]\Theta \vdash [\Omega]s_0 : [\Omega_\Theta]A'\ q \gg C\ \lceil p \rceil$	By above equalities
	$\Theta \vdash s_0 : [\Theta]A'\ q \gg C'\ \lceil p \rceil \dashv \Delta$	By i.h.
☞	$C' = [\Delta]C'$	$''$
☞	$\Delta \longrightarrow \Omega'$	$''$
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	$''$
	$\Omega_\Theta \longrightarrow \Omega'$	$''$
☞	$C = [\Omega']C'$	$''$
	$\Theta \vdash s_0 : A'\ q \gg C'\ \lceil p \rceil \dashv \Delta$	By above equality
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \vdash e_0\ s_0 \Rightarrow C'\ p \dashv \Delta$	By $\rightarrow$E

- **Case**

$$\dfrac{[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A \mathrel{!} \gg C \not\uparrow \qquad \begin{array}{l}\text{for all } C_2.\\ \text{if } [\Omega]\Gamma \vdash [\Omega]s : [\Omega]A \mathrel{!} \gg C_2 \not\uparrow \text{ then } C_2 = C\end{array}}{[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A \mathrel{!} \gg C \lceil ! \rceil}\ \mathsf{DeclSpineRecover}$$

	$\Gamma \longrightarrow \Omega$	Given
	$[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A \mathrel{!} \gg C \not\uparrow$	Subderivation
	$\Gamma \vdash s : [\Gamma]A \mathrel{!} \gg C' \not\uparrow \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega'$	″
☞	$\Omega \longrightarrow \Omega'$	″
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	″
☞	$C = [\Omega']C'$	″
☞	$C' = [\Delta]C'$	″

Suppose, for a contradiction, that $\mathsf{FEV}([\Delta]C') \neq \emptyset$.
That is, there exists some $\hat{\alpha} \in \mathsf{FEV}([\Delta]C')$.

	$\Delta \longrightarrow \Omega_2$	By Lemma 60 (Split Solutions)
	$\underbrace{\Omega'_1[\hat{\alpha} : \kappa = t_1]}_{\Omega_1} \longrightarrow \Omega'$	″
	$\Omega_2 = \Omega'_1[\hat{\alpha} : \kappa = t_2]$	″
	$t_2 \neq t_1$	″
(NEQ)	$[\Omega_2]\hat{\alpha} \neq [\Omega'_1]\hat{\alpha}$	By def. of subst. ($t_2 \neq t_1$)
(EQ)	$[\Omega_2]\hat{\beta} = [\Omega'_1]\hat{\beta}$ for all $\hat{\beta} \neq \hat{\alpha}$	By construction of Ω_2 and Ω_2 canonical

Choose $\hat{\alpha}_R$ such that $\hat{\alpha}_R \in \mathsf{FEV}(C')$ and either $\hat{\alpha}_R = \hat{\alpha}$ or $\hat{\alpha} \in \mathsf{FEV}([\Delta]\hat{\alpha}_R)$.
Then either $\hat{\alpha}_R = \hat{\alpha}$, or $\hat{\alpha}_R$ is declared to the right of $\hat{\alpha}$ in Δ.

$[\Omega_2]C' \neq [\Omega']C'$	From (NEQ) and (EQ)
$\Gamma \vdash s : [\Gamma]A \mathrel{!} \gg C' \not\uparrow \dashv \Delta$	Above
$[\Omega_2]\Gamma \vdash [\Omega_2]s : [\Omega_2][\Gamma]A \mathrel{!} \gg [\Omega_2]C' \not\uparrow$	By Theorem 8

$\Gamma \vdash s : [\Gamma]A \mathrel{!} \gg C' \not\uparrow \dashv \Delta$	Above
$\Gamma \vdash A \mathrel{!}$ *type*	Given
$\Gamma \vdash [\Gamma]A \mathrel{!}$ *type*	By Lemma 13 (Right-Hand Substitution for Typing)
$\mathsf{FEV}([\Gamma]A) = \emptyset$	By inversion
$\mathsf{FEV}([\Gamma]A) \subseteq \mathsf{dom}(\cdot)$	Property of $\subseteq$
$\Delta = (\Delta_L * \Delta_R)$	By Lemma 72 (Separation—Main) (Spines)
$(\Gamma * \cdot) \xrightarrow[*]{} (\Delta_L * \Delta_R)$	″
$\mathsf{FEV}(C') \subseteq \mathsf{dom}(\Delta_R)$	″
$\hat{\alpha}_R \in \mathsf{FEV}(C')$	Above
$\hat{\alpha}_R \in \mathsf{dom}(\Delta_R)$	Property of $\subseteq$
$\mathsf{dom}(\Delta_L) \cap \mathsf{dom}(\Delta_R) = \emptyset$	Δ well-formed
$\hat{\alpha}_R \notin \mathsf{dom}(\Delta_L)$	
$\mathsf{dom}(\Gamma) \subseteq \mathsf{dom}(\Delta_L)$	By Definition 5
$\hat{\alpha}_R \notin \mathsf{dom}(\Gamma)$	

$[\Omega_2]\Gamma \vdash [\Omega_2]s : [\Omega_2][\Gamma]A\ ! \gg [\Omega_2]C'\ \not!$ Above

Ω_2 and Ω_1 differ only at $\hat\alpha$ Above
$\mathsf{FEV}([\Gamma]A) = \emptyset$ Above
$[\Omega_2][\Gamma]A = [\Omega_1][\Gamma]A$ By preceding two lines

$\Gamma \vdash [\Gamma]A$ *type* Above
$\Gamma \longrightarrow \Omega_2$ By Lemma 33 (Extension Transitivity)
$\Omega_2 \vdash [\Gamma]A$ *type* By Lemma 38 (Extension Weakening (Types))
$\mathsf{dom}(\Omega_2) = \mathsf{dom}(\Omega_1)$ Ω_1 and Ω_2 differ only at $\hat\alpha$
$\Omega_1 \vdash [\Gamma]A$ *type* By Lemma 18 (Equal Domains)

$\Gamma \vdash [\Gamma]A$ *type* Above
$\Omega \vdash [\Gamma]A$ *type* By Lemma 38 (Extension Weakening (Types))
$[\Omega_1][\Gamma]A = [\Omega'][\Gamma]A = [\Omega][\Gamma]A$ By Lemma 55 (Completing Completeness) (ii) twice
$= [\Omega]A$ By Lemma 29 (Substitution Monotonicity) (iii)

$[\Omega]\Gamma = [\Omega']\Gamma$ By Lemma 57 (Multiple Confluence)
$= [\Omega_1]\Gamma$ By Lemma 57 (Multiple Confluence)
$= [\Omega_2]\Gamma$ Follows from $\hat\alpha_R \notin \mathsf{dom}(\Gamma)$

$[\Omega_2]s = [\Omega]s$ Ω_2 and Ω differ only in $\hat\alpha$

$[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ ! \gg [\Omega_2]C'\ \not!$ By above equalities

$C = [\Omega']C'$ Above
$[\Omega']C' \neq [\Omega_2]C'$ By def. of subst.
$C \neq [\Omega_2]C'$ By above equality
$C = [\Omega_2]C'$ Instantiating "for all C_2" with $C_2 = [\Omega_2]C'$
$\Rightarrow\Leftarrow$
$\mathsf{FEV}([\Delta]C') = \emptyset$ By contradiction

☞ $\Gamma \vdash s : [\Gamma]A\ ! \gg C'\ \lceil ! \rceil \dashv \Delta$ By SpineRecover

- **Case** $$\frac{[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg C\ q}{[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg C\ \lceil q \rceil}\ \mathsf{DeclSpinePass}$$

  $[\Omega]\Gamma \vdash [\Omega]s : [\Omega]A\ p \gg C\ q$ Subderivation
  $\Gamma \vdash s : [\Gamma]A\ p \gg C'\ q \dashv \Delta$ By i.h.
  ☞ $\Delta \longrightarrow \Omega'$ ″
  ☞ $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ ″
  ☞ $\Omega \longrightarrow \Omega'$ ″
  ☞ $C' = [\Delta]C'$ ″
  ☞ $C = [\Omega']C'$ ″

  We distinguish cases as follows:

  - If $p = \not!$ or $q = !$, then we can just apply SpinePass:

    ☞ $\Gamma \vdash s : [\Gamma]A\ p \gg C'\ \lceil q \rceil \dashv \Delta$ By SpinePass

  - Otherwise, $p = !$ and $q = \not!$. If $\mathsf{FEV}(C) \neq \emptyset$, we can apply SpinePass, as above. If $\mathsf{FEV}(C) = \emptyset$, then we instead apply SpineRecover:

    ☞ $\Gamma \vdash s : [\Gamma]A\ p \gg C'\ \lceil ! \rceil \dashv \Delta$ By SpineRecover

    Here, $q' = !$ and $q = \not!$, so $q' \sqsubseteq q$.

- **Case**
$$\frac{}{[\Omega]\Gamma \vdash \cdot : [\Omega]A\ p \gg [\Omega]A\ p}\ \text{DeclEmptySpine}$$

☞	$\Gamma \vdash \cdot : [\Gamma]A\ p \gg [\Gamma]A\ p \dashv \Gamma$	By EmptySpine
☞	$[\Gamma]A = [\Gamma][\Gamma]A$	By idempotence of substitution
☞	$\Gamma \longrightarrow \Omega$	Given
☞	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given
☞	$[\Omega][\Gamma]A = [\Omega]A$	By Lemma 29 (Substitution Monotonicity) (iii)
☞	$\Omega \longrightarrow \Omega$	By Lemma 32 (Extension Reflexivity)

- **Case**
$$\frac{[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega]A_1\ q \qquad [\Omega]\Gamma \vdash [\Omega]s_0 : [\Omega]A_2\ q \gg B\ p}{[\Omega]\Gamma \vdash [\Omega](e_0\ s_0) : ([\Omega]A_1) \rightarrow ([\Omega]A_2)\ q \gg B\ p}\ \text{Decl}\!\rightarrow\!\text{Spine}$$

	$[\Omega]\Gamma \vdash [\Omega]e_0 \Leftarrow [\Omega]A_1\ q$	Subderivation
	$\Gamma \vdash e_0 \Leftarrow A'\ q \dashv \Theta$	By i.h.
	$\Theta \longrightarrow \Omega_\Theta$	″
	$\Omega \longrightarrow \Omega_\Theta$	″
	$A = [\Omega_\Theta]A'$	″
	$A' = [\Theta]A'$	″
	$[\Omega]\Gamma \vdash [\Omega]s_0 : [\Omega]A_2\ q \gg B\ p$	Subderivation
	$\Gamma \vdash s_0 : A_2\ q \gg B\ p \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega'$	″
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	″
☞	$\Omega \longrightarrow \Omega'$	″
☞	$B' = [\Delta]B'$	″
☞	$B = [\Omega']B'$	″
☞	$\Gamma \vdash e_0\ s_0 : A_1 \rightarrow A_2\ q \gg B\ p \dashv \Delta$	By $\rightarrow$Spine

- **Case** $\dfrac{[\Omega]\Gamma \vdash [\Omega]P\ \mathit{true} \qquad [\Omega]\Gamma \vdash [\Omega]e \Leftarrow [\Omega]A_0\ p}{[\Omega]\Gamma \vdash [\Omega]e \Leftarrow ([\Omega]A_0) \wedge [\Omega]P\ p}$ Decl∧I

  If e not a case, then:

$$
\begin{array}{rll}
& [\Omega]\Gamma \vdash [\Omega]P\ \mathit{true} & \text{Subderivation} \\
& \Gamma \vdash P\ \mathit{true} \dashv \Theta & \text{By Lemma 95 (Completeness of Checkprop)} \\
& \Theta \longrightarrow \Omega_0' & '' \\
& \Omega \longrightarrow \Omega_0' & '' \\[1ex]
& \Gamma \longrightarrow \Omega & \text{Given} \\
& \Gamma \longrightarrow \Omega_0' & \text{By Lemma 33 (Extension Transitivity)} \\
& [\Omega]\Gamma = [\Omega]\Omega & \text{By Lemma 54 (Completing Stability)} \\
& = [\Omega_0']\Omega_0' & \text{By Lemma 55 (Completing Completeness) (iii)} \\
& = [\Omega_0']\Theta & \text{By Lemma 56 (Confluence of Completeness)} \\[1ex]
& \Gamma \vdash A_0 \wedge P\ p\ \mathit{type} & \text{Given} \\
& \Gamma \vdash A_0\ p\ \mathit{type} & \text{By inversion} \\
& [\Omega]A_0 = [\Omega_0']A_0 & \text{By Lemma 55 (Completing Completeness) (ii)} \\[1ex]
& [\Omega]\Gamma \vdash [\Omega]e \Leftarrow [\Omega]A_0\ p & \text{Subderivation} \\
& [\Omega_0']\Theta \vdash [\Omega]e \Leftarrow [\Omega_0']A_0\ p & \text{By above equalities} \\
& \Theta \vdash e \Leftarrow [\Theta]A_0\ p \dashv \Delta & \text{By i.h.} \\
☞ & \Delta \longrightarrow \Omega' & '' \\
☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & '' \\
& \Omega_0' \longrightarrow \Omega' & '' \\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 33 (Extension Transitivity)} \\[1ex]
☞ & \Gamma \vdash e \Leftarrow A_0 \wedge P\ p \dashv \Delta & \text{By } \wedge\text{I}
\end{array}
$$

Otherwise, we have $e = \mathsf{case}(e_0, \Pi)$. Let n be the height of the given derivation.

$$
\begin{array}{lll}
n-1 & [\Omega]\Gamma \vdash [\Omega](\mathsf{case}(e_0, \Pi)) \Leftarrow [\Omega]A_0\ p & \text{Subderivation} \\
n-2 & [\Omega]\Gamma \vdash [\Omega]e_0 \Rightarrow B\ ! & \text{By Lemma 62 (Case Invertibility)} \\
n-2 & [\Omega]\Gamma \vdash [\Omega]\Pi :: B \Leftarrow [\Omega]A_0\ p & '' \\
n-2 & [\Omega]\Gamma \vdash [\Omega]\Pi\ \mathit{covers}\ B & '' \\
n-1 & [\Omega]\Gamma \vdash [\Omega]P\ \mathit{true} & \text{Subderivation} \\
n-1 & [\Omega]\Gamma \vdash [\Omega]\Pi :: B \Leftarrow ([\Omega]A_0) \wedge ([\Omega]P)\ p & \text{By Lemma 61 (Interpolating With and Exists) (1)} \\
n-1 & [\Omega]\Gamma \vdash [\Omega]\Pi :: B \Leftarrow [\Omega](A_0 \wedge P)\ p & \text{By def. of subst.}
\end{array}
$$

$$
\begin{array}{rl}
\Gamma \vdash e_0 \Rightarrow B'\ ! \dashv \Theta & \text{By i.h.} \\
\Theta \longrightarrow \Omega_0' & '' \\
\Omega \longrightarrow \Omega_0' & '' \\
B = [\Omega_0']B' & '' \\
= [\Omega_0'][\Theta]B' & \text{By Lemma 30 (Substitution Invariance)} \\[1ex]
[\Omega]\Gamma = [\Omega_0']\Theta & \text{By Lemma 57 (Multiple Confluence)} \\
[\Omega](A_0 \wedge P) = [\Omega_0'](A_0 \wedge P) & \text{By Lemma 55 (Completing Completeness) (ii)}
\end{array}
$$

$$
\begin{array}{rll}
n-1 & [\Omega_0']\Theta \vdash [\Omega]\Pi :: [\Omega_0'][\Theta]B' \Leftarrow [\Omega_0'](A_0 \wedge P)\ p & \text{By above equalities} \\
& \Theta \vdash \Pi :: [\Theta]B' \Leftarrow A_0 \wedge P\ p \dashv \Delta & \text{By i.h.} \\
☞ & \Delta \longrightarrow \Omega' & '' \\
☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & '' \\
& \Omega_0' \longrightarrow \Omega' & ''
\end{array}
$$

	$\Theta \vdash \Pi\ \mathit{covers}\ [\Theta]B'$	By Theorem 10
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \vdash \mathsf{case}(e_0, \Pi) \Leftarrow A_0 \land P\ p \dashv \Delta$	By Case

- **Case** DeclNil: Similar to the first part of the Decl∧I case.

- **Case**

$$\frac{[\Omega]\Gamma \vdash ([\Omega]t) = \mathsf{succ}(t_2)\ \mathit{true} \qquad \begin{array}{l}[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow [\Omega]A_0\ p \\ [\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow \big(\mathsf{Vec}\ t_2\ [\Omega]A_0\big)\ \not{\!\!\!\!\!\;\;}\end{array}}{[\Omega]\Gamma \vdash ([\Omega]e_1) :: ([\Omega]e_2) \Leftarrow \big(\mathsf{Vec}\ ([\Omega]t)\ [\Omega]A_0\big)\ p}\ \mathsf{DeclCons}$$

Let $\Omega^+ = (\Omega, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} = t_2)$.

	$[\Omega]\Gamma \vdash ([\Omega]t) = \mathsf{succ}(t_2)\ \mathit{true}$	Subderivation
	$[\Omega^+](\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N}) \vdash ([\Omega]t) = [\Omega^+]\mathsf{succ}(\hat\alpha)\ \mathit{true}$	Defs. of extension and subst.
1	$\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} \vdash t = \mathsf{succ}(\hat\alpha)\ \mathit{true} \dashv \Gamma'$	By Lemma 95 (Completeness of Checkprop)
	$\Gamma' \longrightarrow \Omega_0'$	"
	$\Omega^+ \longrightarrow \Omega_0'$	"

$\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} \longrightarrow \Gamma'$	By Lemma 47 (Checkprop Extension)
$\Gamma, \blacktriangleright_{\hat\alpha}, \hat\alpha : \mathbb{N} \longrightarrow \Omega_0'$	By Lemma 33 (Extension Transitivity)
$[\Omega]\Gamma = [\Omega]\Omega$	By Lemma 54 (Completing Stability)
$= [\Omega^+]\Omega^+$	By def. of context application
$= [\Omega_0']\Omega_0'$	By Lemma 55 (Completing Completeness) (iii)
$= [\Omega_0']\Gamma'$	By Lemma 56 (Confluence of Completeness)
$[\Omega]A_0 = [\Omega^+]A_0$	By def. of context application
$= [\Omega_0']A_0$	By Lemma 55 (Completing Completeness) (ii)

	$[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow [\Omega]A_0\ p$	Subderivation
	$[\Omega_0']\Gamma' \vdash [\Omega]e_1 \Leftarrow [\Omega_0']A_0\ p$	By above equalities
2	$\Gamma' \vdash e_1 \Leftarrow [\Gamma']A_0\ p \dashv \Theta$	By i.h.
	$\Theta \longrightarrow \Omega_0''$	"
	$\Omega_0' \longrightarrow \Omega_0''$	"
	$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow \big(\mathsf{Vec}\ t_2\ [\Omega]A_0\big)\ \not{\!\!\!\!\!\;\;}$	Subderivation
	$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow \big(\mathsf{Vec}\ ([\Omega^+]\hat\alpha)\ [\Omega]A_0\big)\ \not{\!\!\!\!\!\;\;}$	By def. of substitution
	$[\Omega_0'']\Theta \vdash [\Omega]e_2 \Leftarrow \big(\mathsf{Vec}\ ([\Omega_0'']\hat\alpha)\ [\Omega_0'']A_0\big)\ \not{\!\!\!\!\!\;\;}$	By lemmas
	$[\Omega_0'']\Theta \vdash [\Omega]e_2 \Leftarrow [\Omega_0'']\big(\mathsf{Vec}\ \hat\alpha\ A_0\big)\ \not{\!\!\!\!\!\;\;}$	By def. of subst.
3	$\Theta \vdash e_2 \Leftarrow [\Theta]A_0\ p \dashv \Delta, \blacktriangleright_{\hat\alpha}, \Delta'$	By i.h.
	$\Delta, \blacktriangleright_{\hat\alpha}, \Delta' \longrightarrow \Omega''$	"
	$\mathsf{dom}(\Delta, \blacktriangleright_{\hat\alpha}, \Delta') = \mathsf{dom}(\Omega'')$	"
	$\Omega_0'' \longrightarrow \Omega''$	"

	$\Omega'' = (\Omega, \blacktriangleright_{\hat\alpha}, \dots)$	By Lemma 22 (Extension Inversion) (ii)
☞	$\Delta \longrightarrow \Omega'$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
	$(\Gamma', \blacktriangleright_{\hat\alpha}, \dots) \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Omega \longrightarrow \Omega'$	By Lemma 22 (Extension Inversion) (ii)
☞	$\Gamma \vdash e_1 :: e_2 \Leftarrow (\mathsf{Vec}\ t\ A_0)\ p \dashv \Delta$	By Cons

- **Case**

$$\frac{[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow A_1'\ p \qquad [\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow A_2'\ p}{[\Omega]\Gamma \vdash \langle [\Omega]e_1, [\Omega]e_2 \rangle \Leftarrow A_1' \times A_2'\ p}\ \mathsf{Decl}{\times}\mathsf{I}$$

Either $[\Gamma]A = A_1 \times A_2$ or $[\Gamma]A = \hat{\alpha} \in \mathsf{unsolved}(\Gamma)$.

- In the first case ($[\Gamma]A = A_1 \times A_2$), we have $A_1' = [\Omega]A_1$ and $A_2' = [\Omega]A_2$.

$$
\begin{array}{rll}
& [\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow A_1'\ p & \text{Subderivation} \\
& [\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow [\Omega]A_1\ p & [\Omega]A_1 = A_1' \\
& \Gamma \vdash e_1 \Leftarrow [\Gamma]A_1\ p \dashv \Theta & \text{By i.h.} \\
& \Theta \longrightarrow \Omega_\Theta & '' \\
& \mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_\Theta) & '' \\
& \Omega \longrightarrow \Omega_\Theta & '' \\
\\
& [\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow A_2'\ p & \text{Subderivation} \\
& [\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow [\Omega]A_2\ p & [\Omega]A_2 = A_2' \\
\\
& \Gamma \longrightarrow \Theta & \text{By Lemma 51 (Typing Extension)} \\
& [\Omega]\Gamma = [\Omega_\Theta]\Theta & \text{By Lemma 57 (Multiple Confluence)} \\
& [\Omega]A_2 = [\Omega_\Theta]A_2 & \text{By Lemma 55 (Completing Completeness) (ii)} \\
\\
& [\Omega_\Theta]\Theta \vdash [\Omega]e_2 \Leftarrow [\Omega_\Theta]A_2\ p & \text{By above equalities} \\
& \Theta \vdash e_2 \Leftarrow [\Gamma]A_2\ p \dashv \Delta & \text{By i.h.} \\
☞ & \Delta \longrightarrow \Omega' & '' \\
☞ & \mathsf{dom}(\Delta) = \mathsf{dom}(\Omega') & '' \\
& \Omega_\Theta \longrightarrow \Omega' & '' \\
☞ & \Omega \longrightarrow \Omega' & \text{By Lemma 33 (Extension Transitivity)} \\
& \Gamma \vdash \langle e_1, e_2 \rangle \Leftarrow ([\Gamma]A_1) \times ([\Gamma]A_2)\ p \dashv \Delta & \text{By } \times\mathsf{I} \\
☞ & \Gamma \vdash \langle e_1, e_2 \rangle \Leftarrow [\Gamma](A_1 \times A_2)\ p \dashv \Delta & \text{By def. of subst.}
\end{array}
$$

- In the second case, where $[\Gamma]A = \hat{\alpha}$, combine the corresponding subcase for Decl+I_k with some straightforward additional reasoning about contexts (because here we have two subderivations, rather than one).

- **Case** $$\dfrac{[\Omega]\Gamma \vdash [\Omega]e_0 \Rightarrow C\ \mathbf{!} \qquad [\Omega]\Gamma \vdash [\Omega]\Pi :: C \Leftarrow [\Omega]A\ p \qquad [\Omega]\Gamma \vdash [\Omega]\Pi\ \textit{covers}\ C}{[\Omega]\Gamma \vdash \mathsf{case}([\Omega]e_0, [\Omega]\Pi) \Leftarrow [\Omega]A\ p}\ \mathsf{DeclCase}$$

$$
\begin{array}{rl}
[\Omega]\Gamma \vdash [\Omega]e_0 \Rightarrow C\ \mathbf{!} & \text{Subderivation} \\
\Gamma \vdash e_0 \Rightarrow C'\ \mathbf{!} \dashv \Theta & \text{By i.h.} \\
\Theta \longrightarrow \Omega_\Theta & '' \\
\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_\Theta) & '' \\
\Omega \longrightarrow \Omega_\Theta & '' \\
C = [\Omega_\Theta]C' & '' \\
\\
\Theta \vdash C'\ \mathbf{!}\ \textit{type} & \text{By Lemma 63 (Well-Formed Outputs of Typing)} \\
\mathsf{FEV}(C') = \emptyset & \text{By inversion} \\
[\Omega_\Theta]C' = C' & \text{By a property of substitution}
\end{array}
$$

	$\Gamma \longrightarrow \Omega$	Given
	$\Delta \longrightarrow \Omega$	Given
	$\Theta \longrightarrow \Omega$	By Lemma 33 (Extension Transitivity)
	$[\Omega]\Gamma = [\Omega]\Theta = [\Omega]\Delta$	By Lemma 56 (Confluence of Completeness)
	$\Gamma \longrightarrow \Theta$	By Lemma 51 (Typing Extension)
	$\Gamma \longrightarrow \Omega_\Theta$	By Lemma 33 (Extension Transitivity)
	$[\Omega]\Gamma = [\Omega_\Theta]\Theta$	By Lemma 57 (Multiple Confluence)
	$\Gamma \vdash A\ \mathit{type}$	Given + inversion
	$\Omega \vdash A\ \mathit{type}$	By Lemma 38 (Extension Weakening (Types))
	$[\Omega]A = [\Omega_\Theta]A$	By Lemma 55 (Completing Completeness) (ii)
	$[\Omega]\Gamma \vdash [\Omega]\Pi :: C \Leftarrow [\Omega]A\ p$	Subderivation
	$[\Omega_\Theta]\Theta \vdash [\Omega]\Pi :: [\Omega_\Theta]C' \Leftarrow [\Omega_\Theta]A\ p$	By above equalities
	$\Theta \vdash \Pi :: C' \Leftarrow [\Theta]A\ p \dashv \Delta$	By i.h. (v)
☞	$\Delta \longrightarrow \Omega'$	″
	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	″
	$\Omega_\Theta \longrightarrow \Omega$	″
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)

$[\Omega]\Gamma \vdash [\Omega]\Pi\ \mathit{covers}\ C$	Subderivation
$[\Omega]\Gamma = [\Omega]\Delta$	Above
$\quad = [\Omega']\Delta$	By Lemma 57 (Multiple Confluence)
$[\Omega']\Delta \vdash [\Omega]\Pi\ \mathit{covers}\ C'$	By above equalities
$\Delta \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)

$\Gamma \vdash C'\ !\ \mathit{type}$	Given
$\Gamma \longrightarrow \Delta$	By Lemma 51 (Typing Extension) & 33
$\Delta \vdash C'\ !\ \mathit{type}$	By Lemma 41 (Extension Weakening for Principal Typing)
$[\Delta]C' = C'$	By $\mathsf{FEV}(C') = \emptyset$ and a property of subst.
$\Delta \vdash \Pi\ \mathit{covers}\ C'$	By Theorem 10

☞ $\Gamma \vdash \mathsf{case}(e_0, \Pi) \Leftarrow [\Gamma]A\ p \dashv \Delta$ By Case

- **Case** $$\dfrac{[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow A_1\ p \qquad [\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow A_2\ p}{[\Omega]\Gamma \vdash \langle [\Omega]e_1, [\Omega]e_2 \rangle \Leftarrow \underbrace{A_1 \times A_2}_{[\Omega]A}\ p}\ \mathsf{Decl}{\times}\mathsf{I}$$

Either $A = \hat{\alpha}$ where $[\Omega]\hat{\alpha} = A_1 \times A_2$, or $A = A_1' \times A_2'$ where $A_1 = [\Omega]A_1'$ and $A_2 = [\Omega]A_2'$.

In the former case ($A = \hat{\alpha}$):

We have $[\Omega]\hat{\alpha} = A_1 \times A_2$. Therefore $A_1 = [\Omega]A_1'$ and $A_2 = [\Omega]A_2'$. Moreover, $\Gamma = \Gamma_0[\hat{\alpha} : \kappa]$.

$[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow [\Omega]A_1'\ p$ Subderivation

Let $\Gamma' = \Gamma_0[\hat{\alpha}_1 : \kappa, \hat{\alpha}_2 : \kappa, \hat{\alpha} : \kappa = \hat{\alpha}_1 + \hat{\alpha}_2]$.

$[\Omega]\Gamma = [\Omega]\Gamma'$	By def. of context substitution
$[\Omega]\Gamma' \vdash [\Omega]e_1 \Leftarrow [\Omega]A_1'\ p$	By above equality
$\Gamma' \vdash e_1 \Leftarrow [\Gamma']A_1'\ p' \dashv \Theta$	By i.h.
$\Theta \longrightarrow \Omega_1$	″
$\Omega \longrightarrow \Omega_1$	″
$\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_1)$	″

$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow [\Omega]A_2'\ p$ Subderivation

	$[\Omega]\Gamma = [\Omega_1]\Theta$	By Lemma 57 (Multiple Confluence)
	$[\Omega]A_2' = [\Omega_1]A_2'$	By Lemma 55 (Completing Completeness) (ii)
	$[\Omega_1]\Theta \vdash [\Omega]e_2 \Leftarrow [\Omega_1]A_2'\ p$	By above equalities
	$\Theta \vdash e_2 \Leftarrow [\Theta]A_2'\ p' \dashv \Delta$	By i.h.
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
☞	$\Delta \longrightarrow \Omega'$	"
	$\Omega_1 \longrightarrow \Omega'$	"
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
☞	$\Gamma \vdash \langle e_1, e_2\rangle \Leftarrow \hat{\alpha}\ p' \dashv \Delta$	By $\times\mathsf{I}\hat{\alpha}$

In the latter case ($A = A_1' \times A_2'$):

	$[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow A_1\ p$	Subderivation
	$[\Omega]\Gamma \vdash [\Omega]e_1 \Leftarrow [\Omega]A_1'\ p$	$A_1 = [\Omega]A_1'$
	$\Gamma \vdash e_1 \Leftarrow [\Gamma]A_1'\ p \dashv \Theta$	By i.h.
	$\Theta \longrightarrow \Omega_0$	"
	$\mathsf{dom}(\Theta) = \mathsf{dom}(\Omega_0)$	"
	$\Omega \longrightarrow \Omega_0$	"
	$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow A_2\ p$	Subderivation
	$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow [\Omega]A_2'\ p$	$A_2 = [\Omega]A_2'$
	$\Gamma \vdash A_1' \times A_2'\ p\ \mathit{type}$	Given ($A = A_1' \times A_2'$)
	$\Gamma \vdash A_2'\ \mathit{type}$	By inversion
	$\Gamma \longrightarrow \Omega$	Given
	$\Gamma \longrightarrow \Omega_0$	By Lemma 33 (Extension Transitivity)
	$\Omega_0 \vdash A_2'\ \mathit{type}$	By Lemma 38 (Extension Weakening (Types))
	$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow [\Omega_0]A_2'\ p$	By Lemma 55 (Completing Completeness)
	$[\Omega]\Gamma \vdash [\Omega]e_2 \Leftarrow [\Omega_0][\Theta]A_2'\ p$	By Lemma 29 (Substitution Monotonicity) (iii)
	$[\Omega]\Theta \vdash [\Omega]e_2 \Leftarrow [\Omega_0][\Theta]A_2'\ p$	By Lemma 57 (Multiple Confluence)
	$\Theta \vdash e_2 \Leftarrow [\Theta]A_2'\ p \dashv \Delta$	By i.h.
☞	$\Delta \longrightarrow \Omega'$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
	$\Omega_0 \longrightarrow \Omega'$	"
☞	$\Omega \longrightarrow \Omega'$	By Lemma 33 (Extension Transitivity)
	$\Gamma \vdash \langle e_1, e_2\rangle \Leftarrow ([\Omega]A_1) \times ([\Omega]A_2)\ p \dashv \Delta$	By $\times\mathsf{I}$
☞	$\Gamma \vdash \langle e_1, e_2\rangle \Leftarrow [\Omega](A_1 \times A_2)\ p \dashv \Delta$	By def. of substitution

Now we turn to parts (v) and (vi), completeness of matching.

- **Case** DeclMatchEmpty: Apply rule MatchEmpty.
- **Case** DeclMatchSeq: Apply the i.h. twice, along with standard lemmas.
- **Case** DeclMatchBase: Apply the i.h. (i) and rule MatchBase.
- **Case** DeclMatchUnit: Apply the i.h. and rule MatchUnit.
- **Case** DeclMatch$\exists$: By i.h. and rule Match$\exists$.
- **Case** DeclMatch$\times$: By i.h. and rule Match$\times$.
- **Case** DeclMatch$+_k$: By i.h. and rule Match$+_k$.

- **Case** $$\dfrac{[\Omega]\Gamma \;/\; P \vdash \vec{\rho} \Rightarrow e :: [\Omega]A, [\Omega]\vec{A} \Leftarrow [\Omega]C\ p}{[\Omega]\Gamma \vdash \vec{\rho} \Rightarrow e :: ([\Omega]A \wedge [\Omega]P), [\Omega]\vec{A} \Leftarrow [\Omega]C\ p}\ \mathsf{DeclMatch}\wedge$$

  To apply the i.h. (vi), we will show (1) $\Gamma \vdash (A, \vec{A})$ **!** *types*, (2) $\Gamma \vdash P$ *prop*, (3) $\mathsf{FEV}(P) = \emptyset$, (4) $\Gamma \vdash C\ p$ *type*, (5) $[\Omega]\Gamma \;/\; [\Omega]P \vdash \vec{\rho} \Rightarrow [\Omega]e :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$, and (6) $p' \sqsubseteq p$.

	$\Gamma \vdash (A \wedge P, \vec{A})$ **!** *types*	Given
	$\Gamma \vdash (A \wedge P)$ **!** *type*	By inversion on PrincipalTypevecWF
	$\Gamma \vdash A$ **!** *type*	By Lemma 42 (Inversion of Principal Typing) (3)
(2)	$\Gamma \vdash P$ *prop*	"
(3)	$\mathsf{FEV}(P) = \emptyset$	By inversion
(1)	$\Gamma \vdash (A, \vec{A})$ **!** *types*	By inversion and PrincipalTypevecWF

(4)	$\Gamma \vdash C\ p$ *type*	Given
(5)	$[\Omega]\Gamma \;/\; P \vdash \vec{\rho} \Rightarrow [\Omega]e :: [\Omega]A, [\Omega]\vec{A} \Leftarrow [\Omega]C\ p$	Subderivation
(6)	$p' \sqsubseteq p$	Given

	$\Gamma \;/\; [\Gamma]P \vdash \vec{\rho} \Rightarrow e :: [\Gamma](A, \vec{A}) \Leftarrow [\Gamma]C\ p' \dashv \Delta$	By i.h. (vi)
☞	$\Delta \longrightarrow \Omega'$	"
☞	$\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$	"
☞	$\Omega \longrightarrow \Omega'$	"

	$\Gamma \;/\; [\Gamma]P \vdash \vec{\rho} \Rightarrow e :: [\Gamma]A, [\Gamma]\vec{A}) \Leftarrow [\Gamma]C\ p' \dashv \Delta$	By def. of subst.
	$\Gamma \vdash \vec{\rho} \Rightarrow e :: ([\Gamma]A \wedge [\Gamma]P), [\Gamma]\vec{A}) \Leftarrow [\Gamma]C\ p' \dashv \Delta$	By Match$\wedge$
☞	$\Gamma \vdash \vec{\rho} \Rightarrow e :: [\Gamma]\big((A \wedge P), \vec{A}\big) \Leftarrow [\Gamma]C\ p' \dashv \Delta$	By def. of subst.

- **Case** DeclMatchNeg: By i.h. and rule MatchNeg.

- **Case** DeclMatchWild: By i.h. and rule MatchWild.

- **Case** DeclMatchNil: Similar to the DeclMatch$\wedge$ case.

- **Case** DeclMatchCons: Similar to the DeclMatch$\exists$ and DeclMatch$\wedge$ cases.

- **Case** $$\dfrac{\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \bot}{[\Omega]\Gamma \;/\; [\Omega]\sigma = [\Omega]\tau \vdash [\Omega](\vec{\rho} \Rightarrow e) :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p}\ \mathsf{DeclMatch}\bot$$

☞	$\Gamma \longrightarrow \Omega$	Given
	$\mathsf{FEV}(\sigma = \tau) = \emptyset$	Given
	$[\Omega]\sigma = [\Gamma]\sigma$	By Lemma 39 (Principal Agreement) (i)
	$[\Omega]\tau = [\Gamma]\tau$	Similar

	$\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \bot$	Given
	$\mathsf{mgu}([\Gamma]\sigma, [\Gamma]\tau) = \bot$	By above equalities
	$\Gamma \;/\; \sigma \mathrel{\dot{=}} \tau : \kappa \dashv \bot$	By Lemma 92 (Completeness of Elimeq) (2)
☞	$\Gamma \;/\; [\Gamma]\sigma = [\Gamma]\tau \vdash \vec{\rho} \Rightarrow e :: [\Gamma]\vec{A} \Leftarrow [\Gamma]C\ p \dashv \Gamma$	By Match$\bot$
☞	$\Omega \longrightarrow \Omega$	By Lemma 32 (Extension Reflexivity)
☞	$\mathsf{dom}(\Gamma) = \mathsf{dom}(\Omega)$	Given

- **Case** $$\dfrac{\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \theta \qquad \theta([\Omega]\Gamma) \vdash \theta(\vec{\rho} \Rightarrow [\Omega]e) :: \theta([\Omega]\vec{A}) \Leftarrow \theta([\Omega]C)\ p}{[\Omega]\Gamma \;/\; [\Omega]\sigma = [\Omega]\tau \vdash \vec{\rho} \Rightarrow [\Omega]e :: [\Omega]\vec{A} \Leftarrow [\Omega]C\ p}\ \mathsf{DeclMatchUnify}$$

$([\Omega]\sigma = [\Gamma]\sigma)$ and $([\Omega]\tau = [\Gamma]\tau)$ — As in DeclMatch⊥ case
$\mathsf{mgu}([\Omega]\sigma, [\Omega]\tau) = \theta$ — Given
$\mathsf{mgu}([\Gamma]\sigma, [\Gamma]\tau) = \theta$ — By above equalities

$\Gamma \ / \ \sigma \mathrel{\overset{\circ}{=}} \tau : \kappa \dashv (\Gamma, \Theta)$ — By Lemma 92 (Completeness of Elimeq) (1)
$\Theta = (\alpha_1 = t_1, \ldots, \alpha_n = t_n)$ — ″
$[\Gamma, \Theta]u = \theta([\Gamma]u)$ — ″ for all $\Gamma \vdash u : \kappa$

$\theta([\Omega]\Gamma) \vdash \theta(\vec{\rho} \Rightarrow [\Omega]e) :: \theta([\Omega]\vec{A}) \Leftarrow \theta([\Omega]C)\ p$ — Subderivation

$\theta([\Omega]\Gamma) = [\Omega, \blacktriangleright_P, \Theta](\Gamma, \blacktriangleright_P, \Theta)$ — By Lemma 93 (Substitution Upgrade) (iii)
$\theta([\Omega]\vec{A}) = [\Omega, \blacktriangleright_P, \Theta]\vec{A}$ — By Lemma 93 (Substitution Upgrade) (i) (over $\vec{A}$)
$\theta([\Omega]C) = [\Omega, \blacktriangleright_P, \Theta]C$ — By Lemma 93 (Substitution Upgrade) (i)
$\theta(\vec{\rho} \Rightarrow [\Omega]e) = [\Omega, \blacktriangleright_P, \Theta](\vec{\rho} \Rightarrow e)$ — By Lemma 93 (Substitution Upgrade) (iv)

$[\Omega, \blacktriangleright_P, \Theta](\Gamma, \blacktriangleright_P, \Theta) \vdash [\Omega, \blacktriangleright_P, \Theta](\vec{\rho} \Rightarrow e) :: [\Omega, \blacktriangleright_P, \Theta]\vec{A} \Leftarrow [\Omega, \blacktriangleright_P, \Theta]C\ p$ — By above equalities

$\Gamma, \blacktriangleright_P, \Theta \vdash (\vec{\rho} \Rightarrow e) :: [\Gamma, \blacktriangleright_P, \Theta]\vec{A} \Leftarrow [\Gamma, \blacktriangleright_P, \Theta]C\ p \dashv \Delta, \blacktriangleright_P, \Delta'$ — By i.h.
$\Delta, \blacktriangleright_P, \Delta' \longrightarrow \Omega', \blacktriangleright_P, \Omega''$ — ″
$\Omega, \blacktriangleright_P, \Theta \longrightarrow \Omega', \blacktriangleright_P, \Omega''$ — ″
$\mathsf{dom}(\Delta, \blacktriangleright_P, \Delta') = \mathsf{dom}(\Omega', \blacktriangleright_P, \Omega'')$ — ″

☞ $\Delta \longrightarrow \Omega'$ — By Lemma 22 (Extension Inversion) (ii)
☞ $\mathsf{dom}(\Delta) = \mathsf{dom}(\Omega')$ — ″
☞ $\Omega \longrightarrow \Omega'$ — By Lemma 22 (Extension Inversion) (ii)

☞ $\Gamma \ / \ [\Gamma]\sigma = [\Gamma]\tau \vdash \vec{\rho} \Rightarrow e :: [\Gamma]\vec{A} \Leftarrow [\Gamma]C\ p \dashv \Delta$ — By MatchUnify ☐