A EXAMPLES IN $F^e_{\sqcup\sqcap}$

Table 1. Examples used in the comparison of TypeScript with $F^e_{\sqcup\sqcap}$, where **V** represents the variant with intersection and union type inference. Record extension is enabled by default in both systems.

$F^e_{\sqcup\sqcap}$ **program**	$F^e_{\sqcup\sqcap}$	**V**
$\mathbf{let}\ \text{f1} : (\text{Label}\ m \to \text{Int}) \to \text{Int} = \lambda o.\ o.m\ \mathbf{in}$ $\mathbf{let}\ \text{o1} = \langle m \mapsto 1, \langle n \mapsto \text{True}, \langle\rangle\rangle\rangle\ \mathbf{in}\ \text{f1 o1 -- ex1\_1}$	✓	✓
$\mathbf{let}\ \text{g1} : (\text{Label}\ n \to \text{Bool}) \to \text{Bool} = \lambda o.\ o.n\ \mathbf{in}$ $\mathbf{let}\ \text{o1} = \langle m \mapsto 1, \langle n \mapsto \text{True}, \langle\rangle\rangle\rangle\ \mathbf{in}\ \text{g1 o1 -- ex1\_2}$	✓	✓
$\mathbf{let}\ \text{h1} : ((\text{Label}\ m \to \text{Int}) \sqcap (\text{Label}\ n \to \text{Int})) \sqcup$ $((\text{Label}\ k \to \text{Int} \to \text{Int}) \sqcap (\text{Label}\ m \to \text{Int})) = \lambda o.\ o.m\ \mathbf{in}\ \ldots$	✓	✓
$\mathbf{let}\ \text{f2} : ((\text{Int} \to \text{Int}) \sqcap (\text{Bool} \to \text{Bool})) \to \text{Bool} = \lambda g.\ g\ \text{True}\ \mathbf{in}\ \ldots$	✓	✓
$(\lambda f.\ f\ 1\ 2) : ((\text{Int} \to \text{Int} \to \text{Int}) \sqcap (\text{Int} \to \text{Bool} \to \text{Int})) \to \text{Int}\ \text{-- f3\_1}$	✓	✓
$(\lambda f.\ f\ 1\ \text{True}) : ((\text{Int} \to \text{Int} \to \text{Int}) \sqcap (\text{Int} \to \text{Bool} \to \text{Int})) \to \text{Int}\ \text{-- f3\_2}$	✓	✓
$\mathbf{let}\ \text{f4} = \Lambda a.\ (\lambda x.\ x) : a \to a\ \mathbf{in}\ \text{f4}\ 1\ \text{-- ex4\_1}$	✓	✓
$\mathbf{let}\ \text{f4} = \Lambda a.\ (\lambda x.\ x) : a \to a\ \mathbf{in}\ \text{f4 @Int}\ 1\ \text{-- ex4\_2}$	✓	✓
$\mathbf{let}\ \text{f5} = \Lambda a.\ (\lambda x.\ \lambda y.\ x) : a \to a \to a\ \mathbf{in}\ \text{f5}\ 1\ \text{True -- ex5\_1}$	✗	✗
$\mathbf{let}\ \text{f5} = \Lambda a.\ (\lambda x.\ \lambda y.\ x) : a \to a \to a\ \mathbf{in}\ \text{f5 @}(\text{Bool} \sqcup \text{Int})\ 1\ \text{True -- ex5\_2}$	✓	✓
$\mathbf{let}\ \text{g5} = \Lambda a.\ (\lambda x.\ \lambda y.\ \lambda z.\ 1) : (a \to \text{Int}) \to (a \to \text{Int}) \to a \to \text{Int}\ \mathbf{in}$ $\text{g5}\ ((\lambda x.\ 1) : \text{Int} \to \text{Int})\ ((\lambda y.\ 2) : \text{Bool} \to \text{Int})\ \text{-- ex5\_3}$	✗	✗
$\text{g5 @}(\text{Int} \sqcap \text{Bool})\ ((\lambda x.\ 1) : \text{Int} \to \text{Int})\ ((\lambda y.\ 2) : \text{Bool} \to \text{Int})\ \text{-- ex5\_4}$	✓	✓
$\mathbf{let}\ \text{f6} = \Lambda a.\ (\lambda x.\ 2) : (a \to \text{Int}) \to \text{Int}\ \mathbf{in}$ $\mathbf{let}\ \text{g6} = (\lambda f.\ 1) : (\text{Int} \to \text{Int}) \to \text{Int}\ \mathbf{in}\ \text{f6 g6 -- ex6}$	✓	✓
$\mathbf{let}\ \text{f7} = \Lambda a.\ (\lambda x.\ x) : a \to a\ \mathbf{in}\ \text{f7 @}(\forall a.\ a \to a)\ \text{f7 -- ex7\_1}$	✓	✓
$\mathbf{let}\ \text{f7} = \Lambda a.\ (\lambda x.\ x) : a \to a\ \mathbf{in}\ \text{f7 @}(\forall a.\ (a \sqcap a) \to (a \sqcup a))\ \text{f7 -- ex7\_2}$	✓	✓
$\mathbf{let}\ \text{f8} : ((\forall a.\ a \to a \to \text{Int}) \to \text{Int}) \to \text{Int} = \lambda x.\ 1\ \mathbf{in}$ $\mathbf{let}\ \text{g8\_1} : (\text{Int} \to (\text{Int} \sqcap \text{Bool}) \to \text{Int}) \to \text{Int} = \lambda f.\ 1\ \mathbf{in}\ \text{f8 g8\_1 -- ex8\_1}$	✓	✓
$\mathbf{let}\ \text{g8\_2} : ((\text{Int} \sqcup \text{Bool}) \to \text{Int} \to \text{Int}) \to \text{Int} = \lambda f.\ 1\ \mathbf{in}\ \text{f8 g8\_2 -- ex8\_2}$	✗	✗
$\mathbf{let}\ \text{g8\_3} : ((\text{Int} \sqcap \text{Bool}) \to \text{Int} \to \text{Int}) \to \text{Int} = \lambda f.\ 1\ \mathbf{in}\ \text{f8 g8\_3 -- ex8\_3}$	✓	✓
$\mathbf{let}\ \text{g8\_4} : (\text{Int} \to (\text{Int} \sqcup \text{Bool}) \to \text{Int}) \to \text{Int} = \lambda f.\ 1\ \mathbf{in}\ \text{f8 g8\_4 -- ex8\_4}$	✗	✓
$\mathbf{let}\ \text{f9} : (\forall a.\ \text{Int} \to a \to \text{Int}) \to \text{Bool} \to \text{Int} = \lambda k.\ k\ 3\ \mathbf{in}$ $\mathbf{let}\ \text{h9} : (\forall a.\ \forall b.\ b \to a \to b) \to \text{Bool} \to \text{Int} = \lambda k.\ \text{f9}\ k\ \mathbf{in}\ \ldots$	✓	✓
$\mathbf{let}\ \text{g9} : (\forall a.\ \text{Int} \to a \to \text{Int}) \to \text{Bool} \to \text{Int} = \lambda k.\ k\ \text{@Bool}\ 3\ \mathbf{in}$ $(\lambda k.\ \text{g9}\ k) : (\forall a.\ \forall b.\ b \to a \to b) \to \text{Bool} \to \text{Int}\ \text{-- ex9\_1}$	✓	✓
$(\lambda k.\ k\ \text{@Bool}\ 3) : (\forall a.\ \forall b.\ b \to a \to b) \to \text{Bool} \to \text{Int}\ \text{-- ex9\_2}$	✓	✓
$\mathbf{let}\ \text{f10} : (\text{Int} \to \text{Int}) \to \text{Int} = \lambda x.\ 1\ \mathbf{in}$ $\mathbf{let}\ \text{h10} = (\lambda x.\ \lambda y.\ y)\ 1\ \mathbf{in}\ \text{f10 h10 -- ex10}$	✓	✓
$\mathbf{let}\ \text{f11} : ((\text{Int} \to \text{Int}) \sqcap (\text{Bool} \to \text{Bool})) \to \text{Int} =$ $\lambda g.\ g\ 1\ \mathbf{in}\ \text{f11}\ (\lambda x.\ x)\ \text{-- ex11}$	✓	✓
$(\lambda x.\ 1) : (\forall a.\ a) \to \text{Int}\ \text{-- ex12\_1}$	✓	✓
$(\lambda x.\ 1) : (\forall a.\ (a \to \text{Bool}) \sqcap (a \to \text{Int})) \to \text{Int}\ \text{-- ex12\_2}$	✓	✓
$(\lambda x.\ x.m)\ \langle m \mapsto 1, \langle\rangle\rangle\ \text{-- ex13}$	✓	✓
$\mathbf{let}\ \text{g14} : (\text{Int} \sqcup \text{Bool}) \to (\text{Int} \sqcup \text{Bool}) \to \text{Int} = \Lambda a.\ \lambda x.\ \lambda y.\ 1\ \mathbf{in}$ $\mathbf{let}\ \text{h14\_1} : \text{Int} \to \text{Bool} \to \text{Int} = \text{g14}\ \mathbf{in}\ \ldots$	✗	✓
$\mathbf{let}\ \text{h14\_2} : \text{Int} \to \text{Bool} \to \text{Int} = \Lambda a.\ (\lambda x.\ \lambda y.\ 1) : a \to a \to \text{Int}\ \mathbf{in}\ \ldots$	✗	✗
$\mathbf{let}\ \text{f15} : ((\forall a.\ a \to a) \to \text{Int}) \to \text{Int} = \lambda x.\ 1\ \mathbf{in}$ $\mathbf{let}\ \text{h15} : (\forall a.\ (a \to a) \to (a \to a)) \to \text{Int} = \lambda x.\ 1\ \mathbf{in}\ \text{f15 h15 -- ex15}$	✗	✗

We include a Haskell prototype implementation in the supplementary materials, which implements all the algorithmic rules in the paper. Additionally, to make the examples more interesting, we have implemented a few extensions. These include polymorphic lists ($[a]$), a case analysis expression for pattern matching on lists, recursion via a fixpoint operator, and recursive let expressions.

B FULL SET OF RULES

B.1 Well-formedness Definitions

$\boxed{\Psi \vdash A}$ $\quad$ A is well-formed in Ψ

$$\frac{}{\Psi \vdash \mathbb{1}}\ \mathtt{wf_d unit} \qquad \frac{}{\Psi \vdash \top}\ \mathtt{wf_d}\top \qquad \frac{}{\Psi \vdash \bot}\ \mathtt{wf_d}\bot \qquad \frac{a \in \Psi \lor \tilde{a} \in \Psi}{\Psi \vdash a}\ \mathtt{wf_d var}$$

$$\frac{\Psi \vdash A \quad \Psi \vdash B}{\Psi \vdash A \to B}\ \mathtt{wf_d}{\to} \qquad \frac{\Psi, a \vdash A \quad a \in^s A}{\Psi \vdash \forall a.\, A}\ \mathtt{wf_d}\forall \qquad \frac{\Psi \vdash A \quad \Psi \vdash B}{\Psi \vdash A \sqcap B}\ \mathtt{wf_d}\sqcap \qquad \frac{\Psi \vdash A \quad \Psi \vdash B}{\Psi \vdash A \sqcup B}\ \mathtt{wf_d}\sqcup$$

$\boxed{\Psi \vdash e}$ $\quad$ e is well-formed in Ψ

$$\frac{x : A \in \Psi}{\Psi \vdash x}\ \mathtt{wf_d tmvar} \qquad \frac{}{\Psi \vdash ()}\ \mathtt{wf_d tmunit} \qquad \frac{\Psi, x : A \vdash e}{\Psi \vdash \lambda x.\, e}\ \mathtt{wf_d abs} \qquad \frac{\Psi \vdash e_1 \quad \Psi \vdash e_2}{\Psi \vdash e_1\ e_2}\ \mathtt{wf_d app}$$

$$\frac{\Psi \vdash A \quad \Psi \vdash e}{\Psi \vdash (e : A)}\ \mathtt{wf_d anno} \qquad \frac{\Psi \vdash A \quad \Psi \vdash e}{\Psi \vdash e\ @A}\ \mathtt{wf_d tApp} \qquad \frac{\Psi, a \vdash A \quad \Psi, a \vdash e \quad a \in^s A}{\Psi \vdash \Lambda a.\, e : A}\ \mathtt{wf_d tLam}$$

$\boxed{\vdash_{\mathtt{t}} \Psi}$ $\quad$ Ψ is a well-formed typing context

$$\frac{}{\vdash_{\mathtt{t}} \cdot}\ \mathtt{wfe}_{\cdot} \qquad \frac{\vdash_{\mathtt{t}} \Psi}{\vdash_{\mathtt{t}} \Psi, a}\ \mathtt{wfe_a} \qquad \frac{\vdash_{\mathtt{t}} \Psi \quad \Psi \vdash A}{\vdash_{\mathtt{t}} \Psi, x : A}\ \mathtt{wfe_{of}}$$

$\boxed{\vdash \Psi}$ $\quad$ Ψ is a well-formed subtyping context

$$\frac{\vdash_{\mathtt{t}} \Psi}{\vdash \Psi}\ \mathtt{wfe_t} \qquad \frac{\vdash \Psi}{\vdash \Psi, \tilde{a}}\ \mathtt{wfe}_{\tilde{a}}$$

Fig. S1. Well-Formedness for the Bidirectional Type System

The well-formedness of the bidirectional type system is presented in Figure S1. The rules of type and expression well-formedness are mostly standard, except the free variable restriction in $\mathtt{wf_d}\forall$ and $\mathtt{wf_d tLam}$. For context well-formedness, we adopt the stronger invariants proposed by Cui et al. [2023] to distinguish the subtyping contexts from the typing context: no subtype variables should appear in the typing context, and no variable and type variable should appear after any subtype variable. However, we do not build a new syntax category for the typing context. We simply choose to employ to set of well-formedness rules:$\vdash_{\mathtt{t}} \Psi$ and $\vdash \Psi$ for the bidirectional context. However, we find this distinction not very helpful and it even complicates the reasoning and weakens the conclusion for certain properties (as the properties are now proved for a more restricted context).

The well-formedness of the algorithmic systems is presented in Figure S3. We also have a syntactic category for algorithmic context. It can be obtained from the work erasure function $\lfloor \Gamma \rfloor$ in Figure S2 removes all works in the worklist. The rules for type and expression well-formedness are defined on the algorithmic context. They are similar to the bidirectional counterparts. The algorithmic system has additional rules for the well-formedness of continuations, works, and worklists.

$\boxed{\lfloor \Gamma \rfloor = \Sigma}$ erase all works in Γ to get Σ

$$\begin{aligned}
\lfloor \cdot \rfloor &= \cdot \\
\lfloor \Gamma, a \rfloor &= \lfloor \Gamma \rfloor \\
\lfloor \Gamma, \tilde{a} \rfloor &= \lfloor \Gamma \rfloor \\
\lfloor \Gamma, \widehat{a} \rfloor &= \lfloor \Gamma \rfloor \\
\lfloor \Gamma, x : A \rfloor &= \lfloor \Gamma \rfloor, x : A \\
\lfloor \Gamma, w \rfloor &= \lfloor \Gamma \rfloor
\end{aligned}$$

Fig. S2. Work erasurement Function

$\boxed{\vdash \Sigma}$ Σ is a well-formed algorithmic context

$$\frac{}{\vdash \cdot}\ \texttt{wfe\_}\cdot \qquad \frac{\vdash \Sigma}{\vdash \Sigma, a}\ \texttt{wfe\_a} \qquad \frac{\vdash \Sigma}{\vdash \Sigma, \tilde{a}}\ \texttt{wfe\_}\tilde{\texttt{a}} \qquad \frac{\vdash \Sigma}{\vdash \Sigma, \widehat{a}}\ \texttt{wfe\_}\widehat{\texttt{a}} \qquad \frac{\vdash \Sigma \quad \Sigma \vdash A}{\vdash \Sigma, x : A}\ \texttt{wfe\_of}$$

Fig. S3. Well-Formedness for the Algorithmic Context

$\boxed{\vdash_{\texttt{t}} \Gamma}$ Γ is a well-formed typing worklist

$$\frac{}{\vdash_{\texttt{t}} \cdot}\ \texttt{wft}\cdot \qquad \frac{\vdash_{\texttt{t}} \Gamma}{\vdash_{\texttt{t}} \Gamma, a}\ \texttt{wft}_{\texttt{a}} \qquad \frac{\vdash_{\texttt{t}} \Gamma}{\vdash_{\texttt{t}} \Gamma, \widehat{a}}\ \texttt{wft}_{\widehat{\texttt{a}}} \qquad \frac{\vdash_{\texttt{t}} \Gamma \quad \lfloor \Gamma \rfloor \vdash_{\texttt{t}} A}{\vdash_{\texttt{t}} \Gamma, x : A}\ \texttt{wft}_{\texttt{of}}$$

$$\frac{\vdash_{\texttt{t}} \Gamma \quad \lfloor \Gamma \rfloor \vdash_{\texttt{t}} w \quad w \neq * \leq *}{\vdash_{\texttt{t}} \Gamma \Vdash w}\ \texttt{wft}_{\texttt{w}}$$

$\boxed{\vdash \Gamma}$ Γ is a well-formed subtyping worklist

$$\frac{\vdash_{\texttt{t}} \Gamma}{\vdash \Gamma}\ \texttt{wf}_{\texttt{t}} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \tilde{a}}\ \texttt{wf}_{\tilde{\texttt{a}}} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \widehat{a}}\ \texttt{wf}_{\widehat{\texttt{a}}} \qquad \frac{\vdash \Gamma \quad \lfloor \Gamma \rfloor \vdash A \quad \lfloor \Gamma \rfloor \vdash B}{\vdash \Gamma \Vdash A \leq B}\ \texttt{wf}_{\leq}$$

Fig. S4. Well-Formedness of Algorithmic Worklist

Due to the invariance imposed for the bidirectional type system to distinguish the typing and subtyping context, the algorithmic system must have that as well. This restriction is defined on algorithmic worklists directly, as shown in Figure S4: a typing worklist ($\vdash_{\texttt{t}} \Gamma$) is dedicated to typing judgments, while the latter part ($\vdash \Gamma$) is for subtyping judgments. Variables and type variables are restricted to appearing only in the typing part, whereas subtyping judgments and subtype variables are confined to the subtyping part.

$\boxed{\Sigma \vdash A}$ A is well-formed in Σ

$$\frac{}{\Sigma \vdash \mathbb{1}}\ \texttt{wf\_unit} \qquad \frac{}{\Sigma \vdash \top}\ \texttt{wf\_}\top \qquad \frac{}{\Sigma \vdash \bot}\ \texttt{wf\_}\bot \qquad \frac{a \in \Sigma \vee \tilde{a} \in \Sigma \vee \widehat{a} \in \Sigma}{\Sigma \vdash a}\ \texttt{wf\_var}$$

$$\frac{\Sigma \vdash A \quad \Sigma \vdash B}{\Sigma \vdash A \to B}\ \texttt{wf\_}\to \qquad \frac{\Sigma, a \vdash A \quad a \in^s A}{\Sigma \vdash \forall a.\, A}\ \texttt{wf\_}\forall \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash B}{\Sigma \vdash A \sqcap B}\ \texttt{wf\_}\sqcap \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash B}{\Sigma \vdash A \sqcup B}\ \texttt{wf\_}\sqcup$$

$\boxed{\Sigma \vdash e}$ e is well-formed in Σ

$$\frac{x : A \in \Sigma}{\Sigma \vdash x}\ \texttt{wf\_tmvar} \qquad \frac{}{\Sigma \vdash ()}\ \texttt{wf\_tmunit} \qquad \frac{\Sigma, x : A \vdash e}{\Sigma \vdash \lambda x.\, e}\ \texttt{wf\_abs} \qquad \frac{\Sigma \vdash e_1 \quad \Sigma \vdash e_2}{\Sigma \vdash e_1\ e_2}\ \texttt{wf\_app}$$

$$\frac{\Sigma \vdash A \quad \Sigma \vdash e}{\Sigma \vdash (e : A)}\ \texttt{wf\_anno} \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash e}{\Sigma \vdash e\ @A}\ \texttt{wf\_tApp} \qquad \frac{\Sigma, a \vdash A \quad \Sigma, a \vdash e \quad a \in^s A}{\Sigma \vdash \Lambda a.\, e : A}\ \texttt{wf\_tLam}$$

$\boxed{\Sigma \vdash \omega^s, \Sigma \vdash \omega^d}$ ω^s, ω^d is well-formed in Σ

$$\frac{\Sigma \vdash A}{\Sigma \vdash \_ \leq A}\ \texttt{wf}^{\texttt{s}} \leq \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash \omega^s}{\Sigma \vdash \_ \circ A \Rrightarrow \omega^s}\ \texttt{wf}^{\texttt{s}}\circ \qquad \frac{\Sigma \vdash \omega^d}{\Sigma \vdash \_ \triangleright \omega^d}\ \texttt{wf}^{\texttt{s}}\triangleright$$

$$\frac{\Sigma \vdash A \quad \Sigma \vdash B \quad \Sigma \vdash \omega^s}{\Sigma \vdash \_ \Cup A \circ B \Rrightarrow \omega^s}\ \texttt{wf}^{\texttt{s}} \Cup\circ \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash \omega^s}{\Sigma \vdash \_ \Cup_\circ A \blacktriangleright \omega^s}\ \texttt{wf}^{\texttt{s}}\Cup_\circ \qquad \frac{\Sigma \vdash e \quad \Sigma \vdash \omega^s}{\Sigma \vdash \_ \to \_ \bullet e \Rrightarrow_\alpha \omega^s}\ \texttt{wf}^{\texttt{d}}\bullet$$

$$\frac{\Sigma \vdash A \quad \Sigma \vdash \omega^d}{\Sigma \vdash \_ \to \_ \Cup A \triangleright \omega^d}\ \texttt{wf}^{\texttt{d}} \Cup\triangleright \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash B \quad \Sigma \vdash \omega^d}{\Sigma \vdash \_ \to \_ \Cup_\triangleright (A \to B) \blacktriangleright \omega^d}\ \texttt{wf}^{\texttt{d}}\Cup_\triangleright$$

$\boxed{\Sigma \vdash w}$ w is well-formed in Σ

$$\frac{\Sigma \vdash A \quad \Sigma \vdash B}{\Sigma \vdash A \leq B}\ \texttt{wf} \leq \qquad \frac{\Sigma \vdash e \quad \Sigma \vdash A}{\Sigma \vdash e \Leftarrow A}\ \texttt{wf} \Leftarrow \qquad \frac{\Sigma \vdash e \quad \Sigma \vdash \omega^s}{\Sigma \vdash e \Rightarrow \omega^s}\ \texttt{wf} \Rightarrow$$

$$\frac{\Sigma \vdash A \quad \Sigma \vdash B \quad \Sigma \vdash \omega^s}{\Sigma \vdash A \circ B \Rrightarrow \omega^s}\ \texttt{wf}\circ \qquad \frac{\Sigma \vdash A \quad \Sigma \vdash \omega^d}{\Sigma \vdash A \triangleright \omega^d}\ \texttt{wf}\triangleright \qquad \frac{\Sigma \vdash C \quad \Sigma \vdash A \quad \Sigma \vdash B \quad \Sigma \vdash \omega^s}{\Sigma \vdash C \Cup A \circ B \Rrightarrow \omega^s}\ \texttt{wf} \Cup\circ$$

$$\frac{\Sigma \vdash B \quad \Sigma \vdash C \quad \Sigma \vdash A \quad \Sigma \vdash \omega^d}{\Sigma \vdash (B \to C) \Cup A \triangleright \omega^d}\ \texttt{wf} \Cup\triangleright \qquad \frac{\Sigma \vdash A_1 \to B_1 \quad \Sigma \vdash A_2 \to B_2 \quad \Sigma \vdash \omega^d}{\Sigma \vdash (A_1 \to B_1) \Cup_\triangleright (A_2 \to B_2) \blacktriangleright \omega^d}\ \texttt{wf}\Cup_\triangleright$$

$$\frac{\Sigma \vdash A_1 \quad \Sigma \vdash A_2 \quad \Sigma \vdash \omega^s}{\Sigma \vdash A_1 \Cup_\circ A_2 \blacktriangleright \omega^s}\ \texttt{wf}\Cup_\circ$$

B.2 Reduction Rules of Bidirectional Worklist

For soundness and completeness proofs, we employ two sets of reduction rules for the bidirectional worklist system: worklist reduction by delegation $\Omega \longrightarrow \Omega'$ and worklist reduction by continuation-passing $\Omega \longrightarrow_{\omega} \Omega'$. The reduction by delegation delegates the current work to the bidirectional system directly, and pass the result (if any) to the following continuation. Their rules are presented in Figure S5 and Figure S6 (we omit the ω in $\longrightarrow_{\omega}$ in Figure S6). The continuation is represented by the defunctionalized style in Figure S5, and HOAS in Figure S6. It should be easy to recover the defunctionalized version of the worklist-reduction by continuation-passing (which is aligned with the actual formalization) by referencing the algorithmic ones shown in Figure S8 and Figure S9.

$\boxed{\lfloor\Omega\rfloor{=}\Psi}$ erase all works in Ω to get Ψ

$$\begin{aligned}
\lfloor \cdot \rfloor &= \cdot \\
\lfloor \Omega, a \rfloor &= \lfloor \Omega \rfloor, a \\
\lfloor \Omega, \tilde{a} \rfloor &= \lfloor \Omega \rfloor, \tilde{a} \\
\lfloor \Omega, x : A \rfloor &= \lfloor \Omega \rfloor, x : A \\
\lfloor \Omega \Vdash w \rfloor &= \lfloor \Omega \rfloor
\end{aligned}$$

$\boxed{\Omega \longrightarrow \Omega'}$ Ω reduces to Ω'

$$\begin{aligned}
\Omega, a &\longrightarrow \Omega \\
\Omega, \tilde{a} &\longrightarrow \Omega \\
\Omega, x : A &\longrightarrow \Omega \\
\Omega \Vdash A \leq B &\longrightarrow \Omega && \text{when } \lfloor\Omega\rfloor \vdash A \leq B \\
\Omega \Vdash e \Leftarrow A &\longrightarrow \Omega && \text{when } \lfloor\Omega\rfloor \vdash e \Leftarrow A \\
\Omega \Vdash e \Rightarrow_{\alpha} \omega^{s} &\longrightarrow \Omega \Vdash \omega^{s} \diamond A && \text{when } \lfloor\Omega\rfloor \vdash e \Rightarrow A \\
\Omega \Vdash A \triangleright_{\alpha,\beta} \omega^{d} &\longrightarrow \Omega \Vdash \omega^{d} \diamond B \diamond C && \text{when } \lfloor\Omega\rfloor \vdash A \triangleright B \to C \\
\Omega \Vdash A \circ B \Rrightarrow_{\alpha} \omega^{s} &\longrightarrow \Omega \Vdash \omega^{s} \diamond C && \text{when } \lfloor\Omega\rfloor \vdash A \circ B \Rrightarrow C \\
\Omega \Vdash A \to B \bullet e \Rrightarrow_{\alpha} \omega^{s} &\longrightarrow \Omega \Vdash \omega^{s} \diamond B && \text{when } \lfloor\Omega\rfloor \vdash e \Leftarrow A \\
\Omega \Vdash C_1 \Cup A_2 \circ B_2 \Rrightarrow \omega^{s} &\longrightarrow \Omega \Vdash \omega^{s} \diamond (C_1 \sqcup C_2) && \text{when } \lfloor\Omega\rfloor \vdash A_2 \circ B_2 \Rrightarrow C_2 \\
\Omega \Vdash (B_1 \to C_1) \Cup A_2 \triangleright \omega^{d} &\longrightarrow \Omega \Vdash \omega^{d} \diamond (B_1 \sqcap B_2) \diamond (C_1 \sqcup C_2) && \text{when } \lfloor\Omega\rfloor \vdash A_2 \triangleright B_2 \to C_2 \\
\Omega \Vdash A_1 \Cup_{\circ} A_2 \blacktriangleright \omega^{s} &\longrightarrow \Omega \Vdash \omega^{s} \diamond (A_1 \sqcup A_2) \\
\Omega \Vdash (A_1 \to B_1) \Cup_{\triangleright} (A_2 \to B_2) &\longrightarrow \Omega \Vdash \omega^{d} \diamond (A_1 \sqcap A_2) \diamond (B_1 \sqcup C_2)
\end{aligned}$$

Fig. S5. Bidirectional Worklist Reduction by Delegation

The connection between the worklist reduction by delegation and the bidirectional system under empty context is straightforward by its definition (stronger equivalence can be established as well):

Theorem B.1 (Soundness and Completeness of Bidirectional Worklist Reduction by Delegation). *Given $\cdot \vdash e$ and $\cdot \vdash A$, then*

(1) $\cdot \vdash e \Leftarrow A$, *iff* $\cdot \Vdash e \Leftarrow A \longrightarrow^{*} \cdot$.
(2) $\cdot \vdash e \Rightarrow A$, *iff* $\cdot \Vdash e \Rightarrow \_ \leq \top \longrightarrow^{*} \cdot$.

$\boxed{\Omega \longrightarrow \Omega'}$ Ω reduces to Ω'

$$\Omega, a : \Box \longrightarrow_1 \Omega \qquad \Omega, a : \tilde{\Box} \longrightarrow_2 \Omega \qquad \Omega, x : A \longrightarrow_3 \Omega$$

$$\Omega \Vdash \mathbb{1} \le \mathbb{1} \longrightarrow_4 \Omega \qquad \Omega \Vdash a \le a \longrightarrow_5 \Omega \qquad \Omega \Vdash A \le \top \longrightarrow_6 \Omega \qquad \Omega \Vdash \bot \le A \longrightarrow_7 \Omega$$

$$\begin{array}{rll}
\Omega \Vdash A_1 \to A_2 \le B_1 \to B_2 & \longrightarrow_8 \Omega B_1 \le A_1 \Vdash A_2 \le B_2 \Vdash & \\
\Omega \Vdash \forall a.\, A \le B & \longrightarrow_9 \Omega \Vdash [\tau/a]A \le B & \text{when } B^{\neq\forall} \\
\Omega \Vdash \forall a.\, A_1 \le \forall a.\, A_2 & \longrightarrow_{10} \Omega, \tilde{a} \Vdash A_1 \le A_2 & \\
\Omega \Vdash A \le B_2 \sqcap B_2 & \longrightarrow_{11} \Omega \Vdash A \le B_1 \Vdash A \le B_2 & \\
\Omega \Vdash A_1 \sqcap A_2 \le B & \longrightarrow_{12} \Omega \Vdash A_1 \le B & \\
\Omega \Vdash A_1 \sqcap A_2 \le B & \longrightarrow_{13} \Omega \Vdash A_2 \le B & \\
\Omega \Vdash A_1 \sqcup A_2 \le B & \longrightarrow_{14} \Omega \Vdash A_1 \le B \Vdash A_2 \le B & \\
\Omega \Vdash A \le B_1 \sqcup B_2 & \longrightarrow_{15} \Omega \Vdash A \le B_1 & \\
\Omega \Vdash A \le B_1 \sqcup B_2 & \longrightarrow_{16} \Omega \Vdash A \le B_2 & \\
\Omega \Vdash e \Leftarrow B & \longrightarrow_{17} \Omega \Vdash e \Rightarrow_\alpha \alpha \le B & \\
\Omega \Vdash \lambda x.\, e \Leftarrow A \to B & \longrightarrow_{18} \Omega, x : A \Vdash e \Leftarrow B & \\
\Omega \Vdash \lambda x.\, e \Leftarrow \top & \longrightarrow_{19} \Omega, x : \bot \vdash e \Leftarrow \top & \\
\Omega \Vdash e \Leftarrow A_1 \sqcap A_2 & \longrightarrow_{20} \Omega \Vdash e \Leftarrow A_1 \Vdash e \Leftarrow A_2 & \\
\Omega \Vdash e \Leftarrow A_1 \sqcup A_2 & \longrightarrow_{21} \Omega \Vdash e \Leftarrow A_1 & \\
\Omega \Vdash e \Leftarrow A_1 \sqcup A_2 & \longrightarrow_{22} \Omega \Vdash e \Leftarrow A_2 & \\
\Omega \Vdash x \Rightarrow_\alpha \omega & \longrightarrow_{23} \Omega \Vdash [A/a]\omega & \text{when } x : A \in \Omega \\
\Omega \Vdash e : A \Rightarrow_\alpha \omega & \longrightarrow_{24} \Omega \Vdash \omega \diamond A \Vdash e \Leftarrow A & \\
\Omega \Vdash (\Lambda a.\, e : A) \Rightarrow_\alpha \omega & \longrightarrow_{25} \Omega \Vdash \omega \diamond (\forall a.\, A), a : \Box \Vdash e \Leftarrow A & \\
\Omega \Vdash () \Rightarrow_\alpha \omega & \longrightarrow_{26} \Omega \Vdash \omega \diamond \mathbb{1} & \\
\Omega \Vdash e_1\ e_2 \Rightarrow_\alpha \omega & \longrightarrow_{27} \Omega \Vdash e_1 \Rightarrow_\beta (\beta \triangleright_{\gamma_1,\gamma_2} (\gamma_1 \to \gamma_2 \bullet e_2 \Rrightarrow_\alpha \omega)) & \\
\Omega \Vdash e\ @A \Rightarrow_\alpha \omega & \longrightarrow_{28} \Omega \Vdash e \Rightarrow_\beta (\beta \circ A \Rrightarrow_\alpha \omega) & \\
\Omega \Vdash \lambda x.\, e \Rightarrow_\alpha \omega & \longrightarrow_{29} \Omega \Vdash \omega \diamond (\tau_1 \to \tau_2), x : \tau_1 \Vdash e \Leftarrow \tau_2 & \\
\Omega \Vdash \bot \triangleright_{\alpha,\beta} \omega & \longrightarrow_{30} \Omega \Vdash \top \to \bot \triangleright_{\alpha,\beta} \omega & \\
\Omega \Vdash A \to B \triangleright_{\alpha,\beta} \omega & \longrightarrow_{31} \Omega \Vdash \omega \diamond A \diamond B & \\
\Omega \Vdash \forall a.\, A \triangleright_{\alpha,\beta} \omega & \longrightarrow_{32} \Omega \Vdash [\tau/a]A \triangleright_{\alpha,\beta} \omega & \\
\Omega \Vdash A_1 \sqcap A_2 \triangleright_{\alpha,\beta} \omega & \longrightarrow_{33} \Omega \Vdash A_1 \triangleright_{\alpha,\beta} \omega & \\
\Omega \Vdash A_1 \sqcap A_2 \triangleright_{\alpha,\beta} \omega & \longrightarrow_{34} \Omega \Vdash A_2 \triangleright_{\alpha,\beta} \omega & \\
\Omega \Vdash A_1 \sqcup A_2 \triangleright_{\alpha_1,\alpha_2} \omega & \longrightarrow_{35} \Omega \Vdash A_1 \triangleright_{\beta_1,\beta_2} (A_2 \triangleright_{\gamma_1,\gamma_2} (\beta_1 \to \beta_2 \Cup_{\triangleright} \gamma_1 \to \gamma_2 \blacktriangleright_{\alpha_1,\alpha_2} \omega)) & \\
\Omega \Vdash (A_1 \to B_1) \Cup_{\triangleright} (A_2 \to B_2) \blacktriangleright \omega & \longrightarrow_{36} \Omega \Vdash \omega \diamond (A_1 \sqcap A_2) \diamond (B_1 \sqcup B_2) & \\
\Omega \Vdash A_1 \to A_2 \bullet e \Rrightarrow_\alpha \omega & \longrightarrow_{37} \Omega \Vdash \omega \diamond A_2 \Vdash e \Leftarrow A_1 & \\
\Omega \Vdash \forall a.\, A \circ B \Rrightarrow_\alpha \omega & \longrightarrow_{38} \Omega \Vdash \omega \diamond ([B/a]A) & \\
\Omega \Vdash \bot \circ A \Rrightarrow_\alpha \omega & \longrightarrow_{39} \Omega \Vdash \omega \diamond \bot & \\
\Omega \Vdash A_1 \sqcap A_2 \circ B \Rrightarrow_\alpha \omega & \longrightarrow_{40} \Omega \Vdash A_1 \circ B \Rrightarrow_\alpha \omega & \\
\Omega \Vdash A_1 \sqcap A_2 \circ B \Rrightarrow_\alpha \omega & \longrightarrow_{41} \Omega \Vdash A_2 \circ B \Rrightarrow_\alpha \omega & \\
\Omega \Vdash A_1 \sqcup A_2 \circ B \Rrightarrow_\alpha \omega & \longrightarrow_{42} \Omega \Vdash A_1 \circ B \Rrightarrow_\beta (A_2 \circ B \Rrightarrow_\gamma \beta \Cup_{\circ} \gamma \blacktriangleright_\alpha \omega) & \\
\Omega \Vdash A_1 \Cup_{\circ} A_2 \blacktriangleright_\alpha \omega & \longrightarrow_{43} \Omega \Vdash \omega \diamond (A_1 \sqcup A_2) &
\end{array}$$

Fig. S6. Bidirectional Worklist Reduction by Continuation-passing

B.3 Syntax-directed Transfer of Continuation and Work

$\boxed{\theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}$ ω^s is transferred to $\omega^{s\prime}$ under θ

$$\frac{\theta \vDash A \rightsquigarrow A'}{\theta \vDash \_ \leq A \rightsquigarrow \_ \leq A'} \qquad \frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash \_ \circ A \Rrightarrow_\alpha \omega^s \rightsquigarrow \_ \circ A' \Rrightarrow_\alpha \omega^{s\prime}} \qquad \frac{\theta \vDash \omega^s \rightsquigarrow \omega^{d\prime}}{\theta \vDash \_ \triangleright_{\alpha,\beta} \omega^d \rightsquigarrow \_ \triangleright_{\alpha,\beta} \omega^{d\prime}}$$

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash \_ \Cup A \circ B \Rrightarrow \omega^s \rightsquigarrow \_ \Cup A' \circ B' \Rrightarrow \omega^{s\prime}} \qquad \frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash \_ \Cup_\circ A \blacktriangleright \omega^s \rightsquigarrow \_ \Cup_\circ A' \blacktriangleright \omega^{s\prime}}$$

$\boxed{\theta \vDash \omega^d \rightsquigarrow \omega^{d\prime}}$ ω^d is transferred to $\omega^{d\prime}$ under θ

$$\frac{\theta \vDash e \rightsquigarrow e' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash \_ \rightarrow \_ \bullet e \Rrightarrow \omega^s} \qquad \frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash \omega^d \rightsquigarrow \omega^{d\prime}}{\theta \vDash \_ \rightarrow \_ \Cup A \triangleright \omega^d \rightsquigarrow \_ \rightarrow \_ \Cup A' \triangleright \omega^{d\prime}}$$

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B' \quad \theta \vDash \omega^d \rightsquigarrow \omega^{d\prime}}{\theta \vDash \_ \rightarrow \_ \Cup_\triangleright (A \rightarrow B) \blacktriangleright \omega^d \rightsquigarrow \_ \rightarrow \_ \Cup_\triangleright (A' \rightarrow B') \blacktriangleright \omega^{d\prime}}$$

$\boxed{\theta \vDash w \rightsquigarrow w'}$ w^d is transferred to w' under θ

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B'}{\theta \vDash A \leq B \rightsquigarrow A' \leq B'} \qquad \frac{\theta \vDash e \rightsquigarrow e' \quad \theta \vDash A \rightsquigarrow A'}{\theta \vDash e \Leftarrow A \rightsquigarrow e \Leftarrow A'} \qquad \frac{\theta \vDash e \rightsquigarrow e' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash e \Rightarrow \omega^s \rightsquigarrow e' \Rightarrow \omega^{s\prime}}$$

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash A \circ B \Rrightarrow \omega^s \rightsquigarrow A \circ B \Rrightarrow \omega^{s\prime}} \qquad \frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash \omega^d \rightsquigarrow \omega^s d'}{\theta \vDash A \triangleright \omega^d \rightsquigarrow A' \triangleright \omega^{d\prime}}$$

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B' \quad \theta \vDash e \rightsquigarrow e' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash A \rightarrow B \bullet e \Rrightarrow \omega^s \rightsquigarrow A' \rightarrow B' \bullet e' \Rrightarrow \omega^{s\prime}}$$

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B' \quad \theta \vDash C \rightsquigarrow C' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash C \Cup A \circ B \Rrightarrow \omega^s \rightsquigarrow C' \Cup A' \circ B' \Rrightarrow \omega^{s\prime}}$$

$$\frac{\theta \vDash A \rightsquigarrow A' \quad \theta \vDash B \rightsquigarrow B' \quad \theta \vDash C \rightsquigarrow C' \quad \theta \vDash \omega^d \rightsquigarrow \omega^{d\prime}}{\theta \vDash (B \rightarrow C) \Cup A \triangleright \omega^d \rightsquigarrow (B' \rightarrow C') \Cup A' \triangleright \omega^{d\prime}}$$

$$\frac{\theta \vDash A_1 \rightsquigarrow A_1' \quad \theta \vDash A_2 \rightsquigarrow A_2' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash A_1 \Cup_\circ A_2 \blacktriangleright \omega^s \rightsquigarrow A_1' \Cup_\circ A_2' \blacktriangleright \omega^{s\prime}}$$

$$\frac{\theta \vDash e \rightsquigarrow e' \quad \theta \vDash \omega^s \rightsquigarrow \omega^{s\prime}}{\theta \vDash (A_1 \rightarrow B_1) \Cup_\triangleright (A_2 \rightarrow B_2) \blacktriangleright \omega^d \rightsquigarrow (A_1' \rightarrow B_1') \Cup_\triangleright (A_2' \rightarrow B_2') \blacktriangleright \omega^{d\prime}}$$

Fig. S7. Syntax-directed Transfer of Continuation and Work

The syntax-directed transfer rules of continuation and work are shown in Figure S7. There is nothing special about these rules, as every rule is just inductively defined and based on the transfer of expressions and types. The transfer of two types of continuation (ω^s and ω^d) needs a mutually inductive definition.

B.4 Defunctionalizing the Continuation

In Section 4, the continuation is simply described as a function (in the meta-language) that takes types as arguments and returns a work. However, it is usually unpleasant to mix HOAS and locally nameless in the binding structure. In addition, reasoning about functions in Coq would probably require the use of some extra axioms. It also ignores the fact that $F^{e}_{\sqcup\sqcap}$ has a fixed structure for continuation. For example, a continuation that returns a subtyping work can only be $\alpha \leq B$, not $A \leq \beta$ or $\alpha \leq \beta$, according to rules that create such continuations. To capture such properties and simplify the proof, we use the following defunctionalized [Reynolds 1972] style to inductively define the continuation (i.e. turning functions to data types).

There are two main differences compared to the syntax used in the previous sections: (1) the continuation is made into an independent syntax category using inductive definitions; (2) continuations that expect one type and two types are distinguished; (3) _ is used at places that expects an unknown result and meta-argument (α, β) is not used any more. The connection between continuation and work is clear: when holes in a continuation are fully applied, the continuation will become a work. The reduction rules guarantee that the continuation is always fully applied.

$$
\begin{array}{llll}
\text{Work} & w & ::= & A \leq B \mid e \Leftarrow A \mid e \Rightarrow \omega^s \mid A \circ B \Rrightarrow \omega^s \mid A \triangleright \omega^d \\
 & & & A \rightarrow B \bullet e \Rrightarrow \omega^s \mid C \Cup A \circ B \Rrightarrow \omega^s \mid (B \rightarrow C) \Cup A \triangleright \omega^d \\
 & & & (A_1 \rightarrow B_1) \Cup_{\triangleright} (A_2 \rightarrow B_2) \blacktriangleright \omega^d \mid A_1 \Cup_{\circ} A_2 \blacktriangleright \omega^s \\
\text{Continuation} & \omega^s & ::= & \_ \leq A \mid \_ \circ A \Rrightarrow \omega^s \mid \_ \triangleright \omega^d \mid \_ \Cup A \circ B \Rrightarrow \omega^s \mid \_ \Cup_{\circ} A \blacktriangleright \omega^s \\
 & \omega^d & ::= & \_ \rightarrow \_ \bullet e \Rrightarrow \omega^s \mid \_ \rightarrow \_ \Cup A \triangleright \omega^d \mid \_ \rightarrow \_ \Cup_{\triangleright} (A \rightarrow B) \blacktriangleright \omega^d
\end{array}
$$

One limitation of the defunctionalized continuations is that the scoping is more rigid: results can only be passed to the next continuation, instead of arbitrary subsequent continuations. Additional effort has to be taken to represent the argument passed to the subsequent continuation used in rule 41 (γ_1 and γ_2) and 49 (β_2) in Figure 4. This marks the third difference in syntax: (3) new intermediate works and continuations are introduced to cache the intermediate results (those marked gray). New reduction rules are introduced to process these works and continuations. Take the later three type-application rules as an example: rule 4 processes the left branch and relays to the second branch; when the result of the first branch is ready, rule 5 switches the focus to the second branch and relays the result of the first branch; when the results of both branches are ready, rule 6 combines them.

$$
\begin{array}{rl}
\Gamma \Vdash A_1 \sqcup A_2 \triangleright \omega & \longrightarrow_1 \Gamma \Vdash A_1 \triangleright (\_ \rightarrow \_ \Cup A_2 \triangleright \omega) \\
\Gamma \Vdash (B_2 \rightarrow C_2) \Cup A_1 \triangleright \omega & \longrightarrow_2 \Gamma \Vdash A_1 \triangleright (\_ \rightarrow \_ \Cup_{\triangleright} (B_2 \rightarrow C_2) \blacktriangleright \omega) \\
\Gamma \Vdash (A_1 \rightarrow B_1) \Cup_{\triangleright} (A_2 \rightarrow B_2) \blacktriangleright \omega & \longrightarrow_3 \Gamma \Vdash \omega \diamond (A_1 \sqcap A_2) \diamond (B_1 \sqcup B_2) \\
\Gamma \Vdash A_1 \sqcup A_2 \circ B \Rrightarrow \omega & \longrightarrow_4 \Gamma \Vdash A_1 \circ B \Rrightarrow (\_ \Cup A_2 \circ B \Rrightarrow \omega) \\
\Gamma \Vdash C_2 \Cup A_1 \circ B_1 \Rrightarrow \omega & \longrightarrow_5 \Gamma \Vdash A_1 \circ B \Rrightarrow (\_ \Cup_{\circ} C_2 \blacktriangleright \omega) \\
\Gamma \Vdash A_1 \Cup_{\circ} A_2 \blacktriangleright \omega & \longrightarrow_6 \Gamma \Vdash \omega \diamond (A_1 \sqcup A_2)
\end{array}
$$

An example of reduction with such a defunctionalized style for type-application judgment with union types is shown below.

$$
\begin{array}{ll}
\Gamma \Vdash ((\forall a.a) \sqcup \bot) \circ \mathbb{1} \Rrightarrow \omega & \\
\longrightarrow^1 \Gamma \Vdash \forall a.a \circ \mathbb{1} \Rrightarrow (\_ \Cup (\bot \circ \mathbb{1}) \Rrightarrow \omega) & \longrightarrow^4 \Gamma \Vdash \bot \Cup_{\circ} \mathbb{1} \blacktriangleright \omega \\
\longrightarrow^2 \Gamma \Vdash \mathbb{1} \Cup (\bot \circ \mathbb{1}) \Rrightarrow \omega & \longrightarrow^5 \Gamma \Vdash \omega \diamond (\mathbb{1} \sqcup \bot) \\
\longrightarrow^3 \Gamma \Vdash \bot \circ \mathbb{1} \Rrightarrow \_ \Cup_{\circ} \mathbb{1} \blacktriangleright \omega &
\end{array}
$$

The full set of algorithmic reduction rules using the defunctionalized continuation is shown in Figure S8 and Figure S9. We additionally make continuation applications $\omega^s \diamond A$ and $\omega^d \diamond A \diamond B$ as works, and design specific rules (rules 53 and 54) to handle their reduction. The reduction rules themselves are trivial: turning the applied continuation to the corresponding work, as described by the side condition $w^s \diamond A = w$. This relation is total and deterministic and could have been modeled as a function directly instead.

$\boxed{\Gamma \longrightarrow \Gamma'}$ Γ worklist-reduces to Γ'

$$
\begin{array}{rcll}
\Gamma, a & \longrightarrow_1 & \Gamma & \\
\Gamma, \tilde{a} & \longrightarrow_2 & \Gamma & \\
\Gamma, \widehat{a} & \longrightarrow_3 & \Gamma & \\
\Gamma, x : A & \longrightarrow_4 & \Gamma & \\
\Gamma \Vdash \mathbb{1} \leq \mathbb{1} & \longrightarrow_5 & \Gamma & \\
\Gamma \Vdash a \leq a & \longrightarrow_6 & \Gamma & \\
\Gamma \Vdash A \leq \top & \longrightarrow_7 & \Gamma & \\
\Gamma \Vdash \bot \leq A & \longrightarrow_8 & \Gamma & \\
\Gamma \Vdash A_1 \to A_2 \leq B_1 \to B_2 & \longrightarrow_9 & \Gamma \Vdash B_1 \leq A_1 \Vdash A_2 \leq B_2 & \\
\Gamma \Vdash \forall a.\, A \leq B & \longrightarrow_{10} & \Gamma, \widehat{a} \Vdash A \leq B & \text{when } B^{\neq\forall} \\
\Gamma \Vdash \forall a.\, A_1 \leq \forall a.\, A_2 & \longrightarrow_{11} & \Gamma, \tilde{a} \Vdash A_1 \leq A_2 & \\
\Gamma \Vdash A \leq B_2 \sqcap B_2 & \longrightarrow_{12} & \Gamma \Vdash A \leq B_1 \Vdash A \leq B_2 & \\
\Gamma \Vdash A_1 \sqcap A_2 \leq B & \longrightarrow_{13} & \Gamma \Vdash A_1 \leq B & \\
\Gamma \Vdash A_1 \sqcap A_2 \leq B & \longrightarrow_{14} & \Gamma \Vdash A_2 \leq B & \\
\Gamma \Vdash A_1 \sqcup A_2 \leq B & \longrightarrow_{15} & \Gamma \Vdash A_1 \leq B \Vdash A_2 \leq B & \\
\Gamma \Vdash A \leq B_1 \sqcup B_2 & \longrightarrow_{16} & \Gamma \Vdash A \leq B_1 & \\
\Gamma \Vdash A \leq B_1 \sqcup B_2 & \longrightarrow_{17} & \Gamma \Vdash A \leq B_2 & \\
\Gamma \Vdash a \leq \tau & \longrightarrow_{18} & \{\tau/a\}\Gamma & \text{when } \widehat{a} \in \Gamma \wedge a \notin \text{fv}(\tau) \\
\Gamma \Vdash \tau \leq a & \longrightarrow_{19} & \{\tau/a\}\Gamma & \text{when } \widehat{a} \in \Gamma \wedge a \notin \text{fv}(\tau) \\
\Gamma \Vdash a \leq A \to B & \longrightarrow_{20} & \{a_1 \to a_2/a\}(\Gamma, \widehat{a}_1, \widehat{a}_2, a \leq A \to B) & \text{when } \widehat{a} \in \Gamma \wedge \Gamma \nvdash^m (A \to B) \\
\Gamma \Vdash A \to B \leq a & \longrightarrow_{21} & \{a_1 \to a_2/a\}(\Gamma, \widehat{a}_1, \widehat{a}_2, A \to B \leq a) & \text{when } \widehat{a} \in \Gamma \wedge \Gamma \nvdash^m (A \to B) \\
\Gamma \Vdash e \Leftarrow B & \longrightarrow_{22} & \Gamma \Vdash e \Rightarrow \_ \leq B & \\
\Gamma \Vdash \lambda x.\, e \Leftarrow A \to B & \longrightarrow_{23} & \Gamma, x : A \Vdash e \Leftarrow B & \\
\Gamma \Vdash \lambda x.\, e \Leftarrow \top & \longrightarrow_{24} & \Gamma, x : \bot \Vdash e \Leftarrow \top & \\
\Gamma \Vdash e \Leftarrow A_1 \sqcap A_2 & \longrightarrow_{25} & \Gamma \Vdash e \Leftarrow A_1 \Vdash e \Leftarrow A_2 & \\
\Gamma \Vdash e \Leftarrow A_1 \sqcup A_2 & \longrightarrow_{26} & \Gamma \Vdash e \Leftarrow A_1 & \\
\Gamma \Vdash e \Leftarrow A_1 \sqcup A_2 & \longrightarrow_{27} & \Gamma \Vdash e \Leftarrow A_2 & \\
\Gamma \Vdash \lambda x.\, e \Leftarrow a & \longrightarrow_{28} & \{a_1 \to a_2/a\}(\Gamma, \widehat{a}_1, \widehat{a}_2 \Vdash \lambda x.e \Leftarrow a) & \text{when } \widehat{a} \in \Gamma \\
\Gamma \Vdash x \Rightarrow \omega^s & \longrightarrow_{29} & \Gamma \Vdash \omega^s \diamond A & \text{when } x : A \in \Gamma \\
\Gamma \Vdash e : A \Rightarrow \omega^s & \longrightarrow_{30} & \Gamma \Vdash \omega \diamond A \Vdash e \Leftarrow A & \\
\Gamma \Vdash (\Lambda a.\, e : A) \Rightarrow \omega^s & \longrightarrow_{31} & \Gamma \Vdash \omega \diamond (\forall a.\, A), a \Vdash e \Leftarrow A & \\
\Gamma \Vdash () \Rightarrow \omega^s & \longrightarrow_{32} & \Gamma \Vdash \omega \diamond \mathbb{1} &
\end{array}
$$

Fig. S8. Algorithmic Worklist Reduction using Defunctionalized Continuation (Part I)

$$
\begin{aligned}
\Gamma \Vdash e_1\ e_2 \Rightarrow \omega^s &\longrightarrow_{33} \Gamma \Vdash e_1 \Rightarrow (\_ \triangleright (\_{\rightarrow}\_ \bullet e_2 \Rrightarrow \omega^s))\\
\Gamma \Vdash e\ @A \Rightarrow \omega^s &\longrightarrow_{34} \Gamma \Vdash e \Rightarrow (\_ \circ A \Rrightarrow \omega^s)\\
\Gamma \Vdash \lambda x.\ e \Rightarrow \omega^s &\longrightarrow_{35} \Gamma, \widehat{a}_1, \widehat{a}_2 \Vdash \omega \diamond (a_1 \rightarrow a_2), x : a_1 \Vdash e \Leftarrow a_2\\
\Gamma \Vdash A \rightarrow B \triangleright \omega^d &\longrightarrow_{36} \Gamma \Vdash \omega^d \diamond A \diamond B\\
\Gamma \Vdash \bot \triangleright \omega^d &\longrightarrow_{37} \Gamma \Vdash \top \rightarrow \bot \triangleright \omega^d\\
\Gamma \Vdash \forall a.\ A \triangleright \omega^d &\longrightarrow_{38} \Gamma, \widehat{a} \Vdash A \triangleright \omega^d\\
\Gamma \Vdash A_1 \sqcap A_2 \triangleright \omega^d &\longrightarrow_{39} \Gamma \Vdash A_1 \triangleright \omega^d\\
\Gamma \Vdash A_1 \sqcap A_2 \triangleright \omega^d &\longrightarrow_{40} \Gamma \Vdash A_2 \triangleright \omega^d\\
\Gamma \Vdash A_1 \sqcup A_2 \triangleright \omega^d &\longrightarrow_{41} \Gamma \Vdash A_1 \triangleright (\_{\rightarrow}\_ \Cup A_2 \triangleright \omega)\\
\Gamma \Vdash a \triangleright \omega^d &\longrightarrow_{42} \{a_1 \rightarrow a_2/a\}(\Gamma, \widehat{a}_1, \widehat{a}_2 \Vdash a \triangleright \omega^d) \qquad \text{when } \widehat{a} \in \Gamma\\
\Gamma \Vdash (B_2 \rightarrow C_2) \Cup A_1 \triangleright \omega^d &\longrightarrow_{43} \Gamma \Vdash A_1 \triangleright (\_{\rightarrow}\_ \Cup_{\triangleright} B_2 \rightarrow C_2 \blacktriangleright \omega)\\
\Gamma \Vdash (A_1 \rightarrow B_1) \Cup_{\triangleright} (A_2 \rightarrow B_2) \blacktriangleright \omega^d &\longrightarrow_{44} \Gamma \Vdash \omega^d \diamond (A_1 \sqcap A_2) \diamond (B_1 \sqcup B_2)\\
\Gamma \Vdash A \rightarrow B \bullet e \Rrightarrow \omega^s &\longrightarrow_{45} \Gamma \Vdash \omega^s \diamond B \Vdash e \Leftarrow A\\
\Gamma \Vdash \forall a.\ A \circ B \Rrightarrow \omega^s &\longrightarrow_{46} \Gamma \Vdash \omega^s \diamond ([B/a]A)\\
\Gamma \Vdash \bot \circ A \Rrightarrow \omega^s &\longrightarrow_{47} \Gamma \Vdash \omega \diamond \bot\\
\Gamma \Vdash A_1 \sqcap A_2 \circ B \Rrightarrow \omega^s &\longrightarrow_{48} \Gamma \Vdash A_1 \circ B \Rrightarrow \omega^s\\
\Gamma \Vdash A_1 \sqcap A_2 \circ B \Rrightarrow \omega^s &\longrightarrow_{49} \Gamma \Vdash A_2 \circ B \Rrightarrow \omega^s\\
\Gamma \Vdash A_1 \sqcup A_2 \circ B \Rrightarrow \omega^s &\longrightarrow_{50} \Gamma \Vdash A_1 \circ B \Rrightarrow (\_ \Cup A_2 \circ B \Rrightarrow \omega^s)\\
\Gamma \Vdash C_2 \Cup A_1 \circ B_1 \Rrightarrow \omega^s &\longrightarrow_{51} \Gamma \Vdash A_1 \circ B \Rrightarrow (\_ \Cup_{\circ} C_2 \blacktriangleright \omega^s)\\
\Gamma \Vdash A_1 \Cup_{\circ} A_2 \blacktriangleright \omega &\longrightarrow_{52} \Gamma \Vdash \omega \diamond (A_1 \sqcup A_2)\\
\Gamma \Vdash \omega^s \diamond A &\longrightarrow_{53} \Gamma \Vdash w \qquad \text{when } \omega^s \diamond A = w\\
\Gamma \Vdash \omega^d \diamond A \diamond B &\longrightarrow_{54} \Gamma \Vdash w \qquad \text{when } \omega^d \diamond A \diamond B = w
\end{aligned}
$$

Fig. S9. Algorithmic Worklist Reduction using Defunctionalized Continuation (Part II)

B.5 Algorithmic Rules of $F^{e}_{\sqcup\sqcap}$ with Records

The algorithmic system of $F^{e}_{\sqcup\sqcap}$ with records still shares the same type and expression as the bidirectional system. The syntax of work extended with a new work for the inference of record extension: inference intersection ($A \Cap B \blacktriangleright_{\alpha} \omega$) and projection inference ($A \rightarrow B \blacksquare C \Rrightarrow_{\alpha} \omega$). Six new rules are added to the algorithmic reduction.

$$\text{Work} \quad w \quad ::= \quad \cdots \mid A \Cap B \blacktriangleright_{\alpha} \omega \mid A \rightarrow B \blacksquare C \Rrightarrow_{\alpha} \omega$$

$$
\begin{aligned}
\Gamma \Vdash \texttt{Label}\ l \leq \texttt{Label}\ l &\longrightarrow_{1} \Gamma\\
\Gamma \Vdash \langle\rangle \Rightarrow_{\alpha} \omega &\longrightarrow_{2} \Gamma \Vdash \omega \diamond \mathbb{1}\\
\Gamma \Vdash \langle l_1 \mapsto e_1, e_2 \rangle \Rightarrow_{\alpha} \omega &\longrightarrow_{3} \Gamma \Vdash e_1 \Rightarrow_{\beta} (e_2 \Rightarrow_{\gamma} (\texttt{Label}\ l_1{\rightarrow}\beta \Cap \gamma \blacktriangleright_{\alpha} \omega))\\
\Gamma \Vdash e.l \Rightarrow_{\alpha} \omega &\longrightarrow_{4} \Gamma \Vdash e \Rightarrow_{\alpha} (\alpha \triangleright_{\beta,\gamma} (\beta \rightarrow \gamma \blacksquare \texttt{Label}\ l \Rrightarrow_{\alpha} \omega))\\
\Gamma \Vdash A_1 \Cap_{\Rightarrow} A_2 \blacktriangleright \omega &\longrightarrow_{5} \Gamma \Vdash \omega \diamond (A_1 \sqcap A_2)\\
\Gamma \Vdash A \rightarrow B \blacksquare C \Rrightarrow_{\alpha} \omega &\longrightarrow_{6} \Gamma \Vdash \omega \diamond B \Vdash C \leq A
\end{aligned}
$$

Since rule 3 also uses nested continuation. Its defunctionalization requires creating a new type of intermediate work to relay the inference result of e_1, similar to the defunctionalization of the union type case of matching and type application.

C ADDITIONAL DISCUSSION OF $F^e_{\sqcup\sqcap}$

Specification of type application with $\bot$. The specification of type application where $\bot$ could appear in A is shown below. There are two differences in the $\circ \Rrightarrow$ to $\leq$ direction: (1). there is an additional $A \leq \bot$ case; (2). the syntactic equality $[B/a]A' = C$ is replaced by the equivalent relation ($\Psi \vdash C \leq [B/a]A'$ and $\Psi \vdash [B/a]A' \leq C$). Though this specification looks complicated, its meaning is intuitive: if A can be type applied, we can find a bot-like or universal-type–like supertype A' of A, whose type-application result is equivalent to A. The $\leq$ to $\circ \Rrightarrow$ remains unchanged.

Lemma C.1 (Specification of Type Application). *Given all contexts and types well-formed,*

(1) *if $\Psi \vdash A \circ B \Rrightarrow C$, then either*
  (a) exists A', $\Psi \vdash A \leq \forall a.A'$, $\Psi \vdash C \leq [B/a]A'$ and $\Psi \vdash [B/a]A' \leq C$; or
  (b) $\Psi \vdash A \leq \bot$ and $\Psi \vdash C \leq \bot$;
(2) *if $\Psi \vdash A \leq \forall a.\ A'$, then exists C s.t. $\Psi \vdash A \circ B \Rrightarrow C$ and $\Psi' \vdash C \leq [B/a]A'$.*

Greediness. The greediness of the algorithm means it always uses the first monotype seen in the subtyping judgment $a \leq \tau$ or $\tau \leq a$ to solve a. However, the "first" subtyping judgment is an implementation-dependent idea, which may differ from the original position a appears in the universal type. In our current algorithmic rule, $\Gamma \Vdash a \to a \leq A \to B \longrightarrow \Gamma \Vdash A \leq a \Vdash a \leq B$ where the codomain type is solved first. In the base $F^e_{\sqcup\sqcap}$ system, this order makes no difference since solving whichever first leads to the same behavior. However, in the $F^e_{\sqcup\sqcap}$ with instantiable intersection and union types, different reduction rules would lead to different examples being accepted/rejected.

The different behavior between our implementation and TypeScript on `ex8_2` ($\forall a.a \to a \to \texttt{Int} \leq \texttt{Int} \to (\texttt{Int}\sqcup\texttt{Bool}) \to \texttt{Int}$) and `ex8_4` ($\forall a.a \to a \to \texttt{Int} \leq (\texttt{Int}\sqcup\texttt{Bool}) \to \texttt{Int} \to \texttt{Int}$) is due to this: our solving prioritizes the codomain type to solve the instantiation while TypeScript prioritizes the domain type. We believe that changing $\Gamma \Vdash A_1 \to B_1 \leq A_2 \to B_2 \longrightarrow \Gamma \Vdash A_2 \leq A_1 \Vdash B_1 \leq B_2$ to $\Gamma \Vdash A_1 \to B_1 \leq A_2 \to B_2 \longrightarrow \Gamma \Vdash B_1 \leq B_2 \Vdash A_2 \leq A_1$ would fix this inconsistency, and it fits people's intuition on greediness better since the domain type also appears first in the original type.

In our algorithmic rules, $\Gamma \Vdash a \leq \tau \longrightarrow \{\tau/a\}\Gamma$ overlaps with $\Gamma \Vdash A \leq B_1 \sqcup B_2 \longrightarrow \Gamma \Vdash A \leq B_1$ and $\Gamma \Vdash A \leq B_1 \sqcup B_2 \longrightarrow \Gamma \Vdash A \leq B_2$. Similarly, $\Gamma \Vdash \tau \leq a \longrightarrow \{\tau/a\}\Gamma$ overlaps with $\Gamma \Vdash A_1 \sqcap A_2 \leq B \longrightarrow \Gamma \Vdash A_1 \leq B$ and $\Gamma \Vdash A_1 \sqcap A_2 \leq B \longrightarrow \Gamma \Vdash A_2 \leq B$. These overlappings (or backtracking) are essential to guarantee this variant strictly accepts more programs. Otherwise, as the rejection of `ex8_3` by TypeScript demonstrates, a fully greedy implementation would reject programs where an instantiation without intersection and union types exists. This algorithm is still greedy in that it still fully relies on the first instantiation found. $\cdot, \widehat{a} \Vdash a \leq \mathbb{1} \Vdash a \leq \mathbb{1} \sqcap (\mathbb{1} \to \mathbb{1})$ reduces, but $\cdot, \widehat{a} \Vdash a \leq \mathbb{1} \sqcap (\mathbb{1} \to \mathbb{1}) \Vdash a \leq \mathbb{1}$ does not, just because the order of instantiation changes. TypeScript does opt for less backtracking even if it rejects some more programs.

D PROOF DETAILS

D.1 Proof Details of Type Safety

$\boxed{A \hookrightarrow A^F}$ A is translated to A^F

$$\frac{}{\mathbb{1} \hookrightarrow \mathbb{1}}\ \hookrightarrow\mathsf{Unit} \qquad \frac{}{a \hookrightarrow a}\ \hookrightarrow\mathsf{Var} \qquad \frac{}{\top \hookrightarrow \mathbb{1}}\ \hookrightarrow\top \qquad \frac{}{\bot \hookrightarrow \forall a.\, a}\ \hookrightarrow\bot \qquad \frac{A \hookrightarrow A^F \quad B \hookrightarrow B^F}{A \to B \hookrightarrow A^F \to B^F}\ \hookrightarrow\to$$

$$\frac{A \hookrightarrow A^F}{\forall a.\, A \hookrightarrow \forall a.\, A^F}\ \hookrightarrow\forall \qquad \frac{A \hookrightarrow A^F \quad B \hookrightarrow B^F}{A \sqcap B \hookrightarrow A^F \times B^F}\ \hookrightarrow\sqcap \qquad \frac{A \hookrightarrow A^F \quad B \hookrightarrow B^F}{A \sqcup B \hookrightarrow A^F + B^F}\ \hookrightarrow\sqcup$$

$\boxed{\Psi \hookrightarrow \Delta}$ Ψ is translated to Δ

$$\frac{}{\cdot \hookrightarrow \cdot}\ \hookrightarrow\cdot \qquad \frac{\Psi \hookrightarrow \Psi^F \quad A \hookrightarrow A^F}{\Psi, x : A \hookrightarrow \Delta, x : A^F}\ \hookrightarrow\mathsf{x} \qquad \frac{\Psi \hookrightarrow \Psi^F}{\Psi, a \hookrightarrow \Delta, a}\ \hookrightarrow\mathsf{a} \qquad \frac{\Psi \hookrightarrow \Psi^F}{\Psi, \tilde{a} \hookrightarrow \Delta, a}\ \hookrightarrow\tilde{\mathsf{a}}$$

Fig. S10. Type and Context Translation

$\boxed{\Psi \vdash A \leq B \ \hookrightarrow co}$ Subtyping $\Psi \vdash A \leq B$ is elaborated to co

$$\frac{}{\Psi \vdash \mathbb{1} \leq \mathbb{1} \ \hookrightarrow \lambda(x : \mathbb{1}).\, x}\ \leq\mathsf{Unit} \qquad \frac{}{\Psi \vdash A \leq \top \ \hookrightarrow \lambda(x : |A|).\, ()}\ \leq\top \qquad \frac{}{\Psi \vdash \bot \leq A \ \hookrightarrow \lambda(x : \forall a.\, a).\, x\ @|A|}\ \leq\bot$$

$$\frac{}{\Psi \vdash a \leq a \ \hookrightarrow \lambda(x : a).\, x}\ \leq\mathsf{Var} \qquad \frac{\Psi \vdash B_1 \leq A_1 \ \hookrightarrow co_1 \quad \Psi \vdash A_2 \leq B_2 \ \hookrightarrow co_2}{\Psi \vdash A_1 \to A_2 \leq B_1 \to B_2 \ \hookrightarrow \lambda(f : |A_1 \to A_2|).\, \lambda(x : |B_1|).\, co_2\ (f\ (co_1\ x))}\ \leq\to$$

$$\frac{\Psi \vdash [\tau/a]A \leq B \ \hookrightarrow co_1 \quad B^{\neq\forall}}{\Psi \vdash \forall a.\, A \leq B \ \hookrightarrow \lambda(x : |\forall a.\, A|).\, co_1\ (x\ @|\tau|)}\ \leq\forall\mathsf{L} \qquad \frac{\Psi, \tilde{a} \vdash A_1 \leq A_2 \ \hookrightarrow co_1}{\Psi \vdash \forall a.\, A_1 \leq \forall a.\, A_2 \ \hookrightarrow \lambda(x : |\forall a.\, A_1|).\, \Lambda a.\, co_1\ (x\ @a)}\ \leq\forall$$

$$\frac{\Psi \vdash A \leq B_1 \ \hookrightarrow co_1 \quad \Psi \vdash A \leq B_2 \ \hookrightarrow co_2}{\Psi \vdash A \leq B_1 \sqcap B_2 \ \hookrightarrow \lambda(x : |A|).\, \langle co_1\ x, co_2\ x\rangle}\ \leq\sqcap\mathsf{R}$$

$$\frac{\Psi \vdash A_1 \leq B \ \hookrightarrow co_1}{\Psi \vdash A_1 \sqcap A_2 \leq B \ \hookrightarrow \lambda(x : |A_1 \sqcap A_2|).\, co_1\ (\pi_1\ x)}\ \leq\sqcap\mathsf{L}_1 \qquad \frac{\Psi \vdash A_2 \leq B \ \hookrightarrow co_1}{\Psi \vdash A_1 \sqcap A_2 \leq B \ \hookrightarrow \lambda(x : |A_1 \sqcap A_2|).\, co_1\ (\pi_2\ x)}\ \leq\sqcap\mathsf{L}_2$$

$$\frac{\Psi \vdash A_1 \leq B \ \hookrightarrow co_1 \quad \Psi \vdash A_2 \leq B \ \hookrightarrow co_2}{\Psi \vdash A_1 \sqcup A_2 \leq B \ \hookrightarrow \lambda(x : |A_1 \sqcup A_2|).\, \mathtt{case}\ \ x\ \ \mathtt{of}\ \ y \mapsto co_1\ y; z \mapsto co_2\ z}\ \leq\sqcup\mathsf{L}$$

$$\frac{\Psi \vdash A \leq B_1 \ \hookrightarrow co_1}{\Psi \vdash A \leq B_1 \sqcup B_2 \ \hookrightarrow \lambda(x : |A|).\, \mathtt{inl}\ (co_1\ x)}\ \leq\sqcup\mathsf{R}_1 \qquad \frac{\Psi \vdash A \leq B_2 \ \hookrightarrow co_1}{\Psi \vdash A \leq B_1 \sqcup B_2 \ \hookrightarrow \lambda(x : |A|).\, \mathtt{inr}\ (co_2\ x)}\ \leq\sqcup\mathsf{R}_2$$

Fig. S11. Elaboration of Bidirectional Subtyping

$\boxed{\Psi \vdash e \Leftrightarrow A \hookrightarrow e'}$ Checking and Inference is elaborated to e'

$$\frac{\Psi, x : A \vdash e \Leftarrow B \hookrightarrow e' \qquad \Psi \vdash A}{\Psi \vdash \lambda x.\, e \Leftarrow A \to B \hookrightarrow \lambda(x : |A|).\, e'}\ \Leftarrow\!\to \qquad \frac{\Psi, x : \bot \vdash e \Leftarrow \top \hookrightarrow *}{\Psi \vdash \lambda x.\, e \Leftarrow \top \hookrightarrow ()}\ \Leftarrow\!\to\!\top$$

$$\frac{\Psi \vdash e \Rightarrow A \hookrightarrow e' \qquad \Psi \vdash A \le B \hookrightarrow co}{\Psi \vdash e \Leftarrow B \hookrightarrow co\ E'}\ \Leftarrow\text{Sub} \qquad \frac{\Psi \vdash e \Leftarrow A_1 \hookrightarrow e_1' \qquad \Psi \vdash e \Leftarrow A_2 \hookrightarrow e_2'}{\Psi \vdash e \Leftarrow A_1 \sqcap A_2 \hookrightarrow \langle e_1', e_2' \rangle}\ \Leftarrow\sqcap$$

$$\frac{\Psi \vdash e \Leftarrow A_1 \hookrightarrow e_1'}{\Psi \vdash e \Leftarrow A_1 \sqcup A_2 \hookrightarrow \texttt{inl}\ e_1'}\ \Leftarrow\sqcup_1 \qquad \frac{\Psi \vdash e \Leftarrow A_2 \hookrightarrow e_2'}{\Psi \vdash e \Leftarrow A_1 \sqcup A_2 \hookrightarrow \texttt{inr}\ e_2'}\ \Leftarrow\sqcup_2$$

$$\frac{(x : A) \in \Psi}{\Psi \vdash x \Rightarrow A \hookrightarrow x}\ \Rightarrow\text{Var} \qquad \frac{\Psi \vdash A \qquad \Psi \vdash e \Leftarrow A \hookrightarrow e'}{\Psi \vdash (e : A) \Rightarrow A \hookrightarrow e'}\ \Rightarrow\text{Anno} \qquad \frac{}{\Psi \vdash () \Rightarrow \mathbb{1} \hookrightarrow ()}\ \Rightarrow\texttt{Unit}$$

$$\frac{\Psi, x : \sigma \vdash e \Leftarrow \tau \hookrightarrow e' \qquad \Psi \vdash \sigma}{\Psi \vdash \lambda x.\, e \Rightarrow \sigma \to \tau \hookrightarrow \lambda(x : |\sigma|).\, e'}\ \Rightarrow\!\to\!\text{Mono} \qquad \frac{\Psi, a \vdash e \Leftarrow A \hookrightarrow e'}{\Psi \vdash \Lambda a.\, e : A \Rightarrow \forall a.\, A \hookrightarrow \Lambda a.\, e'}\ \Rightarrow\Lambda$$

$$\frac{\Psi \vdash e_1 \Rightarrow A \hookrightarrow e_1' \qquad \Psi \vdash A \triangleright B \to C \hookrightarrow co \qquad \Psi \vdash e_2 \Leftarrow B \hookrightarrow e_2'}{\Psi \vdash e_1\ e_2 \Rightarrow C \hookrightarrow (co\ e_1')\ e_2'}\ \Rightarrow\text{App}$$

$$\frac{\Psi \vdash e \Rightarrow A \hookrightarrow e \qquad \Psi \vdash A \circ B \Rrightarrow C \hookrightarrow co_1 \mid co_2}{\Psi \vdash e\ @B \Rightarrow C \hookrightarrow co_2\ (co_1\ e)}\ \Rightarrow\text{TApp}$$

Fig. S12. Elaboration of Checking and Inference

Type variables	a, b		Variables x, y
Types	A, B, C	$::=$	$\mathbb{1} \mid a \mid \forall a.\, A \mid A \to B \mid A + B \mid A \times B$
Expressions	e, t	$::=$	$x \mid () \mid \lambda x : A.\, e \mid e_1\ e_2 \mid e : A \mid e\ @A \mid \Lambda a.\, e \mid \texttt{inl}\ e \mid \texttt{inr}\ e \mid$ $\texttt{case}\ e\ \texttt{of}\ \texttt{inl}\ x \to e_1; \texttt{inr}\ y \to e_2 \mid \Pi_1\ e \mid \Pi_2\ e \mid \langle e1, e2 \rangle$

The type safety is proved by elaborating expressions in $F^e_{\sqcup\sqcap}$ to expressions in System F with a terminal type $\mathbb{1}$, and product ($A \times B$) and sum ($A + B$) type. For brevity, we use $F_{\times+}$ to denote this target type system. The elaboration of intersection and union is similar to the rules by Dunfield [2014] except we do not have an expression-level merge operator. This target system is known to be type-safe.

The type translation is shown in S10. Intersection type $A \sqcap B$ is translated to product type $A \times B$. Union type $A \sqcup B$ is translated to sum type $A + B$. Both $\top$ and $\mathbb{1}$ is translated to $\mathbb{1}$. $\bot$ is translated to $\forall a.\ a$. For context translation, the subtype variable is translated to an ordinary type variable as $\tilde{a}, \Psi \hookrightarrow a, \Delta$.

The main goal of the elaboration on type-level relations (subtyping $A \le B$, matching $A \triangleright B \to C$, and type application $A \circ B \Rrightarrow C$) is to produce a coercion function. For subtyping, this function should be of type $|A| \to |B|$. For matching, this function should be of type $|A| \to |B| \to |C|$. For type application, the situation is a bit complicated as B is included in the relation. It should produce two coercions. co_1 converts $|A|$ to another type $|D|$ that can be type applied (i.e. $|\bot|$ and $|\forall a.\ A'|$), and co_2 coerce type $|D|$ to the desired output type $|C|$ by type applying $|D|$ to $|B|$. There are two tasks for such coercion: (1) inserting explicit type application to places where implicit instantiation is invoked; (2) inserting projections (π_1, π_2) and injections (`inl`, `inr`) where a branch is chosen implicitly by rules related to intersection and union types.

$\boxed{\Psi \vdash A \triangleright B \to C \ \hookrightarrow co}$ Bidirectional Matching is elaborated to *co*

$$\frac{}{\Psi \vdash \bot \triangleright \top \to \bot \ \hookrightarrow \lambda(x : |\bot|).\, x \mathbin{@}(|\top \to \bot|)}\ \triangleright\bot \qquad \frac{}{\Psi \vdash A \to B \triangleright A \to B \ \hookrightarrow \lambda(x : |A \to B|).\, x}\ \triangleright\!\to$$

$$\frac{\Psi \vdash [\tau/a]A \triangleright B \to C \ \hookrightarrow co_1}{\Psi \vdash \forall a.\, A \triangleright B \to C \ \hookrightarrow \lambda(x : |\forall a.\, A|).\, co_1\ (x \mathbin{@}\tau)}\ \triangleright\forall$$

$$\frac{\Psi \vdash A_1 \triangleright B \to C \ \hookrightarrow co}{\Psi \vdash (A_1 \sqcap A_2) \triangleright B \to C \ \hookrightarrow \lambda(x : |A_1 \sqcap A_2|).\, co\ (\pi_1\ x)}\ \triangleright\sqcap_1$$

$$\frac{\Psi \vdash A_2 \triangleright B \to C \ \hookrightarrow co}{\Psi \vdash (A_1 \sqcap A_2) \triangleright B \to C \ \hookrightarrow \lambda(x : |A_1 \sqcap A_2|).\, co\ (\pi_2\ x)}\ \triangleright\sqcap_2$$

$$\frac{\Psi \vdash A_1 \triangleright B_1 \to C_1 \ \hookrightarrow co_1 \qquad \Psi \vdash A_2 \triangleright B_2 \to C_2 \ \hookrightarrow co_2}{\begin{array}{c}\Psi \vdash (A_1 \sqcup A_2) \triangleright (B_1 \sqcap B_2) \to (C_1 \sqcup C_2) \\ \hookrightarrow \lambda(x : |A_1 \sqcup A_2|).\, \lambda(y : |B_1 \sqcap B_2|).\, \mathtt{case}\ \ x\ \ \mathtt{of}\ \ z_1 \mapsto \mathtt{inl}\ \ ((co_1\ z_1)\ (\pi_1\ y)); z_2 \mapsto \mathtt{inr}\ \ ((co_2\ z_2)\ (\pi_2\ y))\end{array}}\ \triangleright\sqcup$$

$\boxed{\Psi \vdash A \circ B \Rrightarrow C \ \hookrightarrow co_1 | co_2}$ type application is elaborated to co_1 and co_2

$$\frac{}{\Psi \vdash \bot \circ A \Rrightarrow \bot \ \hookrightarrow \lambda(x : |\bot|).\, x \mid \lambda(x : |\bot|).\, x}\ \circ\!\Rrightarrow\!\bot$$

$$\frac{}{\Psi \vdash \forall a.\, A \circ B \Rrightarrow [B/a]A \ \hookrightarrow \lambda(x : |\forall a.\, A|).\, x \mid \lambda(x : |\forall a.\, A|).\, (x \mathbin{@}|B|)}\ \circ\!\Rrightarrow\!\forall$$

$$\frac{\Psi \vdash A_1 \circ B \Rrightarrow C \ \hookrightarrow co_1 \mid co_2}{\Psi \vdash (A_1 \sqcap A_2) \circ B \Rrightarrow C \ \hookrightarrow \lambda(x : |A_1 \sqcap A_2|).\, co_1\ (\pi_1\ x) \mid co_2}\ \circ\!\Rrightarrow\!\sqcap_1$$

$$\frac{\Psi \vdash A_2 \circ B \Rrightarrow C \ \hookrightarrow co_1 \mid co_2}{\Psi \vdash (A_1 \sqcap A_2) \circ B \Rrightarrow C \ \hookrightarrow \lambda(x : |A_1 \sqcap A_2|).\, co_1\ (\pi_2\ x) \mid co_2}\ \circ\!\Rrightarrow\!\sqcap_2$$

$$\frac{\Psi \vdash A_1 \circ B \Rrightarrow C_1 \ \hookrightarrow co_1 \mid co_2 \qquad \Psi \vdash A_2 \circ B \Rrightarrow C_2 \ \hookrightarrow co_3 \mid co_4}{\begin{array}{c}\Psi \vdash (A_1 \sqcup A_2) \circ B \Rrightarrow C \\ \hookrightarrow \lambda(x : |A_1 \sqcup A_2|).\, \mathtt{case}\ \ x\ \ \mathtt{of}\ \ y_1 \mapsto \mathtt{inl}\ \ (co_2\ (co_1\ y_1)); y_2 \mapsto \mathtt{inr}\ \ (co_4\ (co_3\ y_2)) \mid \lambda(x : |C_1 \sqcup C_2|).\, x\end{array}}\ \circ\!\Rrightarrow\!\sqcup$$

Fig. S13. Elaboration of Matching and Type Application

Lemma D.1 (Soundness of Subtyping, Matching and Type-application by Elaborating to System $F_{\times+}$). *Given everything well-formed and* $\Psi \hookrightarrow \Delta$,

(1) *If* $\Psi \vdash A \leq B$ *then* $\Delta \vdash_{F_{\times+}} co : |A| \to |B|$.
(2) *If* $\Psi \vdash A \triangleright B \to C \ \hookrightarrow co$ *then* $\Delta \vdash_{F_{\times+}} co : |A| \to |B| \to |C|$.
(3) *If* $\Psi \vdash A \circ B \Rrightarrow C \ \hookrightarrow co_1 \mid co_2$ *then,* $\exists D$*, s.t.* $\Delta \vdash_{F_{\times+}} co_1 : |A| \to |D|$ *and* $\Delta \vdash_{F_{\times+}} co_2 : |D| \to |C|$.

The soundness of elaboration for inference ($\Psi \vdash e \Rightarrow A$) and checking ($\Psi \vdash e \Leftarrow A$) states that the elaborated term e' has type $|A|$.

Theorem D.2 (Type Soundness by Elaborating to System $F_{\times+}$). *Given everything well-formed and* $\Psi \hookrightarrow \Delta$,

(1) *If* $\Psi \vdash e \Rightarrow A \ \hookrightarrow e'$ *then* $\Delta \vdash_{F_{\times+}} e' : |A|$.
(2) *If* $\Psi \vdash e \Leftarrow A \ \hookrightarrow e'$ *then* $\Delta \vdash_{F_{\times+}} e' : |A|$.

D.2 Proof Details of Decidability

In this section, we provide an overview of the decidability. The decidability proof is based on a lexicographic group of 9 measures on the worklist Γ: $(|\Gamma|_e, |\Gamma|_\omega, |\Gamma|_{\twoheadrightarrow\!\!\!\Rightarrow}, |\Gamma|_{\twoheadrightarrow\!\!\!\Rightarrow_\omega}, |\Gamma|_\triangleright, |\Gamma|_{\triangleright_\omega}, |\Gamma|_\rightarrow, |\Gamma|_{\widehat{a}}, |\Gamma|_\leq)$. We will mostly demonstrate the high-level ideas behind the design of these measures since their correctness has been formally checked. The measures presented here are an overestimation, so possibly some of them can be modified to have a more precise overestimation. All the rule indexing in this section is based on the defunctionalized version shown in Figure S8 and Figure S9.

$\boxed{|A|_{\sqcup\sqcap}}$ number of intersections and unions in A

$$
\begin{aligned}
|1|_{\sqcup\sqcap} = |\top|_{\sqcup\sqcap} = |\bot|_{\sqcup\sqcap} = |a|_{\sqcup\sqcap} &= 0 \\
|A \to B|_{\sqcup\sqcap} &= |A|_{\sqcup\sqcap} + |B|_{\sqcup\sqcap} \\
|\forall a.\, A|_{\sqcup\sqcap} &= |A|_{\sqcup\sqcap} \\
|A \sqcap B|_{\sqcup\sqcap} &= |A|_{\sqcup\sqcap} + |B|_{\sqcup\sqcap} + 2 \\
|A \sqcup B|_{\sqcup\sqcap} &= |A|_{\sqcup\sqcap} + |B|_{\sqcup\sqcap} + 2
\end{aligned}
$$

$\boxed{|A|_{\sqcup\sqcap a}, {}_{\Xi}|e|_{\sqcup\sqcap a}}$ number of intersections, unions and bound type variables in A, e

$$
\begin{aligned}
|1|_{\sqcup\sqcap a} = |\top|_{\sqcup\sqcap a} = |\bot|_{\sqcup\sqcap a} = |a|_{\sqcup\sqcap a} &= 0 \\
|A \to B|_{\sqcup\sqcap a} &= |A|_{\sqcup\sqcap a} + |B|_{\sqcup\sqcap a} \\
|\forall a.\, A|_{\sqcup\sqcap a} &= |A|_{\sqcup\sqcap a} + \#a \in A \\
|A \sqcap B|_{\sqcup\sqcap a} &= |A|_{\sqcup\sqcap a} + |B|_{\sqcup\sqcap a} + 2 \\
|A \sqcup B|_{\sqcup\sqcap a} &= |A|_{\sqcup\sqcap a} + |B|_{\sqcup\sqcap a} + 2
\end{aligned}
$$

$$
\begin{aligned}
{}_{\Xi}|()|_{\sqcup\sqcap a} &= 0 \\
{}_{\Xi}|x|_{\sqcup\sqcap a} &= n \quad \text{when } x : n \in \Xi \\
{}_{\Xi}|\lambda x.\, e|_{\sqcup\sqcap a} &= {}_{\Xi, x:0}|e|_{\sqcup\sqcap a} \\
{}_{\Xi}|e_1\ e_2|_{\sqcup\sqcap a} &= 2\ {}_{\Xi}|e_1|_{\sqcup\sqcap a} + {}_{\Xi}|e_2|_{\sqcup\sqcap a} + 1 \\
{}_{\Xi}|\Lambda a.\, e : A|_{\sqcup\sqcap a} &= |\forall a.\, A|_{\sqcup\sqcap a} \\
{}_{\Xi}|e\ @A|_{\sqcup\sqcap a} &= ({}_{\Xi}|e|_{\sqcup\sqcap a} + 1) \cdot (|A|_{\sqcup\sqcap a} + 2) \\
{}_{\Xi}|e : A|_{\sqcup\sqcap a} &= ({}_{\Xi}|e|_{\sqcup\sqcap a} + 1) \cdot (|A|_{\sqcup\sqcap a} + 2)
\end{aligned}
$$

$\boxed{\|\Gamma\| = \Xi}$ compute Ξ ($|\cdot|_{\sqcup\sqcap a}$ context) from Γ

$$
\begin{aligned}
\|\cdot\| &= \cdot \\
\|\Gamma, a\| &= \|\Gamma\| \\
\|\Gamma, \tilde{a}\| &= \|\Gamma\|
\end{aligned}
\qquad
\begin{aligned}
\|\Gamma, \widehat{a}\| &= \|\Gamma\| \\
\|\Gamma, x : A\| &= \|\Gamma\|, x : |A|_{\sqcup\sqcap a} \\
\|\Gamma \Vdash \omega\| &= \|\Gamma\|
\end{aligned}
$$

Fig. S14. Definition of $|\cdot|_{\sqcup\sqcap}$, $|\cdot|_{\sqcup\sqcap a}$ and $\|\Gamma\|$

The definition $|\cdot|_e$ is the most complicated one, due to the intrinsic complication of the duplication of expression. The duplication of expression is caused by the checking introduction rule of intersection ($e \Leftarrow A_1 \sqcap A_2$). It seems that we can easily estimate the duplication from e, A_1, and A_2. However, checking judgment can be created by the inference judgment in rules 30 ($e : A \Rightarrow \omega^s$), 31 ($\Lambda a.\, e : A \Rightarrow \omega^s$), and 33 ($e_1 e_2 \Rightarrow \omega^s$, e_2 will be checked by the domain type matched from the type of e_1). This makes the first difficulty: for inference, the type information is unknown yet, so we must estimate it from the expressions themselves. The estimation for annotation and type abstraction is simple since the type information is fully given. The estimation for application is tricky since

e_1 can be any expression, especially when e_1 is x and type-application. This estimation is finally done by $|\cdot|_{\sqcup\sqcap a}$, and $\|\Gamma\|$, whose definition is in Figure S14. $|e|_{\sqcup\sqcap a}$ roughly estimates the number of duplications if another expression e_2 is checked by the type inferred from e. $|\cdot|_{\sqcup\sqcap a}$ considers the number of intersection and union types and bound type variables that appear in e. The number of bound type variables is considered due to the existence of $e\ @A$, so that the bound type variables could be replaced by A later so that more intersection and union types can be introduced. This measure is employed in many places of the definition. The function $|\Gamma|$ precomputes the size of the type of each variable in the context Γ, which is used in $|\cdot|_e$.

Figure S15 shows the definition of $|\cdot|_e$, which is based on $|\cdot|_{\sqcup\sqcap a}$. The second complication is that the size cannot be simply computed by the pattern matching the expression. Suppose we want to estimate the size of $(e : A) : B$, $e : A$ will be duplicated by a number related to B, and for each duplication, e will be duplicated by a number related to A, which means the estimation must carry the outer duplication information in computing the size of the inner components. This leads to a tail-recursively computed measure ${}_{\Xi}|e|_e^n$, ${}_{\Xi}|\omega^s|_e^n$ and ${}_{\Xi}|\omega^d|_e^{(n_1,n_2)}$, where the left subscript Ξ denotes the precomputed context, and the right superscript is the size carried by tail recursion. For ${}_{\Xi}|\omega^d|_e^{(n_1,n_2)}$ the size carried by its tail recursion, as well as its output, is a pair. The first element of the pair is the size of the negative position, and the second is the size of the positive position. The passed pair enables more precise approximation. To summarize, there are two types of recursion in the computation of $|\cdot|_e$: recursing over the continuation chain in $|\omega|_e$ and recursing over the expression in $|e|_e$.

Figure S16 shows the definition of $|\cdot|_\omega$. Matching and type-application judgments count as zero in this measure as they are reasoned modularly by their own measures. Checking, inference, application inference, and continuation applications are non-zero. Since only application inference has a corresponding continuation, it is the only non-zero measure in $|\omega^s|_\omega$ and $|\omega^d|_\omega$.

Figure S18 shows the definition of $|\cdot|_{\triangleright}$. Figure S17 shows the definition of $|\cdot|_{\twoheadrightarrow}$. Figure S19 shows the definition of $|\cdot|_{\twoheadrightarrow_\omega}$ and $|\cdot|_{\triangleright_\omega}$. Since the size of the type being applied could increase as $(\forall a.\ a \circ ((\forall a.\ a) \sqcup (\forall a.\ a)))$, the information along the continuation chain must be considered in advance to get the correct estimation. Thus, $|\cdot|_{\twoheadrightarrow}$ needs to recurse over the continuation chain. The situation is simpler for matching since the type being matched always gets smaller. $|\cdot|_{\triangleright}$ is simply a summation over all continuation. $|\cdot|_{\twoheadrightarrow_\omega}$ and $|\cdot|_{\triangleright_\omega}$ are mainly introduced due to the defunctionalization and they help reason about the reduction of the intermediate works and continuations that relay the information.

$|\cdot|_{\to}$ is shown in Figure S18. Its meaning is well justified by its name: sum of rank of non-mono components. $|\widehat{a}|_{\leq}$ is a simple measure that counts the existential variable declaration in Γ.

$|\leq|_{\to}$ is for the reasoning of subtyping. Duplication also happens in subtyping in rules 12 and 15. However, the information in subtyping judgment is always full, so the estimation is much simpler than in the case of expression. The key design is $|A \leq B|_{\leq} = |A|_{\leq} \cdot |B|_{\sqcup\sqcap} + |B|_{\leq} \cdot |A|_{\sqcup\sqcap}$ where $|A|_{\sqcup\sqcap}$ is a simpler version of $|A|_{\sqcup\sqcap a}$ that only counts the intersection and union type, but not the bound type variables. As the last measure, $|\leq|_{\to}$ also deals with some miscellaneous cases like the garbage collection of variables and type variables, so it also counts the number of such declarations.

All the first 7 measures are smaller or unchanged under worklist substitution with arbitrary monotype, as formulated by the following theorem. $|\widehat{a}|_{\leq}$ decreases by 1 since the target a is removed from Γ.

Theorem D.3 (Measure Stability under Worklist Substitution). *Given a well-formed Γ, and a well-formed monotype τ under Γ,*

(1) $|\{\tau/a\}\Gamma|_e \leq |\Gamma|_e$

(2) $|\{\tau/a\}\Gamma|_{\omega} \leq |\Gamma|_{\omega}$
(3) $|\{\tau/a\}\Gamma|_{\twoheadrightarrow} \leq |\Gamma|_{\twoheadrightarrow}$
(4) $|\{\tau/a\}\Gamma|_{\twoheadrightarrow_{\omega}} \leq |\Gamma|_{\twoheadrightarrow_{\omega}}$
(5) $|\{\tau/a\}\Gamma|_{\triangleright} \leq |\Gamma|_{\triangleright}$
(6) $|\{\tau/a\}\Gamma|_{\triangleright_{\omega}} \leq |\Gamma|_{\triangleright_{\omega}}$
(7) $|\{\tau/a\}\Gamma|_{\rightarrow} \leq |\Gamma|_{\rightarrow}$

Case Analysis. We summarize the decreasing measure of each rule. This set of measures must decrease after a fixed number of reduction steps, so the reasoning is simple, ignoring the complication of designing the measures.

The measure decreases after a one-step reduction for the following rules.

- For rules 23-27 and 29-35, $|\Gamma|_{e}$ decreases.
- For rules 22, 45, 53, and 54, $|\Gamma|_{\omega}$ decreases.
- For rules 48-50, $|\Gamma|_{\twoheadrightarrow}$ decreases.
- For rule 51, $|\Gamma|_{\twoheadrightarrow_{\omega}}$ decreases.
- For rules 37-42, $|\Gamma|_{\triangleright}$ decreases.
- For rule 43, $|\Gamma|_{\triangleright_{\omega}}$ decreases.
- For rules 10, 20 and 21, $|\Gamma|_{\rightarrow}$ decreases.
- For rules 18 and 19, $|\Gamma|_{\widehat{a}}$ decreases.
- For rules 1-9 and 11-17, $|\Gamma|_{\leq}$ decreases.

In the remaining cases, the measure decreases after a two-step reduction.

- For rule 28, $|\Gamma|_{e}$ decreases.
- For rules 46, 47, $|\Gamma|_{\twoheadrightarrow}$ decreases.
- For rule 52, $|\Gamma|_{\twoheadrightarrow_{\omega}}$ decreases.
- For rules 36 and 44, $|\Gamma|_{\triangleright}$ decreases.

$\boxed{{}_{\Xi}|e|_e^n,\ {}_{\Xi}|\omega^s|_e^n,\ {}_{\Xi}|\omega^d|_e^{(n_1,n_2)},\ {}_{\Xi}|w|_e,\ |\Gamma|_e}$ expression size of e, ω^s, ω^d, w, and Γ

$$
\begin{aligned}
{}_{\Xi}|()|_e^n = {}_{\Xi}|x|_e^n &= n+1 \\
{}_{\Xi}|\lambda x.\ e|_e^n &= {}_{\Xi,x:n}|e|_e^n + n + 1 \\
{}_{\Xi}|e_1\ e_2|_e^n &= {}_{\Xi}|e_1|_e^n + ({}_{\Xi}|e_1|_{\sqcup\sqcap a} + 1) \cdot {}_{\Xi}|e_2|_e^{({}_{\Xi}|e_1|_{\sqcup\sqcap a}+1)\cdot(n+1)-1} + 1 \\
{}_{\Xi}|\Lambda a.\ e : A|_e^n &= (n+1) \cdot ({}_{\Xi}|e|_e^{|A|_{\sqcup\sqcap a}} + 1) \\
{}_{\Xi}|e\ @A|_e^n &= (n+1) \cdot ({}_{\Xi}|e|_e^{(|A|_{\sqcup\sqcap a}+1)\cdot(n+1)-1} + 1) \\
{}_{\Xi}|e : A|_e^n &= (n+1) \cdot ({}_{\Xi}|e|_e^{|A|_{\sqcup\sqcap a}} + 1)
\end{aligned}
$$

$$
\begin{aligned}
{}_{\Xi}|\_ \le A|_e^n &= 0 \\
{}_{\Xi}|\_ \circ A \Rrightarrow \omega^s|_e^n &= {}_{\Xi}|\omega^s|_e^{(|A|_{\sqcup\sqcap a}+2)\cdot n} \\
{}_{\Xi}|\_ \triangleright \omega^d|_e^n &= \sum {}_{\Xi}|\omega^d|_e^{(n,n)} \\
{}_{\Xi}|\_ \Cup A \circ B \Rrightarrow \omega^s|_e^n &= {}_{\Xi}|\omega^s|_e^{|A|_{\sqcup\sqcap a}\cdot(|B|_{\sqcup\sqcap a}+2)+n+2} \\
{}_{\Xi}|\_ \Cup_{\circ} A \blacktriangleright \omega^s|_e^n &= {}_{\Xi}|\omega^s|_e^{|A|_{\sqcup\sqcap a}+n+2}
\end{aligned}
$$

$$
\begin{aligned}
{}_{\Xi}|\_ \to \_ \bullet e \Rrightarrow \omega^s|_e^{(n_1,n_2)} &= (0,\ {}_{\Xi}|e|_e^{n_1} \cdot (n_1+1) + {}_{\Xi}|\omega^s|_e^{n_2}) \\
{}_{\Xi}|\_ \to \_ \Cup A \triangleright \omega^d|_e^{(n_1,n_2)} &= {}_{\Xi}|\omega^d|_e^{(n_1+|A|_{\sqcup\sqcap a}+2,n_2+|A|_{\sqcup\sqcap a}+2)} \\
{}_{\Xi}|\_ \to \_ \Cup_{\triangleright} (A \to B) \blacktriangleright \omega^d|_e^{(n_1,n_2)} &= {}_{\Xi}|\omega^d|_e^{(n_1+|A|_{\sqcup\sqcap a}+2,n_2+|B|_{\sqcup\sqcap a}+2)}
\end{aligned}
$$

$$
\begin{aligned}
{}_{\Xi}|A \le B|_e &= 0 \\
{}_{\Xi}|e \Leftarrow A|_e &= {}_{\Xi}|e|_e^{|A|_{\sqcup\sqcap a}} \\
{}_{\Xi}|e \Rightarrow \omega^s|_e &= {}_{\Xi}|e|_e^0 + {}_{\Xi}|\omega^s|_e^{|e|_{\sqcup\sqcap a}} \\
{}_{\Xi}|A \circ B \Rrightarrow \omega^s|_e &= {}_{\Xi}|\omega^s|_e^{|A|_{\sqcup\sqcap a}\cdot(|B|_{\sqcup\sqcap a}+2)} \\
{}_{\Xi}|A \triangleright \omega^d|_e &= \sum {}_{\Xi}|\omega^d|_e^{(|A|_{\sqcup\sqcap a},|A|_{\sqcup\sqcap a})} \\
{}_{\Xi}|A \to B \bullet e \Rrightarrow \omega^s|_e &= |A|_{\sqcup\sqcap a} \cdot ({}_{\Xi}|e|_e^{|A|_{\sqcup\sqcap a}} + 1) + {}_{\Xi}|\omega^s|_e^{|B|_{\sqcup\sqcap a}} \\
{}_{\Xi}|C \Cup A \circ B \Rrightarrow \omega^s|_e &= {}_{\Xi}|\omega^s|_e^{|A|_{\sqcup\sqcap a}\cdot(|B|_{\sqcup\sqcap a}+2)+|C|_{\sqcup\sqcap a}+2} \\
{}_{\Xi}|(B \to C) \Cup A \triangleright \omega^d|_e &= \sum {}_{\Xi}|\omega^d|_e^{(|B|_{\sqcup\sqcap a}+|A|_{\sqcup\sqcap a}+2,|C|_{\sqcup\sqcap a}+|A|_{\sqcup\sqcap a}+2)} \\
{}_{\Xi}|(A_1 \to B_1) \Cup_{\triangleright} (A_2 \to B_2) \blacktriangleright \omega^d|_e &= \sum {}_{\Xi}|\omega^d|_e^{(|A_1|_{\sqcup\sqcap a}+|A_2|_{\sqcup\sqcap a}+2,|B_1|_{\sqcup\sqcap a}+|B_2|_{\sqcup\sqcap a}+2)} \\
{}_{\Xi}|(A_1 \Cup_{\circ} A_2 \blacktriangleright \omega^s)|_e &= {}_{\Xi}|\omega^s|_e^{(|A_1|_{\sqcup\sqcap a}+|A_2|_{\sqcup\sqcap a}+2)} \\
{}_{\Xi}|\omega^s \diamond A|_e &= {}_{\Xi}|\omega^s|_e^{|A|_{\sqcup\sqcap a}} \\
{}_{\Xi}|\omega^d \diamond A, B|_e &= \sum {}_{\Xi}|\omega^d|_e^{(|A|_{\sqcup\sqcap a},|B|_{\sqcup\sqcap a})}
\end{aligned}
$$

$$
|\Gamma|_e = \sum_{w\in\Gamma} {}_{\|\Gamma\|}|w|_e
$$

Fig. S15. Definition of $|\cdot|_e$

$\boxed{|\omega^s|_\omega, |\omega^d|_\omega}$ number of judgments of ω^s, ω^d

$$\begin{aligned}
|\_ \leq A|_\omega &= 0 \\
|\_ \circ A \Rrightarrow_\alpha \omega^s|_\omega &= |\omega^s|_\omega \\
|\_ \triangleright \omega^d|_\omega &= |\omega^d|_\omega \\
|\_ \Cup A \circ B \Rrightarrow \omega^s|_\omega &= |\omega^s|_\omega \\
|\_ \Cup_\circ A \blacktriangleright \omega^s|_\omega &= |\omega^s|_\omega
\end{aligned}
\qquad
\begin{aligned}
|\_ \rightarrow \_ \bullet e \Rrightarrow \omega^s|_\omega &= |\omega^s|_\omega + 5 \\
|\_ \rightarrow \_ \Cup A \triangleright \omega^d|_\omega &= |\omega^d|_\omega \\
|\_ \rightarrow \_ \Cup_\triangleright (A \rightarrow B) \blacktriangleright \omega^d|_\omega &= |\omega^d|_\omega
\end{aligned}$$

$\boxed{|w|_\omega}$ number of judgments of w

$$\begin{aligned}
|A \leq B|_\omega &= 0 \\
|e \Leftarrow A|_\omega &= 3 \\
|e \Rightarrow \omega^s|_\omega &= |\omega^s|_\omega + 2 \\
|A \circ B \Rrightarrow \omega^s|_\omega &= |\omega^s|_\omega \\
|A \triangleright \omega^d|_\omega &= |\omega^d|_\omega \\
|A \rightarrow B \bullet e \Rrightarrow \omega^s|_\omega &= |\omega^s|_\omega + 5
\end{aligned}
\qquad
\begin{aligned}
|C \Cup A \circ B \Rrightarrow \omega^s|_\omega &= |\omega^s|_\omega \\
|(B \rightarrow C) \Cup A \triangleright \omega^d|_\omega &= |\omega^d|_\omega \\
|(A_1 \rightarrow B_1) \Cup_\triangleright (A_2 \rightarrow B_2) \blacktriangleright \omega^d|_\omega &= |\omega^d|_\omega \\
|(A_1 \Cup_\circ A_2 \blacktriangleright \omega^s)|_\omega &= |\omega^s|_\omega \\
|\omega^s \diamond A|_\omega &= |\omega^s|_\omega + 1 \\
|\omega^d \diamond A \diamond B|_\omega &= |\omega^d|_\omega + 1
\end{aligned}$$

$\boxed{|\Gamma|_\omega}$ number of judgments in Γ

$$|\Gamma|_\omega = \sum_{w \in \Gamma} |w|_\omega$$

Fig. S16. Definition of $|\cdot|_\omega$

$\boxed{|A|_{\Rrightarrow}}$ type size of type-application of A

$$
\begin{aligned}
|1|_{\Rrightarrow} = |\top|_{\Rrightarrow} = |\bot|_{\Rrightarrow} = |A \to B|_{\Rrightarrow} &= 0 \\
|\forall a.\, A|_{\Rrightarrow} &= |A|_{\Rrightarrow} + |A|_a + 1 \\
|A \sqcap B|_{\Rrightarrow} &= |A|_{\Rrightarrow} + |B|_{\Rrightarrow} + 1 \\
|A \sqcup B|_{\Rrightarrow} &= |A|_{\Rrightarrow} + |B|_{\Rrightarrow} + 1
\end{aligned}
$$

$\boxed{|\omega^s|^{n}_{\Rrightarrow}, |\omega^d|^{n}_{\Rrightarrow}}$ type size of type-application of ω^s and ω^d

$$
\begin{aligned}
|\_ \leq A|^{n}_{\Rrightarrow} &= n \\
|\_ \circ A \Rrightarrow \omega^s|^{n}_{\Rrightarrow} &= |\omega^s|^{(|A|_{\Rrightarrow}+2)\cdot n}_{\Rrightarrow} \\
|\_ \triangleright \omega^d|^{n}_{\Rrightarrow} &= |\omega^d|^{n}_{\Rrightarrow} \\
|\_ \Cup A \circ B \Rrightarrow \omega^s|^{n}_{\Rrightarrow} &= |\omega^s|^{|A|_{\Rrightarrow}\cdot(|B|_{\Rrightarrow}+2)+n+1}_{\Rrightarrow} \\
|\_ \Cup_{\circ} A \blacktriangleright \omega^s|^{n}_{\Rrightarrow} &= |\omega^s|^{|A|_{\Rrightarrow}+n+1}_{\Rrightarrow}
\end{aligned}
$$

$$
\begin{aligned}
|\_ \to \_ \bullet e \Rrightarrow \omega^s|^{n}_{\Rrightarrow} &= |\omega^s|^{n}_{\Rrightarrow} \\
|\_ \to \_ \Cup A \triangleright \omega^d|^{n}_{\Rrightarrow} &= |\omega^d|^{n}_{\Rrightarrow} \\
|\_ \to \_ \Cup_{\triangleright} (A \to B) \blacktriangleright \omega^d|^{n}_{\Rrightarrow} &= |\omega^d|^{n}_{\Rrightarrow}
\end{aligned}
$$

$\boxed{|w|_{\Rrightarrow}}$ type size of type-application of w

$$
\begin{aligned}
|A \leq B|_{\Rrightarrow} = |e \Leftarrow A|_{\Rrightarrow} &= 0 \\
|e \Rightarrow \omega^s|_{\Rrightarrow} &= |\omega^s|^{0}_{\Rrightarrow} \\
|A \circ B \Rrightarrow \omega^s|_{\Rrightarrow} &= |\_ \circ B \Rrightarrow \omega^s|^{|A|_{\Rrightarrow}}_{\Rrightarrow} \\
|A \triangleright \omega^d|_{\Rrightarrow} &= |\omega^d|^{0}_{\Rrightarrow} \\
|A \to B \bullet e \Rrightarrow \omega^s|_{\Rrightarrow} &= |\omega^s|^{0}_{\Rrightarrow} \\
|C \Cup A \circ B \Rrightarrow \omega^s|_{\Rrightarrow} &= |\omega^s|^{|C|_{\Rrightarrow}+|A|_{\Rrightarrow}\cdot(|B|_{\Rrightarrow}+2)+1}_{\Rrightarrow} \\
|(B \to C) \Cup A \triangleright \omega^d|_{\Rrightarrow} &= |\omega^d|^{0}_{\Rrightarrow} \\
|(A_1 \to B_1) \Cup_{\triangleright} (A_2 \to B_2) \blacktriangleright \omega^d|_{\Rrightarrow} &= |\omega^d|^{0}_{\Rrightarrow} \\
|(A_1 \Cup_{\circ} A_2 \blacktriangleright \omega^s)|_{\Rrightarrow} &= |\omega^s|^{|A_1|_{\Rrightarrow}+|A_2|_{\Rrightarrow}+1}_{\Rrightarrow}
\end{aligned}
$$

$\boxed{|\Gamma|_{\Rrightarrow}}$ type size of type-application of Γ

$$
|\Gamma|_{\Rrightarrow} = \sum_{w \in \Gamma} |w|_{\Rrightarrow}
$$

Fig. S17. Definition of $|\cdot|_{\Rrightarrow}$

$\boxed{|A|_{\triangleright}, |\omega^s|_{\triangleright}, |\omega^d|_{\triangleright}, |w|_{\triangleright}, |\Gamma|_{\triangleright}}$ type size of matching of A, ω^s, ω^d, w, and Γ

$$\begin{aligned}
|1|_{\triangleright} = |\top|_{\triangleright} &= 0 \\
|\bot|_{\triangleright} = |a|_{\triangleright} &= 2 \\
|A \to B|_{\triangleright} &= 1 \\
|\forall a.\, A|_{\triangleright} &= |A|_{\triangleright} + 1 \\
|A \sqcap B|_{\triangleright} &= |A|_{\triangleright} + |B|_{\triangleright} + 1 \\
|A \sqcup B|_{\triangleright} &= |A|_{\triangleright} + |B|_{\triangleright} + 2
\end{aligned}$$

$$\begin{aligned}
|\_ \triangleright \omega^d|_{\triangleright} &= |\omega^d|_{\triangleright} + 1 \\
|\omega^s|_{\triangleright} &= 0 \quad \text{otherwise} \\
|\_ \!\to\! \_ \bullet e \Rrightarrow \omega^s|_{\triangleright} &= 0 \\
|\_ \!\to\! \_ \Cup A \triangleright \omega^d|_{\triangleright} &= |\omega^d|_{\triangleright} + |A|_{\triangleright} + 1 \\
|\_ \!\to\! \_ \Cup_{\triangleright} (A \to B) \blacktriangleright \omega^d|_{\triangleright} &= |\omega^d|_{\triangleright} + 1
\end{aligned}$$

$$\begin{aligned}
|A \triangleright \omega^d|_{\triangleright} &= |A|_{\triangleright} + |\omega^d|_{\triangleright} \\
|(B \to C) \Cup A \triangleright \omega^d|_{\triangleright} &= |A|_{\triangleright} + |\omega^d|_{\triangleright} + 1 \\
|(A_1 \to B_1) \Cup_{\triangleright} (A_2 \to B_2) \blacktriangleright \omega^d|_{\triangleright} &= |\omega^d|_{\triangleright} + 1 \\
|w|_{\triangleright} &= 0 \quad \text{otherwise}
\end{aligned}$$

$$|\Gamma|_{\triangleright} = \sum_{w \in \Gamma} |w|_{\triangleright}$$

$\boxed{{}_{\Gamma}|A|^{n}_{\to}, {}_{\Gamma}|w|_{\to}, |\Gamma|_{\to}}$ sum of rank non-mono component of A, w, and Γ

$$\begin{aligned}
{}_{\Gamma}|1|^{n}_{\to} &= 0 \\
{}_{\Gamma}|\top|^{n}_{\to} = {}_{\Gamma}|\bot|^{n}_{\to} &= n \\
{}_{\Gamma}|a|^{n}_{\to} &= 0 \quad \text{when } a \in \Gamma \vee \widehat{a} \in \Gamma \\
{}_{\Gamma}|a|^{n}_{\to} &= n \quad \text{when } \tilde{a} \in \Gamma \\
{}_{\Gamma}|A \to B|^{n}_{\to} &= {}_{\Gamma}|A|^{n+1}_{\to} + {}_{\Gamma}|B|^{n+1}_{\to} \\
{}_{\Gamma}|\forall a.\, A|^{n}_{\to} &= {}_{\Gamma}|A|^{n}_{\to} + n \\
{}_{\Gamma}|A \sqcap B|^{n}_{\to} &= {}_{\Gamma}|A|^{n}_{\to} + {}_{\Gamma}|B|^{n}_{\to} + n \\
{}_{\Gamma}|A \sqcup B|^{n}_{\to} &= {}_{\Gamma}|A|^{n}_{\to} + {}_{\Gamma}|B|^{n}_{\to} + n
\end{aligned}$$

$$\begin{aligned}
{}_{\Gamma}|A \le B|_{\to} &= {}_{\Gamma}|A|_{\to} \cdot |B|_{\sqcup\sqcap a} + {}_{\Gamma}|B|_{\to} \cdot |A|_{\sqcup\sqcap a} \\
{}_{\Gamma}|w|_{\to} &= 0 \quad \text{otherwise}
\end{aligned}$$

$$|\Gamma|_{\to} = \sum_{w \in \Gamma} {}_{\Gamma}|w|_{\to}$$

$\boxed{|A|_{\le}, |w|_{\le}, |\Gamma|_{\le}}$ type size in subtyping judgment of A, w, and Γ

$$\begin{aligned}
|1|_{\le} = |\top|_{\le} = |\bot|_{\le} = |a|_{\le} &= 1 \\
|A \to B|_{\le} &= |A|_{\le} + |B|_{\le} + 1 \\
|\forall a.\, A|_{\le} &= |A|_{\le} + 2 \\
|A \sqcap B|_{\le} &= |A|_{\le} + |B|_{\le} + 1 \\
|A \sqcup B|_{\le} &= |A|_{\le} + |B|_{\le} + 1
\end{aligned}$$

$$\begin{aligned}
|A \le B|_{\le} &= |A|_{\le} \cdot |B|_{\sqcup\sqcap} + |B|_{\le} \cdot |A|_{\sqcup\sqcap} \\
|w|_{\le} &= 0 \quad \text{otherwise}
\end{aligned}$$

$$|\Gamma|_{\le} = \sum_{w \in \Gamma} |w|_{\le} + \sum_{x:A \in \Gamma} 1 + \sum_{a \in \Gamma} 1 + \sum_{\tilde{a} \in \Gamma} 1 + \sum_{\widehat{a} \in \Gamma} 1$$

Fig. S18. Definition of $|\cdot|_{\triangleright}$, $|\cdot|_{\to}$ and $|\cdot|_{\le}$

$\boxed{|\omega^s|_{\Rrightarrow_\omega}, |\omega^d|_{\Rrightarrow_\omega}}$ type-application judgment size of ω^s and ω^d

$$\begin{aligned}
|\_ \leq A|_{\Rrightarrow_\omega} &= 0 \\
|\_ \circ A \Rrightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} + 1 \\
|\_ \triangleright \omega^d|_{\Rrightarrow_\omega} &= |\omega^d|_{\Rrightarrow_\omega} \\
|\_ \Cup A \circ B \Rrightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} + 3 \\
|\_ \Cup_\circ A \blacktriangleright \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} + 1
\end{aligned}
\qquad
\begin{aligned}
|\_ \to \_ \bullet e \Rrightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} \\
|\_ \to \_ \Cup A \triangleright \omega^d|_{\Rrightarrow_\omega} &= |\omega^d|_{\Rrightarrow_\omega} \\
|\_ \to \_ \Cup_\triangleright (A \to B) \blacktriangleright \omega^d|_{\Rrightarrow_\omega} &= |\omega^d|_{\Rrightarrow_\omega}
\end{aligned}$$

$\boxed{|w|_{\Rrightarrow_\omega}, |\Gamma|_{\Rrightarrow_\omega}}$ type-application judgment size of w and Γ

$$\begin{aligned}
|A \leq B|_{\Rrightarrow_\omega} = |e \Leftarrow A|_{\Rrightarrow_\omega} &= 0 \\
|e \Rightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} \\
|A \circ B \Rrightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} + 1 \\
|A \triangleright \omega^d|_{\Rrightarrow_\omega} &= |\omega^d|_{\Rrightarrow_\omega} \\
|A \to B \bullet e \Rrightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega}
\end{aligned}
\qquad
\begin{aligned}
|C \Cup A \circ B \Rrightarrow \omega^s|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} + 3 \\
|(B \to C) \Cup A \triangleright \omega^d|_{\Rrightarrow_\omega} &= |\omega^d|_{\Rrightarrow_\omega} \\
|(A_1 \to B_1) \Cup_\triangleright (A_2 \to B_2) \blacktriangleright \omega^d|_{\Rrightarrow_\omega} &= |\omega^d|_{\Rrightarrow_\omega} \\
|(A_1 \Cup_\circ A_2 \blacktriangleright \omega^s)|_{\Rrightarrow_\omega} &= |\omega^s|_{\Rrightarrow_\omega} + 1 \\
|\Gamma|_{\Rrightarrow_\omega} &= \sum_{w \in \Gamma} |w|_{\Rrightarrow_\omega}
\end{aligned}$$

$\boxed{|\omega^s|_{\triangleright_\omega}, |\omega^d|_{\triangleright_\omega}}$ matching judgment size of ω^s and ω^d

$$\begin{aligned}
|\_ \leq A|_{\triangleright_\omega} &= 0 \\
|\_ \circ A \Rrightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|\_ \triangleright \omega^d|_{\triangleright_\omega} &= |\omega^d|_{\triangleright_\omega} + 1 \\
|\_ \Cup A \circ B \Rrightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|\_ \Cup_\circ A \blacktriangleright \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega}
\end{aligned}
\qquad
\begin{aligned}
|\_ \to \_ \bullet e \Rrightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|\_ \to \_ \Cup A \triangleright \omega^d|_{\triangleright_\omega} &= |\omega^d|_{\triangleright_\omega} + 3 \\
|\_ \to \_ \Cup_\triangleright (A \to B) \blacktriangleright \omega^d|_{\triangleright_\omega} &= |\omega^d|_{\triangleright_\omega} + 1
\end{aligned}$$

$\boxed{|w|_{\triangleright_\omega}, |\Gamma|_{\triangleright_\omega}}$ matching judgment size of w and Γ

$$\begin{aligned}
|A \leq B|_{\triangleright_\omega} = |e \Leftarrow A|_{\triangleright_\omega} &= 0 \\
|e \Rightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|A \circ B \Rrightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|A \triangleright \omega^d|_{\triangleright_\omega} &= |\omega^d|_{\triangleright_\omega} + 1 \\
|A \to B \bullet e \Rrightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega}
\end{aligned}
\qquad
\begin{aligned}
|C \Cup A \circ B \Rrightarrow \omega^s|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|(B \to C) \Cup A \triangleright \omega^d|_{\triangleright_\omega} &= |\omega^d|_{\triangleright_\omega} + 3 \\
|(A_1 \to B_1) \Cup_\triangleright (A_2 \to B_2) \blacktriangleright \omega^d|_{\triangleright_\omega} &= |\omega^d|_{\triangleright_\omega} + 1 \\
|(A_1 \Cup_\circ A_2 \blacktriangleright \omega^s)|_{\triangleright_\omega} &= |\omega^s|_{\triangleright_\omega} \\
|\Gamma|_{\triangleright_\omega} &= \sum_{w \in \Gamma} |w|_{\triangleright_\omega}
\end{aligned}$$

Fig. S19. Definition of $|\cdot|_{\Rrightarrow}$ and $|\cdot|_{\triangleright_\omega}$

E FORMALIZATION IN COQ

In this section, we discuss some details of our Coq formalization. One interesting point is that previous work on HRP using the worklist approach was done in Abella [Gacek 2008], which uses higher-order abstract syntax for representing variable binding. In Coq, we need to use a first-order representation, and we have chosen the locally nameless representation [Charguéraud 2012] for that. The choice of first-order variable binding also has an impact on the representation of continuations. File structure of the proof and the corresponding code of the definitions and theorems listed in the paper are also provided.

E.1 Locally Nameless Representation

Since Coq does not have a native support for binding structure as Abella does, we pick the locally nameless representation as the technique to deal with them manually. Compared to De Bruijn indices, locally nameless representation is more flexible in modifying the context. This is crucial for our formalization since worklist substitution often reorders the worklist in a very complex way. We choose Ott [Sewell et al. 2007] and LNgen [Aydemir and Weirich 2010] as the tool chain to automatically generate definitions and lemmas related to locally nameless representations.

Renaming lemma. One special proof burden of locally nameless is the renaming lemma, i.e., a property hold by renaming a freee variable to another fresh name. Though co-finite quantification reduces the need for renaming lemmas in various positions, we find two patterns where the renaming lemma is indispensable: the first being, $P(x) \vee \neg P(x)$; the second being $\exists x, P(x)$, where x could be anything containining something that uses binders. For example, x can be worklist, works, types or expressions. The first pattern appears in the proof of decidability. The induction hypothesis generated for these two patterns are $forall\ a, a \notin \mathsf{L} \supset (P(x'(a)) \vee \neg P(x'(a)))$ and $forall\ a, a \notin \mathsf{L} \supset (exists\ x', P(x'(a)))$. However, what we want is usually $(forall\ a, a \notin \mathsf{L} \supset P(x'(a)) \vee (forall\ a, a \notin \mathsf{L} \supset \neg P(x))$ and $exists\ x', (forall\ a, a \notin \mathsf{L} \supset P(x'(a)))$. Since such a distributivity for logical connective is not true for arbitrary P, we have to manually prove renaming lemmas. For example, the renaming lemma for worklsit reduction, which is used in the decidability proof, is as follows:

Lemma E.1 (Worklist Reduction Renaming). *If* $\vdash \Gamma, b \notin fv(\Gamma)$, *and* $\Gamma \longrightarrow^* \cdot$, *then* $[b/a]\Gamma \longrightarrow^* \cdot$

E.2 Benefits of Coq over Abella

While the variable binding is a strength of Abella compared to Coq, we found the use of Coq beneficial in several ways, including (1). custom tactics; (2). automatic arithmetic reasoning by Lia; (3). functional definition; (4). notation system.

E.3 File Structure of Proof

In this section, we present an overview of our formalization in Coq. Proofs of $F^e_{\sqcup\sqcap}$ and its variants can be found in the supplementary materials. The proof script for the base $F^e_{\sqcup\sqcap}$ consists of 40669 lines of Coq code, 18,262 of which are generated by Ott and LNgen. Two variants consist of fewer lines of code since we do not have decidability proof and type safety proof for them. The file structure of the proofs of base $F^e_{\sqcup\sqcap}$ is shown below, under the `proof/uni` directory. The proof of $F^e_{\sqcup\sqcap}$ with records is under the `proof/uni_rec` directory and the proof of $F^e_{\sqcup\sqcap}$ with records and instantiable intersection/union types is under the `proof/uni_monoiu` directory.

Translation Table for the Proof Scripts. Table 2 and 3 provides a mapping from symbolic names in the paper to the textual names in the proof script. However, thanks to the notations system in Coq, most definitions are already written with a similar sytnax as in the paper.

- `proof`
  - `uni`
    - `decl`
      - `def_extra.v` definitions of matching, type application, checking and inference
      - `prop_basic.v`
      - `prop_rename.v` renaming lemmas
      - `prop_safety.v` definition of elaboration, type safety
      - `prop_subtyping.v` properties of subtyping, including transitivity)
      - `prop_typing.v` properties of subtyping, including checking subsumption
    - `decl_worklist`
      - `def.v` definitions of reduction rules
      - `prop_equiv.v` equivalence between two reduction rules
    - `algo_worklist`
      - `def_extra.v` definitions of reduction rules
      - `prop_basic.v`
      - `prop_completeness.v` completeness of the algorithmic worklist reduction
      - `prop_decidability.v` measure definition and decidability of the algorithmic worklist reduction
      - `prop_rename.v` renaming lemmas
      - `prop_soundness.v` soundness of the algorithmic worklist reduction
      - `transfer.v` definitions and properties of syntax-directed transfer
    - `def_ott.v` syntax definitions generated by Ott
    - `prop_ln.v` locally nameless properties generated by LNgen
    - `notations.v` definitions of notations
    - `ltac_utils.v`
  - `systemf`
    - `def_ott.v` syntax definitions generated by Ott
    - `prop_ln.v` locally nameless properties generated by LNgen

Table 2. Mapping Table from Symbolic Definitions in the Paper to the Definition Names in the Proof

Symbol	Coq name
A	typ
τ	d_mono_typ, a_mono_typ
$A^{\neq\forall}$	d_wneq_forall, a_wneq_forall
$A^{\not\perp}$	bot_free
e	exp
ω^s, ω^d	conts, contd
w	work
Ψ	denv
Σ	aenv
Ξ	nenv
Δ	fenv
$\Psi' \leq \Psi$	d_subenv
$\Psi \vdash A \leq B$	d_sub
$\Psi \vdash e \Leftrightarrow A$	d_chk_inf
$\Psi \vdash A \triangleright B \to C$	d_infabs
$\Psi \vdash A \circ B \Rrightarrow C$	d_inftapp
$\{\tau/a\}\Gamma$	aworklist_subst
$\Gamma \longrightarrow^* \cdot$	a_wl_red
$\Omega \longrightarrow^* \cdot$	d_wl_del_red
$\Omega \longrightarrow^* \cdot$	d_wl_red
θ	subst_set
$\theta \vDash \Gamma \rightsquigarrow \Omega \dashv \theta'$	transfer
$\lvert \cdot \rvert_e$	exp_size, exp_size_conts/d, exp_size_work, exp_size_wl
$\lvert \cdot \rvert_\omega$	judge_size_work, judge_size_wl
$\lvert \cdot \rvert_{\Rrightarrow}$	inftapp_depth, inftapp_depth_conts/d, inftapp_depth_work, inftapp_depth_wl
$\lvert \cdot \rvert_{\Rrightarrow\omega}$	inftapp_judge_size_work, inftapp_judge_size_wl
$\lvert \cdot \rvert_{\triangleright}$	infabs_depth, infabs_depth_conts/d, infabs_depth_work, infabs_depth_wl
$\lvert \cdot \rvert_{\triangleright\omega}$	infabs_judge_size_work, infabs_judge_size_wl
$\lvert \cdot \rvert_{\to}$	split_depth, split_depth_work, split_depth_wl
$\lvert \cdot \rvert_{\widehat{a}}$	num_etvar_wl
$\lvert \cdot \rvert_{\leq}$	weight, weight_work, weight_wl
$\lvert\Psi\rvert \vdash_{F_{\times+}} e : A$	f_typing

Table 3. Mapping Table from Theorems in the Paper to the Theorem Names in the Proof

Symbol	Coq name
Theorem 2.1	d_sub_inst
Theorem 2.2	d_chk_subsumption
Theorem 3.1	d_sub_reflexivity
Theorem 3.2	d_sub_transitivity
Lemma 3.3	d_infabs_soundness, d_infabs_completeness
Lemma 3.4	d_inftapp_soundness1, d_inftapp_completness
Theorem 4.1	a_wl_red_chk_soundness, a_wl_red_inf_soundness
	a_wl_red_chk_completeness, a_wl_red_inf_completeness
Theorem 4.2	a_wf_wl_red_decidable_thm
Theorem 5.1	d_sub_subst_stvar
Theorem 5.2	d_infabs_subsumption_same_env, d_inftapp_subsumption_same_env
Theorem 5.3	d_chk_inf_subsumption
Theorem 5.4	d_chk_inf_elab_sound_f
Theorem 5.5	d_wl_red_sound, d_wl_red_complete
Theorem 5.6	a_wl_red_soundness
Theorem 5.7	a_wl_red_completeness
Lemma 5.8	aworklist_subst_transfer_same_dworklist_rev
	aworklist_subst_transfer_same_dworklist
Lemma 5.9	trans_typ_etvar_s_in_more_num_arrow
	d_sub_more_num_arrow_in_mono_typ

REFERENCES

Brian Aydemir and Stephanie Weirich. 2010. LNgen: Tool support for locally nameless representations. *Draft available at http://www. cis. upenn. edu/˜ sweirich/papers/lngen* 61 (2010), 108.

Arthur Charguéraud. 2012. The Locally Nameless Representation. *Journal of Automated Reasoning* 49, 3 (01 Oct 2012), 363–408. https://doi.org/10.1007/s10817-011-9225-2

Chen Cui, Shengyi Jiang, and Bruno C. d. S. Oliveira. 2023. Greedy Implicit Bounded Quantification. *Proc. ACM Program. Lang.* 7, OOPSLA2, Article 295 (oct 2023), 29 pages. https://doi.org/10.1145/3622871

Jana Dunfield. 2014. Elaborating intersection and union types. *J. Funct. Program.* 24, 2-3 (2014), 133–165. https://doi.org/10.1017/S0956796813000270

Andrew Gacek. 2008. The Abella Interactive Theorem Prover (System Description). In *Automated Reasoning*, Alessandro Armando, Peter Baumgartner, and Gilles Dowek (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 154–161.

John C. Reynolds. 1972. Definitional interpreters for higher-order programming languages. In *Proceedings of the ACM Annual Conference - Volume 2* (Boston, Massachusetts, USA) *(ACM '72)*. Association for Computing Machinery, New York, NY, USA, 717–740. https://doi.org/10.1145/800194.805852

Peter Sewell, Francesco Zappa Nardelli, Scott Owens, Gilles Peskine, Thomas Ridge, Susmit Sarkar, and Rok Strniša. 2007. Ott: effective tool support for the working semanticist. In *Proceedings of the 12th ACM SIGPLAN International Conference on Functional Programming* (Freiburg, Germany) *(ICFP '07)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/1291151.1291155