

Complete and Easy Quantitative Contextual Typing

Wojciech Kołowski
Mateusz Pyzik

Types

Types:

$A, B ::=$

$A \multimap B \mid !_r A \mid A \otimes B \mid A \oplus B \mid \text{Unit} \mid \text{Empty} \mid$
 $a \mid \tilde{a} \mid \hat{a} \mid \forall @a : \text{Type}_r. A \mid \forall a : \text{Type}_r. A$

Monotypes:

$\tau ::=$

$\tau_1 \multimap \tau_2 \mid !_r \tau \mid \tau_1 \otimes \tau_2 \mid \tau_1 \oplus \tau_2 \mid \text{Unit} \mid \text{Empty} \mid$
 $a \mid \hat{a} \mid \forall @a : \text{Type}_r. \tau$

Terms

Terms:

$e ::=$

$(e : A) \mid x \mid \lambda x. e \mid \lambda x : A. e \mid e_1 \ e_2 \mid e \ @A \mid$
 $\Lambda a. e \mid \Lambda a : \text{Type}_r. e \mid e \ A \mid$
 $(e_1, e_2) \mid \text{let } (x, y) = e_1 \text{ in } e_2 \mid$
 $\text{inl } e \mid \text{inr } e \mid \text{case } e \text{ of } \{x.e_1; y.e_2\} \mid$
 $\text{unit} \mid \text{let unit} = e_1 \text{ in } e_2 \mid$
 $\text{Empty-elim } e \mid$
 $\text{let } x = e_1 \text{ in } e_2 \mid \text{let } x : A = e_1 \text{ in } e_2 \mid$

Contexts

Contexts:

$\Gamma ::=$

$\cdot \mid \Gamma, r x : A \mid \Gamma, a : \text{Type}_r \mid \Gamma, \tilde{a} : \text{Type}_r \mid$
 $\Gamma, \hat{a} \mid \Gamma, \hat{a} = \tau \mid \Gamma, \blacktriangleright_a \mid$
 $\Gamma, \square \mid \Gamma, \square : A \mid \Gamma, \square e \mid \Gamma, \square @A \mid \Gamma, \square A$

Context operations and notations

- Γ, \square, Δ is a notation which means that Δ does not contain any hints. It works not only for \square , but for all kinds of hints.
- $\Gamma[\hat{a}]$ is a notation for Γ, \hat{a}, Γ' . We use it for checking variable presence and for updating existential variables in the context.
- $\langle \Gamma \rangle A$ substitutes all equations $\hat{a} = \tau$ from Γ in A .

Judgements

- $\Gamma \vdash r \Rightarrow A \Rightarrow s \dashv \Gamma'$ – kinding
- $\Gamma \vdash e \Rightarrow A \dashv \Gamma'$ – typing
- $\Gamma \vdash A \rightsquigarrow B \dashv \Gamma'$ – matching
- $\Gamma \vdash r \Rightarrow A <: B \Rightarrow s \dashv \Gamma'$ – subtyping
- $\Gamma \vdash r \Rightarrow \hat{a} := A \Rightarrow s \dashv \Gamma'$ – sub-instantiation
- $\Gamma \vdash r \Rightarrow A =: \hat{a} \Rightarrow s \dashv \Gamma'$ – super-instantiation

Kinding judgement

$\Gamma \vdash r \Rightarrow A \Rightarrow s \dashv \Gamma'$ – in context Γ , check if type A is of kind Type_r . If it is, s will be 1. If it is not, s indicates how much A is missing to be in Type_r , i.e. A is in $\text{Type}_{r/s}$. Return output context Γ' .

Note: Γ and r are inputs. A is subject. s and Γ' are outputs.

$\Gamma \vdash r \Rightarrow A \dashv \Gamma'$ – kind checking is a notation for
 $\Gamma \vdash r \Rightarrow A \Rightarrow 1 \dashv \Gamma'$.

$\Gamma \vdash A \Rightarrow r \dashv \Gamma'$ – kind inference is a notation for
 $\Gamma \vdash * \Rightarrow A \Rightarrow s \dashv \Gamma'$ with $r = */s$

Kinding – variables and quantifiers

$$\frac{}{\Gamma[a : \text{Type}_s] \vdash r \Rightarrow a \Rightarrow r/s \dashv \Gamma[a : \text{Type}_s]}$$

$$\frac{}{\Gamma[\tilde{a} : \text{Type}_s] \vdash r \Rightarrow \tilde{a} \Rightarrow r/s \dashv \Gamma[\tilde{a} : \text{Type}_s]}$$

$$\frac{}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} \Rightarrow 1 \dashv \Gamma[\hat{b}, \hat{a} = !_r \hat{b}]}$$

$$\frac{\Gamma, a : \text{Type}_s \vdash r \Rightarrow A \Rightarrow t \dashv \Gamma'}{\Gamma \vdash r \Rightarrow \forall a : \text{Type}_s. A \Rightarrow t \dashv \Gamma'}$$

$$\frac{\Gamma, a : \text{Type}_s \vdash r \Rightarrow A \Rightarrow t \dashv \Gamma'}{\Gamma \vdash r \Rightarrow \forall @a : \text{Type}_s. A \Rightarrow t \dashv \Gamma'}$$

Kinding – functions and boxes

$$\frac{\Gamma \vdash 1 \Rightarrow A \Rightarrow 1 \dashv \Gamma_1 \quad \Gamma_1 \vdash 1 \Rightarrow \langle \Gamma_1 \rangle B \Rightarrow 1 \dashv \Gamma_2}{\Gamma \vdash r \Rightarrow A \multimap B \Rightarrow r \dashv \Gamma_2}$$

$$\frac{\Gamma \vdash 1 \Rightarrow A \Rightarrow t \dashv \Gamma'}{\Gamma \vdash r \Rightarrow !_0 A \Rightarrow 1 \dashv \Gamma'}$$

$$\frac{\Gamma \vdash r/s \Rightarrow A \Rightarrow t \dashv \Gamma'}{\Gamma \vdash r \Rightarrow !_s A \Rightarrow t \dashv \Gamma'}$$

Kinding – remaining type formers

$$\overline{\Gamma \vdash r \Rightarrow \text{Unit} \Rightarrow 1 \dashv \Gamma} \quad \overline{\Gamma \vdash r \Rightarrow \text{Empty} \Rightarrow 1 \dashv \Gamma}$$

$$\frac{\Gamma \vdash r \Rightarrow A \Rightarrow s_A \dashv \Gamma_1 \quad \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle B \Rightarrow s_B \dashv \Gamma_2}{\Gamma \vdash r \Rightarrow A \otimes B \Rightarrow s_A \sqcap s_B \dashv \Gamma_2}$$

$$\frac{\Gamma \vdash r \Rightarrow A \Rightarrow s_A \dashv \Gamma_1 \quad \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle B \Rightarrow s_B \dashv \Gamma_2}{\Gamma \vdash r \Rightarrow A \oplus B \Rightarrow s_A \sqcap s_B \dashv \Gamma_2}$$

Typing judgement

$\Gamma \vdash e \Rightarrow A \dashv \Gamma'$ – in context Γ , term e infers type A , returning output context Γ'

Note: Γ is input. e is subject. A and Γ' are outputs.

Note: “inference mode” is $\Gamma, \square \vdash e \Rightarrow A \dashv \Gamma'$, as it requires the empty hint.

Note: “checking mode” is $\Gamma, \square : A \vdash e \Rightarrow _ \dashv \Gamma'$, as it requires a type hint $\square : A$, but does not use the inferred type.

Typing – variables and annotations

$$\frac{\Gamma(x) = A \quad 0\Gamma + x \vdash \langle 0\Gamma + x \rangle A \rightsquigarrow B \dashv \Gamma'}{\Gamma \vdash_c x \Rightarrow B \dashv \Gamma' + \Gamma}$$

$$\frac{\Gamma(x) = A \quad \Gamma \vdash \langle \Gamma \rangle A \rightsquigarrow B \dashv \Gamma'}{\Gamma \vdash_{nc} x \Rightarrow B \dashv \Gamma'}$$

$$\frac{\Gamma \vdash 1 \Rightarrow A \Rightarrow 1 \dashv \Gamma_1 \quad 0\Gamma_1, \square : A \vdash e \Rightarrow - \dashv \Gamma_2 \quad \Gamma_2 \vdash \langle \Gamma_2 \rangle A \rightsquigarrow B \dashv \Gamma_3}{\Gamma \vdash (e : A) \Rightarrow B \dashv \Gamma_3 + \Gamma_1}$$

Typing – stationary rules

$$\frac{\Gamma, \Delta, a : \text{Type}_r, \square : A \vdash e \Rightarrow _ \dashv \Gamma', a : \text{Type}_r, \Theta}{\Gamma, \square : \forall a : \text{Type}_r. A, \Delta \vdash e \Rightarrow \forall a : \text{Type}_r. A \dashv \Gamma'}$$

$$\frac{0\Gamma, 0\Delta, \square : A \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma, \square : !_r A, \Delta \vdash e \Rightarrow !_r A \dashv r\Gamma' + (\Gamma, \Delta)}$$

Typing – application(s)

$$\frac{\Gamma, \Box e_2 \vdash e_1 \Rightarrow A \dashv \Gamma'}{\Gamma \vdash e_1 e_2 \Rightarrow A \dashv \Gamma'}$$

$$\frac{\Gamma, \Box @A \vdash e \Rightarrow B \dashv \Gamma'}{\Gamma \vdash e @A \Rightarrow B \dashv \Gamma'}$$

$$\frac{\Gamma, \Box A \vdash e \Rightarrow B \dashv \Gamma'}{\Gamma \vdash e A \Rightarrow B \dashv \Gamma'}$$

Typing – unannotated abstraction

$$\frac{\begin{array}{l} \Gamma, \Delta, 0x : A, \square : B \vdash e \Rightarrow _ \dashv \Gamma_1, rx : A, \Theta \\ \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle A \Rightarrow 1 \dashv \Gamma_2 \end{array}}{\Gamma, \square : A \multimap B, \Delta \vdash \lambda x. e \Rightarrow A \multimap B \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \multimap \hat{a}_2], \Delta, 0x : \hat{a}_1, \square : \hat{a}_2 \vdash e \Rightarrow _ \dashv \Gamma_1, rx : \hat{a}_1, \Theta \\ \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle \hat{a}_1 \Rightarrow 1 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}], \square : \hat{a}, \Delta \vdash \lambda x. e \Rightarrow \hat{a} \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} 0\Gamma, \square \vdash e' \Rightarrow A \dashv \Gamma_1 \quad 0\Gamma_1, \Delta, 0x : A \vdash e \Rightarrow B \dashv \Gamma_2, rx : A, \Theta \\ \Gamma_2 \vdash r \Rightarrow \langle \Gamma_2 \rangle A \Rightarrow s \dashv \Gamma_3 \end{array}}{\Gamma, \square e', \Delta \vdash \lambda x. e \Rightarrow B \dashv \Gamma_3 + s \Gamma_1 + \Gamma}$$

$$\frac{\begin{array}{l} \Gamma, \Delta \blacktriangleright_{\hat{a}}, \hat{a}, 0x : \hat{a}, \square \vdash e \Rightarrow A \dashv \Gamma_1, rx : \hat{a}, \Gamma_4 \\ \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle \hat{a} \Rightarrow 1 \dashv \Gamma_2, \blacktriangleright_{\hat{a}}, \Gamma_3 \\ B = \forall \text{unsolved}(\Gamma_3). \langle \Gamma_3 \rangle (\hat{a} \multimap \forall \text{unsolved}(\Gamma_4). \langle \Gamma_4 \rangle A) \end{array}}{\Gamma, \square, \Delta \vdash \lambda x. e \Rightarrow B \dashv \Gamma_2}$$

Typing – annotated abstraction

$$\begin{array}{c}
 \Gamma \vdash * \Rightarrow A \Rightarrow r \dashv \Gamma_1 \\
 0 \Gamma_1, \square : A \vdash e' \Rightarrow _ \dashv \Gamma_2 \\
 0 \Gamma_2, \Delta, 0x : A \vdash e \Rightarrow B \dashv \Gamma_3, sx : A, \Theta \\
 */r \sqsubseteq s \\
 \hline
 \Gamma, \square e', \Delta \vdash \lambda x : A. e \Rightarrow B \dashv \Gamma_3 + s/(*/r) \Gamma_2 + \Gamma_1
 \end{array}$$

$$\begin{array}{c}
 \Gamma \vdash * \Rightarrow A \Rightarrow r \dashv \Gamma_1 \\
 \Gamma_1, \Delta, 0x : A, \square \vdash e \Rightarrow B \dashv \Gamma_2, sx : A, \Theta \\
 */r \sqsubseteq s \\
 \hline
 \Gamma, \square, \Delta \vdash \lambda x : A. e \Rightarrow A \multimap B \dashv \Gamma_2
 \end{array}$$

Typing – let

- We define $\text{let } x = e_1 \text{ in } e_2$ as $(\lambda x. e_2) e_1$
- We define $\text{let } x : A = e_1 \text{ in } e_2$ as $(\lambda x : A. e_2) e_1$.

The following rules are then admissible:

$$\begin{array}{c}
 0 \Gamma, \square \vdash e_1 \Rightarrow A \dashv \Gamma_1 \\
 0 \Gamma_1, 0 x : A \vdash e_2 \Rightarrow B \dashv \Gamma_2, r x : A \\
 \Gamma_2 \vdash r \Rightarrow A \Rightarrow s \dashv \Gamma_3 \\
 \hline
 \Gamma \vdash \text{let } x = e_1 \text{ in } e_2 \Rightarrow B \dashv \Gamma_3 + s \Gamma_1 + \Gamma
 \end{array}$$

$$\begin{array}{c}
 0 \Gamma, \square : A \vdash e_1 \Rightarrow _ \dashv \Gamma_1 \\
 0 \Gamma_1, 0 x : A \vdash e_2 \Rightarrow B \dashv \Gamma_2, r x : A \\
 \Gamma_2 \vdash r \Rightarrow A \Rightarrow s \dashv \Gamma_3 \\
 \hline
 \Gamma \vdash \text{let } x : A = e_1 \text{ in } e_2 \Rightarrow B \dashv \Gamma_3 + s \Gamma_1 + \Gamma
 \end{array}$$

Typing – explicit quantification

$$\frac{\Gamma, \Delta, a : \text{Type}_r, \square : A \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma, \square : \forall @a : \text{Type}_r. A, \Delta \vdash \Lambda a. e \Rightarrow \forall @a : \text{Type}_r. A \dashv \Gamma'}$$

$$\frac{\Gamma, \Delta, a : \text{Type}_r, \square \vdash e \Rightarrow A \dashv \Gamma'}{\Gamma, \square, \Delta \vdash \Lambda a : \text{Type}_r. e \Rightarrow \forall @a : \text{Type}_r. A \dashv \Gamma'}$$

$$\frac{\Gamma \vdash * \Rightarrow B \Rightarrow r \dashv \Gamma_1 \quad \Gamma_1, \Delta, a : \text{Type}_{*/r} \vdash e \Rightarrow A \dashv \Gamma_2}{\Gamma, \square B, \Delta \vdash \Lambda a. e \Rightarrow \forall @a : \text{Type}_{*/r}. A \dashv \Gamma_2}$$

Typing – Unit

$$\frac{\Gamma \vdash \text{Unit} \rightsquigarrow _ \dashv \Gamma'}{\Gamma \vdash \text{unit} \Rightarrow \text{Unit} \dashv \Gamma'}$$

$$\frac{\Gamma, \Box : \text{Unit} \vdash e_1 \Rightarrow _ \dashv \Gamma_1 \quad \Gamma_1 \vdash e_2 \Rightarrow A \dashv \Gamma_2}{\Gamma \vdash \text{let unit} = e_1 \text{ in } e_2 \Rightarrow A \dashv \Gamma_2}$$

Typing – short weak rules for constants

$$\frac{\Gamma \vdash \text{Empty} \multimap \forall a : \text{Type}_1. a \rightsquigarrow C \dashv \Gamma'}{\Gamma \vdash \text{Empty-elim} \Rightarrow C \dashv \Gamma'}$$

$$\frac{\Gamma \vdash \forall a : \text{Type}_1. a \multimap \forall b : \text{Type}_1. b \multimap a \otimes b \rightsquigarrow C \dashv \Gamma'}{\Gamma \vdash \text{pair} \Rightarrow C \dashv \Gamma'}$$

$$\frac{\Gamma \vdash \forall a : \text{Type}_1. a \multimap \forall b : \text{Type}_1. a \oplus b \rightsquigarrow C \dashv \Gamma'}{\Gamma \vdash \text{inl} \Rightarrow C \dashv \Gamma'}$$

$$\frac{\Gamma \vdash \forall b : \text{Type}_1. b \multimap \forall a : \text{Type}_1. a \oplus b \rightsquigarrow C \dashv \Gamma'}{\Gamma \vdash \text{inr} \Rightarrow C \dashv \Gamma'}$$

Typing – short weak rules for constants (explanation)

The above rules treat introduction/elimination forms for various types as constants. In each rule, the type of the constant only needs to be matched with hints from the context. These rules are short and elegant, but somewhat weak, because the quantifiers can only be instantiated with monotypes. For this reason, below we will also give longer, but more powerful rules that treat these introduction/elimination forms as primitives.

Typing – empty elimination

$$\frac{\Gamma, \Delta, \square : \text{Empty} \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma, \square : A, \Delta \vdash \text{Empty-elim } e \Rightarrow A \dashv \Gamma'}$$

$$\frac{\Gamma, \square \vdash e' \Rightarrow A \dashv \Gamma_1 \quad \Gamma_1, \Delta \vdash \text{Empty-elim } e \Rightarrow B \dashv \Gamma_2}{\Gamma, \square e', \Delta \vdash \text{Empty-elim } e \Rightarrow B \dashv \Gamma_2}$$

$$\frac{\Gamma \vdash 1 \Rightarrow A \Rightarrow 1 \dashv \Gamma_1 \quad \Gamma_1, \Delta \vdash \text{Empty-elim } e \Rightarrow B \dashv \Gamma_2}{\Gamma, \square @A, \Delta \vdash \text{Empty-elim } e \Rightarrow B \dashv \Gamma_2}$$

$$\frac{\Gamma \vdash 1 \Rightarrow A \Rightarrow 1 \dashv \Gamma_1 \quad \Gamma_1, \Delta \vdash \text{Empty-elim } e \Rightarrow B \dashv \Gamma_2}{\Gamma, \square A, \Delta \vdash \text{Empty-elim } e \Rightarrow B \dashv \Gamma_2}$$

Typing – pairs

$$\frac{\Gamma, \Delta, \square : A \vdash e_1 \Rightarrow _ \dashv \Gamma_1 \quad \Gamma_1, \square : B \vdash e_2 \Rightarrow _ \dashv \Gamma_2}{\Gamma, \square : A \otimes B, \Delta \vdash (e_1, e_2) \Rightarrow A \otimes B \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \otimes \hat{a}_2], \Delta, \square : \hat{a}_1 \vdash e_1 \Rightarrow _ \dashv \Gamma_1 \\ \Gamma_1, \square : \hat{a}_2 \vdash e_2 \Rightarrow _ \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}], \square : \hat{a}, \Delta \vdash (e_1, e_2) \Rightarrow \hat{a} \dashv \Gamma_2}$$

$$\frac{\Gamma, \Delta, \square \vdash e_1 \Rightarrow A \dashv \Gamma_1 \quad \Gamma_1, \square \vdash e_2 \Rightarrow B \dashv \Gamma_2}{\Gamma, \square, \Delta \vdash (e_1, e_2) \Rightarrow A \otimes B \dashv \Gamma_2}$$

Typing – product elimination

$$0 \Gamma, \square \vdash e_1 \Rightarrow A \otimes B \dashv \Gamma_1$$

$$0 \Gamma_1, 0 x : A, 0 y : B \vdash e_2 \Rightarrow C \dashv \Gamma_2, r_A x : A, r_B y : B$$

$$\Gamma_2 \vdash r_A \Rightarrow A \Rightarrow s_A \dashv \Gamma_3$$

$$\Gamma_3 \vdash r_B \Rightarrow B \Rightarrow s_B \dashv \Gamma_4$$

$$\frac{}{\Gamma \vdash \text{let } (x, y) = e_1 \text{ in } e_2 \Rightarrow C \dashv \Gamma_4 + (s_A \sqcap s_B) \Gamma_1 + \Gamma}$$

Typing – sum constructors

$$\frac{\Gamma, \Delta, \square : A \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma, \square : A \oplus B, \Delta \vdash \text{inl } e \Rightarrow A \oplus B \dashv \Gamma'}$$

$$\frac{\Gamma, \Delta, \square : B \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma, \square : A \oplus B, \Delta \vdash \text{inr } e \Rightarrow A \oplus B \dashv \Gamma'}$$

$$\frac{\Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \oplus \hat{a}_2], \Delta, \square : \hat{a}_1 \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma[\hat{a}], \square : \hat{a}, \Delta \vdash \text{inl } e \Rightarrow \hat{a} \dashv \Gamma'}$$

$$\frac{\Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \oplus \hat{a}_2], \Delta, \square : \hat{a}_2 \vdash e \Rightarrow _ \dashv \Gamma'}{\Gamma[\hat{a}], \square : \hat{a}, \Delta \vdash \text{inr } e \Rightarrow \hat{a} \dashv \Gamma'}$$

Typing – sum constructors

$$\frac{\Gamma, \square, \Delta \vdash e \Rightarrow A \dashv \Gamma'}{\Gamma, \square, \Delta \vdash \text{inl } e \Rightarrow \forall b : \text{Type}_1. A \oplus b \dashv \Gamma'}$$

$$\frac{\Gamma, \square, \Delta \vdash e \Rightarrow B \dashv \Gamma'}{\Gamma, \square \Delta \vdash \text{inr } e \Rightarrow \forall a : \text{Type}_1. a \oplus B \dashv \Gamma'}$$

Typing – sum elimination

$$0\Gamma, \square \vdash e \Rightarrow A \oplus B \dashv \Gamma_1$$

$$0\Gamma_1, 0x : A \vdash e_1 \Rightarrow C_1 \dashv \Gamma_2, r_A x : A$$

$$0\Gamma_1, 0y : B \vdash e_2 \Rightarrow C_2 \dashv \Gamma_3, r_B y : B$$

$$\Gamma_4 = \Gamma_2 \sqcap \Gamma_3$$

$$\Gamma_4 \vdash 1 \Rightarrow C_1 <: C_2 \Rightarrow 1 \dashv \Gamma_5$$

$$\Gamma_5 \vdash 1 \Rightarrow C_2 <: C_1 \Rightarrow 1 \dashv \Gamma_6$$

$$\Gamma_6 \vdash r_A \Rightarrow A \Rightarrow s_A \dashv \Gamma_7$$

$$\Gamma_7 \vdash r_B \Rightarrow B \Rightarrow s_B \dashv \Gamma_8$$

$$\frac{}{\Gamma \vdash \text{case } e \text{ of } \{x.e_1; y.e_2\} \Rightarrow C_1 \dashv \Gamma_8 + (s_A \sqcap s_B) \Gamma_1 + \Gamma}$$

Matching judgement

Matching: $\Gamma \vdash A \rightsquigarrow B \dashv \Gamma'$ – type A matches context Γ , with output type B and output context Γ'

Note: Γ and A are inputs. B and Γ' are outputs.

Matching – inference and checking

$$\overline{\Gamma, \Box, \Delta \vdash A \rightsquigarrow A \dashv \Gamma, \Delta}$$

$$\frac{\Gamma, \Delta \vdash 1 \Rightarrow A <: B \Rightarrow r \dashv \Gamma'}{\Gamma, \Box : B, \Delta \vdash A \rightsquigarrow B \dashv r \Gamma'}$$

Matching – term argument hints

$$\frac{\Gamma, \Box : A \vdash e \Rightarrow _ \dashv \Gamma_1 \quad \Gamma_1, \Delta \vdash B \rightsquigarrow C \dashv \Gamma_2}{\Gamma, \Box e, \Delta \vdash A \multimap B \rightsquigarrow C \dashv \Gamma_2}$$

$$\frac{\Gamma, \Box e, \Delta \vdash A \rightsquigarrow C \dashv \Gamma'}{\Gamma, \Box e, \Delta \vdash !_r A \rightsquigarrow C \dashv \Gamma'}$$

$$\frac{\Gamma, \Box e, \Delta, \hat{a} \vdash A[a := !_r \hat{a}] \rightsquigarrow C \dashv \Gamma'}{\Gamma, \Box e, \Delta \vdash \forall a : \text{Type}_r. A \rightsquigarrow C \dashv \Gamma'}$$

Matching – implicit type argument hints

$$\frac{\Gamma \vdash r \Rightarrow B \Rightarrow 1 \dashv \Gamma_1 \quad \Gamma_1, \Delta \vdash A[a := B] \rightsquigarrow C \dashv \Gamma_2}{\Gamma, \square @B, \Delta \vdash \forall a : \text{Type}_r. A \rightsquigarrow C \dashv \Gamma_2}$$

$$\frac{\Gamma, \square @B, \Delta \vdash A \rightsquigarrow C \dashv \Gamma'}{\Gamma, \square @B, \Delta \vdash !_r A \rightsquigarrow C \dashv \Gamma'}$$

Matching – explicit type argument hints

$$\frac{\Gamma \vdash r \Rightarrow B \Rightarrow 1 \dashv \Gamma_1 \quad \Gamma_1, \Delta \vdash A[a := B] \rightsquigarrow C \dashv \Gamma_2}{\Gamma, \Box B, \Delta \vdash \forall @a : \text{Type}_r. A \rightsquigarrow C \dashv \Gamma_2}$$

$$\frac{\Gamma, \Box B, \Delta \vdash A \rightsquigarrow C \dashv \Gamma'}{\Gamma, \Box B, \Delta \vdash !_r A \rightsquigarrow C \dashv \Gamma'}$$

$$\frac{\Gamma, \Box B, \Delta, \hat{a} \vdash A[a := !_r \hat{a}] \rightsquigarrow C \dashv \Gamma'}{\Gamma, \Box B, \Delta \vdash \forall a : \text{Type}_r. A \rightsquigarrow C \dashv \Gamma'}$$

Subtyping judgement

$\Gamma \vdash r \Rightarrow A <: B \Rightarrow s \dashv \Gamma'$ – in context Γ , A is a subtype of $!_{r/s} B$, with output context Γ' .

Note: Γ , r , A and B are inputs. s and Γ' are outputs.

Note: If A is a subtype of $!_r B$, then s is 1. If it is not, then s indicates how much is missing for the subtyping to be the case.

Subtyping – variables

$$\frac{}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} <: \hat{a} \Rightarrow 1 \dashv \vdash \Gamma[\hat{b}, \hat{a}!_{[r]}b]}$$

$$\frac{}{\Gamma[a : \text{Type}_s] \vdash r \Rightarrow a <: a \Rightarrow r/s \dashv \vdash \Gamma[a : \text{Type}_s]}$$

$$\frac{}{\Gamma[\tilde{a} : \text{Type}_s] \vdash r \Rightarrow \tilde{a} <: \tilde{a} \Rightarrow r/s \dashv \vdash \Gamma[\tilde{a} : \text{Type}_s]}$$

Subtyping – quantifiers

$$\frac{\Gamma, \tilde{a} : \text{Type}_s \vdash r \Rightarrow A[a := \tilde{a}] <: B[a := \tilde{a}] \Rightarrow t \dashv \Gamma', \tilde{a}, \Theta}{\Gamma \vdash r \Rightarrow \forall a : \text{Type}_s. A <: \forall a : \text{Type}_s. B \Rightarrow t \dashv \Gamma'}$$

$$\frac{\begin{array}{l} a \in \text{FV}(A) \\ B \neq \forall b : \text{Type}_{s'}. B' \end{array} \quad \Gamma, \blacktriangleright_{\hat{a}}, \hat{a} \vdash r \Rightarrow A[a := !_s \hat{a}] <: B \Rightarrow t \dashv \Gamma', \blacktriangleright_{\hat{a}}, \Theta}{\Gamma \vdash r \Rightarrow \forall a : \text{Type}_s. A <: B \Rightarrow t \dashv \Gamma'}$$

$$\frac{\Gamma, a : \text{Type}_s \vdash r \Rightarrow A <: B \Rightarrow t \dashv \Gamma', a : \text{Type}_s, \Theta}{\Gamma \vdash r \Rightarrow \forall @a : \text{Type}_s. A <: \forall @a : \text{Type}_s. B \Rightarrow t \dashv \Gamma'}$$

Subtyping – boxes

$$\frac{\Gamma \vdash r \cdot s \Rightarrow A <: B \Rightarrow t \dashv \Gamma'}{\Gamma \vdash r \Rightarrow A <: !_s B \Rightarrow t \dashv \Gamma'}$$

$$\frac{B \neq !_s B' \quad \Gamma \vdash r/s \Rightarrow A <: B \Rightarrow t \dashv \Gamma'}{\Gamma \vdash r \Rightarrow !_s A <: B \Rightarrow t \dashv \Gamma'}$$

Subtyping – type formers

$$\overline{\Gamma \vdash r \Rightarrow \text{Unit} <: \text{Unit} \Rightarrow 1 \dashv \Gamma}$$

$$\overline{\Gamma \vdash r \Rightarrow \text{Empty} <: \text{Empty} \Rightarrow 1 \dashv \Gamma}$$

$$\frac{\Gamma \vdash r \Rightarrow A_1 <: A_2 \Rightarrow s_A \dashv \Gamma_1 \quad \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle B_1 <: \langle \Gamma_1 \rangle B_2 \Rightarrow s_B \dashv \Gamma_2}{\Gamma \vdash r \Rightarrow A_1 \otimes B_1 <: A_2 \otimes B_2 \Rightarrow s_A \sqcap s_B \dashv \Gamma_2}$$

$$\frac{\Gamma \vdash r \Rightarrow A_1 <: A_2 \Rightarrow s_A \dashv \Gamma_1 \quad \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle B_1 <: \langle \Gamma_1 \rangle B_2 \Rightarrow s_B \dashv \Gamma_2}{\Gamma \vdash r \Rightarrow A_1 \oplus B_1 <: A_2 \oplus B_2 \Rightarrow s_A \sqcap s_B \dashv \Gamma_2}$$

$$\frac{\Gamma \vdash 1 \Rightarrow A_2 <: A_1 \Rightarrow 1 \dashv \Gamma_1 \quad \Gamma_1 \vdash 1 \Rightarrow \langle \Gamma_1 \rangle B_1 <: \langle \Gamma_1 \rangle B_2 \Rightarrow 1 \dashv \Gamma_2}{\Gamma \vdash r \Rightarrow A_1 \multimap B_1 <: A_2 \multimap B_2 \Rightarrow r \dashv \Gamma_2}$$

Subtyping – switch to instantiation

$$\frac{\hat{a} \notin \text{FV}(A) \quad \Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := A \Rightarrow s \vdash \Gamma'}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} <: A \Rightarrow s \vdash \Gamma'}$$

$$\frac{\hat{a} \notin \text{FV}(A) \quad \Gamma[\hat{a}] \vdash r \Rightarrow A =: \hat{a} \Rightarrow s \vdash \Gamma'}{\Gamma[\hat{a}] \vdash r \Rightarrow A <: \hat{a} \Rightarrow s \vdash \Gamma'}$$

Instantiation judgements

$\Gamma \vdash r \Rightarrow \hat{a} := A \Rightarrow s \dashv \Gamma'$ – in context Γ , instantiate the existential variable \hat{a} to a subtype of $!_{r/s} A$, with output context Γ' .

$\Gamma \vdash r \Rightarrow A =: \hat{a} \Rightarrow s \dashv \Gamma'$ – in context Γ , instantiate the existential variable \hat{a} , so that $A <: !_{r/s} \hat{a}$, with output context Γ .

Note: in both cases, Γ , r , \hat{a} and A are inputs, whereas s and Γ' are outputs.

Sub-instantiation

$$\frac{\Gamma \vdash r \Rightarrow \tau \Rightarrow s \vdash \Gamma_1}{\Gamma, \hat{a}, \Gamma' \vdash r \Rightarrow \hat{a} := \tau \Rightarrow 1 \vdash \Gamma_1, \hat{a} = !_s \tau, \Gamma'}$$

$$\frac{}{\Gamma[\hat{a}][\hat{b}] \vdash r \Rightarrow \hat{a} := \hat{b} \Rightarrow r \vdash \Gamma[\hat{a}][\hat{b} = \hat{a}]}$$

$$\frac{}{\Gamma[b : \text{Type}_s][\hat{a}] \vdash r \Rightarrow \hat{a} := b \Rightarrow 1 \vdash \Gamma[b : \text{Type}_s][\hat{a} = !_s b]}$$

$$\frac{\Gamma[\hat{a}] \vdash r \cdot s \Rightarrow \hat{a} := A \Rightarrow t \vdash \Gamma'}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := !_s A \Rightarrow t \vdash \Gamma'}$$

Note: no rule for implicit quantifier.

Sub-instantiation

$$\overline{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := \text{Unit} \Rightarrow 1 \dashv \Gamma[\hat{a} = \text{Unit}]}$$

$$\overline{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := \text{Empty} \Rightarrow 1 \dashv \Gamma[\hat{a} = \text{Empty}]}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \otimes \hat{a}_2] \vdash r \Rightarrow \hat{a}_1 := A \Rightarrow s_1 \dashv \Gamma_1 \\ \Gamma_1 \vdash r \Rightarrow \hat{a}_2 := \langle \Gamma_1 \rangle B \Rightarrow s_2 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := A \otimes B \Rightarrow s_1 \sqcap s_2 \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \oplus \hat{a}_2] \vdash r \Rightarrow \hat{a}_1 := A \Rightarrow s_1 \dashv \Gamma_1 \\ \Gamma_1 \vdash r \Rightarrow \hat{a}_2 := \langle \Gamma_1 \rangle B \Rightarrow s_2 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := A \oplus B \Rightarrow s_1 \sqcap s_2 \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = !_r(\hat{a}_1 \multimap \hat{a}_2)] \vdash 1 \Rightarrow A =: \hat{a}_1 \Rightarrow 1 \dashv \Gamma_1 \\ \Gamma_1 \vdash 1 \Rightarrow \hat{a}_2 := \langle \Gamma_1 \rangle B \Rightarrow 1 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}] \vdash r \Rightarrow \hat{a} := A \multimap B \Rightarrow 1 \dashv \Gamma_2}$$

Super-instantiation

$$\frac{\Gamma \vdash r \Rightarrow \tau \Rightarrow s \dashv \Gamma_1}{\Gamma, \hat{a}, \Gamma' \vdash r \Rightarrow \tau =: \hat{a} \Rightarrow s \dashv \Gamma_1, \hat{a} = \tau, \Gamma'}$$

$$\frac{}{\Gamma[\hat{a}][\hat{b}] \vdash r \Rightarrow \hat{b} =: \hat{a} \Rightarrow 1 \dashv \Gamma[\hat{a}][\hat{b} = !_r \hat{a}]}$$

$$\frac{}{\Gamma[b : \text{Type}_s][\hat{a}] \vdash r \Rightarrow b =: \hat{a} \Rightarrow r/s \dashv \Gamma[b : \text{Type}_s][\hat{a} = b]}$$

$$\frac{b \in \text{FV}(B) \quad \Gamma[\hat{a}], \blacktriangleright_{\hat{b}}, \hat{b} \vdash r \Rightarrow B[b := \hat{b}] =: \hat{a} \Rightarrow t \dashv \Gamma', \blacktriangleright_{\hat{b}}, \Theta}{\Gamma[\hat{a}] \vdash r \Rightarrow \forall b : \text{Type}_s. B =: \hat{a} \Rightarrow t \dashv \Gamma'}$$

$$\frac{\Gamma[\hat{a}] \vdash r/s \Rightarrow A =: \hat{a} \Rightarrow t \dashv \Gamma'}{\Gamma[\hat{a}] \vdash r \Rightarrow !_s A =: \hat{a} \Rightarrow t \dashv \Gamma'}$$

Super-instantiation

$$\frac{}{\Gamma[\hat{a}] \vdash r \Rightarrow \text{Unit} =: \hat{a} \Rightarrow 1 \dashv \Gamma[\hat{a} = \text{Unit}]}$$

$$\frac{}{\Gamma[\hat{a}] \vdash r \Rightarrow \text{Empty} =: \hat{a} \Rightarrow 1 \dashv \Gamma[\hat{a} = \text{Empty}]}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \otimes \hat{a}_2] \vdash r \Rightarrow A =: \hat{a}_1 \Rightarrow s_1 \dashv \Gamma_1 \\ \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle B =: \hat{a}_2 \Rightarrow s_2 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}] \vdash r \Rightarrow A \otimes B =: \hat{a} \Rightarrow s_1 \sqcap s_2 \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \oplus \hat{a}_2] \vdash r \Rightarrow A =: \hat{a}_1 \Rightarrow s_1 \dashv \Gamma_1 \\ \Gamma_1 \vdash r \Rightarrow \langle \Gamma_1 \rangle B =: \hat{a}_2 \Rightarrow s_2 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}] \vdash r \Rightarrow A \oplus B =: \hat{a} \Rightarrow s_1 \sqcap s_2 \dashv \Gamma_2}$$

$$\frac{\begin{array}{l} \Gamma[\hat{a}_1, \hat{a}_2, \hat{a} = \hat{a}_1 \multimap \hat{a}_2] \vdash 1 \Rightarrow \hat{a}_1 := A \Rightarrow 1 \dashv \Gamma_1 \\ \Gamma_1 \vdash 1 \Rightarrow \langle \Gamma_1 \rangle B =: \hat{a}_2 \Rightarrow 1 \dashv \Gamma_2 \end{array}}{\Gamma[\hat{a}] \vdash r \Rightarrow A \multimap B =: \hat{a} \Rightarrow r \dashv \Gamma_2}$$

Monotypes and explicit quantifiers

Note: we use the monotype rule to handle the difficult case when $\tau = \forall @a : \text{Type}_r. A$. If we had a separate rule for this, then the monotype rule would be admissible. Alternatively, we can write down separate rules to handle the explicit quantifier in a naive way.

$$\frac{\Gamma \vdash r \Rightarrow \forall @a : \text{Type}_s. \tau \Rightarrow t \vdash \Gamma_1}{\Gamma, \hat{a}, \Gamma' \vdash r \Rightarrow \hat{a} := \forall @a : \text{Type}_s. \tau \Rightarrow 1 \vdash \Gamma_1, \hat{a} = !_t \forall @a : \text{Type}_s. \tau, \Gamma'}$$

$$\frac{\Gamma \vdash r \Rightarrow \forall @a : \text{Type}_s. \tau \Rightarrow t \vdash \Gamma_1}{\Gamma, \hat{a}, \Gamma' \vdash r \Rightarrow \forall @a : \text{Type}_s. \tau =: \hat{a} \Rightarrow t \vdash \Gamma_1, \hat{a} = \forall @a : \text{Type}_s. \tau, \Gamma'}$$