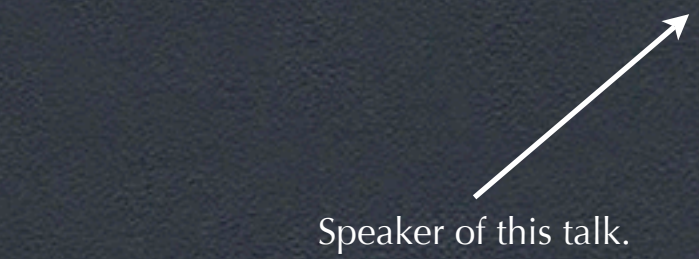
Safe Compositional Specification

of Network Systems

Azer Bestavros, Adam D. Bradley, Yarom Gabay,
Assaf J. Kfoury, Likai Liu, and Ibrahim Matta



Outline

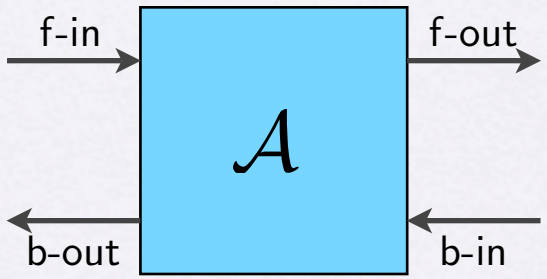
- Introduction: on our way to a domain specific language.
- Syntax and Types
- Compositional Analysis
- Future Work

Introduction

- Problem: Internet is a heterogeneous network, with unpredictable performance and feedback.
- Goal: to build a network with known “flow” properties.
- Properties: bandwidth, congestion control response, quality of service capabilities, ...

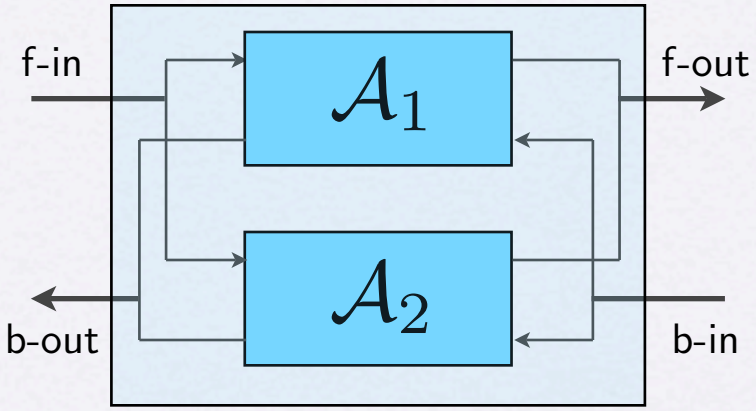
Introduction

- Simplified network model: boxes with forward and backward direction inputs and outputs.
- Simplified network arrangements: parallel and sequential connections.
- Flow “variable”

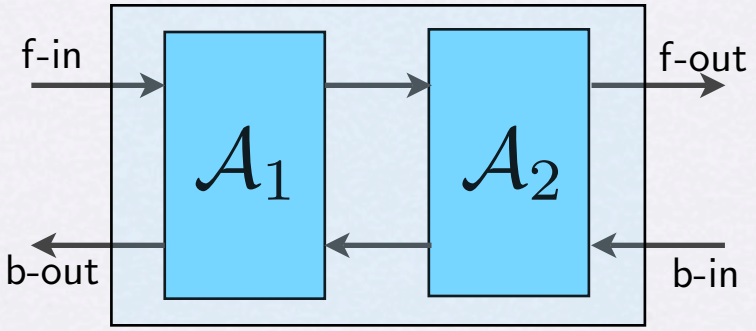


Introduction

- Simplified network model: boxes with forward and backward direction inputs and outputs.
- Simplified network arrangements: parallel and sequential connections.
- Flow "variable"



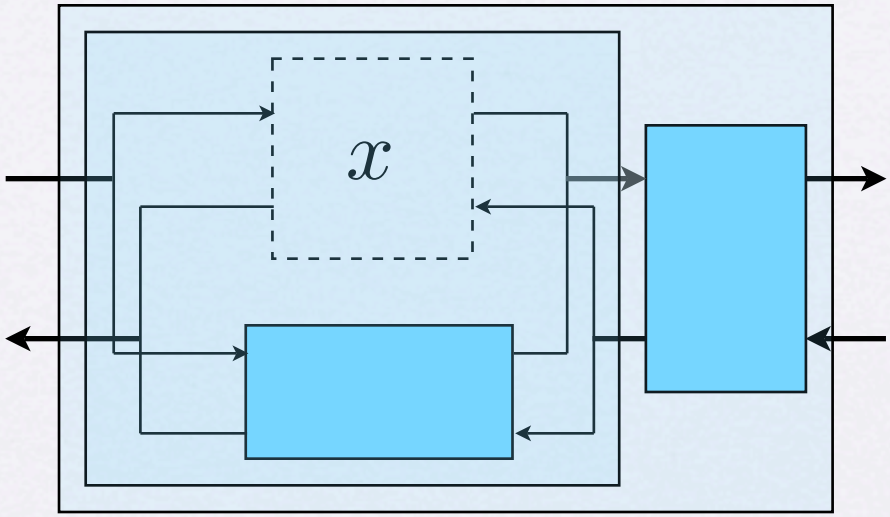
$\mathcal{A}_1 \| \mathcal{A}_2$ (parallel)



$\mathcal{A}_1 ; \mathcal{A}_2$ (sequential)

Introduction

- Simplified network model: boxes with forward and backward direction inputs and outputs.
- Simplified network arrangements: parallel and sequential connections.
- Flow “variable”



Outline

- Introduction: on our way to a domain specific language.
- Syntax and Types
- Compositional Analysis
- Future Work

Syntax and Types

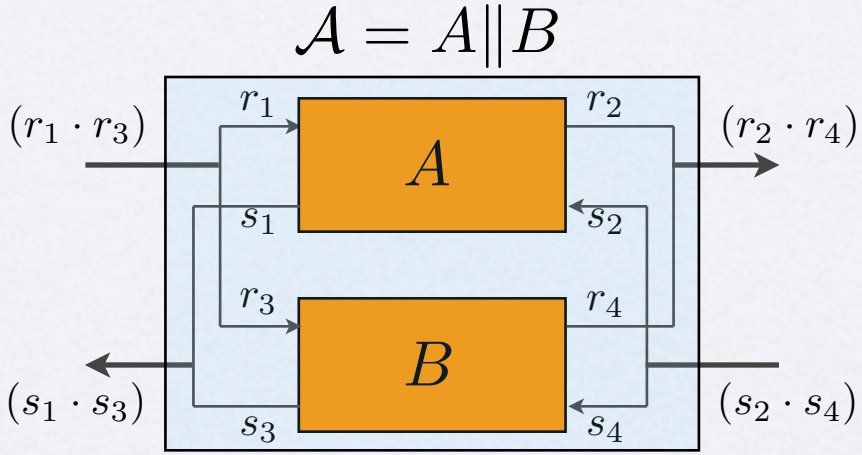
$$
\begin{array}{lll}
x, y, z & \in \mathsf{FlowVar} & \text{flow variable} \\
A, B, C & \in \mathsf{LocalFlow} & \text{local flow} \\
\mathcal{A}, \mathcal{B}, \mathcal{C} & \in \mathsf{GlobalFlow} ::= A \mid x & \\
& \mid \mathcal{A}; \mathcal{B} & \text{sequential flow} \\
& \mid \mathcal{A} \| \mathcal{B} & \text{parallel flow} \\
& \mid \textbf{let}\ \ x = \mathcal{A}\ \ \textbf{in}\ \ \mathcal{B} & \text{let-binding} \\
& \mid \textbf{let}\ \ x \in \mathcal{A}_1, \ldots, \mathcal{A}_n\ \ \textbf{in}\ \ \mathcal{B} & \text{multiple-choice let}
\end{array}
$$

Evalution:

$$\textbf{let}\ \ x = \mathcal{A}\ \ \textbf{in}\ \ \mathcal{B} \to [x := \mathcal{A}]\mathcal{B}$$

Syntax and Types

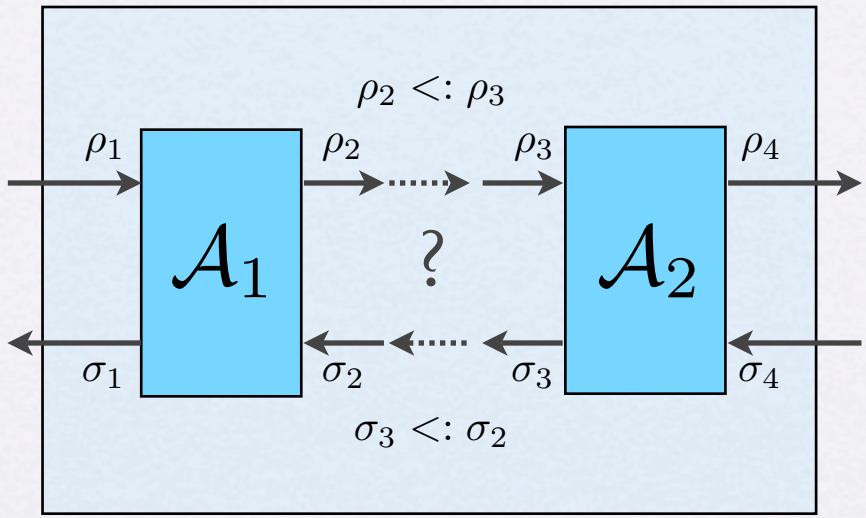
$$
\begin{array}{rcll}
r & \in & \text{FwSocketType} & \\
s & \in & \text{BwSocketType} & \\
t & \in & \text{SocketType} & ::= \quad r \mid s \\
\rho & \in & \text{FwType} & ::= \quad r \mid (\rho_1 \cdot \rho_2) \\
\sigma & \in & \text{BwType} & ::= \quad s \mid (\sigma_1 \cdot \sigma_2) \\
\tau & \in & \text{Type} & ::= \quad \rho \mid \sigma \\
T & \in & \text{FlowType} & ::= \quad \begin{bmatrix} \rho_1 & \rho_2 \\ \sigma_1 & \sigma_2 \end{bmatrix}
\end{array}
$$



$$
\mathcal{A} : \begin{bmatrix} (r_1 \cdot r_3) & (r_2 \cdot r_4) \\ (s_1 \cdot s_3) & (s_2 \cdot s_4) \end{bmatrix}
$$

$$
\begin{bmatrix} \rho_1 & \rho_2 \\ \sigma_1 & \sigma_2 \end{bmatrix} \bullet \begin{bmatrix} \rho_3 & \rho_4 \\ \sigma_3 & \sigma_4 \end{bmatrix} = \begin{bmatrix} (\rho_1 \cdot \rho_3) & (\rho_2 \cdot \rho_4) \\ (\sigma_1 \cdot \sigma_3) & (\sigma_2 \cdot \sigma_4) \end{bmatrix}
$$

Subtyping


$\rho_2 <: \rho_3$
ρ_1
ρ_2
ρ_3
ρ_4
$\mathcal{A}_1$
?
$\mathcal{A}_2$
σ_1
σ_2
σ_3
σ_4
$\sigma_3 <: \sigma_2$

Subtyping

$\Delta \subseteq \mathsf{FwSocketType} \times \mathsf{FwSocketType} \cup \mathsf{BwSocketType} \times \mathsf{BwSocketType}$

$$\frac{\{t_1 <: t_2\} \subseteq \Delta}{\Delta \vdash t_1 <: t_2} \text{ (stype)}$$

$$\frac{t \in \mathsf{SocketType}}{\Delta \vdash t <: t} \text{ (stype-refl)}$$

$$\frac{\Delta \vdash t_1 <: t_2 \quad \Delta \vdash t_2 <: t_3}{\Delta \vdash t_1 <: t_3} \text{ (stype-trans)}$$

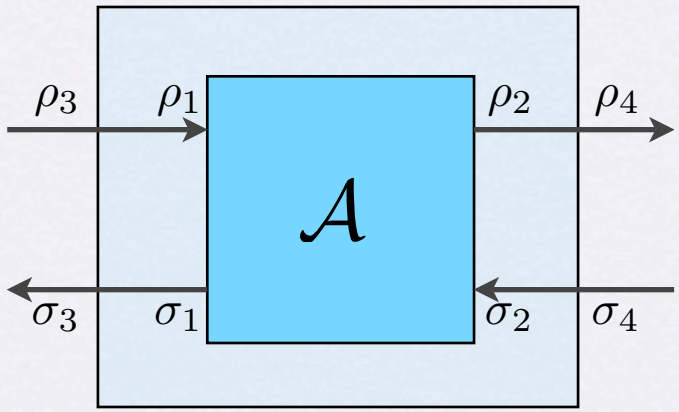
$\forall t_1, t_2 \in \mathsf{SocketType}.$ if $\Delta \vdash t_1 <: t_2$ and $\Delta \vdash t_2 <: t_1$, then $t_1 = t_2$
(stype-anti)

Δ is a partial order!

Lifting

Translation of subtyping at a “higher level” to subtyping of a “lower level”

$$\frac{\Delta \vdash \rho_3 <: \rho_1 \quad \Delta \vdash \rho_2 <: \rho_4 \quad \Delta \vdash \sigma_1 <: \sigma_3 \quad \Delta \vdash \sigma_4 <: \sigma_2}{\Delta \vdash \begin{bmatrix} \rho_1 & \rho_2 \\ \sigma_1 & \sigma_2 \end{bmatrix} <: \begin{bmatrix} \rho_3 & \rho_4 \\ \sigma_3 & \sigma_4 \end{bmatrix}} \text{(ftype-lift)}$$



Lifting

$$\frac{\Delta \vdash \rho_1 <: \rho_1' \quad \Delta \vdash \rho_2 <: \rho_2'}{\Delta \vdash (\rho_1 \cdot \rho_2) <: (\rho_1' \cdot \rho_2')} \text{ (fwtype-lift)}$$

$$\frac{\Delta \vdash \sigma_1 <: \sigma_1' \quad \Delta \vdash \sigma_2 <: \sigma_2'}{\Delta \vdash (\sigma_1 \cdot \sigma_2) <: (\sigma_1' \cdot \sigma_2')} \text{ (bwtype-lift)}$$

Recall:

$$\rho \in \mathsf{FwType} ::= r \mid (\rho_1 \cdot \rho_2)$$
$$\sigma \in \mathsf{BwType} ::= s \mid (\sigma_1 \cdot \sigma_2)$$

We can easily prove that subtyping of Type and FlowType is also a partial order relation.

Type Judgment

- As usual, we have a type environment:

  $\Gamma = \{x_1 : T_1, \ldots, x_m : T_m\}$

- Also have a local-flow lookup function:

  $\mathsf{type} = \{A_1 : T'_1, \ldots, A_n : T'_n\}$

Type Judgment

$$\frac{\Gamma(x) = T}{\Gamma \vdash x : T}\ (\text{var}) \qquad \frac{\text{type}(A) = T}{\Gamma \vdash A : T}\ (\text{local})$$

$$\frac{\Gamma, \Delta \vdash \mathcal{A} : T \quad \Gamma, \Delta \vdash \mathcal{B} : T'}{\Gamma, \Delta \vdash \mathcal{A} \| \mathcal{B} : T \bullet T'}\ (\text{par})$$

$$\frac{\Gamma, \Delta \vdash \mathcal{A} : \begin{bmatrix} \rho_1 & \rho_2 \\ \sigma_1 & \sigma_2 \end{bmatrix} \quad \Gamma, \Delta \vdash \mathcal{B} : \begin{bmatrix} \rho_3 & \rho_4 \\ \sigma_3 & \sigma_4 \end{bmatrix} \quad \Delta \vdash \rho_2 <: \rho_3 \quad \Delta \vdash \sigma_3 <: \sigma_2}{\Gamma, \Delta \vdash \mathcal{A}\,; \mathcal{B} \ : \begin{bmatrix} \rho_1 & \rho_4 \\ \sigma_1 & \sigma_4 \end{bmatrix}}\ (\text{seq})$$

$$\frac{\Gamma, \Delta \vdash \mathcal{A} : T \quad \Gamma \cup \{x : T\}, \Delta \vdash \mathcal{B} : T'}{\Gamma, \Delta \vdash \mathbf{let} \ \ x = \mathcal{A} \ \ \mathbf{in} \ \ \mathcal{B} : T'}\ (\text{let})$$

Type Judgment

Lemma (Type Preservation). *If the judgment* $\Gamma, \Delta \vdash \mathcal{A} : T$ *is derivable and* $\mathcal{A} \rightarrow \mathcal{B}$*, then* $\Gamma, \Delta \vdash \mathcal{B} : T$*. In particular, if* $\mathcal{A}$ *type checks, then so does* $\mathcal{B}$*.*

- At this point, we can devise an algorithm $\mathcal{P}_{NC}(\mathcal{A})$ to follow the judgment rules in an outside-in manner. Easy to show it is correct.
- The algorithm can only check closed specifications, i.e., with no free variable occurrences.

This limitation gives rise to...

Compositional Analysis

Outline

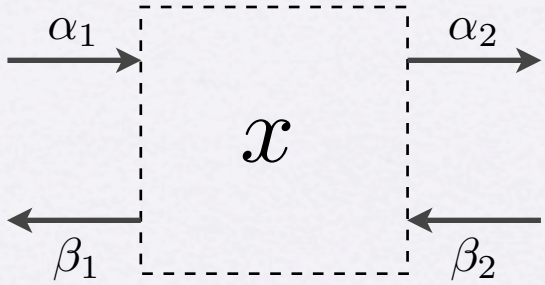
- Introduction: on our way to a domain specific language.
- Syntax and Types
- Compositional Analysis
- Future Work

Compositional Analysis

- Idea: build judgment from the “leaf.”
- Free variables are assigned types using type variables, which are substituted to a concrete type at a later point.
- Subtyping premises may or may not be satisfied. Depends on substitution later.
- To show the algorithm is correct, we show that it satisfies “principal typing.”

Type Variables

$$
\begin{array}{lll}
\alpha & \in \mathsf{FwTypeVar} & \\
\beta & \in \mathsf{BwTypeVar} & \\
\tilde{\rho} & \in \mathsf{FwType}^{\sim} & ::= \; r \mid \alpha \mid (\tilde{\rho}_1 \cdot \tilde{\rho}_2) \\
\tilde{\sigma} & \in \mathsf{BwType}^{\sim} & ::= \; s \mid \beta \mid (\tilde{\sigma}_1 \cdot \tilde{\sigma}_2) \\
\tilde{\tau} & \in \mathsf{Type}^{\sim} & ::= \; \tilde{\rho} \mid \tilde{\sigma} \\
\tilde{T} & \in \mathsf{FlowType}^{\sim} & ::= \; \begin{bmatrix} \tilde{\rho}_1 & \tilde{\rho}_2 \\ \tilde{\sigma}_1 & \tilde{\sigma}_2 \end{bmatrix}
\end{array}
$$



Set of Constraints: $C = \{\alpha <: \alpha', \beta <: \beta', \tilde{\rho} <: \tilde{\rho}', \tilde{\sigma} <: \tilde{\sigma}', \dots\}$

Substitution: $S = \{\alpha \mapsto \tilde{\rho}, \beta \mapsto \tilde{\sigma}\}$

Example: $S(C) = \{\tilde{\rho} <: \alpha', \tilde{\sigma} <: \beta', \tilde{\rho} <: \tilde{\rho}', \tilde{\sigma} <: \tilde{\sigma}', \dots\}$

Compositional Analysis

Theorem (Solvability and Principality). *Let $\mathcal{A}$ be a global-flow specification. Suppose $\mathcal{P}$ terminates and returns the triple $\mathcal{P}(\mathcal{A}) = \langle \tilde{\Gamma}, C, \tilde{T} \rangle$. It then holds that:*

1. *There is a typing derivation with final judgment*

$$\tilde{\Gamma}, \Delta, C \vdash \mathcal{A} : \tilde{T}$$

i.e., *there is a typing for $\mathcal{A}$ that assigns it the type $\tilde{T}$.*

Compositional Analysis

Theorem (Solvability and Principality). *Let $\mathcal{A}$ be a global-flow specification. Suppose $\mathcal{P}$ terminates and returns the triple $\mathcal{P}(\mathcal{A}) = \langle \tilde{\Gamma}, C, \tilde{T} \rangle$. It then holds that:*

2. *For every substitution S, not necessarily closed, if $S(C)$ is consistent then there is a typing derivation where the final judgment is*

$$S(\tilde{\Gamma}), \Delta, S(C) \vdash \mathcal{A} : S(\tilde{T})$$

Compositional Analysis

Theorem (Solvability and Principality). *Let $\mathcal{A}$ be a global-flow specification. Suppose $\mathcal{P}$ terminates and returns the triple $\mathcal{P}(\mathcal{A}) = \langle \tilde{\Gamma}, C, \tilde{T} \rangle$. It then holds that:*

3. *For every typing derivation with final judgment $\Gamma, \Delta \vdash \mathcal{A} : T$, there is a closed substitution S such that*

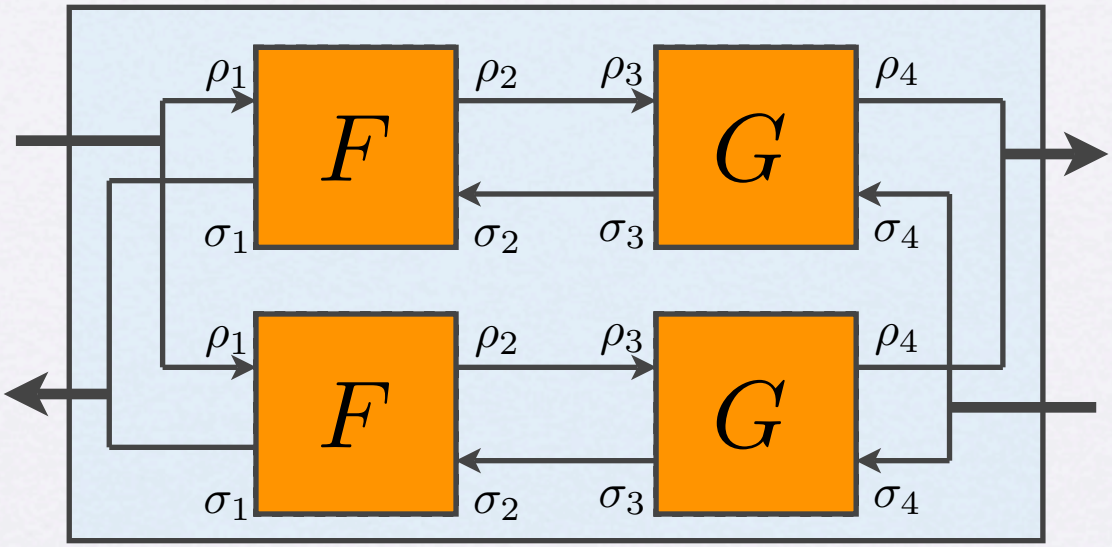
   (a) $\Gamma = S(\tilde{\Gamma})$,

   (b) $\Delta \vdash S(C)$,

   (c) $T = S(\tilde{T})$.

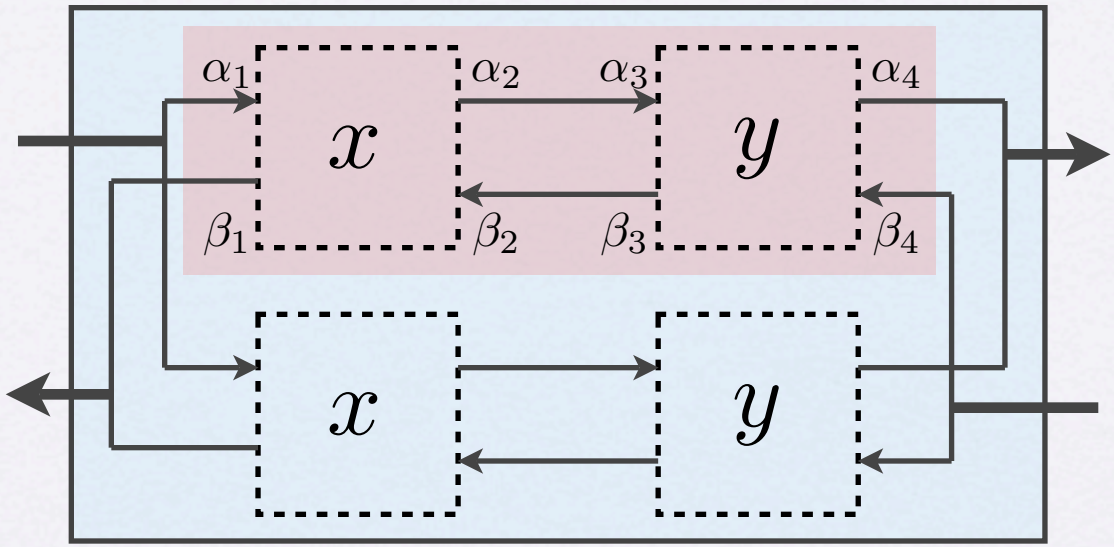
Example

$$\textbf{let}\ \ x = F\ \ \textbf{in}$$
$$\textbf{let}\ \ y = G\ \ \textbf{in}$$
$$(x;y) \| (x;y)$$



Example

$$\textbf{let}\quad x = F\quad \textbf{in}$$
$$\textbf{let}\quad y = G\quad \textbf{in}$$
$$(x;y)\|(x;y)$$

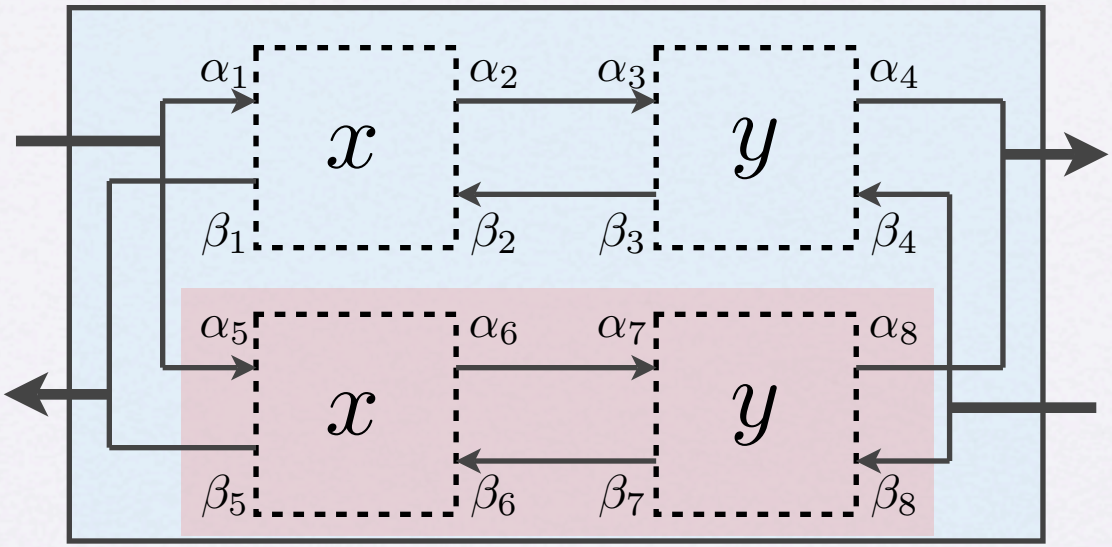


$$\frac{\tilde{\Gamma}_1(x) = \begin{bmatrix}\alpha_1 & \alpha_2\\ \beta_1 & \beta_2\end{bmatrix} \qquad \tilde{\Gamma}_2(y) = \begin{bmatrix}\alpha_3 & \alpha_4\\ \beta_3 & \beta_4\end{bmatrix}}{\dfrac{\tilde{\Gamma}_1, \Delta, \varnothing \vdash x : \begin{bmatrix}\alpha_1 & \alpha_2\\ \beta_1 & \beta_2\end{bmatrix} \qquad \tilde{\Gamma}_2, \Delta, \varnothing \vdash y : \begin{bmatrix}\alpha_3 & \alpha_4\\ \beta_3 & \beta_4\end{bmatrix}}{\tilde{\Gamma}_{1,2} = \{x : \begin{bmatrix}\alpha_1 & \alpha_2\\ \beta_1 & \beta_2\end{bmatrix}, y : \begin{bmatrix}\alpha_3 & \alpha_4\\ \beta_3 & \beta_4\end{bmatrix}\}, \Delta, \vdash (x;y) : \begin{bmatrix}\alpha_1 & \alpha_4\\ \beta_1 & \beta_4\end{bmatrix}}$$

$$C = \{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2\}$$

Example

$$\textbf{let}\quad x = F\quad \textbf{in}$$
$$\textbf{let}\quad y = G\quad \textbf{in}$$
$$(x;y) \| (x;y)$$

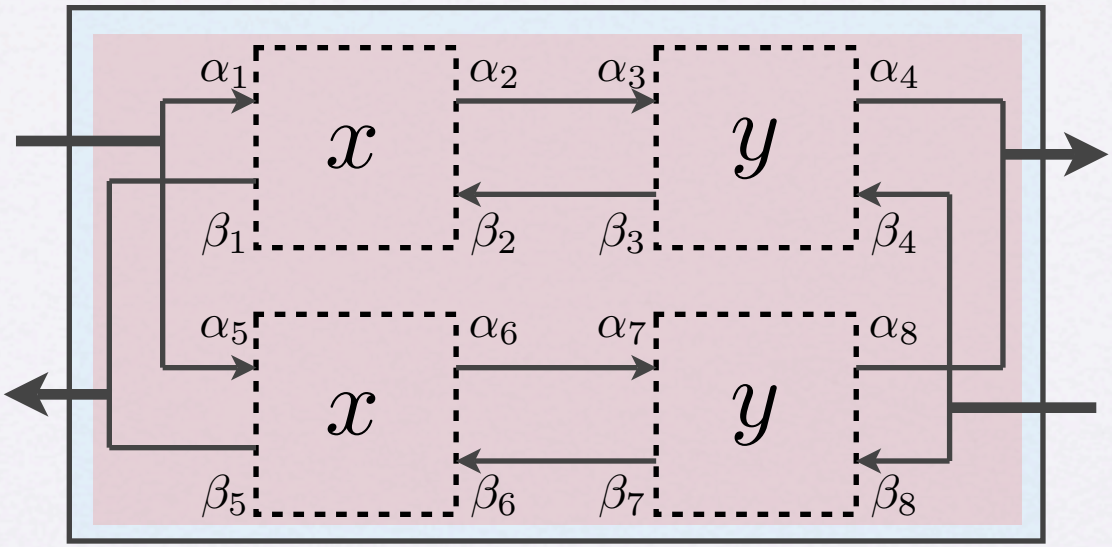


$$\frac{\tilde{\Gamma}_3(x) = \begin{bmatrix}\alpha_5 & \alpha_6\\ \beta_5 & \beta_6\end{bmatrix} \qquad \tilde{\Gamma}_4(y) = \begin{bmatrix}\alpha_7 & \alpha_8\\ \beta_7 & \beta_8\end{bmatrix}}{\dfrac{\tilde{\Gamma}_3, \Delta, \varnothing \vdash x : \begin{bmatrix}\alpha_5 & \alpha_6\\ \beta_5 & \beta_6\end{bmatrix} \qquad \tilde{\Gamma}_4, \Delta, \varnothing \vdash y : \begin{bmatrix}\alpha_7 & \alpha_8\\ \beta_7 & \beta_8\end{bmatrix}}{\tilde{\Gamma}_{3,4} = \{x : \begin{bmatrix}\alpha_5 & \alpha_6\\ \beta_5 & \beta_6\end{bmatrix}, y : \begin{bmatrix}\alpha_7 & \alpha_8\\ \beta_7 & \beta_8\end{bmatrix}\}, \Delta, \vdash (x;y) : \begin{bmatrix}\alpha_5 & \alpha_8\\ \beta_5 & \beta_8\end{bmatrix}}$$

$$C = \{\alpha_6 <: \alpha_7, \beta_7 <: \beta_6\}$$

Example

$$\textbf{let}\quad x = F\quad \textbf{in}$$
$$\textbf{let}\quad y = G\quad \textbf{in}$$
$$(x;y)\|(x;y)$$



$$\tilde{\Gamma}_{1,2}, \Delta, \{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2\} \vdash (x;y) : \begin{bmatrix} \alpha_1 & \alpha_4 \\ \beta_1 & \beta_4 \end{bmatrix}$$

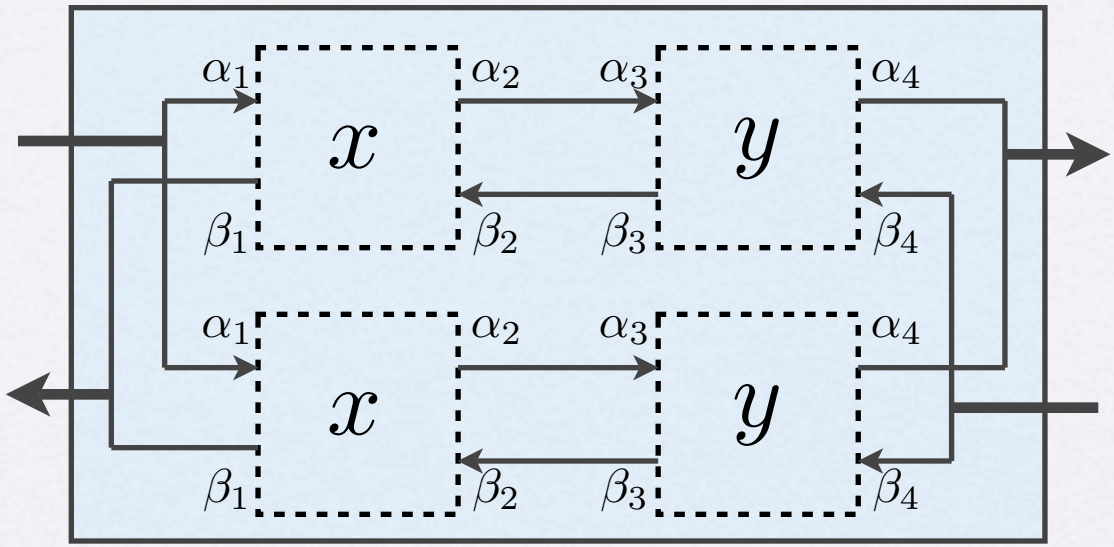
$$\tilde{\Gamma}_{3,4}, \Delta, \{\alpha_6 <: \alpha_7, \beta_7 <: \beta_6\} \vdash (x;y) : \begin{bmatrix} \alpha_5 & \alpha_8 \\ \beta_5 & \beta_8 \end{bmatrix}$$

$$\tilde{\Gamma}_5 = \{x : \begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{bmatrix}, y : \begin{bmatrix} \alpha_3 & \alpha_4 \\ \beta_3 & \beta_4 \end{bmatrix}\}, \Delta$$

$$\{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2, \alpha_6 <: \alpha_7, \beta_7 <: \beta_6,\ \alpha_1 \doteq \alpha_5, \alpha_2 \doteq \alpha_6, \beta_1 \doteq \beta_5, \beta_2 \doteq \beta_6,\ \alpha_3 \doteq \alpha_7, \alpha_4 \doteq \alpha_8, \beta_3 \doteq \beta_7, \beta_4 \doteq \beta_8\} \vdash (x;y)\|(x;y) : \begin{bmatrix} (\alpha_1 \cdot \alpha_5) & (\alpha_4 \cdot \alpha_8) \\ (\beta_1 \cdot \beta_5) & (\beta_4 \cdot \beta_8) \end{bmatrix}$$

Example

$$\textbf{let}\quad x = F\quad \textbf{in}$$
$$\textbf{let}\quad y = G\quad \textbf{in}$$
$$(x;y)\|(x;y)$$



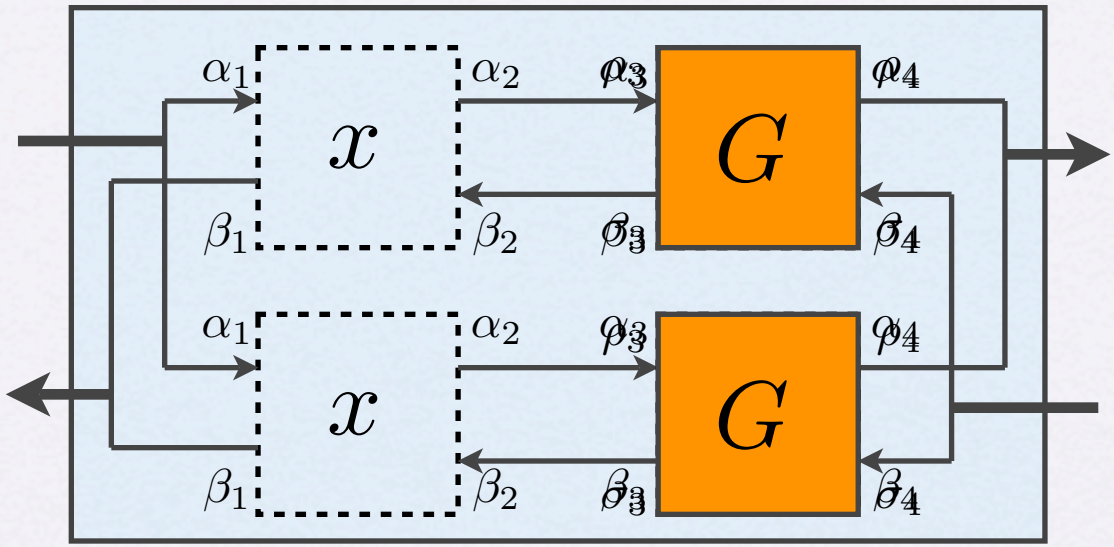
$$\tilde{\Gamma}_{1,2}, \Delta, \{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2\} \vdash (x;y) : \begin{bmatrix} \alpha_1 & \alpha_4 \\ \beta_1 & \beta_4 \end{bmatrix}$$

$$\tilde{\Gamma}_{3,4}, \Delta, \{\alpha_6 <: \alpha_7, \beta_7 <: \beta_6\} \vdash (x;y) : \begin{bmatrix} \alpha_5 & \alpha_8 \\ \beta_5 & \beta_8 \end{bmatrix}$$

$$\tilde{\Gamma}_5 = \{x : \begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{bmatrix}, y : \begin{bmatrix} \alpha_3 & \alpha_4 \\ \beta_3 & \beta_4 \end{bmatrix}\}, \Delta \quad \vdash (x;y)\|(x;y) : \begin{bmatrix} (\alpha_1 \cdot \alpha_1) & (\alpha_4 \cdot \alpha_4) \\ (\beta_1 \cdot \beta_1) & (\beta_4 \cdot \beta_4) \end{bmatrix}$$
$$\{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2\}$$

Example

$$\begin{array}{l}\textbf{let}\quad x = F\quad \textbf{in}\\ \textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array}$$



$$\frac{\mathsf{type}(G) = \begin{bmatrix}\rho_3 & \rho_4\\ \sigma_3 & \sigma_4\end{bmatrix}}{\varnothing, \Delta, \varnothing \vdash G : \begin{bmatrix}\rho_3 & \rho_4\\ \sigma_3 & \sigma_4\end{bmatrix}}$$

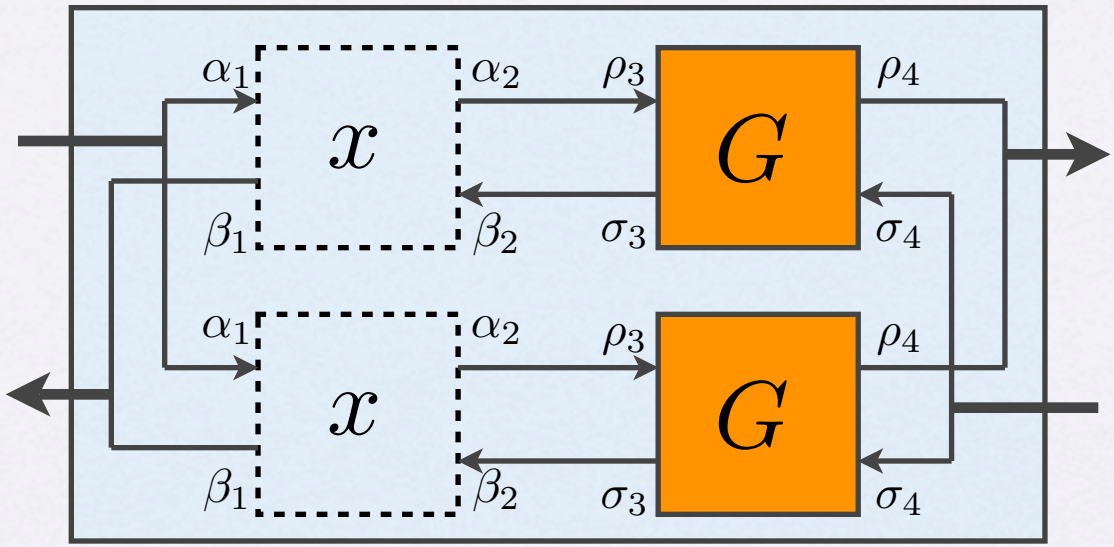
$$\tilde{\Gamma}_5, \Delta, \{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2\} \vdash (x;y)\|(x;y) : \begin{bmatrix}(\alpha_1 \cdot \alpha_1) & (\alpha_4 \cdot \alpha_4)\\ (\beta_1 \cdot \beta_1) & (\beta_4 \cdot \beta_4)\end{bmatrix}$$

$$\tilde{\Gamma}_6 = \{x : \begin{bmatrix}\alpha_1 & \alpha_2\\ \beta_1 & \beta_2\end{bmatrix}\}, \Delta,$$
$$\{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2,$$
$$\alpha_3 \doteq \rho_3, \alpha_4 \doteq \rho_4, \beta_3 \doteq \sigma_3, \beta_4 \doteq \sigma_4\}$$

$$\vdash \begin{array}{l}\textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array} : \begin{bmatrix}(\alpha_1 \cdot \alpha_1) & (\alpha_4 \cdot \alpha_4)\\ (\beta_1 \cdot \beta_1) & (\beta_4 \cdot \beta_4)\end{bmatrix}$$

Example

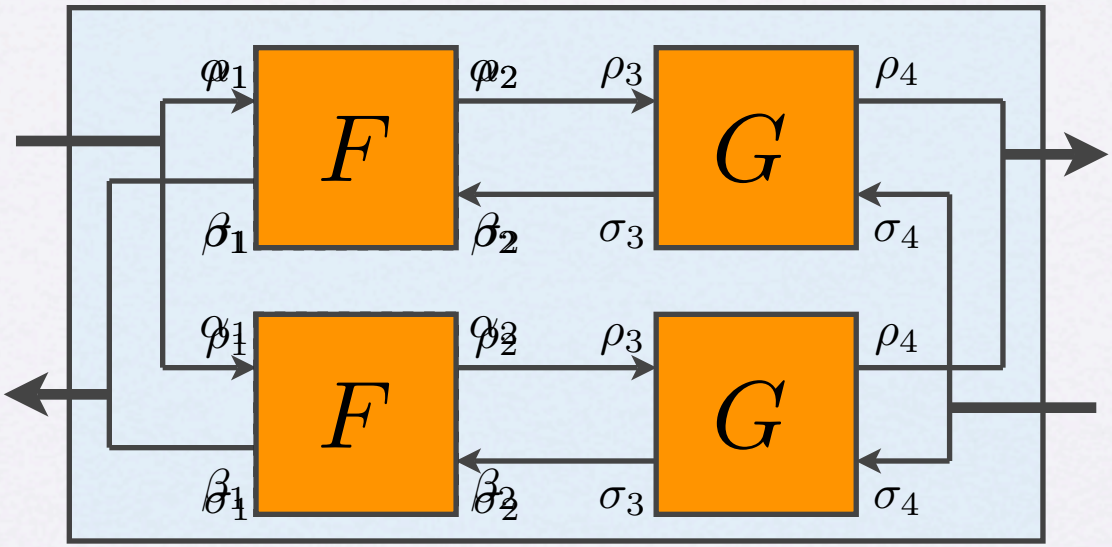
$$\begin{array}{l}\textbf{let}\quad x = F\quad \textbf{in}\\ \textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array}$$



$$\dfrac{\dfrac{\mathsf{type}(G) = \begin{bmatrix}\rho_3 & \rho_4\\ \sigma_3 & \sigma_4\end{bmatrix}}{\varnothing, \Delta, \varnothing \vdash G : \begin{bmatrix}\rho_3 & \rho_4\\ \sigma_3 & \sigma_4\end{bmatrix}} \qquad \tilde{\Gamma}_5, \Delta, \{\alpha_2 <: \alpha_3, \beta_3 <: \beta_2\} \vdash (x;y)\|(x;y) : \begin{bmatrix}(\alpha_1 \cdot \alpha_1) & (\alpha_4 \cdot \alpha_4)\\ (\beta_1 \cdot \beta_1) & (\beta_4 \cdot \beta_4)\end{bmatrix}}{\tilde{\Gamma}_6 = \{x : \begin{bmatrix}\alpha_1 & \alpha_2\\ \beta_1 & \beta_2\end{bmatrix}\}, \Delta, \{\alpha_2 <: \rho_3, \sigma_3 <: \beta_2\} \vdash \begin{array}{l}\textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array} : \begin{bmatrix}(\alpha_1 \cdot \alpha_1) & (\rho_4 \cdot \rho_4)\\ (\beta_1 \cdot \beta_1) & (\sigma_4 \cdot \sigma_4)\end{bmatrix}}$$

Example

$$\begin{array}{l}\textbf{let}\quad x = F\quad \textbf{in}\\ \textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array}$$

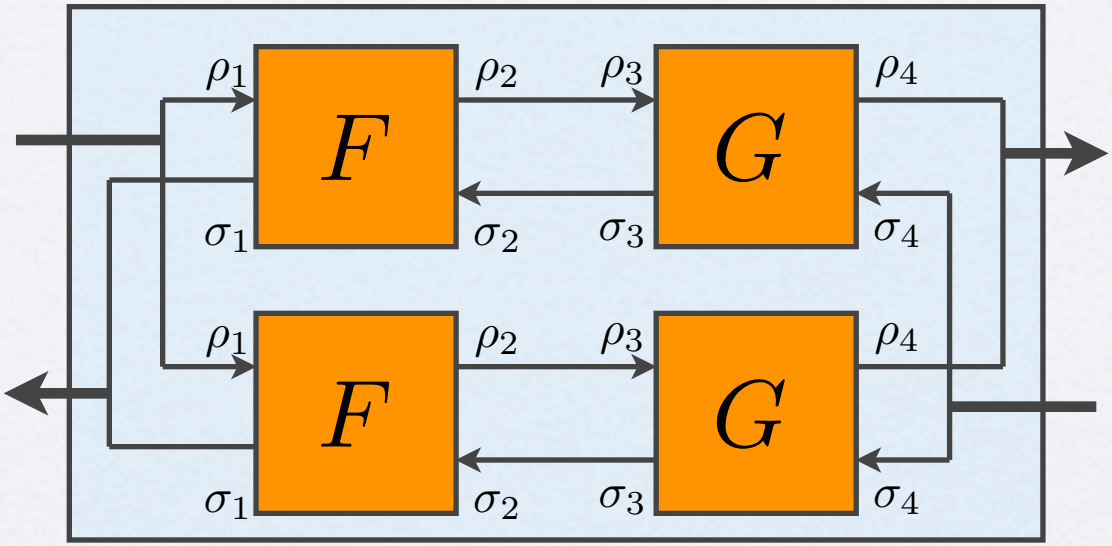


$$\frac{\mathsf{type}(F) = \begin{bmatrix}\rho_1 & \rho_2\\ \sigma_1 & \sigma_2\end{bmatrix}}{\varnothing, \Delta, \varnothing \vdash F : \begin{bmatrix}\rho_1 & \rho_2\\ \sigma_1 & \sigma_2\end{bmatrix}} \qquad \tilde{\Gamma}_6, \Delta, \{\alpha_2 <: \rho_3, \sigma_3 <: \beta_2\} \vdash \begin{array}{l}\textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array} : \begin{bmatrix}(\alpha_1 \cdot \alpha_1) & (\rho_4 \cdot \rho_4)\\ (\beta_1 \cdot \beta_1) & (\sigma_4 \cdot \sigma_4)\end{bmatrix}$$

$$\varnothing, \Delta, \{\rho_2 <: \rho_3, \sigma_3 <: \sigma_2, \ \alpha_1 \doteq \rho_1, \alpha_2 \doteq \rho_2, \beta_1 \doteq \sigma_1, \beta_2 \doteq \sigma_2\} \vdash \begin{array}{l}\textbf{let}\quad x = F\quad \textbf{in}\\ \textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array} : \begin{bmatrix}(\rho_1 \cdot \rho_1) & (\rho_4 \cdot \rho_4)\\ (\sigma_1 \cdot \sigma_1) & (\sigma_4 \cdot \sigma_4)\end{bmatrix}$$

Example

$$\begin{array}{l}\textbf{let}\quad x = F\quad \textbf{in}\\ \textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array}$$



$$\dfrac{\mathsf{type}(F) = \begin{bmatrix}\rho_1 & \rho_2\\ \sigma_1 & \sigma_2\end{bmatrix}}{\varnothing, \Delta, \varnothing \vdash F : \begin{bmatrix}\rho_1 & \rho_2\\ \sigma_1 & \sigma_2\end{bmatrix}} \qquad \begin{array}{l}\tilde{\Gamma}_6, \Delta, \{\alpha_2 <: \rho_3, \sigma_3 <: \beta_2\} \vdash\\ \quad \begin{array}{l}\textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array} : \begin{bmatrix}(\alpha_1 \cdot \alpha_1) & (\rho_4 \cdot \rho_4)\\ (\beta_1 \cdot \beta_1) & (\sigma_4 \cdot \sigma_4)\end{bmatrix}\end{array}$$

$$\varnothing, \Delta, \{\rho_2 <: \rho_3, \sigma_3 <: \sigma_2\} \vdash \begin{array}{l}\textbf{let}\quad x = F\quad \textbf{in}\\ \textbf{let}\quad y = G\quad \textbf{in}\\ (x;y)\|(x;y)\end{array} : \begin{bmatrix}(\rho_1 \cdot \rho_1) & (\rho_4 \cdot \rho_4)\\ (\sigma_1 \cdot \sigma_1) & (\sigma_4 \cdot \sigma_4)\end{bmatrix}$$

Outline

- Introduction: on our way to a domain specific language.
- Syntax and Types
- Compositional Analysis
- Future Work

Future Work

Multiple-choice Let: $\textbf{let}\ \ x \in \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \dots\}\ \ \textbf{in}\ \ \mathcal{B}$

$$\frac{\Gamma, \Delta \vdash \mathcal{A}_i : T_i \text{ for every } i \in 1 \dots n \qquad T = \mathsf{LCSup}\{T_1, \dots, T_n\} \qquad \Gamma \cup \{x : T\}, \Delta \vdash \mathcal{B} : T'}{\Gamma, \Delta \vdash \textbf{let}\ \ x \in \{\mathcal{A}_1, \dots, \mathcal{A}_n\}\ \ \textbf{in}\ \ \mathcal{B} : T'}\ \text{(let-choice-st)}$$

$$\frac{\Gamma, \Delta \vdash \mathcal{A}_i : T_i \quad \Gamma \cup \{x : T_i\}, \Delta \vdash \mathcal{B} : T'_i \text{ for every } i \in 1 \dots n \qquad T' = \mathsf{LCSup}\{T'_1, \dots, T'_n\}}{\Gamma, \Delta \vdash \textbf{let}\ \ x \in \{\mathcal{A}_1, \dots, \mathcal{A}_n\}\ \ \textbf{in}\ \ \mathcal{B} : T'}\ \text{(let-choice-bt)}$$

Future Work

Error Analysis:

- The language of global-flow is simple. Potentially easy to reason about conditions that cause error.
- Error slicing: highlight two or more portions of the specification that contribute to inconsistent constraints.

The End