Session Subtyping and Multiparty Compatibility using Circular Sequents

31st International Conference on Concurrency Theory (CONCUR 2020)
Adapted for Mobility Reading group 22/10/2020.

Ross Horne

Computer Science, University of Luxembourg

1-4 September 2020

Criticism 1: Deep Inference.

Criticism 1: Deep Inference.

> *"I must confess we were a little bit hampered by our lack of familiarity with the calculus of structures."*

Criticism 1: Deep Inference.

> *"I must confess we were a little bit hampered by our lack of familiarity with the calculus of structures."*

The calculus of structures: Developed over the past 20 years. Enables the design of **analytic proof systems** for non-commutative logics. It's novelty is the use of *deep inference* — rules can be applied in *any context*.

Criticism 1: Deep Inference.

> *"I must confess we were a little bit hampered by our lack of familiarity with the calculus of structures."*

The calculus of structures: Developed over the past 20 years. Enables the design of **analytic proof systems** for non-commutative logics. It's novelty is the use of *deep inference* — rules can be applied in *any context*.

$$\frac{\vdash \mathcal{C}\{\,\mathsf{T} \otimes (\mathsf{U} \parr \mathsf{V})\,\}}{\vdash \mathcal{C}\{\,(\mathsf{T} \otimes \mathsf{U}) \parr \mathsf{V}\,\}}$$

Criticism 1: Deep Inference.

> *"I must confess we were a little bit hampered by our lack of familiarity with the calculus of structures."*

The calculus of structures: Developed over the past 20 years. Enables the design of **analytic proof systems** for non-commutative logics. It's novelty is the use of *deep inference* — rules can be applied in *any context*.

$$\frac{\vdash \mathcal{C}\{\, \mathsf{T} \otimes (\mathsf{U} \parr \mathsf{V}) \,\}}{\vdash \mathcal{C}\{\, (\mathsf{T} \otimes \mathsf{U}) \parr \mathsf{V} \,\}}$$

The sequent calculus: The original **analytic proof calculus** of Gentzen. Published in 1934, so is widely understood. Rules are applied to the *root connective* of a formula selected from a sequence of formulae.

Criticism 1: Deep Inference.

> *"I must confess we were a little bit hampered by our lack of familiarity with the calculus of structures."*

The calculus of structures: Developed over the past 20 years. Enables the design of **analytic proof systems** for non-commutative logics. It's novelty is the use of *deep inference* — rules can be applied in *any context*.

$$\frac{\vdash \mathcal{C}\{\, \mathsf{T} \otimes (\mathsf{U} \parr \mathsf{V}) \,\}}{\vdash \mathcal{C}\{\, (\mathsf{T} \otimes \mathsf{U}) \parr \mathsf{V} \,\}}$$

The sequent calculus: The original **analytic proof calculus** of Gentzen. Published in 1934, so is widely understood. Rules are applied to the *root connective* of a formula selected from a sequence of formulae.

$$\frac{\vdash \mathsf{T}\,,\, \mathsf{U}\,,\, \Gamma}{\vdash \mathsf{T} \parr \mathsf{U}\,,\, \Gamma} \qquad \frac{\vdash \mathsf{T}\,,\, \Gamma \quad \vdash \mathsf{U}\,,\, \Delta}{\vdash \mathsf{T} \otimes \mathsf{U}\,,\, \Gamma, \Delta}$$

[Times]
$$\frac{T\,,\,U\,,\,\Gamma \vdash}{T \otimes U\,,\,\Gamma \vdash}$$

[Par]
$$\frac{T\,,\,\Gamma_1 \vdash \quad U\,,\,\Gamma_2 \vdash}{T \parr U\,,\,\Gamma_1\,,\,\Gamma_2 \vdash}$$

[OK]
$$\mathtt{OK}\,,\,\mathtt{OK}\,,\,\ldots\,\mathtt{OK} \vdash$$

[TIMES]
$$\frac{\mathsf{T} \,,\, \mathsf{U} \,,\, \Gamma \vdash}{\mathsf{T} \otimes \mathsf{U} \,,\, \Gamma \vdash}$$

[PAR]
$$\frac{\mathsf{T} \,,\, \Gamma_1 \vdash \quad \mathsf{U} \,,\, \Gamma_2 \vdash}{\mathsf{T} \parr \mathsf{U} \,,\, \Gamma_1 \,,\, \Gamma_2 \vdash}$$

[OK]
$$\mathtt{OK} \,,\, \mathtt{OK} \,,\, \ldots \mathtt{OK} \vdash$$

[JOIN]
$$\frac{!\lambda_j;\mathsf{T}_j \,,\, \Gamma \vdash \quad \text{for } \textbf{all } j \in I}{\bigvee_{i \in I} !\lambda_j;\mathsf{T}_i \,,\, \Gamma \vdash}$$

[MEET]
$$\frac{?\lambda_j;\mathsf{T}_j \,,\, \Gamma \vdash \quad \text{for } \textbf{some } j \in I}{\bigwedge_{i \in I} ?\lambda_j;\mathsf{T}_i \,,\, \Gamma \vdash}$$

[PREFIX]
$$\frac{\mathsf{T} \,,\, \mathsf{U} \,,\, \Gamma \vdash}{!\lambda;\mathsf{T} \,,\, ?\lambda;\mathsf{U} \,,\, \Gamma \vdash}$$

$$[\textsc{Times}]\quad \frac{\mathsf{T}\,,\,\mathsf{U}\,,\,\Gamma \vdash}{\mathsf{T} \otimes \mathsf{U}\,,\,\Gamma \vdash}
\qquad
[\textsc{Par}]\quad \frac{\mathsf{T}\,,\,\Gamma_1 \vdash \quad \mathsf{U}\,,\,\Gamma_2 \vdash}{\mathsf{T} \parr \mathsf{U}\,,\,\Gamma_1\,,\,\Gamma_2 \vdash}
\qquad
[\mathrm{OK}]\quad \mathtt{OK}\,,\,\mathtt{OK}\,,\,\ldots\,\mathtt{OK} \vdash$$

$$[\textsc{Join}]\quad \frac{!\lambda_j;\mathsf{T}_j\,,\,\Gamma \vdash \quad \text{for } \textbf{all } j \in I}{\bigvee_{i \in I} !\lambda_j;\mathsf{T}_i\,,\,\Gamma \vdash}
\qquad
[\textsc{Meet}]\quad \frac{?\lambda_j;\mathsf{T}_j\,,\,\Gamma \vdash \quad \text{for } \textbf{some } j \in I}{\bigwedge_{i \in I} ?\lambda_j;\mathsf{T}_i\,,\,\Gamma \vdash}$$

$$[\textsc{Prefix}]\quad \frac{\mathsf{T}\,,\,\mathsf{U}\,,\,\Gamma \vdash}{!\lambda;\mathsf{T}\,,\,?\lambda;\mathsf{U}\,,\,\Gamma \vdash}$$

$$[\textsc{Intr}]\quad \frac{I \subseteq J \quad \mathsf{T}_k\,,\,\mathsf{U}_k\,,\,\Gamma \vdash \quad \text{for } \textbf{all } k \in I}{\bigvee_{i \in I} !\lambda_i;\mathsf{T}_i\,,\,\bigwedge_{j \in J} ?\lambda_j;\mathsf{U}_j\,,\,\Gamma \vdash}$$

Criticism 2: Session type systems should feature recursion.[1]

[1] Thanks to discussions with Mariangiola Dezani-Ciancaglini and Paola Giannini.

Criticism 2: Session type systems should feature recursion.[1]

Observation 1: Most session calculi are restricted to a **regular** setting — a bounded number of single threaded participants.

[1] Thanks to discussions with Mariangiola Dezani-Ciancaglini and Paola Giannini.

Criticism 2: Session type systems should feature recursion.[1]

Observation 1: Most session calculi are restricted to a **regular** setting — a bounded number of single threaded participants.

Observation 2: In the regular setting, we can use **equirecursion** from type theory — fixed points are equivalent to their infinite unfoldings.

[1] Thanks to discussions with Mariangiola Dezani-Ciancaglini and Paola Giannini.

Criticism 2: Session type systems should feature recursion.[1]

Observation 1: Most session calculi are restricted to a **regular** setting — a bounded number of single threaded participants.

Observation 2: In the regular setting, we can use **equirecursion** from type theory — fixed points are equivalent to their infinite unfoldings.

Observation 3: In proof theory, such regular recursive proofs are **circular proofs**.

[1] Thanks to discussions with Mariangiola Dezani-Ciancaglini and Paola Giannini.

Criticism 2: Session type systems should feature recursion.[1]

Observation 1: Most session calculi are restricted to a **regular** setting — a bounded number of single threaded participants.

Observation 2: In the regular setting, we can use **equirecursion** from type theory — fixed points are equivalent to their infinite unfoldings.

Observation 3: In proof theory, such regular recursive proofs are **circular proofs**.

Design choice: Apply an algorithmic approach to equirecursive subtyping, due to Pierce and Sangiorgi, to make proofs in the sequent calculus circular.

$$\text{[Fix-}\mu\text{]}\ \frac{[\Theta\ ]\!\![\ \mu\mathbf{t}.\mathsf{T}\ ,\ \Gamma]\ \mathsf{T}\left\{{}^{\mu\mathbf{t}.\mathsf{T}}\!/\!_{\mathbf{t}}\right\}\ ,\ \Gamma \vdash}{[\Theta]\ \mu\mathbf{t}.\mathsf{T}\ ,\ \Gamma \vdash} \qquad \text{[Leaf]}\ [\Theta\ ]\!\![\ \Gamma]\ \Gamma \vdash$$

[1] Thanks to discussions with Mariangiola Dezani-Ciancaglini and Paola Giannini.

An Example

$$\dfrac{}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash} \; [\textsc{Times}]$$

Abbreviations :

An Example

$$\dfrac{\dfrac{}{\mathsf{U}\ ,\ \mathsf{V}\ ,\ \mathsf{T} \vdash}\ [\textsc{Fix}\text{-}\mu]}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash}\ [\textsc{Times}]$$

Abbreviations :

$$\begin{aligned} \mathsf{U} &= \quad \mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \\ \mathsf{V} &= \quad \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \\ \mathsf{T} &= \quad \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \end{aligned}$$

An Example

$$\cfrac{\cfrac{\overline{[\Gamma]\ \mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_1;\mathsf{T} \vee !\lambda_2;\mathsf{T} \vdash}\ [\textsc{Join}]}{\mathsf{U}\ ,\ \mathsf{V}\ ,\ \mathsf{T} \vdash}\ [\textsc{Fix}\text{-}\mu]}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash}\ [\textsc{Times}]$$

Abbreviations :

$$\begin{aligned} \mathsf{U} &= \mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \\ \mathsf{V} &= \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \\ \mathsf{T} &= \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \\ \Gamma &= \mathsf{U}\ ,\ \mathsf{V}\ ,\ \mathsf{T} \end{aligned}$$

An Example

$$
\dfrac{
  \dfrac{
    \dfrac{\dfrac{}{[\Gamma]\, \mathsf{U} , \mathsf{V} , !\lambda_1;\mathsf{T} \vdash}\,[\textsc{Fix}\text{-}\mu]
      \qquad
      \dfrac{}{[\Gamma]\, \mathsf{U} , \mathsf{V} , !\lambda_2;\mathsf{T} \vdash}\,[\textsc{Fix}\text{-}\mu]}
    {[\Gamma]\, \mathsf{U} , \mathsf{V} , !\lambda_1;\mathsf{T} \vee !\lambda_2;\mathsf{T} \vdash}\,[\textsc{Join}]
  }{\mathsf{U} , \mathsf{V} , \mathsf{T} \vdash}\,[\textsc{Fix}\text{-}\mu]
}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash}\,[\textsc{Times}]
$$

Abbreviations :

$$
\begin{aligned}
\mathsf{U} &= \mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \\
\mathsf{V} &= \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \\
\mathsf{T} &= \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \\
\Gamma &= \mathsf{U} , \mathsf{V} , \mathsf{T}
\end{aligned}
$$

An Example

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{[\Gamma' \,][\, \Gamma]\ ?\lambda_1;\mathsf{U} \,,\, \mathsf{V} \,,\, !\lambda_1;\mathsf{T} \vdash}\,[\textsc{Prefix}]}{[\Gamma]\ \mathsf{U} \,,\, \mathsf{V} \,,\, !\lambda_1;\mathsf{T} \vdash}\,[\textsc{Fix-}\mu] \qquad \dfrac{\dfrac{}{[\Gamma'' \,][\, \Gamma]\ \mathsf{U} \,,\, ?\lambda_2;\mathsf{V} \,,\, !\lambda_2;\mathsf{T} \vdash}\,[\textsc{Prefix}]}{[\Gamma]\ \mathsf{U} \,,\, \mathsf{V} \,,\, !\lambda_2;\mathsf{T} \vdash}\,[\textsc{Fix-}\mu]}{\dfrac{[\Gamma]\ \mathsf{U} \,,\, \mathsf{V} \,,\, !\lambda_1;\mathsf{T} \vee !\lambda_2;\mathsf{T} \vdash}{\mathsf{U} \,,\, \mathsf{V} \,,\, \mathsf{T} \vdash}\,[\textsc{Fix-}\mu]}\,[\textsc{Join}]}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash}\,[\textsc{Times}]
$$

Abbreviations :

$$
\begin{aligned}
\mathsf{U} &= \mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \\
\mathsf{V} &= \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \\
\mathsf{T} &= \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \\
\Gamma &= \mathsf{U} \,,\, \mathsf{V} \,,\, \mathsf{T} \\
\Gamma' &= \mathsf{U} \,,\, \mathsf{V} \,,\, !\lambda_1;\mathsf{T} \\
\Gamma'' &= \mathsf{U} \,,\, \mathsf{V} \,,\, !\lambda_2;\mathsf{T}
\end{aligned}
$$

An Example

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{[\Gamma' \,][\, \Gamma]\ \Gamma \vdash}\;[\textsc{Leaf}]
      }{[\Gamma' \,][\, \Gamma]\ ?\lambda_1;\mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_1;\mathsf{T} \vdash}\;[\textsc{Prefix}]
    }{[\Gamma]\ \mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_1;\mathsf{T} \vdash}\;[\textsc{Fix}\text{-}\mu]
    \qquad
    \dfrac{
      \dfrac{
        \dfrac{}{[\Gamma'' \,][\, \Gamma]\ \Gamma \vdash}\;[\textsc{Leaf}]
      }{[\Gamma'' \,][\, \Gamma]\ \mathsf{U}\ ,\ ?\lambda_2;\mathsf{V}\ ,\ !\lambda_2;\mathsf{T} \vdash}\;[\textsc{Prefix}]
    }{[\Gamma]\ \mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_2;\mathsf{T} \vdash}\;[\textsc{Fix}\text{-}\mu]
  }{
    \dfrac{[\Gamma]\ \mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_1;\mathsf{T} \vee !\lambda_2;\mathsf{T} \vdash}{\mathsf{U}\ ,\ \mathsf{V}\ ,\ \mathsf{T} \vdash}\;[\textsc{Fix}\text{-}\mu]
  }\;[\textsc{Join}]
}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash}\;[\textsc{Times}]
$$

Abbreviations :

$$
\begin{array}{rcl}
\mathsf{U} & = & \mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \\
\mathsf{V} & = & \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \\
\mathsf{T} & = & \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \\
\Gamma & = & \mathsf{U}\ ,\ \mathsf{V}\ ,\ \mathsf{T} \\
\Gamma' & = & \mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_1;\mathsf{T} \\
\Gamma'' & = & \mathsf{U}\ ,\ \mathsf{V}\ ,\ !\lambda_2;\mathsf{T}
\end{array}
$$

An Example

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{[\Gamma'][\Gamma]\;\Gamma \vdash}\;[\textsc{Leaf}]
}{[\Gamma'][\Gamma]\;?\lambda_1;\mathsf{U}\,,\,\mathsf{V}\,,\,!\lambda_1;\mathsf{T} \vdash}\;[\textsc{Prefix}]
}{[\Gamma]\;\mathsf{U}\,,\,\mathsf{V}\,,\,!\lambda_1;\mathsf{T} \vdash}\;[\textsc{Fix-}\mu]
\qquad
\dfrac{
\dfrac{
\dfrac{}{[\Gamma''][\Gamma]\;\Gamma \vdash}\;[\textsc{Leaf}]
}{[\Gamma''][\Gamma]\;\mathsf{U}\,,\,?\lambda_2;\mathsf{V}\,,\,!\lambda_2;\mathsf{T} \vdash}\;[\textsc{Prefix}]
}{[\Gamma]\;\mathsf{U}\,,\,\mathsf{V}\,,\,!\lambda_2;\mathsf{T} \vdash}\;[\textsc{Fix-}\mu]
}{
\dfrac{[\Gamma]\;\mathsf{U}\,,\,\mathsf{V}\,,\,!\lambda_1;\mathsf{T} \vee !\lambda_2;\mathsf{T} \vdash}{\mathsf{U}\,,\,\mathsf{V}\,,\,\mathsf{T} \vdash}\;[\textsc{Fix-}\mu]
}\;[\textsc{Join}]
}{
\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash
}\;[\textsc{Times}]
$$

Abbreviations :

$$
\begin{aligned}
\mathsf{U} &= \mu\mathbf{u}.(?\lambda_1;\mathbf{u})\\
\mathsf{V} &= \mu\mathbf{v}.(?\lambda_2;\mathbf{v})\\
\mathsf{T} &= \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t})\\
\Gamma &= \mathsf{U}\,,\,\mathsf{V}\,,\,\mathsf{T}\\
\Gamma' &= \mathsf{U}\,,\,\mathsf{V}\,,\,!\lambda_1;\mathsf{T}\\
\Gamma'' &= \mathsf{U}\,,\,\mathsf{V}\,,\,!\lambda_2;\mathsf{T}
\end{aligned}
$$

Multiparty Compatibility: Proves the following threads are multiparty compatible.

$$\mu Y.(?\lambda_1;Y) \quad \| \quad \mu Z.(?\lambda_2;Z) \quad \| \quad \mu X.(!\lambda_1;X \oplus !\lambda_2;X)$$

An Example

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\dfrac{}{[\Gamma' \rrbracket\, \Gamma]\; \Gamma \vdash}\,[\textsc{Leaf}]}{[\Gamma' \rrbracket\, \Gamma]\; ?\lambda_1;\mathsf{U}\;,\;\mathsf{V}\;,\;!\lambda_1;\mathsf{T} \vdash}\,[\textsc{Prefix}]
}{[\Gamma]\;\mathsf{U}\;,\;\mathsf{V}\;,\;!\lambda_1;\mathsf{T} \vdash}\,[\textsc{Fix-}\mu]
\qquad
\dfrac{
\dfrac{\dfrac{}{[\Gamma'' \rrbracket\, \Gamma]\; \Gamma \vdash}\,[\textsc{Leaf}]}{[\Gamma'' \rrbracket\, \Gamma]\; \mathsf{U}\;,\;?\lambda_2;\mathsf{V}\;,\;!\lambda_2;\mathsf{T} \vdash}\,[\textsc{Prefix}]
}{[\Gamma]\;\mathsf{U}\;,\;\mathsf{V}\;,\;!\lambda_2;\mathsf{T} \vdash}\,[\textsc{Fix-}\mu]
}{[\Gamma]\;\mathsf{U}\;,\;\mathsf{V}\;,\;!\lambda_1;\mathsf{T} \vee !\lambda_2;\mathsf{T} \vdash}\,[\textsc{Join}]
}{
\dfrac{\mathsf{U}\;,\;\mathsf{V}\;,\;\mathsf{T} \vdash}{\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \otimes \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \otimes \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t}) \vdash}\,[\textsc{Times}]
}\,[\textsc{Fix-}\mu]
$$

Abbreviations :

$$
\begin{aligned}
\mathsf{U} &= \mu\mathbf{u}.(?\lambda_1;\mathbf{u})\\
\mathsf{V} &= \mu\mathbf{v}.(?\lambda_2;\mathbf{v})\\
\mathsf{T} &= \mu\mathbf{t}.(!\lambda_1;\mathbf{t} \vee !\lambda_2;\mathbf{t})\\
\Gamma &= \mathsf{U}\;,\;\mathsf{V}\;,\;\mathsf{T}\\
\Gamma' &= \mathsf{U}\;,\;\mathsf{V}\;,\;!\lambda_1;\mathsf{T}\\
\Gamma'' &= \mathsf{U}\;,\;\mathsf{V}\;,\;!\lambda_2;\mathsf{T}
\end{aligned}
$$

Multiparty Compatibility: Proves the following threads are multiparty compatible.

$$\mu Y.(?\lambda_1;Y) \quad \| \quad \mu Z.(?\lambda_2;Z) \quad \| \quad \mu X.(!\lambda_1;X \oplus !\lambda_2;X)$$

Subtyping: Establishes the following subtype relation ($\mathsf{U} \otimes \mathsf{V} \leq \mathsf{T}$ iff $\mathsf{U} \otimes \mathsf{V} \otimes \overline{\mathsf{T}} \vdash$).

$$\mu\mathbf{u}.(?\lambda_1;\mathbf{u}) \quad \otimes \quad \mu\mathbf{v}.(?\lambda_2;\mathbf{v}) \quad \leq \quad \mu\mathbf{t}.(?\lambda_1;\mathbf{t} \wedge ?\lambda_2;\mathbf{t})$$

The Cut Elimination "Gold Mine" (again)

Theorem (cut elimination)

The rule $$\frac{\Gamma_1 \, , \, T \vdash \qquad \overline{T} \, , \, \Gamma_2 \vdash}{\Gamma_1 \, , \, \Gamma_2 \vdash}\,[\textsc{Cut}]$$ *is admissible in Session.*

The Cut Elimination "Gold Mine" (again)

Theorem (cut elimination)

The rule $\dfrac{\Gamma_1 \,,\, \mathsf{T} \vdash \qquad \overline{\mathsf{T}} \,,\, \Gamma_2 \vdash}{\Gamma_1 \,,\, \Gamma_2 \vdash}\,[\textsc{Cut}]$ *is admissible in Session.*

Corollary (algorithmic subtyping)

Subtyping is a decidable preorder.

Theorem (algorithmic typing)

All instances of $\dfrac{\Delta \vdash P : \mathsf{T} \qquad \mathsf{T} \leq \mathsf{U}}{\Delta \vdash P : \mathsf{U}}\,[\textsc{subsumption}]$ *can be pushed to the bottom of a type derivation.*

Theorem (deadlock freedom)

*Any **race-free** multiparty-compatible network satisfies deadlock freedom.*

Corollary (substitution principle)

P can replace Q while preserving multiparty compatibility,
whenever $\mathsf{T} \leq \mathsf{U}$, *where* $\vdash P : \mathsf{T}$ *and* $\vdash Q : \mathsf{U}$.

Now I see! So, what cool things can you do?

Owner:

$$
\begin{array}{l}
?\mathit{login\_page}(\mathit{app\_ID}, \mathit{scope});\\
!\mathit{deny} \oplus !\mathit{authorise}(\mathit{name}, \mathit{password})
\end{array}
$$

Trusted App:

$$
\begin{array}{l}
!\mathit{login\_page}(\mathit{app\_ID}, \mathit{scope});\\
?\mathit{deny};!\mathit{release}\\
+\ ?\mathit{authorise}(\mathit{name}, \mathit{password});\\
\quad \mathtt{rec}Y.\ !\mathit{release}\\
\qquad\qquad \oplus\ !\mathit{request}(\mathit{token});\\
\qquad\qquad\quad ?\mathit{revoke} + ?\mathit{response}(\mathit{data});Y
\end{array}
$$

Resource:

$$
\begin{array}{l}
\mathtt{rec}X.\ ?\mathit{release}\\
\qquad + \ ?\mathit{request}(\mathit{token});\\
\qquad\quad !\mathit{revoke} \oplus !\mathit{response}(\mathit{data});X
\end{array}
$$

Trusted App:

$$
\begin{array}{l}
!\mathit{login\_page}(\mathit{app\_ID}, \mathit{scope});\\
?\mathit{deny}; !\mathit{release}\\
+\ \ ?\mathit{authorise}(\mathit{name}, \mathit{password});\\
\quad\ \ \mathtt{rec}\,Y.!\mathit{release}\\
\quad\ \ \oplus\ \ !\mathit{request}(\mathit{token});\\
\qquad\quad ?\mathit{revoke} + ?\mathit{response}(\mathit{data}); Y
\end{array}
$$

$\mapsto$

OAuth 2.0 Server:

$$
\begin{array}{l}
?\mathit{initiate}(\mathit{app\_ID}, \mathit{scope});\\
!\mathit{login\_page}(\mathit{app\_ID}, \mathit{scope});\\
(?\mathit{deny}; !\mathit{close}; !\mathit{release})\\
+\ \ ?\mathit{authorise}(\mathit{name}, \mathit{password});\\
\quad\ \ (!\mathit{close}; !\mathit{release})\\
\quad\ \ \oplus\ \ !\mathit{authorisation\_code}(\mathit{code});\\
\qquad\quad ?\mathit{exchange}(\mathit{app\_ID}, \mathit{secret}, \mathit{code});\\
\qquad\quad (!\mathit{close}; !\mathit{release})\\
\qquad\quad \oplus\ \ !\mathit{access\_token}(\mathit{token})
\end{array}
$$

Untrusted App:

$$
\begin{array}{l}
!\mathit{initiate}(\mathit{add\_ID}, \mathit{scope});\\
?\mathit{close}\\
+\ \ ?\mathit{authorisation\_code}(\mathit{code});\\
\quad\ \ !\mathit{exchange}(\mathit{app\_ID}, \mathit{secret}, \mathit{code});\\
\quad\ \ ?\mathit{close}\\
\quad\ \ +\ \ ?\mathit{access\_token}(\mathit{token});\\
\qquad\quad \mathtt{rec}\,Y.\ \ !\mathit{request}(\mathit{token});\\
\qquad\qquad\qquad ?\mathit{revoke}\\
\qquad\qquad\qquad +\ \ ?\mathit{response}(\mathit{data}); Y
\end{array}
$$

Trusted App:

$$
\begin{array}{l}
!\mathit{login\_page}(\mathit{app\_ID}, \mathit{scope});\\
?\mathit{deny}; !\mathit{release}\\
+\ \ ?\mathit{authorise}(\mathit{name}, \mathit{password});\\
\quad \mathtt{rec}\, Y. !\mathit{release}\\
\quad \oplus\ \ !\mathit{request}(\mathit{token});\\
\qquad ?\mathit{revoke} + ?\mathit{response}(\mathit{data}); Y
\end{array}
$$

$\mapsto$

OAuth 2.0 Server:

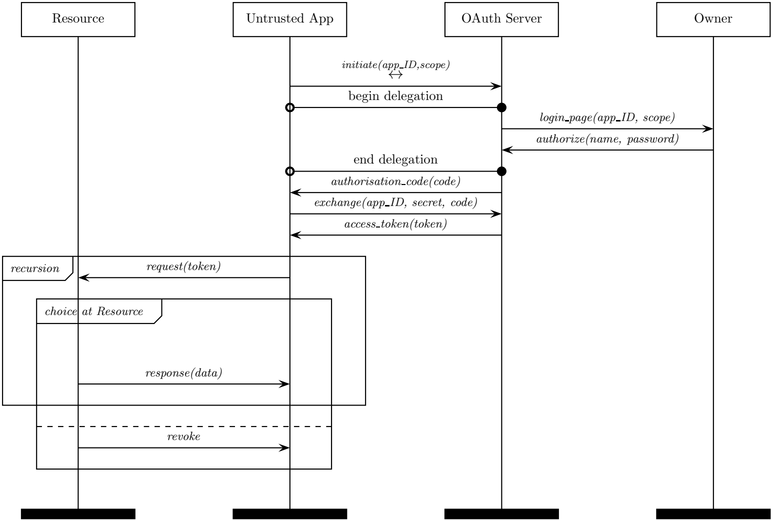
$$
\begin{array}{l}
?\mathit{initiate}(\mathit{app\_ID}, \mathit{scope});\\
!\mathit{login\_page}(\mathit{app\_ID}, \mathit{scope});\\
(?\mathit{deny}; !\mathit{close}; !\mathit{release})\\
+\ \ ?\mathit{authorise}(\mathit{name}, \mathit{password});\\
\quad (!\mathit{close}; !\mathit{release})\\
\quad \oplus\ \ !\mathit{authorisation\_code}(\mathit{code});\\
\qquad ?\mathit{exchange}(\mathit{app\_ID}, \mathit{secret}, \mathit{code});\\
\qquad (!\mathit{close}; !\mathit{release})\\
\qquad \oplus\ \ !\mathit{access\_token}(\mathit{token})
\end{array}
$$

Untrusted App:

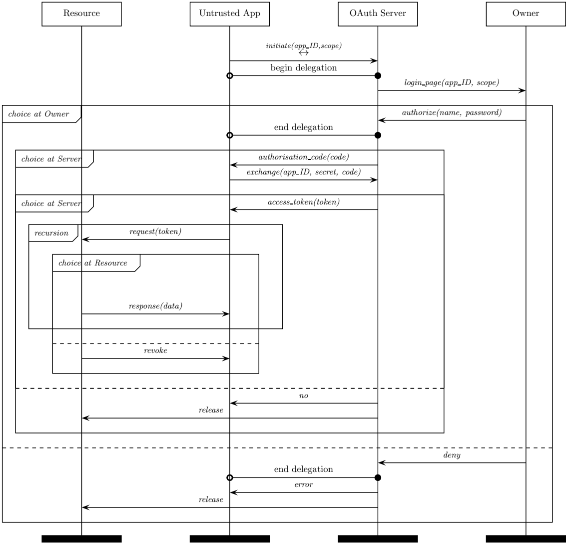
$$
\begin{array}{l}
!\mathit{initiate}(\mathit{add\_ID}, \mathit{scope});\\
?\mathit{close}\\
+\ \ ?\mathit{authorisation\_code}(\mathit{code});\\
\quad !\mathit{exchange}(\mathit{app\_ID}, \mathit{secret}, \mathit{code});\\
\quad ?\mathit{close}\\
\quad +\ \ ?\mathit{access\_token}(\mathit{token});\\
\qquad \mathtt{rec}\, Y.\ !\mathit{request}(\mathit{token});\\
\qquad\qquad ?\mathit{revoke}\\
\qquad\qquad +\ \ ?\mathit{response}(\mathit{data}); Y
\end{array}
$$

$$\boxed{\textit{Untrusted App} \otimes \textit{OAuth Server} \leq \textit{Trusted App}}$$

An application that delegates to an Oauth 2.0 server



Allowing the deputy to make a choice is useful



Internal delegation may liberate multiparty subtyping with roles.

Trusted App:

$$
\begin{array}{l}
\mathit{Owner}!\mathit{login\_page}(\mathit{app\_ID},\mathit{scope});\\
?\mathit{deny};!\mathit{release}\\
+\ \ \mathit{Owner}?\mathit{authorise}(\mathit{name},\mathit{password});\\
\quad \mathtt{rec}\,Y.!\mathit{release}\\
\quad \oplus\ \ \mathit{Resource}!\mathit{request}(\mathit{token});\\
\qquad ?\mathit{revoke}+\mathit{Resource}?\mathit{response}(\mathit{data});Y
\end{array}
$$

$$\longmapsto$$

$$
\begin{array}{l}
\mathit{App}?\mathit{initiate}(\mathit{app\_ID},\mathit{scope});\\
\mathit{App}\circ\langle\!\langle\bullet;\\
\mathit{Owner}!\mathit{login\_page}(\mathit{app\_ID},\mathit{scope});\\
(?\mathit{deny};\bullet\rangle\!\rangle\circ\mathit{App};!\mathit{close};!\mathit{release})\\
+\ \ \mathit{Owner}?\mathit{authorise}(\mathit{name},\mathit{password});\\
\quad \bullet\rangle\!\rangle\circ\mathit{App};\\
\quad (!\mathit{close};!\mathit{release})\\
\quad \oplus\ \ \mathit{App}!\mathit{authorisation\_code}(\mathit{code});\\
\qquad \mathit{App}?\mathit{exchange}(\mathit{app\_ID},\mathit{secret},\mathit{code});\\
\qquad (!\mathit{close};!\mathit{release})\\
\qquad \oplus\ \ \mathit{App}!\mathit{access\_token}(\mathit{token})
\end{array}
$$

$$
\begin{array}{l}
\mathit{OAuth}!\mathit{initiate}(\mathit{add\_ID},\mathit{scope});\\
\circ\langle\!\langle\bullet\mathit{OAuth};\ \mathit{OAuth}\bullet\rangle\!\rangle\circ;\\
?\mathit{close}\\
+\ \ \mathit{OAuth}?\mathit{authorisation\_code}(\mathit{code});\\
\quad \mathit{OAuth}!\mathit{exchange}(\mathit{app\_ID},\mathit{secret},\mathit{code});\\
\quad ?\mathit{close}\\
\quad +\ \ \mathit{OAuth}?\mathit{access\_token}(\mathit{token});\\
\qquad \mathtt{rec}\,Y.\,\mathit{OAuth}!\mathit{request}(\mathit{token});\\
\qquad\qquad ?\mathit{revoke}\\
\qquad\qquad +\ \ \mathit{Resource}?\mathit{response}(\mathit{data});Y
\end{array}
$$

Conclusion and discussion

Conclusion: Non-commutative logic + race-freedom provides us with rich notions of multiparty compatibility and subtyping.

Conclusion and discussion

Conclusion: Non-commutative logic + race-freedom provides us with rich notions of multiparty compatibility and subtyping.

Discussion: The follow has no global type, but is deadlock free and both "Kobayashi" and "Padovani" live (LIVE and LIVE+ respectively in POPL'19). System Session verifies this (but only guarantees deadlock freedom without further modifications).

$$\mu X.\,(!\lambda_1\,;X \oplus !\lambda_2) \parallel \mu Y.\,(?\lambda_1;\,Y + ?\lambda_2) \parallel \mu X.\,(!\lambda_3;\,X \oplus !\lambda_4) \parallel \mu Y.\,(?\lambda_3;\,Y + ?\lambda_4)$$

Conclusion and discussion

Conclusion: Non-commutative logic + race-freedom provides us with rich notions of multiparty compatibility and subtyping.

Discussion: The follow has no global type, but is deadlock free and both "Kobayashi" and "Padovani" live (LIVE and LIVE+ respectively in POPL'19). System SESSION verifies this (but only guarantees deadlock freedom without further modifications).

$$\mu X.\,(!\lambda_1\,;\,X \oplus !\lambda_2) \parallel \mu Y.\,(?\lambda_1\,;\,Y + ?\lambda_2) \parallel \mu X.\,(!\lambda_3;\,X \oplus !\lambda_4) \parallel \mu Y.\,(?\lambda_3;\,Y + ?\lambda_4)$$

Question for the Mobility Reading Group: What established extensions of global types allow the above to be typed and also guarantee livelock freedom (or, at least, deadlock freedom)?