Session Types for Orchestrated Interactions

Franco Barbanera, Ugo de'Liguoro

University of Catania, University of Torino

ICE - Neuchâtel, June 2017

Overview

Overview

- Retractable/speculative contracts

Overview

- Retractable/speculative contracts
- Session Types with speculative selection

Overview

- Retractable/speculative contracts
- Session Types with speculative selection
- Orchestrated Compliance and Session Types

Session Contracts (highlight)

Session Contracts (highlight)

- A “session” restriction of contracts.

Session Contracts (highlight)

- A "session" restriction of contracts.
- $$\begin{array}{rcll} \sigma, \tau & ::= & \mathbf{1} & \textit{inaction} \\ & | & a_1.\sigma_1 + \cdots + a_n.\sigma_n & \textit{external choice} \\ & | & \overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n & \textit{internal choice} \end{array}$$

  the a_i and the $\overline{a}_i$ are, respectively, pairwise distinct; recursion is *guarded*.

Session Contracts (highlight)

- A "session" restriction of contracts.
- $$\begin{array}{rcll} \sigma, \tau & ::= & \mathbf{1} & \textit{inaction} \\ & | & a_1.\sigma_1 + \cdots + a_n.\sigma_n & \textit{external choice} \\ & | & \overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n & \textit{internal choice} \end{array}$$

  the a_i and the $\overline{a}_i$ are, respectively, pairwise distinct; recursion is *guarded*.
- $\alpha.\rho + \rho' \parallel \overline{\alpha}.\sigma \quad \rightarrow \quad \rho \parallel \sigma$

Session Contracts (highlight)

- A "session" restriction of contracts.
-

$$\begin{array}{rcll} \sigma, \tau & ::= & \mathbf{1} & \textit{inaction} \\ & | & a_1.\sigma_1 + \cdots + a_n.\sigma_n & \textit{external choice} \\ & | & \overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n & \textit{internal choice} \end{array}$$

  the a_i and the $\overline{a}_i$ are, respectively, pairwise distinct;
  recursion is *guarded*.
- $\alpha.\rho + \rho' \parallel \overline{\alpha}.\sigma \quad \rightarrow \quad \rho \parallel \sigma$
- Used to model session types and their subtype relation:

Session Contracts (highlight)

- A "session" restriction of contracts.
- $$\begin{array}{lll} \sigma, \tau \;::=\; & \mathbf{1} & \textit{inaction} \\ & \mid\; a_1.\sigma_1 + \cdots + a_n.\sigma_n & \textit{external choice} \\ & \mid\; \overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n & \textit{internal choice} \end{array}$$

  the a_i and the $\overline{a}_i$ are, respectively, pairwise distinct;
  recursion is *guarded*.
- $\alpha.\rho + \rho' \parallel \overline{\alpha}.\sigma \quad \rightarrow \quad \rho \parallel \sigma$
- Used to model session types and their subtype relation:

References

- Barbanera and de' Liguoro: *Sub-behaviour relations for session-based client/server systems* (with higher-order)
- Bernardi and Hennessy: *Modelling session types using contracts*

Compliance

Compliance

Central notion in contract theory.

Compliance

Central notion in contract theory.

The *client* ρ is compliant with the *server* σ, $\rho \dashv \sigma$, if

$$\rho \parallel \sigma \xrightarrow{\;*\;} \rho' \parallel \sigma' \not\longrightarrow \quad \text{implies} \quad \rho' = \mathbf{1}$$

Compliance

Central notion in contract theory.

The *client* ρ is compliant with the *server* σ, $\rho \dashv \sigma$, if

$$\rho \parallel \sigma \xrightarrow{\ *\ } \rho' \parallel \sigma' \not\longrightarrow \quad \text{implies} \quad \rho' = \mathbf{1}$$

Examples (without trailing $\mathbf{1}$'s):

- $\mathbf{1} \dashv \sigma$
- $\overline{a} \dashv a.b$ but $\overline{a}.\overline{b} \not\dashv a$
- $\overline{a} \dashv a + b$ and $a + b \dashv \overline{a}$ but $\overline{a} \oplus \overline{b} \not\dashv a$ and $a \not\dashv \overline{a} \oplus \overline{b}$
- $\mathrm{rec}\, x.\ \overline{a}.x \oplus \overline{b} \dashv \mathrm{rec}\, x.\ a.x + b$ but $\mathrm{rec}\, x.\ \overline{a}.x \oplus \overline{b} \not\dashv \mathrm{rec}\, x.\ a.x$

Relaxing the Compliance relation

Relaxing the Compliance relation

Compliance:

Relaxing the Compliance relation

Compliance:

Any internal choice in $\overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n$ *must find a dual match in the partner.*

Relaxing the Compliance relation

Compliance:

Any internal choice in $\overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n$ *must find a dual match in the partner.*

Among several possibilities:

Relaxing the Compliance relation

Compliance:

Any internal choice in $\overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n$ *must find a dual match in the partner.*

Among several possibilities:

- Retractable session contracts [BDLdL Places 2015, BLdL Coordination 2017]
  Particular internal choices (the retractable ones) can be "tested" for compliance. Backtrack in case of failure.
  failure = no possible synchronization.

Relaxing the Compliance relation

Compliance:
Any internal choice in $\overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n$ *must find a dual match in the partner.*

Among several possibilities:

- RETRACTABLE SESSION CONTRACTS [BDLdL Places 2015, BLdL Coordination 2017]
  Particular internal choices (the retractable ones) can be "tested" for compliance. Backtrack in case of failure.
  failure = no possible synchronization.
- ORCHESTRATED SESSION CONTRACTS [BdL ICE 2016] inspired to [Padovani 2010]
  Particular internal choices are "driven" by a mediating process (the orchestrator) in order to get compliance.

Relaxing the Compliance relation

Compliance:

Any internal choice in $\overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n$ *must find a dual match in the partner.*

Among several possibilities:

- RETRACTABLE SESSION CONTRACTS [BDLdL Places 2015, BLdL Coordination 2017]
  Particular internal choices (the retractable ones) can be "tested" for compliance. Backtrack in case of failure.
  failure = no possible synchronization.
- ORCHESTRATED SESSION CONTRACTS [BdL ICE 2016] inspired to [Padovani 2010]
  Particular internal choices are "driven" by a mediating process (the orchestrator) in order to get compliance.

Using *contracts as games on event structures* [Bartoletti et al. 2016]:

Theorem (Coordination 2016)

retractable compliance = *orchestrated compliance*

Moving to Session Types

Moving to Session Types

TYPES

Standard:

- $?[T].S \quad (![T].S)$ — Input(Output) type (also HO)
- $\bigoplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Selection type
- $\&\{l_1{:}S'_1, \ldots, l_n{:}S'_n\}$ — Branch type

Moving to Session Types

TYPES

Standard:

- $?[T].S \quad (![T].S)$ — Input(Output) type (also HO)
- $\bigoplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Selection type
- $\&\{l_1{:}S'_1, \ldots, l_n{:}S'_n\}$ — Branch type

NEW:

- $\boxplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Speculative selection type

Moving to Session Types

TYPES

Standard:

- $?[T].S \quad (![T].S)$ — Input(Output) type (also HO)
- $\bigoplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Selection type
- $\&\{l_1{:}S'_1, \ldots, l_n{:}S'_n\}$ — Branch type

NEW:

- $\boxplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Speculative selection type

PROCESSES

Standard (but for minor differences):

Moving to Session Types

TYPES

Standard:

- $?[T].S \quad (![T].S)$ — Input(Output) type (also HO)
- $\bigoplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Selection type
- $\&\{l_1{:}S'_1, \ldots, l_n{:}S'_n\}$ — Branch type

NEW:

- $\boxplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Speculative selection type

PROCESSES

Standard (but for minor differences):

- $request_S(k)P$ — Session request (usually $\overline{a}(k)P$)
- $accept_S(k)P$ — Session accept (usually $a(k)P$)

Moving to Session Types

TYPES

Standard:

- $?[T].S \quad (![T].S)$ — Input(Output) type (also HO)
- $\bigoplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Selection type
- $\&\{l_1{:}S'_1, \ldots, l_n{:}S'_n\}$ — Branch type

NEW:

- $\boxplus\{l_1{:}S_1, \ldots, l_n{:}S_n\}$ — Speculative selection type

PROCESSES

Standard (but for minor differences):

- $request_S(k)P$ — Session request (usually $\overline{a}(k)P$)
- $accept_S(k)P$ — Session accept (usually $a(k)P$)

NEW:

- $k \triangleleft [l_i{:}P_i]_{i \in I}$ — Speculative selection

Example

Let $\quad Buyer = request_{\text{BuyerType}}(k)P$

$$\text{BuyerType} = \bigoplus \left\{ \begin{array}{ll} belt & :?[Int].\bigoplus\left\{ \begin{array}{ll} card & : \text{PAYcard}, \\ cash & : \text{PAYcash}, \\ no & : \mathbf{1} \end{array} \right\} \\ bag & :?[Int].\bigoplus\left\{ \begin{array}{ll} card & : \text{PAYcard}, \\ cash & : \text{PAYcash}, \\ no & : \mathbf{1} \end{array} \right\} \end{array} \right\}$$

Example

Let $\quad Buyer = request_{\text{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\text{SellerType}}(k)Q$

$$
\begin{array}{lll}
\text{BuyerType} = \bigoplus \{ & belt :?[\mathsf{I}nt].\bigoplus\{ & card : \text{PAYcard}, \\
 & & cash : \text{PAYcash}, \\
 & & no : \mathbf{1} \ \} \\
 & bag :?[\mathsf{I}nt].\bigoplus\{ & card : \text{PAYcard}, \\
 & & cash : \text{PAYcash}, \\
 & & no : \mathbf{1} \ \} \ \}
\end{array}
$$

$$
\begin{array}{lll}
\text{SellerType} = \&\{ & belt :![\mathsf{I}nt].\&\{ & cash : \text{GETcash}, \\
 & & no : \mathbf{1} \\
 & bag :![\mathsf{I}nt].\&\{ & card : \text{GETcard}, \\
 & & cash : \text{GETcash}, \\
 & & no : \mathbf{1}\} \ \}
\end{array}
$$

Example

Let $\quad Buyer = request_{\mathsf{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\mathsf{SellerType}}(k)Q$

$$\mathsf{BuyerType} = \bigoplus \left\{ \begin{array}{ll} belt & :?[\mathsf{I}nt].\bigoplus\{ \begin{array}{l} card : \mathsf{PAYcard}, \\ cash : \mathsf{PAYcash}, \\ no : \mathbf{1} \ \} \end{array} \\ bag & :?[\mathsf{I}nt].\bigoplus\{ \begin{array}{l} card : \mathsf{PAYcard}, \\ cash : \mathsf{PAYcash}, \\ no : \mathbf{1} \ \} \end{array} \end{array} \right\}$$

$$\mathsf{SellerType} = \&\left\{ \begin{array}{ll} belt & :![\mathsf{I}nt].\&\{ \begin{array}{l} cash : \mathsf{GETcash}, \\ no : \mathbf{1} \end{array} \\ bag & :![\mathsf{I}nt].\&\{ \begin{array}{l} card : \mathsf{GETcard}, \\ cash : \mathsf{GETcash}, \\ no : \mathbf{1}\} \end{array} \end{array} \right\}$$

No safe interaction between *Buyer* and *Seller*

Example

Let $\quad Buyer = request_{\mathsf{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\mathsf{SellerType}}(k)Q$

$$
\begin{aligned}
\mathsf{BuyerType} = \bigoplus \{ \;\; & \mathit{belt} : ?[\mathsf{Int}]. \bigoplus \{ \; \mathit{card} : \mathsf{PAYcard}, \\
& \qquad\qquad\qquad\quad \mathit{cash} : \mathsf{PAYcash}, \\
& \qquad\qquad\qquad\quad \mathit{no} \;\; : \mathbf{1} \; \} \\
& \mathit{bag} \;\; : ?[\mathsf{Int}]. \bigoplus \{ \; \mathit{card} : \mathsf{PAYcard}, \\
& \qquad\qquad\qquad\quad \mathit{cash} : \mathsf{PAYcash}, \\
& \qquad\qquad\qquad\quad \mathit{no} \;\; : \mathbf{1} \; \} \;\; \}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{SellerType} = \&\{ \;\; & \mathit{belt} : ![\mathsf{Int}]. \&\{ \; \mathit{cash} : \mathsf{GETcash}, \\
& \qquad\qquad\qquad \mathit{no} \;\; : \mathbf{1} \\
& \mathit{bag} \;\; : ![\mathsf{Int}]. \&\{ \; \mathit{card} : \mathsf{GETcard}, \\
& \qquad\qquad\qquad \mathit{cash} : \mathsf{GETcash}, \\
& \qquad\qquad\qquad \mathit{no} : \mathbf{1} \} \;\; \}
\end{aligned}
$$

No safe interaction between *Buyer* and *Seller*

Example

Let $\quad Buyer = request_{\mathsf{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\mathsf{SellerType}}(k)Q$

$$\begin{array}{lll} \mathsf{BuyerType} = \bigoplus \{ & belt \ :?[\mathsf{I}nt].\bigoplus\{ & card : \mathsf{PAYcard}, \\ & & cash : \mathsf{PAYcash}, \\ & & no \ \ : \mathbf{1} \ \} \\ & bag \ \ :?[\mathsf{I}nt].\bigoplus\{ & card : \mathsf{PAYcard}, \\ & & cash : \mathsf{PAYcash}, \\ & & no \ \ : \mathbf{1} \ \} \ \ \} \end{array}$$

$$\begin{array}{lll} \mathsf{SellerType} = \&\{ & belt \ :![\mathsf{I}nt].\&\{ & cash : \mathsf{GETcash}, \\ & & no \ \ : \mathbf{1} \\ & bag \ \ :![\mathsf{I}nt].\&\{ & card : \mathsf{GETcard}, \\ & & cash : \mathsf{GETcash}, \\ & & no : \mathbf{1} \} \ \ \} \end{array}$$

No safe interaction between *Buyer* and *Seller*

Example

Let $\quad Buyer = request_{\mathsf{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\mathsf{SellerType}}(k)Q$

$$\begin{aligned}\mathsf{BuyerType} = \bigoplus \{ \; & \textcolor{blue}{belt} : ?[\mathsf{Int}]. \bigoplus \{ \; \textcolor{red}{card} : \mathsf{PAYcard}, \quad \textcolor{red}{\text{Synch Failure}} \\ & \qquad\qquad\qquad\quad cash : \mathsf{PAYcash}, \\ & \qquad\qquad\qquad\quad no : \mathbf{1} \; \} \\ & bag : ?[\mathsf{Int}]. \bigoplus \{ \; card : \mathsf{PAYcard}, \\ & \qquad\qquad\qquad\quad cash : \mathsf{PAYcash}, \\ & \qquad\qquad\qquad\quad no : \mathbf{1} \; \} \; \}\end{aligned}$$

$$\begin{aligned}\mathsf{SellerType} = \& \{ \; & \textcolor{blue}{belt} : ![\mathsf{Int}]. \& \{ \; cash : \mathsf{GETcash}, \\ & \qquad\qquad\qquad no : \mathbf{1} \\ & bag : ![\mathsf{Int}]. \& \{ \; card : \mathsf{GETcard}, \\ & \qquad\qquad\qquad cash : \mathsf{GETcash}, \\ & \qquad\qquad\qquad no : \mathbf{1} \} \; \}\end{aligned}$$

! No safe interaction between *Buyer* and *Seller*

Example

Let $\quad Buyer = request_{\mathrm{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\mathrm{SellerType}}(k)Q$

$$\begin{aligned}\mathrm{BuyerType} = \boxplus \Big\{\ & belt :?[\mathrm{Int}].\bigoplus\{\ card : \mathrm{PAYcard},\\ & \qquad\qquad\qquad\ \ cash : \mathrm{PAYcash},\\ & \qquad\qquad\qquad\ \ no \ \ : \mathbf{1}\ \}\\ & bag :?[\mathrm{Int}].\bigoplus\{\ card : \mathrm{PAYcard},\\ & \qquad\qquad\qquad\ \ cash : \mathrm{PAYcash},\\ & \qquad\qquad\qquad\ \ no \ \ : \mathbf{1}\ \}\ \Big\}\end{aligned}$$

$$\begin{aligned}\mathrm{SellerType} = \&\Big\{\ & belt :![\mathrm{Int}].\&\{\ cash : \mathrm{GETcash},\\ & \qquad\qquad\qquad\ no \ \ : \mathbf{1}\\ & bag :![\mathrm{Int}].\&\{\ card : \mathrm{GETcard},\\ & \qquad\qquad\qquad\ cash : \mathrm{GETcash},\\ & \qquad\qquad\qquad\ no \ \ : \mathbf{1}\end{aligned}$$

Example

Let $\quad Buyer = request_{\text{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\text{SellerType}}(k)Q$

$$\text{BuyerType} = \boxplus \; \{\; belt : ?[\text{Int}].\bigoplus\{\; card : \text{PAYcard},\; cash : \text{PAYcash},\; no : \mathbf{1} \;\} \quad bag : ?[\text{Int}].\bigoplus\{\; card : \text{PAYcard},\; cash : \text{PAYcash},\; no : \mathbf{1} \;\} \;\}$$

$$\text{SellerType} = \&\{\; belt : ![\text{Int}].\&\{\; cash : \text{GETcash},\; no : \mathbf{1} \quad bag : ![\text{Int}].\&\{\; card : \text{GETcard},\; cash : \text{GETcash},\; no : \mathbf{1}$$

$\boxplus \;=\;$ Retractable choice
Orchestrated choice

Example

Let $\quad Buyer = request_{\mathsf{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\mathsf{SellerType}}(k)Q$

$$
\begin{aligned}
\mathsf{BuyerType} = \boxplus \; \{\; & belt : ?[\mathsf{Int}].\bigoplus\{\, card : \mathsf{PAYcard},\\
& \qquad\qquad\qquad\quad cash : \mathsf{PAYcash},\\
& \qquad\qquad\qquad\quad no \;\;: \mathbf{1} \,\}\\
& bag : ?[\mathsf{Int}].\bigoplus\{\, card : \mathsf{PAYcard},\\
& \qquad\qquad\qquad\quad cash : \mathsf{PAYcash},\\
& \qquad\qquad\qquad\quad no \;\;: \mathbf{1} \,\} \;\}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{SellerType} = \&\{\; & belt : ![\mathsf{Int}].\&\{\, cash : \mathsf{GETcash},\\
& \qquad\qquad\qquad\; no \;\;: \mathbf{1}\\
& bag : ![\mathsf{Int}].\&\{\, card : \mathsf{GETcard},\\
& \qquad\qquad\qquad\; cash : \mathsf{GETcash},\\
& \qquad\qquad\qquad\; no \;\;: \mathbf{1}
\end{aligned}
$$

$\boxplus \;=\;$ Retractable choice
Orchestrated choice

Equivalent session-contract semantics

Example

Let $\quad Buyer = request_{\text{BuyerType}}(k)P \quad$ and $\quad Seller = accept_{\text{SellerType}}(k)Q$

$$
\begin{aligned}
\text{BuyerType} = \boxplus \ \{ \ & belt : ?[\text{Int}]. \bigoplus\{ \ card : \text{PAYcard}, \\
& \qquad\qquad\qquad\quad cash : \text{PAYcash}, \\
& \qquad\qquad\qquad\quad no \quad : \mathbf{1} \ \} \\
& bag : ?[\text{Int}]. \bigoplus\{ \ card : \text{PAYcard}, \\
& \qquad\qquad\qquad\quad cash : \text{PAYcash}, \\
& \qquad\qquad\qquad\quad no \quad : \mathbf{1} \ \} \ \ \}
\end{aligned}
$$

$$
\begin{aligned}
\text{SellerType} = \&\{ \ & belt : ![\text{Int}].\&\{ \ cash : \text{GETcash}, \\
& \qquad\qquad\qquad no \quad : \mathbf{1} \\
& bag : ![\text{Int}].\&\{ \ card : \text{GETcard}, \\
& \qquad\qquad\qquad cash : \text{GETcash}, \\
& \qquad\qquad\qquad no \quad : \mathbf{1}
\end{aligned}
$$

$\boxplus \quad =$ Retractable choice
Orchestrated choice ✓

Equivalent session-contract semantics

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}$$

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}$$

where $\mathsf{f} \; = \; \textsc{bag}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no})$

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}$$

where $\mathsf{f} \;=\; \textsc{bag}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no})$

f is computable out of BuyerType and SellerType

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}$$

where $\mathsf{f} \;=\; \textsc{bag}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no})$

f is computable out of BuyerType and SellerType

Compatibility **cannot** be Duality

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}$$

where $\mathsf{f} \;=\; \textsc{bag}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no})$

$$\mathsf{f} \text{ is computable out of } \mathsf{BuyerType} \text{ and } \mathsf{SellerType}$$

Compatibility **cannot** be Duality

Sintactic Duality broken by $\boxplus$ cannot be recovered in a natural way.

Orchestrated Compliance for Session-types

$\dashv$ = orchestrated compatibility (compliance)

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}$$

where $\mathsf{f} \;=\; \textsc{bag}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no})$

f is computable out of BuyerType and SellerType

Compatibility **cannot** be Duality

Sintactic Duality broken by $\boxplus$ cannot be recovered in a natural way.

$\dashv$ for session types inspired by
retractable(orchestrated) compliance for session contracts.

Definition (Orchestrated compliance)

The relation $\mathsf{f} : S \dashv S'$ among the orchestrator f and session types S, S' is the least one such that:

1. $\mathbb{1} : \mathbf{1} \dashv S$, for any S,
2. if $\mathsf{f} : S \dashv S'$ then $\bullet.\mathsf{f} : \mathbf{?}[G].S \dashv \mathbf{!}[G].S'$ and $\bullet.\mathsf{f} : \mathbf{!}[T].S \dashv \mathbf{?}[T].S'$
3. if $\mathsf{f}_i : S_i \dashv S_i'$ for all $i \in I$ then, $\Sigma_{i \in I} l_i.\mathsf{f}_i : \bigoplus\{l_i{:}S_i\}_{i \in I} \dashv \&\{l_j{:}S_j'\}_{j \in I \cup J}$ and $\Sigma_{i \in I} l_i.\mathsf{f}_i : \&\{l_j{:}S_j\}_{j \in I \cup J} \dashv \bigoplus\{l_i{:}S_i'\}_{i \in I}$;
4. if $H \subseteq I \cap J$ with $H \neq \emptyset$ and $\mathsf{f}_h : S_h \dashv S_h'$ for all $h \in H$ then $\oplus_{h \in H} l_h.\mathsf{f}_h : \boxplus\{l_i{:}S_i\}_{i \in I} \dashv \&\{l_j{:}S_j'\}_{j \in J}$ and $\oplus_{h \in H} l_h.\mathsf{f}_h : \&\{l_j{:}S_j\}_{j \in J} \dashv \boxplus\{l_i{:}S_i'\}_{i \in I}$.

Definition (Orchestrated compliance)

The relation $\mathsf{f} : S \dashv S'$ among the orchestrator f and session types S, S' is the least one such that:

1. $\mathbb{1} : \mathbf{1} \dashv S$, for any S,
2. if $\mathsf{f} : S \dashv S'$ then $\bullet.\mathsf{f} : \mathbf{?}[G].S \dashv \mathbf{!}[G].S'$ and $\bullet.\mathsf{f} : \mathbf{!}[T].S \dashv \mathbf{?}[T].S'$
3. if $\mathsf{f}_i : S_i \dashv S_i'$ for all $i \in I$ then, $\Sigma_{i\in I} l_i.\mathsf{f}_i : \bigoplus\{l_i{:}S_i\}_{i\in I} \dashv \&\{l_j{:}S_j'\}_{j\in I\cup J}$ and $\Sigma_{i\in I} l_i.\mathsf{f}_i : \&\{l_j{:}S_j\}_{j\in I\cup J} \dashv \bigoplus\{l_i{:}S_i'\}_{i\in I}$;
4. if $H \subseteq I \cap J$ with $H \neq \emptyset$ and $\mathsf{f}_h : S_h \dashv S_h'$ for all $h \in H$ then
$\oplus_{h\in H} l_h.\mathsf{f}_h : \boxplus\{l_i{:}S_i\}_{i\in I} \dashv \&\{l_j{:}S_j'\}_{j\in J}$ and
$\oplus_{h\in H} l_h.\mathsf{f}_h : \&\{l_j{:}S_j\}_{j\in J} \dashv \boxplus\{l_i{:}S_i'\}_{i\in I}$.

- ASYMMETRIC (client-biased)

$$\mathsf{SellerType}' = \&\{\ \mathit{shoes} : \ldots\ , \mathit{belt} : \ldots, \mathit{bag} : \mathbf{!}[\mathsf{Int}].\&\{\ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash},\ \mathit{no} : \mathbf{!}[\mathit{coupon}]\ \}$$

Definition (Orchestrated compliance)

The relation $\mathsf{f} : S \dashv S'$ among the orchestrator f and session types S, S' is the least one such that:

1. $\mathbb{1} : \mathbf{1} \dashv S$, for any S,
2. if $\mathsf{f} : S \dashv S'$ then $\bullet.\mathsf{f} : ?[G].S \dashv ![G].S'$ and $\bullet.\mathsf{f} : ![T].S \dashv ?[T].S'$
3. if $\mathsf{f}_i : S_i \dashv S_i'$ for all $i \in I$ then, $\Sigma_{i\in I} l_i.\mathsf{f}_i : \bigoplus\{l_i{:}S_i\}_{i\in I} \dashv \&\{l_j{:}S_j'\}_{j\in I\cup J}$ and $\Sigma_{i\in I} l_i.\mathsf{f}_i : \&\{l_j{:}S_j\}_{j\in I\cup J} \dashv \bigoplus\{l_i{:}S_i'\}_{i\in I}$;
4. if $H \subseteq I \cap J$ with $H \neq \emptyset$ and $\mathsf{f}_h : S_h \dashv S_h'$ for all $h \in H$ then
   $\oplus_{h\in H} l_h.\mathsf{f}_h : \boxplus\{l_i{:}S_i\}_{i\in I} \dashv \&\{l_j{:}S_j'\}_{j\in J}$ and
   $\oplus_{h\in H} l_h.\mathsf{f}_h : \&\{l_j{:}S_j\}_{j\in J} \dashv \boxplus\{l_i{:}S_i'\}_{i\in I}$.

- ASYMMETRIC (client-biased)

$$\mathsf{SellerType}' = \&\{\ \mathit{shoes} : \ldots\ , \mathit{belt} : \ldots, \mathit{bag} : ![\mathsf{Int}].\&\{\ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash},\ \mathit{no} : ![\mathit{coupon}]\ \}$$

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}'$$

Definition (Orchestrated compliance)

The relation $\mathsf{f} : S \dashv S'$ among the orchestrator f and session types S, S' is the least one such that:

1. $\mathbb{1} : \mathbf{1} \dashv S$, for any S,
2. if $\mathsf{f} : S \dashv S'$ then $\bullet.\mathsf{f} : ?[G].S \dashv ![G].S'$ and $\bullet.\mathsf{f} : ![T].S \dashv ?[T].S'$
3. if $\mathsf{f}_i : S_i \dashv S'_i$ for all $i \in I$ then, $\Sigma_{i\in I} l_i.\mathsf{f}_i : \bigoplus\{l_i{:}S_i\}_{i\in I} \dashv \&\{l_j{:}S'_j\}_{j\in I\cup J}$ and $\Sigma_{i\in I} l_i.\mathsf{f}_i : \&\{l_j{:}S_j\}_{j\in I\cup J} \dashv \bigoplus\{l_i{:}S'_i\}_{i\in I}$;
4. if $H \subseteq I \cap J$ with $H \neq \emptyset$ and $\mathsf{f}_h : S_h \dashv S'_h$ for all $h \in H$ then $\oplus_{h\in H} l_h.\mathsf{f}_h : \boxplus\{l_i{:}S_i\}_{i\in I} \dashv \&\{l_j{:}S'_j\}_{j\in J}$ and $\oplus_{h\in H} l_h.\mathsf{f}_h : \&\{l_j{:}S_j\}_{j\in J} \dashv \boxplus\{l_i{:}S'_i\}_{i\in I}$.

- ASYMMETRIC (client-biased)

$$\mathsf{SellerType}' = \&\{\ shoes : \dots\ , belt : \dots, bag : ![\mathsf{Int}].\&\{\ card : \mathsf{GETcard},\ cash : \mathsf{GETcash},\ no : \ ![coupon]\ \}$$

$$\mathsf{f} : \mathsf{BuyerType} \dashv \mathsf{SellerType}'$$

- SYMMETRY obtainable by a simple restriction.

Processes operational sematics

$$Buyer = request_{\mathsf{BuyerType}}(k)P \qquad Seller = accept_{\mathsf{SellerType}}(k)Q$$

$$Buyer \mid Seller \quad \rightarrow \quad (\nu k)(\ \langle k \rangle \mathsf{f} \ \mid \ P \ \mid \ Q\ )$$

All the interactions on k are now *mediated* by f.

Processes operational sematics

$$Buyer = request_{\mathsf{BuyerType}}(k)P \qquad Seller = accept_{\mathsf{SellerType}}(k)Q$$

$$Buyer \mid Seller \quad \rightarrow \quad (\nu k)(\ \langle k \rangle \mathsf{f} \ \mid \ P \ \mid \ Q\ )$$

All the interactions on k are now *mediated* by f.

$\boxplus$-& REDUCTION

$$(\nu k)(\ \langle k \rangle \oplus_{h \in H} l_h.\mathsf{f}_h \ \mid \ k^p \triangleleft [l_j : P_j]_{j \in J} \ \mid \ k^{\overline{p}} \triangleright \{l_i{:}Q_i\}_{i \in I}\ )$$

Processes operational sematics

$$Buyer = request_{\mathsf{BuyerType}}(k)P \qquad Seller = accept_{\mathsf{SellerType}}(k)Q$$

$$Buyer \mid Seller \quad \rightarrow \quad (\nu k)(\ \langle k\rangle \mathsf{f} \ \mid \ P \ \mid \ Q\ )$$

All the interactions on k are now *mediated* by f.

⊞-& REDUCTION

$$(\nu k)(\ \langle k\rangle \oplus_{h\in H} l_h.\mathsf{f}_h \ \mid \ k^{p} \triangleleft [l_j : P_j]_{j\in J} \ \mid \ k^{\overline{p}} \triangleright \{l_i{:}Q_i\}_{i\in I}\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle \mathsf{f}_c \ \mid \ P_c \ \mid \ Q_c\ )$$

$$\text{if} \quad c \in H \cap I \cap J$$

Type system

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta}\ (\textsc{Acc-T}) \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta}\ (\textsc{Req-T})$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta}\ (\text{Acc-T}) \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta}\ (\text{Req-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta}\ (\text{CRes-T})$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta} \text{(Acc-T)} \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta} \text{(Req-T)}$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta} \text{(CRes-T)}$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ *for some* Γ *and* Δ *and* $P \xrightarrow{*} R$*. Then* R *does not contain any error.*

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta} \text{(Acc-T)} \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta} \text{(Req-T)}$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta} \text{(CRes-T)}$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ for some Γ and Δ and $P \xrightarrow{*} R$. Then R does not contain any error.

$$(\nu k)(\ k^+![\mathsf{e}].R \mid k^-![\mathsf{e}'].R' \mid \langle k \rangle \mathsf{f}\ )$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta} \ (\text{Acc-T}) \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta} \ (\text{Req-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta} \ (\text{CRes-T})$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ for some Γ and Δ and $P \xrightarrow{*} R$. Then R does not contain any error.

$$(\nu k)(\ k^+![\mathsf{e}].R \mid k^-![\mathsf{e}'].R' \mid \langle k \rangle \mathsf{f}\ )$$

$$(\nu k)(\ k^+ \lhd l.R \mid k^+ \rhd \{l_j{:}R_j\}_{j \in J} \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta} \text{(ACC-T)} \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta} \text{(REQ-T)}$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta} \text{(CRES-T)}$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ for some Γ and Δ and $P \xrightarrow{*} R$. Then R does not contain any error.

$$(\nu k)(\ k^+![e].R \mid k^-![e'].R' \mid \langle k \rangle \mathsf{f}\ )$$

$$(\nu k)(\ k^+ \triangleleft l.R \mid k^+ \triangleright \{l_j{:}R_j\}_{j \in J} \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

$$(\nu k)(\ k^+![e].R \mid k^-?(x).R' \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash \mathit{accept}_S(k)P \triangleright \Delta} \;(\text{Acc-T}) \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash \mathit{request}_S(k)P \triangleright \Delta} \;(\text{Req-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k\rangle \mathsf{f} \mid P) \triangleright \Delta} \;(\text{CRes-T})$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ for some Γ and Δ and $P \xrightarrow{*} R$. Then R does not contain any error.

$$(\nu k)(\; k^+![e].R \mid k^-![e'].R' \mid \langle k\rangle \mathsf{f}\;)$$

$$(\nu k)(\; k^+ \triangleleft l.R \mid k^+ \triangleright \{l_j{:}R_j\}_{j\in J} \mid \langle k\rangle \Sigma_{i\in I} l_i.\mathsf{f}_i\;)$$

$$(\nu k)(\; k^+![e].R \mid k^-?(x).R' \mid \langle k\rangle \Sigma_{i\in I} l_i.\mathsf{f}_i\;)$$

$$(\nu k)(\; k^-![e].R \mid R' \mid \langle k\rangle \mathbf{1}\;)$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash \mathit{accept}_S(k)P \triangleright \Delta}\ (\textsc{Acc-T}) \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash \mathit{request}_S(k)P \triangleright \Delta}\ (\textsc{Req-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta}\ (\textsc{CRes-T})$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ for some Γ and Δ and $P \xrightarrow{*} R$. Then R does not contain any error.

$$(\nu k)(\ k^+![e].R \mid k^-![e'].R' \mid \langle k \rangle \mathsf{f}\ )$$

$$(\nu k)(\ k^+ \triangleleft l.R \mid k^+ \triangleright \{l_j{:}R_j\}_{j \in J} \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

$$(\nu k)(\ k^+![e].R \mid k^-?(x).R' \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

$$(\nu k)(\ k^-![e].R \mid R' \mid \langle k \rangle \mathbf{1}\ )$$

✗ $$(\nu k)(\ k^+![e].R \mid \langle k \rangle \mathbf{1}\ )$$

Type system

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta} \ (\text{Acc-T}) \qquad \frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta} \ (\text{Req-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} : S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta} \ (\text{CRes-T})$$

Theorem (Error freeness)

Let $\Gamma \Vdash P \triangleright \Delta$ for some Γ and Δ and $P \xrightarrow{*} R$. Then R does not contain any error.

$$(\nu k)(\ k^+![e].R \mid k^-![e'].R' \mid \langle k \rangle \mathsf{f}\ )$$

$$(\nu k)(\ k^+ \triangleleft l.R \mid k^+ \triangleright \{l_j{:}R_j\}_{j\in J} \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

$$(\nu k)(\ k^+![e].R \mid k^-?(x).R' \mid \langle k \rangle \Sigma_{i \in I} l_i.\mathsf{f}_i\ )$$

$$(\nu k)(\ k^-![e].R \mid R' \mid \langle k \rangle \mathbf{1}\ )$$

✗ $(\nu k)(\ k^+![e].R \mid \langle k \rangle \mathbf{1}\ )$

unavoidable! Depends on $\dashv$ asimmetry

$\dashv$ -dependent deadlocks

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes

but

peculiar stuck states (blocking residual server's actions)

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes

but

peculiar stuck states (blocking residual server's actions)

$$P \;=\; \mathit{request}_{\mathsf{A}}(k)k![4].\mathit{request}_{\mathsf{B}}(k')k'![\mathrm{True}].$$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; \mathit{request}_{\mathsf{A}}(k)k![4].\mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}].$$

$\mathsf{A} = \mathbf{!}[\mathit{Nat}] \qquad \mathsf{B} = \mathbf{!}[\mathit{Bool}]$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathrm{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = \mathbf{!}[Nat] \qquad \mathsf{B} = \mathbf{!}[Bool]$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; \mathit{request}_{\mathsf{A}}(k)k![4].\mathit{request}_{\mathsf{B}}(k')k'![\mathrm{True}].$$
$$Q \;=\; \mathit{accept}_{\mathsf{C}}(k)k?(x).k![x].\mathit{accept}_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = \mathbf{!}[\mathit{Nat}]$ $\quad$ $\mathsf{B} = \mathbf{!}[\mathit{Bool}]$ $\quad$ $\mathsf{C} = \mathbf{?}[\mathit{Nat}].\mathbf{!}[\mathit{Nat}]$ $\quad$ $\mathsf{D} = \mathbf{?}[\mathit{Bool}]$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathrm{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = \mathbf{!}[Nat]$ $\mathsf{B} = \mathbf{!}[Bool]$ $\mathsf{C} = \mathbf{?}[Nat].\mathbf{!}[Nat]$ $\mathsf{D} = \mathbf{?}[Bool]$

$$P \mid Q$$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathsf{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = ![Nat]$ $\quad$ $\mathsf{B} = ![Bool]$ $\quad$ $\mathsf{C} = ?[Nat].![Nat]$ $\quad$ $\mathsf{D} = ?[Bool]$

$$P \mid Q$$

$$\downarrow \text{(since } \bullet.\mathbf{1} : \mathsf{A} \dashv \mathsf{C})$$

$$(\nu k)(\ \langle k \rangle \bullet .\mathbf{1} \ \mid\ k^{-}![4].request_{\mathsf{B}}(k')k'![\mathsf{True}].\ \mid\ k^{+}?(x).k^{+}![x].accept_{\mathsf{D}}(k')k'?(y)\ )$$

$\dashv$ -dependent deadlocks

asymmetric compliance ⇒ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathrm{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = \mathbf{!}[Nat]$ $\mathsf{B} = \mathbf{!}[Bool]$ $\mathsf{C} = \mathbf{?}[Nat].\mathbf{!}[Nat]$ $\mathsf{D} = \mathbf{?}[Bool]$

$$P \mid Q$$
$$\downarrow (\text{since } \bullet.\mathbf{1} : \mathsf{A} \dashv \mathsf{C})$$
$$(\nu k)(\ \langle k \rangle \bullet .\mathbf{1} \ \mid\ k^{-}![4].request_{\mathsf{B}}(k')k'![\mathrm{True}].\ \mid\ k^{+}?(x).k^{+}![x].accept_{\mathsf{D}}(k')k'?(y)\ )$$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathsf{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = \mathbf{!}[Nat]$ $\quad \mathsf{B} = \mathbf{!}[Bool]$ $\quad \mathsf{C} = \mathbf{?}[Nat].\mathbf{!}[Nat]$ $\quad \mathsf{D} = \mathbf{?}[Bool]$

$$P \mid Q$$
$$\downarrow (\text{since } \bullet.\mathbb{1} : \mathsf{A} \dashv \mathsf{C})$$
$$(\nu k)(\; \langle k \rangle \bullet .\mathbb{1} \;\mid\; k^{-}![4].request_{\mathsf{B}}(k')k'![\mathsf{True}]. \;\mid\; k^{+}?(x).k^{+}![x].accept_{\mathsf{D}}(k')k'?(y) \;)$$
$$\downarrow$$
$$(\nu k)(\; \langle k \rangle \mathbb{1} \;\mid\; request_{\mathsf{B}}(k')k'![\mathsf{True}]. \;\mid\; k^{+}![4]accept_{\mathsf{D}}(k')k'?(y) \;)$$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathrm{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = ![Nat]$ $\quad$ $\mathsf{B} = ![Bool]$ $\quad$ $\mathsf{C} = ?[Nat].![Nat]$ $\quad$ $\mathsf{D} = ?[Bool]$

$$P \mid Q$$

$$\downarrow (\text{since } \bullet.\mathbf{1} : \mathsf{A} \dashv \mathsf{C})$$

$$(\nu k)(\ \langle k\rangle \bullet .\mathbf{1} \ \mid\ k^{-}![4].request_{\mathsf{B}}(k')k'![\mathrm{True}]. \ \mid\ k^{+}?(x).k^{+}![x].accept_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle \mathbf{1} \ \mid\ request_{\mathsf{B}}(k')k'![\mathrm{True}]. \ \mid\ k^{+}![4].accept_{\mathsf{D}}(k')k'?(y)\ )$$

$\dashv$ -dependent deadlocks

asymmetric compliance $\Rightarrow$ more compatible processes
but
peculiar stuck states (blocking residual server's actions)

$$P \;=\; request_{\mathsf{A}}(k)k![4].request_{\mathsf{B}}(k')k'![\mathsf{True}].$$
$$Q \;=\; accept_{\mathsf{C}}(k)k?(x).k![x].accept_{\mathsf{D}}(k')k'?(y)$$

$\mathsf{A} = \mathbf{!}[Nat]$ $\quad$ $\mathsf{B} = \mathbf{!}[Bool]$ $\quad$ $\mathsf{C} = \mathbf{?}[Nat].\mathbf{!}[Nat]$ $\quad$ $\mathsf{D} = \mathbf{?}[Bool]$

$$P \mid Q$$

$$\downarrow \text{(since } \bullet.\mathbf{1} : \mathsf{A} \dashv \mathsf{C})$$

$$(\nu k)(\ \langle k\rangle \bullet .\mathbf{1} \ \mid\ k^{-}![4].request_{\mathsf{B}}(k')k'![\mathsf{True}].\ \mid\ k^{+}?(x).k^{+}![x].accept_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle\mathbf{1} \ \mid\ request_{\mathsf{B}}(k')k'![\mathsf{True}].\ \mid\ k^{+}![4].accept_{\mathsf{D}}(k')k'?(y)\ )$$

$\not\downarrow$ stuck!
even if $\bullet.\mathbf{1} : \mathsf{B} \dashv \mathsf{D}$

Clean-up (abort) reductions

Clean-up (abort) reductions

$$(\nu k)(\langle k \rangle \mathbb{1} \mid \pi.R' \mid R) \quad \rightarrow \quad (\nu k)(\langle k \rangle \mathbb{1} \mid R' \mid R)$$

$$\text{where} \quad \pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$$

Clean-up (abort) reductions

$$(\nu k)(\langle k \rangle \mathbb{1} \mid \pi.R' \mid R) \quad \rightarrow \quad (\nu k)(\langle k \rangle \mathbb{1} \mid R' \mid R)$$

$$\text{where} \quad \pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$$

$$(\nu k)(\ \langle k \rangle \mathbb{1} \mid request_{\mathrm{B}}(k')k'![\mathrm{True}]. \mid k^+![4].accept_{\mathrm{D}}(k')k'?(y)\ )$$

Clean-up (abort) reductions

$$(\nu k)(\langle k \rangle \mathbf{1} \mid \pi.R' \mid R) \quad \rightarrow \quad (\nu k)(\langle k \rangle \mathbf{1} \mid R' \mid R)$$

where $\pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$

$$(\nu k)(\ \langle k \rangle \mathbf{1} \mid \mathit{request}_{\mathrm{B}}(k')k'![\mathrm{True}]. \mid k^+![4].\mathit{accept}_{\mathrm{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k \rangle \mathbf{1} \mid \mathit{request}_{\mathrm{B}}(k')k'![\mathrm{True}]. \mid \mathit{accept}_{\mathrm{D}}(k')k'?(y)\ )$$

Clean-up (abort) reductions

$$(\nu k)(\langle k \rangle \mathbf{1} \mid \pi.R' \mid R) \quad \rightarrow \quad (\nu k)(\langle k \rangle \mathbf{1} \mid R' \mid R)$$

$$\text{where} \quad \pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$$

$$(\nu k)(\ \langle k \rangle \mathbf{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid k^+![4].\mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k \rangle \mathbf{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid \mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow \qquad (\text{since } \bullet.\mathbf{1} : \mathsf{B} \dashv \mathsf{D})$$

$$(\nu k)(\ \langle k \rangle \mathbf{1} \mid (\nu k')(\ \langle k' \rangle \bullet .\mathbf{1} \mid k'![\mathsf{True}]. \mid k'?(y)\ )\ )$$

Clean-up (abort) reductions

$$(\nu k)(\langle k\rangle \mathbb{1} \mid \pi.R' \mid R) \quad \to \quad (\nu k)(\langle k\rangle \mathbb{1} \mid R' \mid R)$$

$$\text{where} \quad \pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid k^+![4].\mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid \mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow \qquad (\text{since } \bullet.\mathbb{1} : \mathsf{B} \dashv \mathsf{D})$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid (\nu k')(\ \langle k'\rangle \bullet .\mathbb{1} \mid k'![\mathsf{True}]. \mid k'?(y)\ )\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid (\nu k')(\ \langle k'\rangle \mathbb{1} \mid \mathbf{0} \mid \mathbf{0}\ )\ ).$$

Clean-up (abort) reductions

$$(\nu k)(\langle k\rangle\mathbb{1} \mid \pi.R' \mid R) \quad\rightarrow\quad (\nu k)(\langle k\rangle\mathbb{1} \mid R' \mid R)$$

$$\text{where} \quad \pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$$

$$(\nu k)(\ \langle k\rangle\mathbb{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid k^+![4].\mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle\mathbb{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid \mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow \qquad (\text{since } \bullet.\mathbb{1} : \mathsf{B} \dashv \mathsf{D})$$

$$(\nu k)(\ \langle k\rangle\mathbb{1} \mid (\nu k')(\ \langle k'\rangle \bullet .\mathbb{1} \mid k'![\mathsf{True}]. \mid k'?(y)\ )\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle\mathbb{1} \mid (\nu k')(\ \langle k'\rangle\mathbb{1} \mid \mathbf{0} \mid \mathbf{0}\ )\ ).$$

Subject reduction and Error Freeness still hold

Clean-up (abort) reductions

$$(\nu k)(\langle k\rangle \mathbb{1} \mid \pi.R' \mid R) \quad \rightarrow \quad (\nu k)(\langle k\rangle \mathbb{1} \mid R' \mid R)$$

$$\text{where} \quad \pi \in \{k^+![e], k^+?(x), k^+ \triangleleft l.\}$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid k^+![4].\mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid \mathit{request}_{\mathsf{B}}(k')k'![\mathsf{True}]. \mid \mathit{accept}_{\mathsf{D}}(k')k'?(y)\ )$$

$$\downarrow \qquad (\text{since } \bullet.\mathbb{1} : \mathsf{B} \dashv \mathsf{D})$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid (\nu k')(\ \langle k'\rangle \bullet .\mathbb{1} \mid k'![\mathsf{True}]. \mid k'?(y)\ )\ )$$

$$\downarrow$$

$$(\nu k)(\ \langle k\rangle \mathbb{1} \mid (\nu k')(\ \langle k'\rangle \mathbb{1} \mid \mathbf{0} \mid \mathbf{0}\ )\ ).$$

Subject reduction and Error Freeness still hold

obviously no $\dashv$-dependent deadlocks.

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \begin{Bmatrix} \mathit{belt} : ?[\mathsf{Int}]. \bigoplus\{ \mathit{card} : \mathsf{PAYcard}, \mathit{cash} : \mathsf{PAYcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ?[\mathsf{Int}]. \bigoplus\{ \mathit{card} : \mathsf{PAYcard}, \mathit{cash} : \mathsf{PAYcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ?[\mathsf{I}\mathit{nt}]. \bigoplus\{ \mathit{card} : \mathsf{PAYcard}, \mathit{cash} : \mathsf{PAYcash}, \mathit{no} : \mathbf{1} \} \end{Bmatrix}$$

$$\begin{aligned} \text{SellerType2} = \&\{ \; & \mathit{belt} : ![\mathsf{Int}].\&\{ \mathit{cash} : \mathsf{GETcash}, \mathit{no} : \mathbf{1} \} \\ & \mathit{bag} : ![\mathsf{Int}].\&\{ \mathit{card} : \mathsf{GETcard}, \mathit{cash} : \mathsf{GETcash}, \mathit{no} : \mathbf{1} \} \\ & \mathit{hat} : ![\mathsf{I}\mathit{nt}].\&\{ \mathit{card} : \mathsf{GETcard}, \mathit{cash} : \mathsf{GETcash}, \mathit{no} : \mathbf{1} \} \; \} \end{aligned}$$

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \left\{ \begin{array}{l} \mathit{belt} : ?[\text{Int}]. \bigoplus\{ \mathit{card} : \text{PAYcard},\ \mathit{cash} : \text{PAYcash},\ \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ?[\text{Int}]. \bigoplus\{ \mathit{card} : \text{PAYcard},\ \mathit{cash} : \text{PAYcash},\ \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ?[\text{I}\mathit{nt}]. \bigoplus\{ \mathit{card} : \text{PAYcard},\ \mathit{cash} : \text{PAYcash},\ \mathit{no} : \mathbf{1} \} \end{array} \right\}$$

$$\text{SellerType2} = \& \left\{ \begin{array}{l} \mathit{belt} : ![\text{Int}].\&\{ \mathit{cash} \text{: GETcash},\ \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ![\text{Int}].\&\{ \mathit{card} : \text{GETcard},\ \mathit{cash} : \text{GETcash},\ \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ![\text{I}\mathit{nt}].\&\{ \mathit{card} : \text{GETcard},\ \mathit{cash} : \text{GETcash}, \mathit{no} : \mathbf{1} \} \end{array} \right\}$$

$$\text{BuyerType2} \dashv \text{SellerType2}$$

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \left\{ \begin{array}{l} \mathit{belt} : ?[\mathsf{Int}]. \bigoplus\{ \ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1} \ \} \\ \mathit{bag} : ?[\mathsf{Int}]. \bigoplus\{ \ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1} \ \} \\ \mathit{hat} : ?[Int]. \bigoplus\{ \ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1} \ \} \end{array} \right\}$$

$$\text{SellerType2} = \&\left\{ \begin{array}{l} \mathit{belt} : ![\mathsf{Int}].\&\{ \ \mathit{cash} : \mathsf{GETcash},\ \mathit{no} : \mathbf{1} \ \} \\ \mathit{bag} : ![\mathsf{Int}].\&\{ \ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash},\ \mathit{no} : \mathbf{1} \ \} \\ \mathit{hat} : ![Int].\&\{ \ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash}, \mathit{no} : \mathbf{1} \ \} \end{array} \right\}$$

$$\mathsf{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

$$\begin{array}{rl} \mathbf{f} = & \textsc{bag}. \bullet .(\textsc{card}.\mathrm{g}_1 + \textsc{cash}.\mathrm{g}_2 + \textsc{no}) \\ & \oplus \\ & \textsc{hat}. \bullet .(\textsc{card}.\mathrm{g}_1 + \textsc{cash}.\mathrm{g}_2 + \textsc{no}) \end{array}$$

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \left\{ \begin{array}{l} \mathit{belt} :?[\mathsf{Int}].\bigoplus\{\ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{bag} :?[\mathsf{Int}].\bigoplus\{\ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{hat} :?[\mathsf{Int}].\bigoplus\{\ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \end{array} \right\}$$

$$\text{SellerType2} = \&\left\{ \begin{array}{l} \mathit{belt} :![\mathsf{Int}].\&\{\ \mathit{cash}: \mathsf{GETcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{bag} :![\mathsf{Int}].\&\{\ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{hat} :![\mathsf{Int}].\&\{\ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash}, \mathit{no} : \mathbf{1}\ \} \end{array} \right\}$$

$$\mathsf{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

orchestration-non-determinism elimination:

$$\begin{array}{rl} \mathbf{f} = & \textsc{bag}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no}) \\ & \oplus \\ & \textsc{hat}.\bullet.(\textsc{card}.\mathsf{g}_1 + \textsc{cash}.\mathsf{g}_2 + \textsc{no}) \end{array}$$

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \left\{ \begin{array}{l} \mathit{belt} : ?[\text{Int}]. \bigoplus\{ \mathit{card} : \text{PAYcard}, \mathit{cash} : \text{PAYcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ?[\text{Int}]. \bigoplus\{ \mathit{card} : \text{PAYcard}, \mathit{cash} : \text{PAYcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ?[Int]. \bigoplus\{ \mathit{card} : \text{PAYcard}, \mathit{cash} : \text{PAYcash}, \mathit{no} : \mathbf{1} \} \end{array} \right\}$$

$$\text{SellerType2} = \&\left\{ \begin{array}{l} \mathit{belt} : ![\text{Int}].\&\{ \mathit{cash}: \text{GETcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ![\text{Int}].\&\{ \mathit{card} : \text{GETcard}, \mathit{cash} : \text{GETcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ![Int].\&\{ \mathit{card} : \text{GETcard}, \mathit{cash} : \text{GETcash}, \mathit{no} : \mathbf{1} \} \end{array} \right\}$$

$$\text{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

orchestration-non-determinism elimination:

- **type level** *(types defining priorities)*

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \left\langle\!\!\left\langle \begin{array}{l} \mathit{belt} :?[\text{Int}].\bigoplus\{\ \mathit{card} : \text{PAYcard},\ \mathit{cash} : \text{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{bag} :?[\text{Int}].\bigoplus\{\ \mathit{card} : \text{PAYcard},\ \mathit{cash} : \text{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{hat} :?[\mathit{Int}].\bigoplus\{\ \mathit{card} : \text{PAYcard},\ \mathit{cash} : \text{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \end{array} \right\rangle\!\!\right\rangle$$

$$\text{SellerType2} = \&\left\{ \begin{array}{l} \mathit{belt} :![\text{Int}].\&\{\ \mathit{cash}: \text{GETcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{bag} :![\text{Int}].\&\{\ \mathit{card} : \text{GETcard},\ \mathit{cash} : \text{GETcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{hat} :![\mathit{Int}].\&\{\ \mathit{card} : \text{GETcard},\ \mathit{cash} : \text{GETcash}, \mathit{no} : \mathbf{1}\ \} \end{array} \right\}$$

$$\text{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

orchestration-non-determinism elimination:

- **type level** *(types defining priorities)*

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \langle\!\langle \begin{array}{l} \textit{belt} :?[\text{Int}].\bigoplus\{\ \textit{card} : \text{PAYcard},\ \textit{cash} : \text{PAYcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{bag} :?[\text{Int}].\bigoplus\{\ \textit{card} : \text{PAYcard},\ \textit{cash} : \text{PAYcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{hat} :?[Int].\bigoplus\{\ \textit{card} : \text{PAYcard},\ \textit{cash} : \text{PAYcash},\ \textit{no} : \mathbf{1}\ \} \end{array} \rangle\!\rangle$$

$$\text{SellerType2} = \&\{\ \begin{array}{l} \textit{belt} :![\text{Int}].\&\{\ \textit{cash}: \text{GETcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{bag} :![\text{Int}].\&\{\ \textit{card} : \text{GETcard},\ \textit{cash} : \text{GETcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{hat} :![Int].\&\{\ \textit{card} : \text{GETcard},\ \textit{cash} : \text{GETcash}, \textit{no} : \mathbf{1}\ \}\ \} \end{array}$$

$$\text{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

orchestration-non-determinism elimination:

- **type level** *(types defining priorities)*

$$\mathbf{f} = \text{BAG}.\bullet.(\text{CARD}.\text{g}_1 + \text{CASH}.\text{g}_2 + \text{NO})$$

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \left\{ \begin{array}{l} \mathit{belt} : ?[\text{Int}]. \bigoplus\{ \mathit{card} : \text{PAYcard}, \mathit{cash} : \text{PAYcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ?[\text{Int}]. \bigoplus\{ \mathit{card} : \text{PAYcard}, \mathit{cash} : \text{PAYcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ?[Int]. \bigoplus\{ \mathit{card} : \text{PAYcard}, \mathit{cash} : \text{PAYcash}, \mathit{no} : \mathbf{1} \} \end{array} \right\}$$

$$\text{SellerType2} = \& \left\{ \begin{array}{l} \mathit{belt} : ![\text{Int}].\&\{ \mathit{cash}: \text{GETcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{bag} : ![\text{Int}].\&\{ \mathit{card} : \text{GETcard}, \mathit{cash} : \text{GETcash}, \mathit{no} : \mathbf{1} \} \\ \mathit{hat} : ![Int].\&\{ \mathit{card} : \text{GETcard}, \mathit{cash} : \text{GETcash}, \mathit{no} : \mathbf{1} \} \end{array} \right\}$$

$$\text{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

orchestration-non-determinism elimination:

- **process level**

Non-determinism in orchestration

$$\mathsf{BuyerType2} = \boxplus \left\{ \begin{array}{l} \mathit{belt} :?[\mathsf{Int}]. \bigoplus\{\ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{bag} :?[\mathsf{Int}]. \bigoplus\{\ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{hat} :?[\mathit{Int}]. \bigoplus\{\ \mathit{card} : \mathsf{PAYcard},\ \mathit{cash} : \mathsf{PAYcash},\ \mathit{no} : \mathbf{1}\ \} \end{array} \right\}$$

$$\mathsf{SellerType2} = \&\left\{ \begin{array}{l} \mathit{belt} :![\mathsf{Int}].\&\{\ \mathit{cash}: \mathsf{GETcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{bag} :![\mathsf{Int}].\&\{\ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash},\ \mathit{no} : \mathbf{1}\ \} \\ \mathit{hat} :![\mathit{Int}].\&\{\ \mathit{card} : \mathsf{GETcard},\ \mathit{cash} : \mathsf{GETcash}, \mathit{no} : \mathbf{1}\ \} \end{array} \right\}$$

$$\mathsf{f} : \mathsf{BuyerType2} \dashv \mathsf{SellerType2}$$

orchestration-non-determinism elimination:

- **process level**

$$\mathbf{f} = \begin{array}{l} \textsc{bag}. \bullet .(\textsc{card}.g_1 + \textsc{cash}.g_2 + \textsc{no}) \\ \oplus \\ \textsc{hat}. \bullet .(\textsc{card}.g_1 + \textsc{cash}.g_2 + \textsc{no}) \end{array}$$

Non-determinism in orchestration

$$\text{BuyerType2} = \boxplus \begin{Bmatrix} \textit{belt} :?[\text{Int}].\bigoplus\{\ \textit{card} : \text{PAYcard},\ \textit{cash} : \text{PAYcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{bag} :?[\text{Int}].\bigoplus\{\ \textit{card} : \text{PAYcard},\ \textit{cash} : \text{PAYcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{hat} :?[\text{Int}].\bigoplus\{\ \textit{card} : \text{PAYcard},\ \textit{cash} : \text{PAYcash},\ \textit{no} : \mathbf{1}\ \} \end{Bmatrix}$$

$$\text{SellerType2} = \&\{\ \textit{belt} :![\text{Int}].\&\{\ \textit{cash}: \text{GETcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{bag} :![\text{Int}].\&\{\ \textit{card} : \text{GETcard},\ \textit{cash} : \text{GETcash},\ \textit{no} : \mathbf{1}\ \} \\ \textit{hat} :![\text{Int}].\&\{\ \textit{card} : \text{GETcard},\ \textit{cash} : \text{GETcash}, \textit{no} : \mathbf{1}\ \}\ \}$$

$$\text{f} : \text{BuyerType2} \dashv \text{SellerType2}$$

orchestration-non-determinism elimination:

- **process level**

$$\mathbf{f} = \textsc{bag}.\bullet.(\textsc{card}.g_1 + \textsc{cash}.g_2 + \textsc{no}) \\ \oplus \\ \textsc{hat}.\bullet.(\textsc{card}.g_1 + \textsc{cash}.g_2 + \textsc{no})$$

$$\textit{Buyer2} = \ldots k \triangleleft \langle\!\langle \textsc{belt} : P_1, \textsc{hat} : P_2, \textsc{bag} : P_3 \rangle\!\rangle \ldots$$

$$request_S(k)P \ \mid\ accept_{S'}(k)Q \ \rightarrow\ (\nu k)(\langle k\rangle \mathsf{f} \ \mid\ P\{k^-/k\} \ \mid\ Q\{k^+/k\}) \quad \text{if } \mathsf{f} = SynthUD(S, S') \neq \textbf{fail}$$

$$(\nu k)(\langle k\rangle \bigoplus_{i\in I} \mathsf{f}_i \ \mid\ k \triangleleft \langle\!\langle l_1{:}P_1, \ldots, l_n{:}P_n\rangle\!\rangle \ \mid\ k \triangleright \{l_j{:}Q_j\}_{j\in J}) \ \rightarrow\ (\nu k)(\langle k\rangle \mathsf{f}_c \mid P_m \mid Q_c$$

$$\text{if } c \in I \cap J,\ 1 \leq m \leq n \text{ and } l_c = l_m,\ \text{where } m = \min\{h \mid l_m \in \{l_i\}_{i\in I}\}$$

Future work

Future work

- Recursion (easy, just some complexity to cope with)

Future work

- Recursion (easy, just some complexity to cope with)
- Subtyping (relation easy to define, but functors on orchestrators needed)

Future work

- Recursion (easy, just some complexity to cope with)
- Subtyping (relation easy to define, but functors on orchestrators needed)
- Multiparty

Future work

- Recursion (easy, just some complexity to cope with)
- Subtyping (relation easy to define, but functors on orchestrators needed)
- Multiparty

Thanks

$$\frac{\Delta \text{ completed}}{\Gamma \Vdash \mathbf{0} \triangleright \Delta} \text{(INACT-T)}$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \quad \Gamma \Vdash Q \triangleright \Delta'}{\Gamma \Vdash P \mid Q \triangleright \Delta \cdot \Delta'} \text{(CONC-T)}$$

$$\frac{\Gamma \Vdash P\{k^+/k\} \triangleright \Delta \cdot k^+ : S}{\Gamma \Vdash accept_S(k)P \triangleright \Delta} \text{(ACC-T)}$$

$$\frac{\Gamma \Vdash P\{k^-/k\} \triangleright \Delta \cdot k^- : S}{\Gamma \Vdash request_S(k)P \triangleright \Delta} \text{(REQ-T)}$$

$$\frac{\Gamma, x : G \Vdash P \triangleright \Delta \cdot k^p : S}{\Gamma \Vdash k^p?(x).P \triangleright \Delta \cdot k^p : ?[G].S} \text{(REC-T)}$$

$$\frac{\Gamma \vdash e : G \quad \Gamma \Vdash P \triangleright \Delta \cdot k^p : S}{\Gamma \Vdash k^p![e]..P \triangleright \Delta \cdot k^p : ![G].S} \text{(SEND-T)}$$

Figure: The type system 1/2.

$$\frac{\Gamma \Vdash P\{k'^q/k'\} \triangleright \Delta \cdot k^p : S_2 \cdot k'^q : S_1}{\Gamma \Vdash \mathsf{catch}\ k^p(k').P \triangleright \Delta \cdot k^p : ?[S_1^q].S_2} \ (\text{CAT-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^p : S_2}{\Gamma \Vdash \mathsf{throw}\ k^p[k'^q].P \triangleright \Delta \cdot k^p : ![S_1^q].S_2 \cdot k'^q : S_1} \ (\text{THR-T})$$

$$\frac{\forall i \in I \supseteq J \quad \Gamma \Vdash P_i \triangleright \Delta \cdot k^p : S_i}{\Gamma \Vdash k^p \triangleright \{l_i : P_i\}_{i \in I} \triangleright \Delta \cdot k^p : \&\{l_j{:}S_j\}_{j \in J}} \ (\text{BR-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^p : S_j \quad j \in I}{\Gamma \Vdash k^p \triangleleft l_j..P \triangleright \Delta \cdot k^p : \bigoplus\{l_i{:}S_i\}_{i \in I}} \ (\text{SEL-T})$$

$$\frac{\forall i \in I \quad \Gamma \Vdash P_i \triangleright \Delta \cdot k^p : S_i}{\Gamma \Vdash k^p \triangleleft [l_i : P_i]_{i \in I} \triangleright \Delta \cdot k^p : \boxplus\{l_i{:}S_i\}_{i \in I}} \ (\text{SSEL-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \cdot k^- : S_1 \cdot k^+ : S_2 \qquad \mathsf{f} :\ S_1 \dashv S_2}{\Gamma \Vdash (\nu k)(\langle k \rangle \mathsf{f} \mid P) \triangleright \Delta} \ (\text{CRES-T})$$

$$\frac{\Gamma \Vdash P \triangleright \Delta \qquad k^+, k^- \notin (\Delta)}{\Gamma \Vdash (\nu k)P \triangleright \Delta} \ (\text{CRES'-T})$$

Figure: The type system 2/2.

Definition (Orchestrators)

$$
\begin{array}{rclll}
\mathsf{f}, \mathsf{g} & ::= & \mathbf{1} & & \text{idle} \\
& | & \bullet.\mathsf{f} & & \text{I/O prefix} \\
& | & l.\mathsf{f} & & \text{selection prefix} \\
& | & l.\mathsf{f} + l'.\mathsf{g} & (l \neq l') & \text{external choice} \\
& | & l.\mathsf{f} \oplus l'.\mathsf{g} & (l \neq l') & \text{internal choice}
\end{array}
$$