Introduction to (Finite) Binary Session Types

Hernán Melgratti

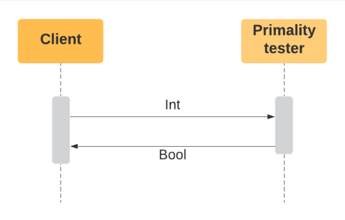
ICC University of Buenos Aires-Conicet

11 February 2020 @ Pisa

Informally

- A session type defines a communication protocol
- In the binary case, it describes the messages exchanged between two parties

First example

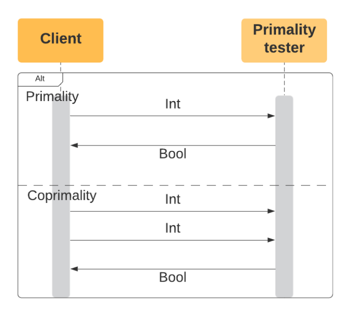


- We rely on a textual description; the flow is described from the point of view of one of the participants
- $\texttt{Tester} = \texttt{?int.!bool.end}$

  $?t$: a receive of a value of type t
  $\_.\_$: followed by
  $!t$: a send of a value of type t
  $\texttt{end}$: a terminated session
- $\texttt{Client} = \texttt{!int.?bool.end}$
- $\texttt{Tester}$ and $\texttt{Client}$ behave dually

Informally

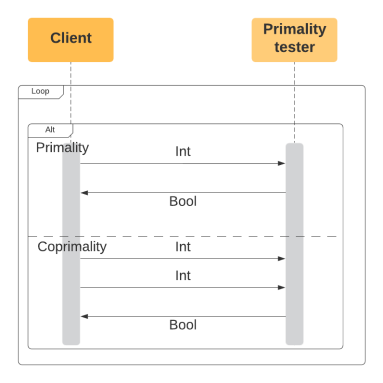
Choices



- $\mathtt{Tester} = \&[\mathtt{Pr} : ?\mathtt{int}.!\mathtt{bool}.\mathtt{end}, \mathtt{Co} : ?\mathtt{int}.?\mathtt{int}.!\mathtt{bool}.\mathtt{end}]$
  - $\&[\mathtt{l}_i : T_i]_{i \in I}$: **Offering** several alternatives, each of them identified by the *label* $\mathtt{l}_i$
- $\mathtt{Client} = \oplus[\mathtt{Pr} : !\mathtt{int}.?\mathtt{bool}.\mathtt{end}, \mathtt{Co} : !\mathtt{int}.!\mathtt{int}.?\mathtt{bool}.\mathtt{end}]$
  - $\oplus[\mathtt{l}_i : T_i]_{i \in I}$: **Selecting** one of the alternatives identified by the *labels* $\mathtt{l}_i$
- `Tester` and `Client` behave dually

Informally

Infinite interactions



- $\mathtt{Tester} = \&[\mathtt{Pr} : ?\mathtt{int}.!\mathtt{bool}.\mathtt{Tester},$
  $\mathtt{Co} : ?\mathtt{int}.?\mathtt{int}.!\mathtt{bool}.\mathtt{Tester}]$
- $\mathtt{Client} = \oplus[\mathtt{Pr} : !\mathtt{int}.?\mathtt{bool}.\mathtt{Client},$
  $\mathtt{Co} : !\mathtt{int}.!\mathtt{int}.?\mathtt{bool}.\mathtt{Client}]$

Modelling a function

f : int $\to$ bool

$\mathtt{f} = \mathtt{?int.!bool.end}$

Invocation

$\mathtt{inv} = \mathtt{!int.?bool.end}$

Modelling an object (Typestate)

File

$$\texttt{File} = ?\texttt{mode}.\texttt{Opened}$$

$$\texttt{Opened} = \&[\mathit{read} : \oplus[\mathit{eof} : \texttt{Opened}, \mathit{val} : !\texttt{string}.\texttt{Opened}], \mathit{close} : \texttt{end}]$$

Client

$$\texttt{Client} = !\texttt{mode}.\texttt{Reading}$$

$$\texttt{Reading} = \oplus[\mathit{read} : \&[\mathit{eof} : \texttt{Reading}, \mathit{val} : !\texttt{string}.\texttt{Reading}], \mathit{close} : \texttt{end}]$$

Syntax of Types

Session Types

$S, T ::=$		$\mathtt{end}$	terminated session
	$\mid$	$?t.S$	receive (input)
	$\mid$	$!t.S$	send (output)
	$\mid$	$\&[\mathtt{l}_i : T_i]_{i \in I}$	branch
	$\mid$	$\oplus[\mathtt{l}_i : T_i]_{i \in I}$	select
	$\mid$	$\mu X.S$	recursive session type
	$\mid$	X	session type variable
$s, t ::=$		S	A session type
	$\mid$	$\mathtt{int}, \mathtt{bool}$	basic types
	$\mid$	$\ldots$	other types
$\mathcal{L} \;=$		$\{\mathtt{l}, \mathtt{l}_1, \ldots\}$	Set of labels

Remark

- The grammar allows terms like $?S.T$
- For instance, $?(?\mathtt{int}.\mathtt{end}).!\mathtt{bool}.\mathtt{end}$ vs $?\mathtt{int}.!\mathtt{bool}.\mathtt{end}$

Examples

$f : \mathtt{int} \to \mathtt{bool}$

$$\mathtt{f} = ?\mathtt{int}.!\mathtt{bool}.\mathtt{end}$$
$$\mathtt{g} = ?\mathtt{f}.!\mathtt{bool}.\mathtt{end}$$

It resembles

$$g : (\mathtt{int} \to \mathtt{bool}) \to \mathtt{bool}$$

but it is not the same (more to come)

File

$$\mathtt{File} = ?\mathtt{mode}.\mathtt{Opened}$$

$$\mathtt{Opened} = \&[\mathit{read} : \oplus[\mathit{eof} : \mathtt{Opened}, \mathit{val} : !\mathtt{string}.\mathtt{Opened}], \mathit{close} : \mathtt{end}]$$

Function that processes a file

$$\mathtt{Client}_1 = !\mathtt{File}.?\mathtt{int}.\mathtt{end}$$
$$\mathtt{Client}_2 = !\mathtt{Opened}.?\mathtt{int}.\mathtt{end}$$

Duality

$\overline{S}$ is the dual of S

$$
\begin{aligned}
\overline{\mathsf{end}} &= \mathsf{end} \\
\overline{?t.S} &= !t.\overline{S} \\
\overline{!t.S} &= ?t.\overline{S} \\
\overline{\&[\mathtt{l}_i : T_i]_{i \in I}} &= \oplus[\mathtt{l}_i : \overline{T_i}]_{i \in I} \\
\overline{\oplus[\mathtt{l}_i : T_i]_{i \in I}} &= \&[\mathtt{l}_i : \overline{T_i}]_{i \in I}
\end{aligned}
$$

Typing

Goal

Determine whether a program implements a protocol (a session type)

1. Fix a language for writing programs
2. Define a relation between programs and session types that states that a program behaves as prescribed by the types

We choose [1]

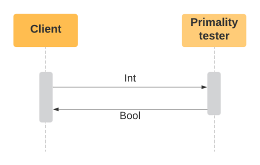
1. A language with message-passing communication based on synchronous channels
2. Session types are associated with channels

[1] Simon J. Gay, Malcolm Hole: Subtyping for session types in the pi calculus. Acta Inf. (2005)

Programs

- Roughly, each participant is implemented by a process (i.e., a thread)
- Processes communicate through *session channels*
- A session channel x has two endpoints x^+ and x^-
- A process sends and receives messages on a session endpoint

Tester



$$\mathtt{Tester} = ?\mathtt{int}.!\mathtt{bool}.\mathtt{end}$$

- We give an implementation over the session endpoints x^+ (for the server) and x^- (for the client)

  $P_{\mathtt{server}} = x^+?(y{:}\mathtt{int}).x^+!\mathtt{true}.0$ *(faulty)*

  $P_{\mathtt{client}} = x^-!1.x^-?(z{:}\mathtt{bool}).Q$
- The system is the parallel composition of the two processes

$$(\nu x{:}\mathtt{Tester})(P_{\mathtt{server}} \mid P_{\mathtt{client}})$$

Syntax of Processes

Polarities

$p ::= + \mid - \mid \epsilon$		Optional polarities

Values (more in general expressions)

$v, w ::=$	$x^p, y^q, \ldots$	(polarised) variables $\mathcal{X} = \{x, y, \ldots\}$
$\mid$	$()$	unit vaue
$\mid$	$\mathtt{true}, \mathtt{false}$	boolean values
$\mid$	$\ldots$	expressions

Processes

$P, Q ::=$	0	terminated process
$\mid$	$x^p\,?(y{:}\mathtt{t}).P$	input
$\mid$	$x^p\,!\,v\,.P$	output
$\mid$	$x^p \triangleright [\mathtt{l}_i : P_i]_{i \in I}$	branch
$\mid$	$x^p \triangleleft \mathtt{l}.P$	select
$\mid$	$P \mid Q$	parallel composition
$\mid$	$(\nu x{:}S)P$	channel creation
$\mid$	$!\,P$	replication

Syntax of Types

Session Types

$S, T ::=$		$\mathtt{end}$	terminated session
	$\mid$	$?t.S$	receive (input)
	$\mid$	$!t.S$	send (output)
	$\mid$	$\&[\mathtt{l}_i : T_i]_{i \in I}$	branch
	$\mid$	$\oplus[\mathtt{l}_i : T_i]_{i \in I}$	select
	$\mid$	$\mu X.S$	recursive session type
	$\mid$	X	session type variable
$s, t ::=$		S	A session type
	$\mid$	$\mathtt{int}, \mathtt{bool}$	basic types
	$\mid$	$\ldots$	other types

Notation

- for a polarity p, we write $\overline{p}$ for the complementary endpoint

$$\overline{+} = - \qquad\qquad \overline{-} = + \qquad\qquad \overline{\epsilon} = \epsilon$$

- we identify x^{ϵ} with x

Typing

Goal

Determine whether a program implements a protocol (a session type)

1. Fix a language for writing programs
2. Define a relation between programs and session types that states that a program behaves as prescribed by the types

Operational semantics

Given in terms of a *Labelled Transition System* (LTS) $(P, \longrightarrow)$ where

- $\longrightarrow \subseteq P \times (\mathcal{X} \cup \{\tau\}) \times (\mathcal{L} \cup \{-\}) \times P$

- $(P, \alpha, \mathtt{l}, Q) \in \longrightarrow$
  - means P evolves to Q after communicating the choice $\mathtt{l}$ on the session α
  - is abbreviated as $P \xrightarrow{\alpha, \mathtt{l}} Q$
- τ stands for a hidden session
- $-$ for no choice

Operational semantics

$$x^p \,!\, v \,.\, P \mid x^{\overline{p}} ?(y{:}\mathtt{t}) \,.\, Q \xrightarrow{x,-} P \mid Q\{v/y\} \text{ [R-Comm]}$$

Substitution

$$\begin{aligned}
x\{v/x\} &= v \\
x^p\{v/y\} &= x^p && \text{if } x \neq y \\
\\
0\{v/y\} &= 0 \\
(P|Q)\{v/y\} &= P\{v/y\}|Q\{v/y\} \\
(x^p ?(z{:}\mathtt{t}).P)\{v/y\} &= x^p\{v/y\} ?(z{:}\mathtt{t}).P\{v/y\} && \text{if } z \notin \mathrm{fn}(v) \cup \{y\}
\end{aligned}$$

Free names

fn

$$
\begin{aligned}
\mathsf{fn}(\mathtt{true}) = \mathsf{fn}(\mathtt{false}) \;&=\; \mathsf{fn}(()) \;=\; \emptyset \\
\mathsf{fn}(x^p) &= \{x^p\} \\
\\
\mathsf{fn}(0) &= \emptyset \\
\mathsf{fn}(P|Q) &= \mathsf{fn}(Q) \cup \mathsf{fn}(P) \\
\mathsf{fn}(x^p\,\mathtt{?}(y{:}\mathtt{t})\,\mathtt{.}\,P) &= \{x^p\} \cup (\mathsf{fn}(P) \setminus \{y\}) \\
\mathsf{fn}(x^p\,\mathtt{!}\,v\,\mathtt{.}\,P) &= \{x^p\} \cup \mathsf{fn}(v) \cup \mathsf{fn}(P) \\
\mathsf{fn}(x^p \triangleright [\mathtt{l}_i : P_i]_{i \in I}) &= \{x^p\} \cup (\bigcup_i \mathsf{fn}(P_i)) \\
\mathsf{fn}(x^p \triangleleft \mathtt{l}\,\mathtt{.}\,P) &= \{x^p\} \cup \mathsf{fn}(P) \\
\mathsf{fn}((\nu x{:}S)P) &= \mathsf{fn}(P) \setminus \{x, x^+, x^-\}
\end{aligned}
$$

Operational semantics

$$x^p\,!\,v\,.\,P \mid x^{\overline{p}}?(y{:}\mathtt{t})\,.\,Q \xrightarrow{x,-} P \mid Q\{v/y\} \quad \text{[R-Comm]}$$

Substitution

$$\begin{aligned}
x\{v/x\} &= v & \\
x^p\{v/y\} &= x^p & \text{if } x \neq y \\
\\
0\{v/y\} &= 0 & \\
(P|Q)\{v/y\} &= P\{v/y\}|Q\{v/y\} & \\
(x^p?(z{:}\mathtt{t}).P)\{v/y\} &= x^p\{v/y\}?(z{:}\mathtt{t}).P\{v/y\} & \text{if } z \notin \mathrm{fn}(v) \cup \{y\} \\
(x^p\,!\,w\,.\,P)\{v/y\} &= x^p\{v/y\}\,!\,w\{v/y\}\,.\,P\{v/y\} & \\
(x^p \triangleright [\mathtt{l}_i : P_i]_{i\in I})\{v/y\} &= x^p\{v/y\} \triangleright [\mathtt{l}_i : P_i\{v/y\}]_{i\in I} & \\
(x^p \triangleleft \mathtt{l}.P)\{v/y\} &= x^p\{v/y\} \triangleleft \mathtt{l}.P\{v/y\} & \\
((\nu x{:}S)P)\{v/y\} &= (\nu x{:}S)P\{v/y\} & \text{if } x \notin \mathrm{fn}(v) \cup \{y\}
\end{aligned}$$

Operational semantics

$$x^p \,!\, v \,.\, P \mid x^{\overline{p}}?(y{:}\mathtt{t}) \,.\, Q \xrightarrow{x,-} P \mid Q\{v/y\} \quad \text{[R-Comm]}$$

$$x^p \triangleleft \mathtt{l}_i \,.\, P \mid x^{\overline{p}} \triangleright [\mathtt{l}_j : Q_j]_{j \in I} \xrightarrow{x,\mathtt{l}_i} P \mid Q_i$$

Operational semantics

$$x^p\,!\,v\,.\,P \mid x^{\overline{p}}\,?(y{:}\mathtt{t})\,.\,Q \;\xrightarrow{x,-}\; P \mid Q\{v/y\} \quad \text{[R-Comm]}$$

$$\frac{i \in I}{x^p \triangleleft \mathtt{l}_i\,.\,P \mid x^{\overline{p}} \triangleright [\mathtt{l}_j : Q_j]_{j\in I} \;\xrightarrow{x,\mathtt{l}_i}\; P \mid Q_i} \quad \text{[R-Select]}$$

Operational semantics

$$x^p\,!\,v\,.\,P \mid x^{\overline{p}}?(y{:}\mathtt{t})\,.\,Q \xrightarrow{x,-} P \mid Q\{v/y\} \text{ [R-Comm]}$$

$$\frac{p \in \{+,-\} \qquad i \in I}{x^p \triangleleft \mathtt{l}_i\,.\,P \mid x^{\overline{p}} \triangleright [\mathtt{l}_j : Q_j]_{j\in I} \xrightarrow{x,\mathtt{l}_i} P \mid Q_i} \text{ [R-Select]}$$

$$\frac{P \xrightarrow{x,\mathtt{l}} P' \qquad S \xrightarrow{\mathtt{l}} T}{(\nu x{:}S)P \xrightarrow{\tau,-} (\nu x{:}T)P'} \text{ [R-NewS]}$$

Semantics of Types

$$?t\,.\,S \xrightarrow{-} S \qquad !t\,.\,S \xrightarrow{-} S$$

$$\&[\mathtt{l}_i : T_i]_{i\in I} \xrightarrow{\mathtt{l}_i} T_i \qquad \oplus[\mathtt{l}_i : T_i]_{i\in I} \xrightarrow{\mathtt{l}_i} T_i$$

Operational semantics

$$x^p\,!\,v\,.\,P \mid x^{\overline{p}}\,?(y{:}\mathtt{t})\,.\,Q \ \xrightarrow{x,-}\ P \mid Q\{v/y\} \text{ [R-Comm]}$$

$$\frac{i \in I}{x^p \triangleleft \mathtt{l}_i\,.\,P \mid x^{\overline{p}} \triangleright [\mathtt{l}_j : Q_j]_{j \in I} \ \xrightarrow{x,\mathtt{l}_i}\ P \mid Q_i} \text{ [R-Select]}$$

$$\frac{P \xrightarrow{x,\mathtt{l}} P' \qquad S \xrightarrow{x,\mathtt{l}} T}{(\nu x{:}S)P \ \xrightarrow{\tau,-}\ (\nu x{:}T)P'} \text{ [R-NewS]}$$

$$\frac{P \xrightarrow{\alpha,\mathtt{l}} P' \qquad \alpha \neq x}{(\nu x{:}S)P \ \xrightarrow{\alpha,\mathtt{l}}\ (\nu x{:}S)P'} \text{ [R-New]}$$

$$\frac{P \xrightarrow{\alpha,\mathtt{l}} P'}{P|Q \xrightarrow{\alpha,\mathtt{l}} P'|Q} \text{ [R-Par]}$$

Structural equivalence

$$
\begin{array}{rcll}
P|0 & \equiv & P & \\
P|Q & \equiv & Q|P & \\
(P|Q)|R & \equiv & Q|(P|R) & \\
(\nu x{:}S)(\nu y{:}T)P & \equiv & (\nu y{:}T)(\nu x{:}S)P & \\
(\nu x{:}S)P|Q & \equiv & (\nu x{:}S)(P|Q) & \text{if } x^p \notin \mathrm{fn}(Q) \\
(\nu x{:}S)0 & \equiv & 0 & \text{if } S = \mathbf{end}
\end{array}
$$

Operational semantics

$$x^p\,!\,v\,.\,P \mid x^{\overline{p}}?(y{:}\mathtt{t})\,.\,Q \;\xrightarrow{x,-}\; P \mid Q\{v/y\} \quad \text{[R-Comm]}$$

$$\frac{i \in I}{x^p \triangleleft \mathtt{l}_i\,.\,P \mid x^{\overline{p}} \triangleright [\mathtt{l}_j : Q_j]_{j\in I} \;\xrightarrow{x,\mathtt{l}_i}\; P \mid Q_i} \text{[R-Select]}$$

$$\frac{P \xrightarrow{x,\mathtt{l}} P' \qquad S \xrightarrow{x,\mathtt{l}} T}{(\nu x{:}S)P \;\xrightarrow{\tau,-}\; (\nu x{:}T)P'} \text{[R-NewS]}$$

$$\frac{P \xrightarrow{\alpha,\mathtt{l}} P' \qquad \alpha \neq x}{(\nu x{:}S)P \;\xrightarrow{\alpha,\mathtt{l}}\; (\nu x{:}S)P'} \text{[R-New]}$$

$$\frac{P \xrightarrow{\alpha,\mathtt{l}} P'}{P|Q \xrightarrow{\alpha,\mathtt{l}} P'|Q} \text{[R-Par]}$$

$$\frac{P \equiv Q \qquad Q \xrightarrow{\alpha,\mathtt{l}} Q' \qquad Q' \equiv P'}{P \xrightarrow{\alpha,\mathtt{l}} P'} \text{[R-Cong]}$$

Typing

Type Judgement

$$\Gamma \vdash P$$

P uses channels as specified by Γ

Environments Γ

- Partial function from polarized names to types
- Written $x_1^{p_1} : t_1, x_2^{p_2} : t_1, \ldots, x_n^{p_n} : t_1$
- Its satisfies one of the following conditions
  - $x^+, x^-, x \notin dom(\Gamma)$
  - $x \in dom(\Gamma)$ and $x^+, x^- \notin dom(\Gamma)$
  - $x^p \in dom(\Gamma)$ and $p \in \{+, -\}$ and $x^{\overline{p}}, x \notin dom(\Gamma)$
  - $x^+, x^- \in dom(\Gamma)$ and $x \notin dom(\Gamma)$

Typing

$x^+ : \texttt{?int.!bool.end} \vdash x^+\texttt{?}(y{:}\texttt{int}).x^+\texttt{!true}.0$

$x^+ : \texttt{?int.!bool.end} \nvdash x^+\texttt{?}(y{:}\texttt{int}).x^+\texttt{!}y.0$

$x^+ : \texttt{?int.end}, y^- : \texttt{!int.end} \vdash x^+\texttt{?}(z{:}\texttt{int}).y^-\texttt{!}z.0$

$x^+ : \texttt{?int.end}, y^- : \texttt{!bool.end} \nvdash 0$

$\vdash (\nu x{:}\texttt{?int.end})(x^+\texttt{?}(z{:}\texttt{int}).0 \mid x^-\texttt{!}1.0)$

$\nvdash (\nu x{:}\texttt{?int.end})(x^+\texttt{?}(z{:}\texttt{int}).0)$

$\nvdash (\nu x{:}\texttt{?int.end})(x^+\texttt{?}(z{:}\texttt{int}).0 \mid x^-\texttt{!}1.0 \mid x^-\texttt{!}2.0)$

Typing

$$\not\vdash (\nu x{:}?\mathtt{int}.?\mathtt{int}.\mathtt{end})(x^{+}?(z{:}\mathtt{int}).x^{+}?(z{:}\mathtt{int}).0 \mid x^{-}!1.0 \mid x^{-}!2.0)$$

Think about

$$\begin{aligned}
(\nu x{:}?\mathtt{int}.!\mathtt{int}.\mathtt{end})(\\
& x^{+}?(z{:}\mathtt{int}).x^{+}!(z+1).0 \mid \\
& x^{+}?(z{:}\mathtt{int}).x^{+}!(z+1).0 \mid \\
& x^{-}!1.x^{-}?(z{:}\mathtt{int}).Q_1 \mid \\
& x^{-}!2.x^{-}?(z{:}\mathtt{int}).Q_2 \quad)
\end{aligned}$$

Typing

$$\frac{\Gamma_1 \vdash P_1 \qquad \Gamma_2 \vdash P_2}{\Gamma_1 + \Gamma_2 \vdash P_1 | P_2}\,\text{[T-Par]}$$

$$x^+ : \mathtt{Tester}, x^- : \overline{\mathtt{Tester}} \vdash P_{\mathtt{server}} \mid P_{\mathtt{client}}$$

where

$\mathtt{Tester} = \mathtt{?int.!bool.end}$

$P_{\mathtt{server}} = x^+\mathtt{?}(y{:}\mathtt{int}).x^+\mathtt{!true}.0$ *(faulty)*

$P_{\mathtt{client}} = x^-\mathtt{!1}.x^-\mathtt{?}(z{:}\mathtt{bool}).\mathtt{Q}$

Typing

$$\frac{\Gamma_1 \vdash P_1 \qquad \Gamma_2 \vdash P_2}{\Gamma_1 + \Gamma_2 \vdash P_1 | P_2}\text{[T-Par]}$$

$$x^+ : \mathtt{Tester}, x^- : \overline{\mathtt{Tester}} \nvdash P_{\mathtt{server}} \mid P_{\mathtt{client}} \mid P_{\mathtt{client}}$$

where

$\mathtt{Tester} = \mathtt{?int.!bool.end}$

$P_{\mathtt{server}} = x^+\mathtt{?}(y{:}\mathtt{int}).x^+\mathtt{!true}.0$ *(faulty)*

$P_{\mathtt{client}} = x^-\mathtt{!1}.x^-\mathtt{?}(z{:}\mathtt{bool}).\mathtt{Q}$

Typing

$$\frac{\Gamma_1 \vdash P_1 \qquad \Gamma_2 \vdash P_2}{\Gamma_1 + \Gamma_2 \vdash P_1 | P_2}\,\text{[T-Par]}$$

Context split

$$\begin{array}{rcll} \Gamma + x^{+} : t & = & \Gamma, x^{+} : t & \text{if } x, x^{+} \notin \mathrm{dom}(\Gamma) \\ \Gamma + x^{-} : t & = & \Gamma, x^{-} : t & \text{if } x, x^{-} \notin \mathrm{dom}(\Gamma) \\ (\Gamma, x : t) + \ x : t & = & \Gamma, x : t & \text{if } t \text{ is not a session type} \end{array}$$

Extended on context as

$$\begin{array}{rcl} \Gamma + \qquad \emptyset & = & \Gamma \\ \Gamma + (x^{p} : t, \Delta) & = & (\Gamma + x^{p} : t) + \Delta \end{array}$$

Linear usage of session endpoints

Typing

$$\frac{\Gamma_1 \vdash P_1 \qquad \Gamma_2 \vdash P_2}{\Gamma_1 + \Gamma_2 \vdash P_1 | P_2}\,[\text{T-Par}] \qquad \frac{\Gamma, x^+ : S, x^- : \overline{S} \vdash P}{\Gamma \vdash (\nu x{:}S)P}\,[\text{T-Res}]$$

$$(\nu x{:}\mathtt{Tester})(P_{\mathtt{server}} \mid P_{\mathtt{client}})$$

where

$\mathtt{Tester} = \mathtt{?int.!bool.end}$

$P_{\mathtt{server}} = x^+\mathtt{?}(y{:}\mathtt{int}).x^+\mathtt{!true}.0$ *(faulty)*

$P_{\mathtt{client}} = x^-\mathtt{!1}.x^-\mathtt{?}(z{:}\mathtt{bool}).\mathtt{Q}$

Typing

$$\frac{\Gamma_1 \vdash P_1 \qquad \Gamma_2 \vdash P_2}{\Gamma_1 + \Gamma_2 \vdash P_1 | P_2}\,[\text{T-Par}] \qquad \frac{\Gamma, x^+ : S, x^- : \overline{S} \vdash P}{\Gamma \vdash (\nu x{:}S)P}\,[\text{T-Res}]$$

$$\frac{\Gamma, x^p : S, y : t \vdash P}{\Gamma, x^p : ?t.S \vdash x^p?(y{:}t).P}\,[\text{T-In}] \qquad \frac{\Gamma_1 \vdash v : t \qquad \Gamma_2, x^p : S \vdash P}{\Gamma_1 + (\Gamma_2, x^p : !t.S) \vdash x^p!v.P}\,[\text{T-Out}]$$

Auxiliary Typing on expressions $\Gamma \vdash v : t$

$\emptyset \vdash \texttt{true} : \texttt{bool} \qquad \emptyset \vdash \texttt{false} : \texttt{bool}$

$\emptyset \vdash () : \texttt{unit} \qquad x^p : t \vdash x^p : t$

Typing

$$\frac{\Gamma_1 \vdash P_1 \qquad \Gamma_2 \vdash P_2}{\Gamma_1 + \Gamma_2 \vdash P_1 | P_2}\,[\text{T-Par}]$$

$$\frac{\Gamma, x^+ : S, x^- : \overline{S} \vdash P}{\Gamma \vdash (\nu x{:}S)P}\,[\text{T-Res}]$$

$$\frac{\Gamma, x^p : S, y : t \vdash P}{\Gamma, x^p : ?t.S \vdash x^p?(y{:}\mathtt{t}).P}\,[\text{T-In}]$$

$$\frac{\Gamma_1 \vdash v : t \qquad \Gamma_2, x^p : S \vdash P}{\Gamma_1 + (\Gamma_2, x^p : !t.S) \vdash x^p!v.P}\,[\text{T-Out}]$$

$$\frac{\Gamma, x^p : S_j \vdash P \qquad j \in I}{\Gamma, x^p : \oplus[\mathtt{l}_i : S_i]_{i \in I} \vdash x^p \triangleleft \mathtt{l}_j.P}\,[\text{T-Choice}]$$

$$\frac{\Gamma, x^p : S_i \vdash P_i \quad \forall i \in I}{\Gamma, x^p : \&[\mathtt{l}_i : S_i]_{i \in I} \vdash x^p \triangleright [\mathtt{l}_i : P_i]_{i \in I}}\,[\text{T-Branch}]$$

$$\frac{\Gamma \text{ completed}}{\Gamma \vdash 0}\,[\text{T-Nil}]$$

Γ completed if $\Gamma(x^p) = S$ implies $S = \mathtt{end}$

Terminology

We say P is *well-typed* if there exists Γ s.t $\Gamma \vdash P$

Example

Is P well-typed?

- $P = x^{+}?(y{:}\mathtt{int}).x^{+}!\mathtt{true}.0$
- Yes! take $\Gamma = x^{+} : ?\mathtt{int}.!\mathtt{bool}.\mathsf{end}$

$$\dfrac{\dfrac{\emptyset \vdash \mathtt{true} : \mathtt{bool} \qquad \dfrac{x^{+} : \mathsf{end}, y : \mathtt{int}\ \text{completed}}{x^{+} : \mathsf{end}, y : \mathtt{int} \vdash 0}\,[\text{T-Nil}]}{x^{+} : !\mathtt{bool}.\mathsf{end}, y : \mathtt{int} \vdash x^{+}!\mathtt{true}.0}\,[\text{T-Out}]}{x^{+} : ?\mathtt{int}.!\mathtt{bool}.\mathsf{end} \vdash x^{+}?(y{:}\mathtt{int}).x^{+}!\mathtt{true}.0}\,[\text{T-In}]$$

Example

Is P well-typed?

- $P = x^{+}?(y{:}!\mathtt{bool}.\mathtt{end}).y!\mathtt{true}.0$
- Yes! take $\Gamma = x^{+} : ?(!\mathtt{bool}.\mathtt{end}).\mathtt{end}$

$$\dfrac{\emptyset \vdash \mathtt{true} : \mathtt{bool} \qquad \dfrac{\dfrac{x^{+} : \mathtt{end}, y : \mathtt{end}\ \text{completed}}{x^{+} : \mathtt{end}, y : \mathtt{end} \vdash 0}\,\text{[T-Nil]}}{}}{\dfrac{x^{+} : \mathtt{end}, y : !\mathtt{bool}.\mathtt{end} \vdash y!\mathtt{true}.0}{x^{+} : ?(!\mathtt{bool}.\mathtt{end}).\mathtt{end} \vdash x^{+}?(y{:}!\mathtt{bool}.\mathtt{end}).y!\mathtt{true}.0}\,\text{[T-In]}}\,\text{[T-Out]}$$

Example

Is P well-typed

- $P = x^+?(y{:}!\mathtt{int}.\mathsf{end}).y!\mathtt{true}.0$
  No!
  Try with $\Gamma = x^+ : ?(!\mathtt{int}.\mathsf{end}).\mathsf{end}$

$$\dfrac{\emptyset \nvdash \mathtt{true} : \mathtt{int} \qquad \dfrac{\dfrac{x^+ : \mathsf{end}, y : \mathsf{end} \text{ completed}}{x^+ : \mathsf{end}, y : \mathsf{end} \vdash 0}\,\text{[T-Nil]}}{x^+ : \mathsf{end}, y : !\mathtt{int}.\mathsf{end} \vdash y!1.0}\,\text{[T-Out]}}{x^+ : ?(!\mathtt{int}.\mathsf{end}).\mathsf{end} \vdash x^+?(y{:}!\mathtt{int}.\mathsf{end}).y!1.0}\,\text{[T-In]}$$

Example

Is P well-typed?

- $P = x^{+}?(y{:}?\mathtt{int}.\mathtt{end}).y!1.0$
  No! Try with $\Gamma = x^{+} : ?(?\mathtt{int}.\mathtt{end}).\mathtt{end}$

$$\dfrac{\dfrac{\text{There is a mismatch: } y : ?\mathtt{int}.\mathtt{end} \text{ and } y!1.0}{x^{+} : \mathtt{end}, y : ?\mathtt{int}.\mathtt{end} \vdash y!1.0}\,[\text{T-Out}]}{x^{+} : ?(?\mathtt{int}.\mathtt{end}).\mathtt{end} \vdash x^{+}?(y{:}?\mathtt{int}.\mathtt{end}).y!1.0}\,[\text{T-In}]$$

Rethinking $(\mathtt{int} \to \mathtt{bool}) \to \mathtt{bool}$

$?(?\mathtt{int}.!\mathtt{bool}.\mathtt{end}).!\mathtt{bool}.\mathtt{end}$ is not $(\mathtt{int} \to \mathtt{bool}) \to \mathtt{bool}$

$g \;:\; (\mathtt{int} \to \mathtt{bool}) \to \mathtt{bool}$
$\mathtt{g}\ \mathtt{f} = \mathtt{f}\ 1$

We may implement g as

$$P_g = x^+?(y{:}\mathtt{t_f}).y!1.y?(z{:}\mathtt{bool}).x^+!z.0$$

where $\mathtt{t_f} = ?\mathtt{int}.!\mathtt{bool}.\mathtt{end}$

But

$$x^+ : ?\mathtt{t_f}.!\mathtt{bool}.\mathtt{end} \nvdash P_g$$

However

$$x^+ : ?\overline{\mathtt{t_f}}.!\mathtt{bool}.\mathtt{end} \vdash P_g$$

What about ?(!int.?bool.end).!bool.end?

g : (int $\to$ bool) $\to$ bool
g y = (y 1) & (y 2)

We may implement g as

$$P_g = x^+?(y{:}\overline{\mathtt{t_f}}).y!1.y?(z_1{:}\mathtt{bool}).y!2.y?(z_2{:}\mathtt{bool}).x^+!z_1\&z_2.0$$

where $\mathtt{t_f} = ?\mathtt{int}.!\mathtt{bool}.\mathtt{end}$

However

$$x^+ : ?\overline{\mathtt{t_f}}.!\mathtt{bool}.\mathtt{end} \not\vdash P_g$$

The parameter y **must** be used just for **one** application

On linearity

- Consider $P = x^{+}\,!\,y^{+}\,.\,y^{+}\,!\,1\,.\,0$.
- Does the following hold?

$$\Gamma, x^{+} : !(!\mathtt{int}.\mathtt{end}).\mathtt{end}, y^{+} : !\mathtt{int}.\mathtt{end} \vdash P$$

$$\frac{\Gamma_1 \vdash v : t \qquad \Gamma_2, x^p : S \vdash P}{\Gamma_1 + (\Gamma_2, x^p : !t.S) \vdash x^p\,!\,v\,.\,P}\;\text{[T-Out]}$$

- No. Why does [T-Out] ban P?
- Take

$$Q = (\nu y{:}!\mathtt{int}.\mathtt{end})(\nu x{:}!(!\mathtt{int}.\mathtt{end}).\mathtt{end})(\\ P \mid x^{-}?(z{:}!\mathtt{int}.\mathtt{end}).z\,!\,2\,.\,0 \mid y^{-}?(z{:}\mathtt{int}).0)$$

- $Q \xrightarrow{\tau,-} Q'$ where

$$Q' = (\nu y{:}!\mathtt{int}.\mathtt{end})(\nu x{:}\mathtt{end})(y^{+}\,!\,1\,.\,0 \mid y^{+}\,!\,2\,.\,0 \mid y^{-}?(z{:}\mathtt{int}).0)$$

  where two processes concurrently send on y^{+}
- A process does not use a session endpoint after *delegating* it (i.e., sending it over a different session endpoint)

Results

Theorem (Type Preservation)

- *If* $\Gamma \vdash P$ *and* $P \xrightarrow{\tau,-} Q$ *then* $\Gamma \vdash Q$.
- *If* $\Gamma, x^p : S, x^{\overline{p}} : \overline{S} \vdash P$ *and* $P \xrightarrow{x,\mathtt{l}} Q$ *then* $S \xrightarrow{x,\mathtt{l}} T$ *and* $\Gamma, x^p : T, x^{\overline{p}} : \overline{T} \vdash Q$.

Theorem (Type Safety)

Let $\Gamma \vdash P$ *where* Γ *balanced* [2]

- *If* $P \equiv (\nu \tilde{z}{:}\tilde{S})(x^p \,!\, v \,.\, P_1 \mid x^{\overline{p}}?(y{:}\mathtt{t}) \,.\, P_2 \mid Q)$ *with* $p \in \{+,-\}$ *then* $x, x^+, x^- \notin \mathrm{fn}(Q)$ *and* $\Gamma, \tilde{z} : \tilde{S} \vdash v : t$
- *If* $P \equiv (\nu \tilde{z}{:}\tilde{S})(x^p \triangleleft \mathtt{l}_j \,.\, R \mid x^{\overline{p}} \triangleright [\mathtt{l}_i : P_i]_{i \in I} \mid Q)$ *with* $p \in \{+,-\}$ *then* $x, x^+, x^- \notin \mathrm{fn}(Q)$ *and* $j \in I$.

[2] Γ is balanced if $x^p : S$ and $x^{\overline{p}} : T$ implies $S = \overline{T}$

Properties

Does the following hold?

$$\vdash (\nu x{:}?\mathtt{int}.\mathtt{end})(x^{+}?(z{:}\mathtt{int}).x^{-}!z.0)$$

Yes!

- The process is well-typed and deadlocked

The type system ensures

- *Type Safety* in communication (e.g., received values are of the expected type)
- *Session Fidelity* (e.g., communication follows the flow described by the session type)
- The type system does not ensure deadlock-freedom

Deadlock

Deadlocked Process

$$P = x^{+}?(z{:}\mathtt{int}).y^{-}!1.0 \quad | \quad y^{+}?(z{:}\mathtt{int}).x^{-}!1.0$$

Is P well-typed?

$$\vdash (\nu x{:}?\mathtt{int}.\mathtt{end})(x^{+}?(z{:}\mathtt{int}).x^{-}!z.0)$$

Yes!

- The process is well-typed and deadlocked
- The type system does not check the dependencies between different sessions

Deadlock-freedom by design (linear logic approaches)

- Connection drawn between linear logic and session-typed pi-calculus gave rise to type systems that guarantee deadlock-freedom
  - Luís Caires, Frank Pfenning: Session Types as Intuitionistic Linear Propositions. CONCUR 2010.
  - Philip Wadler: Propositions as sessions. ICFP 2012.
- The type system imposes some structural constraint on programs
  - two processes share at most one channel
  - Hence, there are no circular dependencies
- Key typing rule (presentation recast)

$$\frac{\Gamma_1, x^p : S \vdash P \quad \Gamma_1, x^{\overline{p}} : \overline{S} \vdash Q}{\Gamma_1, \Gamma_2 \vdash (\nu x{:}S)(P \mid Q)}\,\text{[T-Cut]}$$