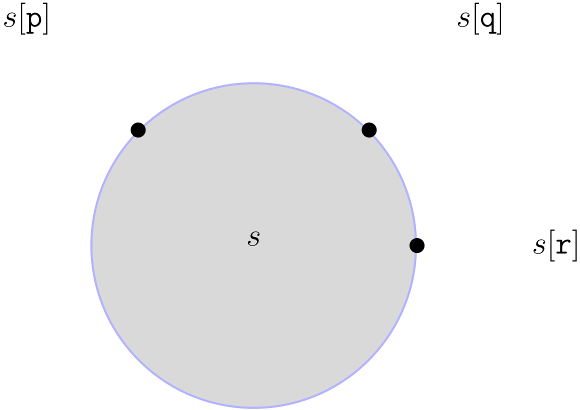
Fair Subtyping for Multi-Party Session Types

Luca Padovani

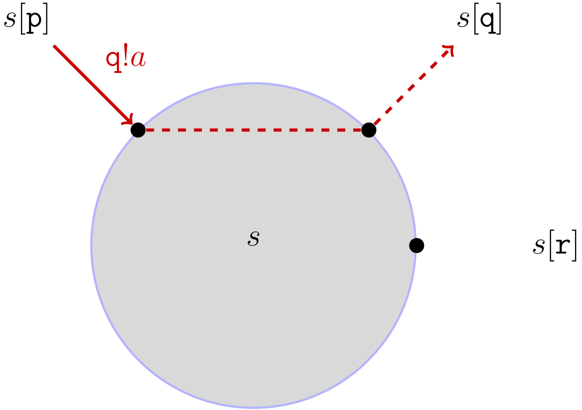
Dipartimento di Informatica, Università di Torino

Sessions and session types



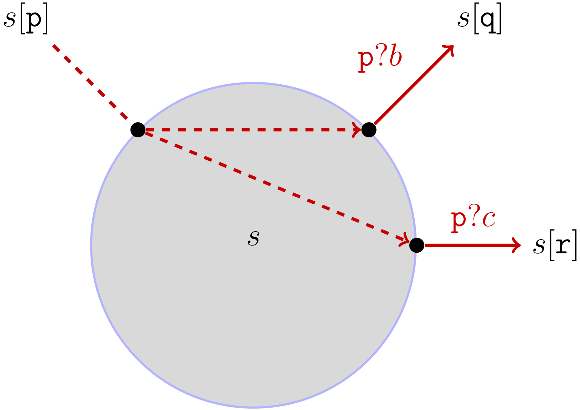
- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$

Sessions and session types



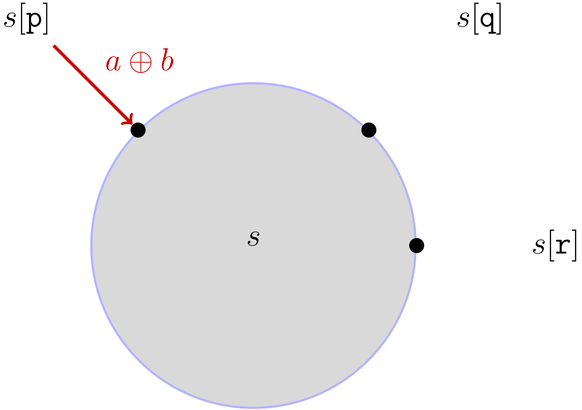
- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$

Sessions and session types



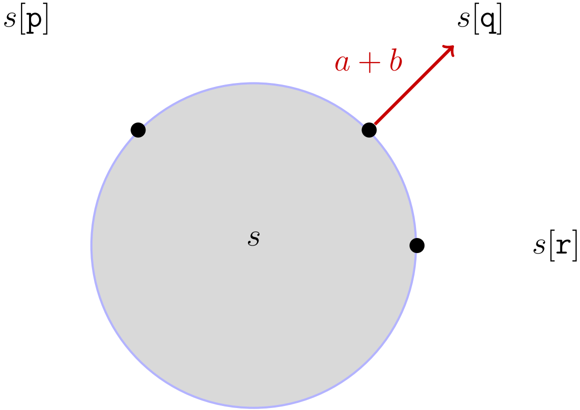
- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$

Sessions and session types



- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$

Sessions and session types



- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$

Digression

$$t \to s$$

$$!t.?s.\mathsf{end}$$

Session correctness = safety + liveness

Safety

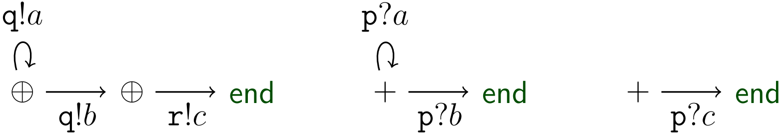
- no message of unexpected type is ever sent

Liveness

- all non-terminated participants (eventually) make progress

Example: multi-party session

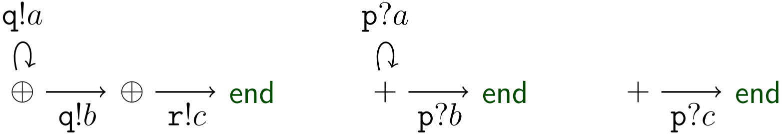
- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$



Is this session correct?

Example: multi-party session

- $s[\mathtt{p}] : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $s[\mathtt{q}] : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $s[\mathtt{r}] : \mathtt{p}?c.\mathsf{end}$



Is this session correct? **Yes, under a fairness assumption**

Subtyping for session types

- Simon Gay, Malcolm Hole, **Subtyping for session types in the pi calculus**, 2005

$$\mathsf{end} \leqslant_{\mathsf{GH}} \mathsf{end}$$

$$\frac{T_1 \leqslant_{\mathsf{GH}} S_1}{\mathrm{p}?a.T_1 \leqslant_{\mathsf{GH}} \mathrm{p}?a.S_1 + \mathrm{p}?b.S_2} \qquad \frac{T_1 \leqslant_{\mathsf{GH}} S_1}{\mathrm{p}!a.T_1 \oplus \mathrm{p}!b.T_2 \leqslant_{\mathsf{GH}} \mathrm{p}!a.S_1}$$

Subtyping for session types

- Simon Gay, Malcolm Hole, **Subtyping for session types in the pi calculus**, 2005

$$\mathsf{end} \leqslant_{\mathsf{GH}} \mathsf{end}$$

$$\frac{T_1 \leqslant_{\mathsf{GH}} S_1}{\mathrm{p}?a.T_1 \leqslant_{\mathsf{GH}} \mathrm{p}?a.S_1 + \mathrm{p}?b.S_2} \qquad \frac{T_1 \leqslant_{\mathsf{GH}} S_1}{\mathrm{p}!a.T_1 \oplus \mathrm{p}!b.T_2 \leqslant_{\mathsf{GH}} \mathrm{p}!a.S_1}$$

covariant input

Subtyping for session types

- Simon Gay, Malcolm Hole, **Subtyping for session types in the pi calculus**, 2005

$$\mathsf{end} \leqslant_{\mathsf{GH}} \mathsf{end}$$

$$\frac{T_1 \leqslant_{\mathsf{GH}} S_1}{\mathrm{p}?a.T_1 \leqslant_{\mathsf{GH}} \mathrm{p}?a.S_1 + \mathrm{p}?b.S_2} \qquad \frac{T_1 \leqslant_{\mathsf{GH}} S_1}{\mathrm{p}!a.T_1 \oplus \mathrm{p}!b.T_2 \leqslant_{\mathsf{GH}} \mathrm{p}!a.S_1}$$

contravariant output

Digression

$$\texttt{int} \leqslant \texttt{real}$$

- it is safe to use a value of type `int` where a value of type `real` is expected

$$!\texttt{real} \leqslant_{\mathsf{GH}} !\texttt{int}$$

- it is safe to use a channel of type `!int` where a channel of type `!real` is expected, or
- it is safe to use a process that sends `int`'s where a process that sends `real`'s is expected

Digression

$$\texttt{int} \leqslant \texttt{real}$$

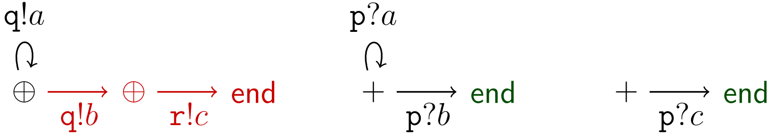
- it is safe to use a value of type `int` where a value of type `real` is expected

$$!\texttt{real} \leqslant_{\mathsf{GH}} !\texttt{int}$$

- it is safe to use a channel of type `!int` where a channel of type `!real` is expected, or
- it is safe to use a process that sends `int`'s where a process that sends `real`'s is expected

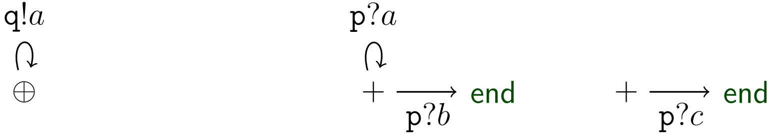
Example: multi-party session (and subtyping)

- $\mathtt{p} : T = \mathtt{q}!a.T \oplus \mathtt{q}!b.\mathtt{r}!c.\mathsf{end}$
- $\mathtt{q} : S = \mathtt{p}?a.S + \mathtt{p}?b.\mathsf{end}$
- $\mathtt{r} : \mathtt{p}?c.\mathsf{end}$



Example: multi-party session (and subtyping)

- $\mathtt{p} : T = \mathtt{q}!a.T$
- $\mathtt{q} : S = \mathtt{p}?a.S + \mathtt{p}?b.\text{end}$
- $\mathtt{r} : \mathtt{p}?c.\text{end}$



Is this session correct?

Dyadic vs multi-party sessions

In the dyadic setting...

- $\leqslant_{\mathsf{GH}}$ preserves both safety and liveness

In the multi-party setting...

- $\leqslant_{\mathsf{GH}}$ preserves safety
- $\leqslant_{\mathsf{GH}}$ does not (necessarily) preserve liveness

How to fix subtyping

session $\quad M = T_1 \mid \cdots \mid T_n$

Definition (correct session)

- M **correct** if $M \Longrightarrow N$ implies $N \Longrightarrow \mathsf{end} \mid \cdots \mid \mathsf{end}$

Definition (fair subtyping)

- $[\![T]\!] = \{M \mid (T \mid M) \text{ is } \textbf{correct}\}$
- $T \leqslant S \quad \text{iff} \quad [\![T]\!] \subseteq [\![S]\!]$

How to fix subtyping

session $\quad M = T_1 \mid \cdots \mid T_n$

Definition (**correct** session)

- M **correct** if $M \Longrightarrow N$ implies $N \Longrightarrow \mathsf{end} \mid \cdots \mid \mathsf{end}$

Definition (fair subtyping)

- $[\![T]\!] = \{M \quad | \quad (T \mid M) \text{ is } \textbf{correct}\}$
- $T \leqslant S \quad \text{iff} \quad [\![T]\!] \subseteq [\![S]\!]$

Dilemma

$$\leqslant_{\mathsf{GH}} \qquad \text{versus} \qquad \leqslant$$

- $\leqslant_{\mathsf{GH}}$ is intuitive but unsound
- $\leqslant$ is sound but obscure

$\leqslant_{\mathsf{GH}}$ and $\leqslant$ are incomparable

$$T = \mathtt{p}!a.T \qquad S = \mathtt{q}?b.S$$

$$T \leqslant S$$

$$[\![T]\!] = [\![S]\!] = \emptyset$$

$$T \not\leqslant_{\mathsf{GH}} S \qquad S \not\leqslant_{\mathsf{GH}} T$$

$\leqslant_{\mathsf{GH}}$ and $\leqslant$ are incomparable

$$T = \mathtt{p}!a.T \qquad S = \mathtt{q}?b.S$$

$$T \lesseqgtr S$$

$$[\![T]\!] = [\![S]\!] = \emptyset$$

$$T \not\leqslant_{\mathsf{GH}} S \qquad S \not\leqslant_{\mathsf{GH}} T$$

not viable $[\![\mathsf{fail}]\!] = [\![T]\!] = [\![S]\!] = \cdots = \emptyset$

$[\![\ldots]\!] \neq \emptyset$ viable $T \leqslant S \Rightarrow T \leqslant_{\mathsf{GH}} S$

When does $\leqslant_{\mathsf{GH}}$ imply $\leqslant$?

$$T \leqslant S \qquad \text{implies} \qquad \mathsf{traces}(S) \subseteq \mathsf{traces}(T)$$

When does $\leqslant_{\mathsf{GH}}$ imply $\leqslant$?

$T \leqslant S$ implies $\mathsf{traces}(S) \subseteq \mathsf{traces}(T)$

almost, anyway

When does $\leqslant_{\mathsf{GH}}$ imply $\leqslant$?

$$T \leqslant S \qquad \text{implies} \qquad \mathsf{traces}(S) \subseteq \mathsf{traces}(T)$$

Idea

1. *define* $T - S$ *so that* $\mathsf{traces}(T - S) = \mathsf{traces}(T) \setminus \mathsf{traces}(S)$
2. *check whether* $T - S$ *is viable*
   - *if* $M \mid (T - S)$ *is correct, then* M *does not "use" any trace in* $\mathsf{traces}(T)$ *if it is also in* $\mathsf{traces}(S)$

When does $\leqslant_{\mathsf{GH}}$ imply $\leqslant$?

$$T \leqslant S \qquad \text{implies} \qquad \mathsf{traces}(S) \subseteq \mathsf{traces}(T)$$

Idea

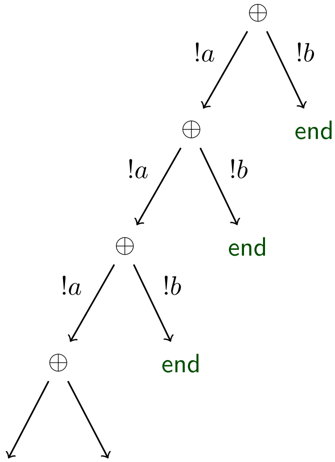
1. *define* $T - S$ *so that* $\mathsf{traces}(T - S) = \mathsf{traces}(T) \setminus \mathsf{traces}(S)$
2. *check whether* $T - S$ *is viable*
   - *if* $M \mid (T - S)$ *is correct, then* M *does not "use" any trace in* $\mathsf{traces}(T)$ *if it is also in* $\mathsf{traces}(S)$

Theorem

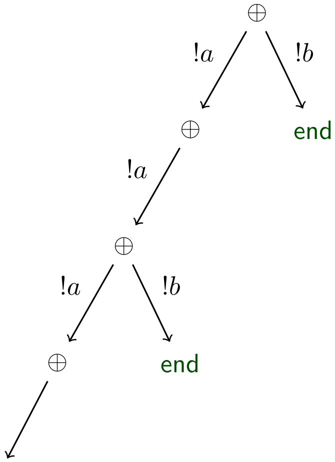
$T \leqslant S$ *iff* $T \leqslant_{\mathsf{GH}} S$ *and* $T - S$ *is not viable*

Experiment 1

$T = !a.T \oplus !b.\text{end}$

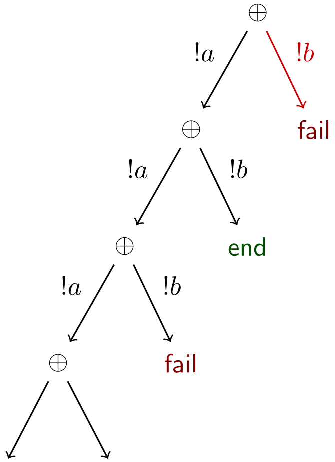


$S = !a.!a.S \oplus !b.\text{end}$

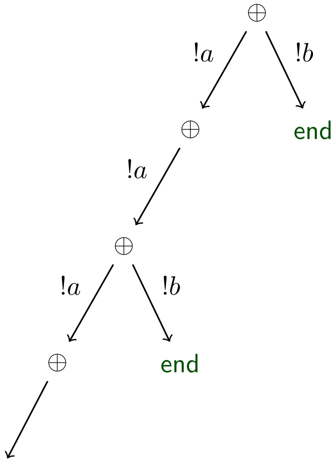


Experiment 1

$T - S \lessgtr ?$



$S = !a.!a.S \oplus !b.\mathsf{end}$

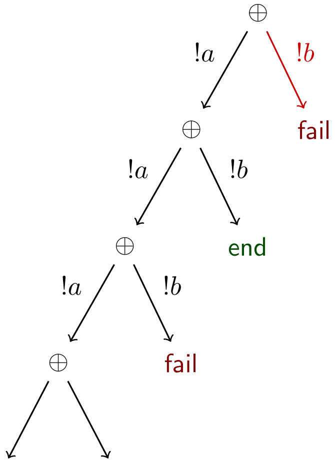


Murphy's law

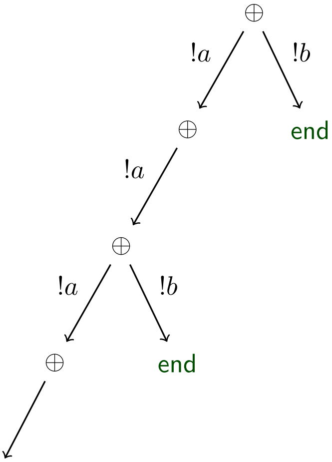
If you can fail, you will fail

Experiment 1

$T - S \lesssim$ fail



$S = !a.!a.S \oplus !b.$end



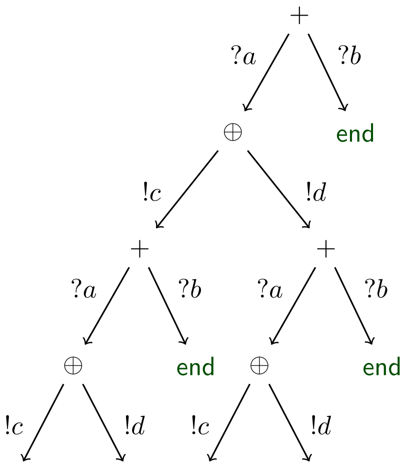
Murphy's law

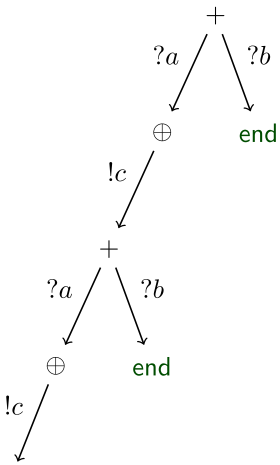
If you can fail, you will fail

Experiment 2

$T = ?a.(!c.T \oplus !d.T) + ?b.\text{end}$

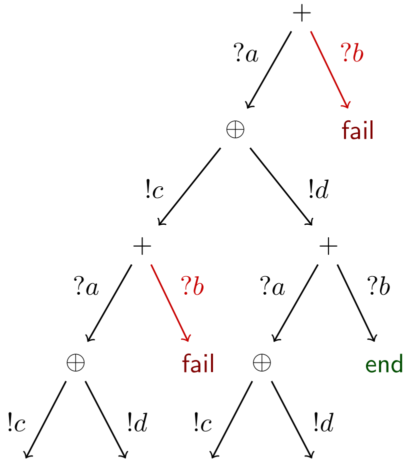
$S = ?a.!c.S + ?b.\text{end}$



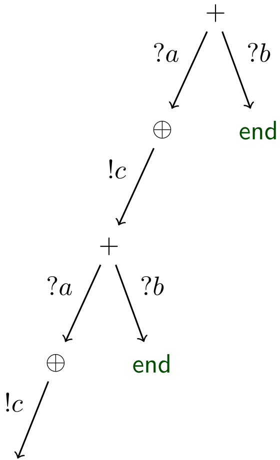


Experiment 2

$T - S \lessgtr ?$



$S = ?a.!c.S + ?b.\mathsf{end}$

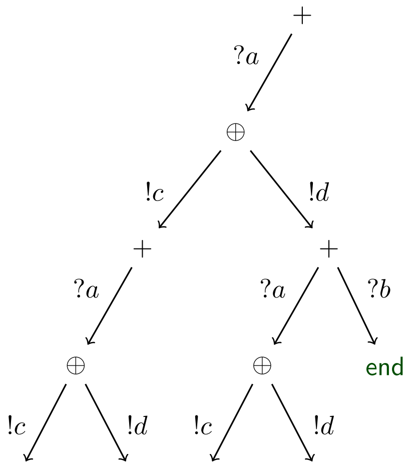


Dual of Murphy's law

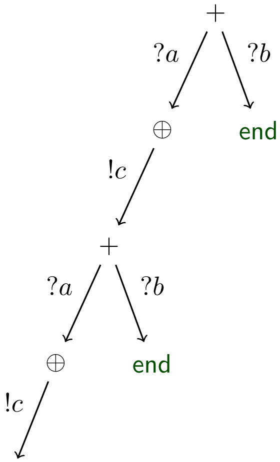
If you don't fail enough, your partner won't fail

Experiment 2

$T - S \lesssim ?a.(!c.(T-S) \oplus !d.T)$



$S = ?a.!c.S + ?b.\mathsf{end}$



Dual of Murphy's law

If you don't fail enough, your partner won't fail

Axiomatization of $\leqslant$

(S-Fail)
$\text{fail} \leqslant T$

(S-End)
$\text{end} \leqslant \text{end}$

(S-Input)

$$\frac{T_1 \leqslant S_1}{\mathtt{p}?a.T_1 \leqslant \mathtt{p}?a.S_1 + \mathtt{p}?b.S_2}$$

(S-Output)

$$\frac{T_1 \leqslant S_1 \qquad T - S \leqslant \text{fail}}{T = \mathtt{p}!a.T_1 \oplus \mathtt{p}!b.T_2 \leqslant \mathtt{p}!a.S_1 = S}$$

Axiomatization of $\leqslant$

(S-Fail) $\textsf{fail} \leqslant T$

(S-End) $\textsf{end} \leqslant \textsf{end}$

(S-Input)

$$\frac{T_1 \leqslant S_1}{\texttt{p}?a.T_1 \leqslant \texttt{p}?a.S_1 + \texttt{p}?b.S_2}$$

(S-Output)

$$\frac{T_1 \leqslant S_1 \qquad T - S \leqslant \textsf{fail}}{T = \texttt{p}!a.T_1 \oplus \texttt{p}!b.T_2 \leqslant \texttt{p}!a.S_1 = S}$$

Axiomatization of $\leqslant$

(S-Fail)
$$\mathsf{fail} \leqslant T$$

(S-End)
$$\mathsf{end} \leqslant \mathsf{end}$$

(S-Input)
$$\frac{T_1 \leqslant S_1}{\mathsf{p}?a.T_1 \leqslant \mathsf{p}?a.S_1 + \mathsf{p}?b.S_2}$$

(S-Output)
$$\frac{T_1 \leqslant S_1 \qquad T - S \leqslant \mathsf{fail}}{T = \mathsf{p}!a.T_1 \oplus \mathsf{p}!b.T_2 \leqslant \mathsf{p}!a.S_1 = S}$$

(S-Path)
$$\frac{\exists S \in \pi : S \leqslant \mathsf{fail} \quad ^{(T\downarrow\pi)}}{T \leqslant \mathsf{fail}}$$

(S-Murphy)
$$\frac{k \in \{1,2\} \qquad T_k \leqslant \mathsf{fail}}{\mathsf{p}!a_1.T_1 \oplus \mathsf{p}!a_2.T_2 \leqslant \mathsf{fail}}$$

Axiomatization of $\leqslant$

(S-Fail)
$$\text{fail} \leqslant T$$

(S-End)
$$\text{end} \leqslant \text{end}$$

(S-Input)
$$\frac{T_1 \leqslant S_1}{\mathtt{p}?a.T_1 \leqslant \ \ldots}$$

(S-Output)
$$\frac{T_1 \leqslant S_1 \qquad T - S \leqslant \text{fail}}{\ldots_1 \oplus \mathtt{p}!b.T_2 \leqslant \mathtt{p}!a.S_1 = S}$$

every path π to success. . .

(S-Path)
$$\frac{\exists S \in \pi : S \leqslant \text{fail} \ \ {}^{(T\downarrow\pi)}}{T \leqslant \text{fail}}$$

(S-Murphy)
$$\frac{k \in \{1, 2\} \qquad T_k \leqslant \text{fail}}{\mathtt{p}!a_1.T_1 \oplus \mathtt{p}!a_2.T_2 \leqslant \text{fail}}$$

Axiomatization of $\leqslant$

(S-Fail)
$\text{fail} \leqslant T$

(S-End)
$\text{end} \leqslant \text{end}$

(S-Input)
$$\frac{T_1 \leqslant S_1}{\mathrm{p}?a.T_1 \leqslant \ \ldots}$$

(S-Output)
$$\frac{T_1 \leqslant S_1 \qquad T - S \leqslant \text{fail}}{\ldots_1 \oplus \mathrm{p}!b.T_2 \leqslant \mathrm{p}!a.S_1 = S}$$

every path π to success. . .

. . . goes through some node S that. . .

(S-Path)
$$\frac{\exists S \in \pi : S \leqslant \text{fail} \quad ^{(T\downarrow\pi)}}{T \leqslant \text{fail}}$$

(S-Murphy)
$$\frac{k \in \{1, 2\} \qquad T_k \leqslant \text{fail}}{\mathrm{p}!a_1.T_1 \oplus \mathrm{p}!a_2.T_2 \leqslant \text{fail}}$$

Axiomatization of $\leqslant$

(S-Fail)
$\text{fail} \leqslant T$

(S-End)
$\text{end} \leqslant \text{end}$

(S-Input)

$$\frac{T_1 \leqslant S_1}{\mathtt{p}?a.T_1 \leqslant \cdots}$$

(S-Output)

$$\frac{T_1 \leqslant S_1 \qquad T - S \leqslant \text{fail}}{\cdots_1 \oplus \mathtt{p}!b.T_2 \leqslant \mathtt{p}!a.S_1 = S}$$

every path π to success. . .

. . . goes through some node S that. . .

(S-Path)

$$\frac{\exists S \in \pi : S \leqslant \text{fail} \quad ^{(T\downarrow\pi)}}{T \leqslant \text{fail}}$$

(S-Murphy)

$$\frac{k \in \{1,2\} \qquad T_k \leqslant \text{fail}}{\mathtt{p}!a_1.T_1 \oplus \mathtt{p}!a_2.T_2 \leqslant \text{fail}}$$

. . . leads to failure

(Fair) subtyping = (fair) testing preorder

- P passes test T
- $P \sqsubseteq Q$ iff P passes test T implies Q passes test T

"Unfair" testing

- De Nicola, Hennessy, **Testing equivalences for processes**, 1983
- ... similar properties of $\leqslant_{\mathsf{GH}}$

Fair testing

- Cleaveland, Natarajan, **Divergence and fair testing**, 1995
- Rensink, Vogler, **Fair testing**, 2007

Fair testing vs fair subtyping

Fair testing

+ generic processes
− denotational (= obscure) characterization
− no complete axiomatization
− exponential

Fair subtyping

− simple processes (session types)
+ operational (= hopefully less obscure) characterization
+ complete axiomatization
+ polynomial

More on fair subtyping

- Padovani, **Fair Subtyping for Multi-Party Session Types**, COORDINATION 2011
  - + formal definitions and proofs
  - + algorithms (viability, normal form, subtyping)

- long version coming up on my home page in **3 days**
  - + higher-order session types
  - + details of axiomatization

Challenges

"fair" type checking?

Fair Subtyping for Multi-Party Session Types

Luca Padovani

Dipartimento di Informatica, Università di Torino