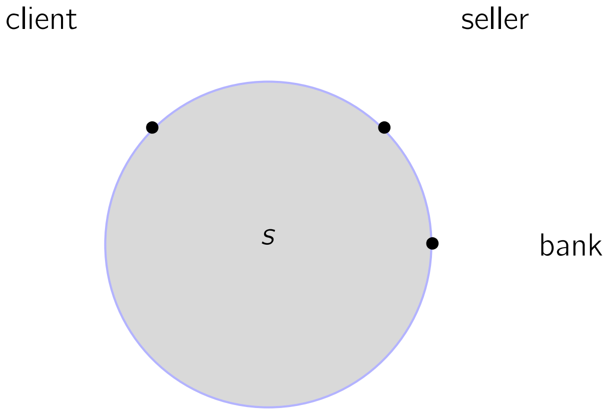
Fair Subtyping for Open Session Types

Luca Padovani

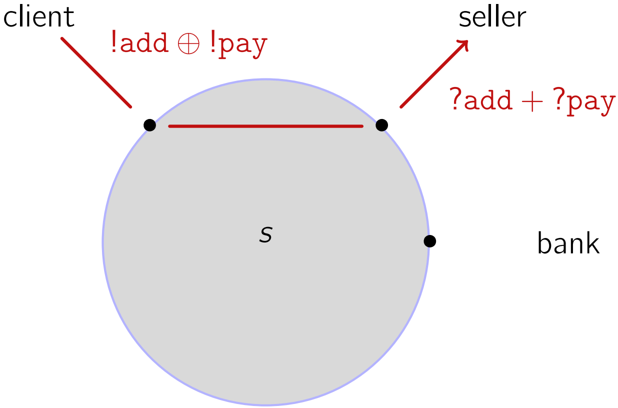
Dipartimento di Informatica, Università di Torino, Italy

Sessions



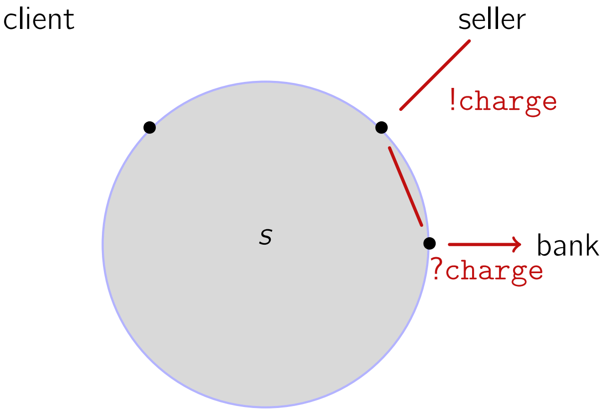
- private communication channel between processes
- two or more endpoints

Sessions



- private communication channel between processes
- two or more endpoints

Sessions



- private communication channel between processes
- two or more endpoints

Session correctness = safety + liveness

Safety: no unexpected message is **ever** sent

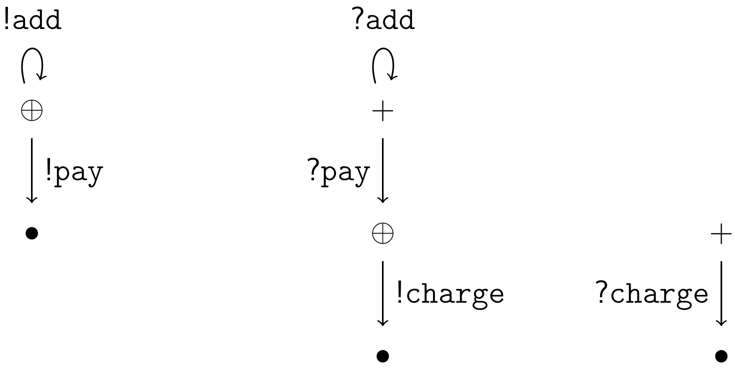
“after paying the client won’t add more items to the cart”

Liveness: all non-terminated participants **eventually** make progress

“the seller will receive payment through client’s bank”

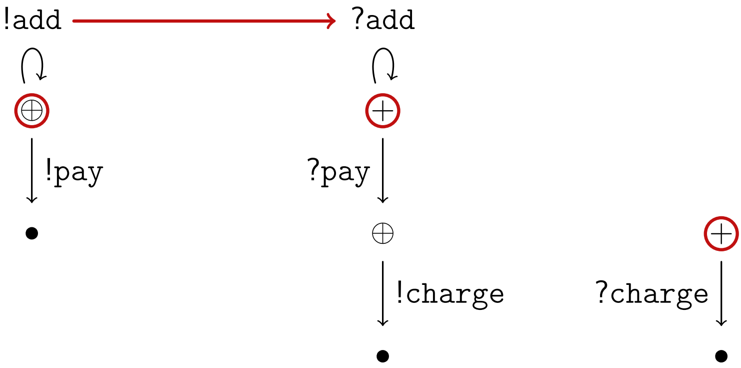
A correct session (under fairness assumptions)

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(?\mathtt{add}.x + ?\mathtt{pay}.!\mathtt{charge})$ $?\mathtt{charge}$



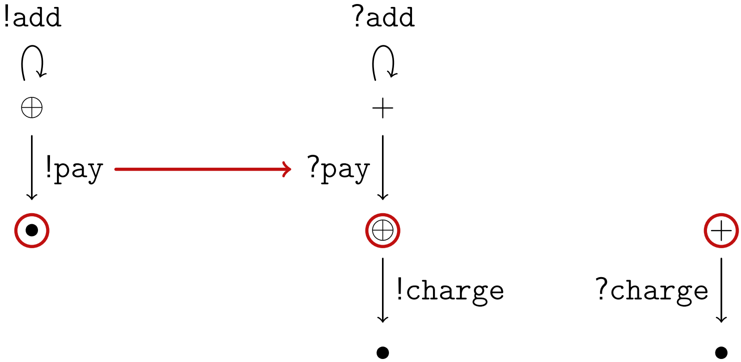
A correct session (under fairness assumptions)

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(?\mathtt{add}.x + ?\mathtt{pay}.!\mathtt{charge})$ $?\mathtt{charge}$



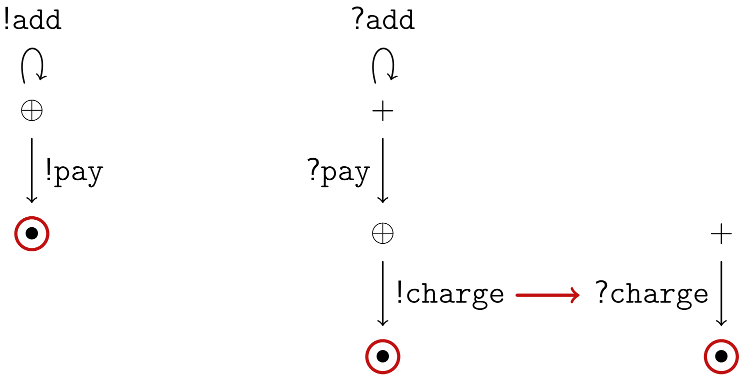
A correct session (under fairness assumptions)

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(?\mathtt{add}.x + ?\mathtt{pay}.!\mathtt{charge})$ $?\mathtt{charge}$



A correct session (under fairness assumptions)

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(?\mathtt{add}.x + ?\mathtt{pay}.!\mathtt{charge})$ $?\mathtt{charge}$



Session type checking

$$k : \mu x.(!\mathtt{add}.x \oplus !\mathtt{pay}) \vdash \mathtt{rec}\ P.k!\langle m \rangle.P$$

Session type checking

$$\frac{P \mapsto \{k : x\}; k : !\mathtt{add}.x \oplus !\mathtt{pay} \vdash k!\langle m\rangle.P}{k : \mu x.(!\mathtt{add}.x \oplus !\mathtt{pay}) \vdash \mathtt{rec}\ P.k!\langle m\rangle.P}\ [\text{t-rec}]$$

Session type checking

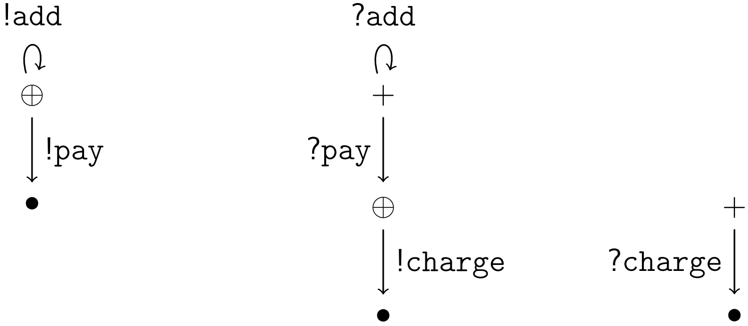
$$\dfrac{\dfrac{\vdash m : \texttt{add} \qquad P \mapsto \{k : x\}; k : x \vdash P}{P \mapsto \{k : x\}; k : !\texttt{add}.x \oplus !\texttt{pay} \vdash k!\langle m \rangle.P}\text{[t-output]}}{k : \mu x.(!\texttt{add}.x \oplus !\texttt{pay}) \vdash \texttt{rec}\ P.k!\langle m \rangle.P}\text{[t-rec]}$$

Session type checking

$$
\dfrac{\vdash m : \texttt{add} \qquad \dfrac{}{P \mapsto \{k : x\}; k : x \vdash P}\,[\text{t-var}]}{\dfrac{P \mapsto \{k : x\}; k : !\texttt{add}.x \oplus !\texttt{pay} \vdash k!\langle m \rangle.P}{k : \mu x.(!\texttt{add}.x \oplus !\texttt{pay}) \vdash \texttt{rec}\ P.k!\langle m \rangle.P}\,[\text{t-rec}]}\,[\text{t-output}]
$$

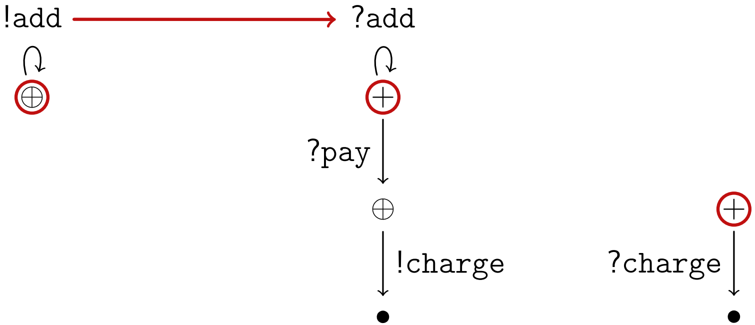
Session type checking

$$\dfrac{\dfrac{\vdash m : \mathtt{add} \qquad \dfrac{}{P \mapsto \{k : x\}; k : x \vdash P}\,[\text{t-var}]}{P \mapsto \{k : x\}; k : !\mathtt{add}.x \oplus !\mathtt{pay} \vdash k!\langle m \rangle.P}\,[\text{t-output}]}{k : \mu x.(!\mathtt{add}.x \oplus !\mathtt{pay}) \vdash \mathtt{rec}\ P.k!\langle m \rangle.P}\,[\text{t-rec}]$$

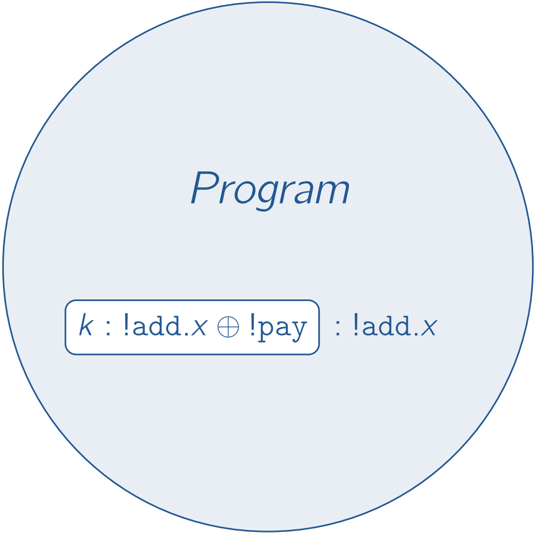


Session type checking, **flawed?**

$$\dfrac{\dfrac{\vdash m : \mathtt{add} \qquad \dfrac{}{P \mapsto \{k : x\}; k : x \vdash P}\,[\text{t-var}]}{P \mapsto \{k : x\}; k : !\mathtt{add}.x \oplus !\mathtt{pay} \vdash k!\langle m \rangle.P}\,[\text{t-output}]}{k : \mu x.(!\mathtt{add}.x \oplus !\mathtt{pay}) \vdash \mathtt{rec}\ P.k!\langle m \rangle.P}\,[\text{t-rec}]$$



Identifying the problem



$$!\mathsf{add}.x \oplus !\mathsf{pay} \overset{?}{\leqslant} !\mathsf{add}.x$$

Subtyping for session types

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

$$!\mathtt{add}.x \oplus !\mathtt{pay} \leqslant_{\mathsf{U}} !\mathtt{add}.x$$

- $\leqslant_{\mathsf{U}}$ subtyping preserves safety...
- ...but not necessarily liveness

Subtyping for session types

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

$$!\mathtt{add}.x \oplus !\mathtt{pay} \leqslant_{\mathsf{U}} !\mathtt{add}.x$$

- $\leqslant_{\mathsf{U}}$ subtyping preserves safety…
- …but not necessarily liveness

What about the coarsest liveness-preserving subtyping?

Defining fair subtyping, the easy way

Idea

- session type $T \sim$ sequential ccs process
- session $M \sim \prod$ session types

Definition

1. M is correct if $M \Longrightarrow N$ implies $N \stackrel{!\mathtt{OK}}{\Longrightarrow}$
2. $T \leqslant S \stackrel{\text{def}}{\iff} \forall \mathcal{C}, M : \mathcal{C}[T] \mid M$ correct implies $\mathcal{C}[S] \mid M$ correct

☺ coarsest liveness-preserving subtyping, by definition

☹ uninformative

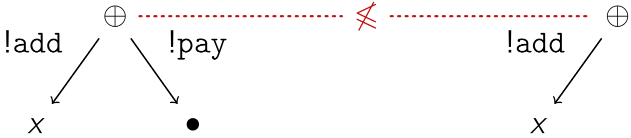
Why is fair subtyping hard to characterize?

Recursion

- for **finite** types, $\leqslant \; = \; \leqslant_{\mathsf{U}}$

Context dependency

- the **same** types **may** or **may not** be related by $\leqslant$



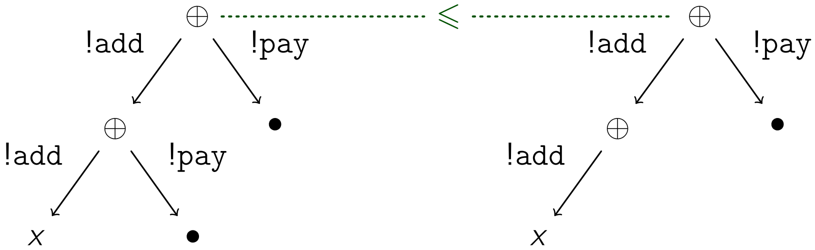
Why is fair subtyping hard to characterize?

Recursion

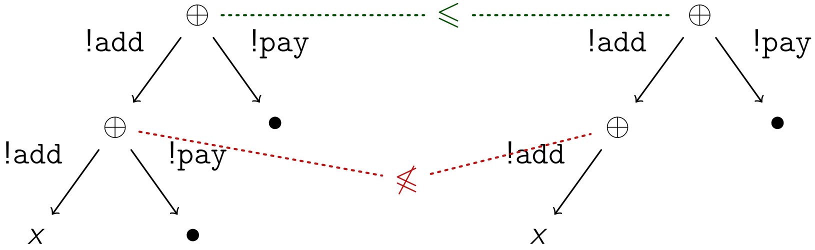
- for **finite** types, $\leqslant \, = \, \leqslant_{\mathsf{U}}$

Context dependency

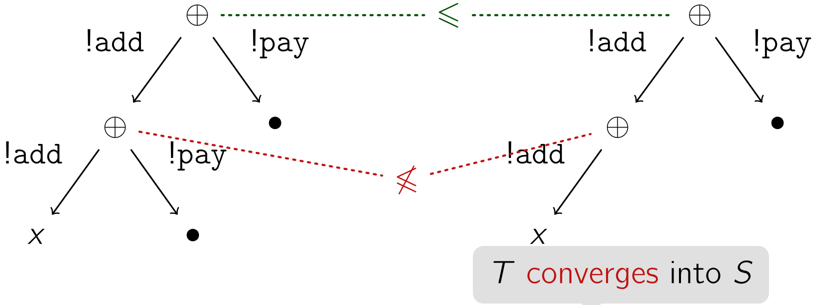
- the **same** types **may** or **may not** be related by $\leqslant$



Subtyping and trace convergence



Subtyping and trace convergence



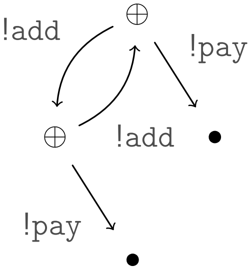
$$T \leqslant S \iff T \leqslant_{\mathsf{U}} S \wedge T \sqsubseteq S$$

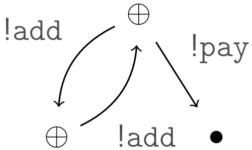
- $\sqsubseteq$ is weaker than trace inclusion
- $\sqsubseteq$ always holds for finite (closed) types

Example

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$

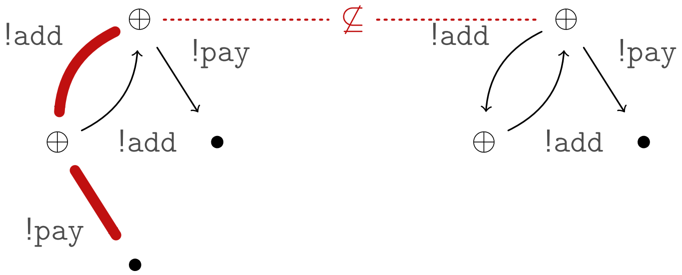
$\mu x.(!\mathtt{add}.!\mathtt{add}.x \oplus !\mathtt{pay})$





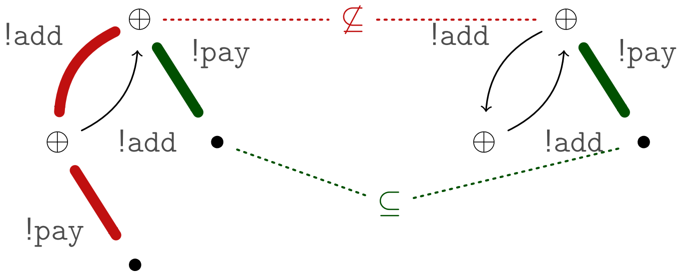
Example

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(!\mathtt{add}.!\mathtt{add}.x \oplus !\mathtt{pay})$



Example

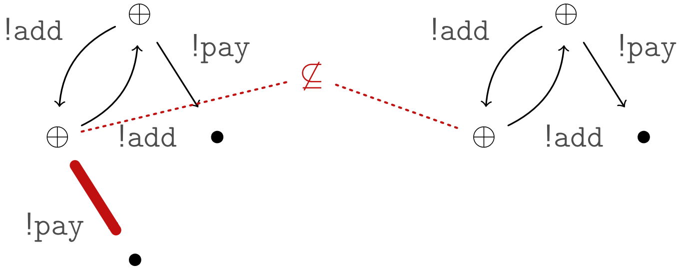
$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(!\mathtt{add}.!\mathtt{add}.x \oplus !\mathtt{pay})$



Example

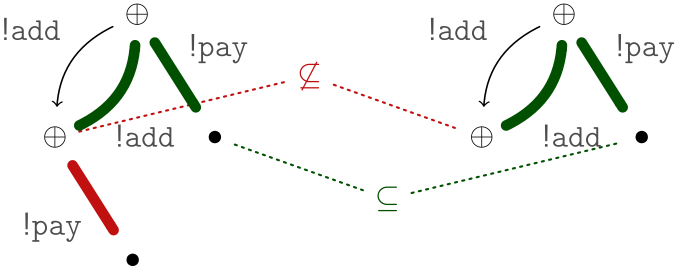
$\mu x.(\texttt{!add}.x \oplus \texttt{!pay})$

$\mu x.(\texttt{!add}.\texttt{!add}.x \oplus \texttt{!pay})$

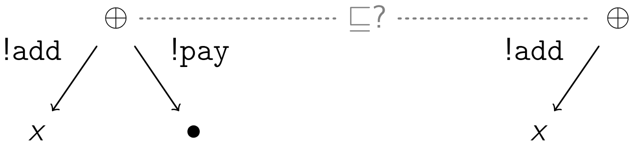


Example

$\mu x.(!\mathtt{add}.x \oplus !\mathtt{pay})$ $\mu x.(!\mathtt{add}.!\mathtt{add}.x \oplus !\mathtt{pay})$

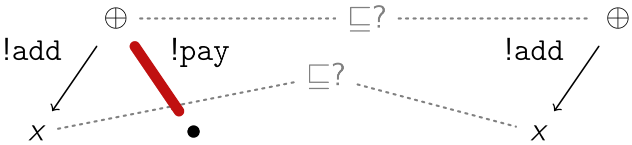


Convergence and open types



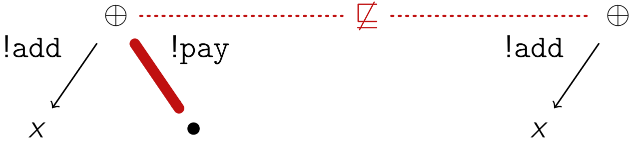
- no coinductive reasoning allowed!

Convergence and open types



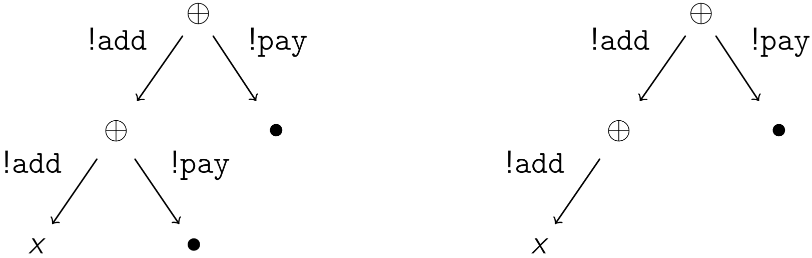
- no coinductive reasoning allowed!

Convergence and open types



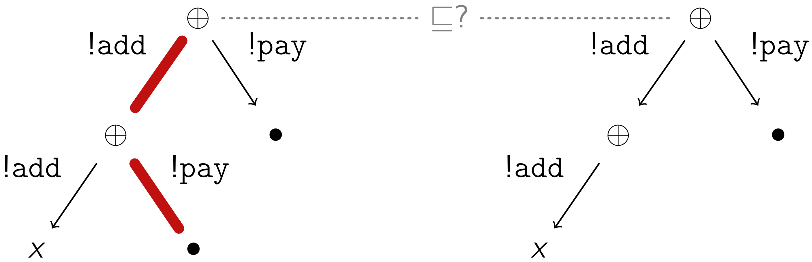
- no coinductive reasoning allowed!

Convergence and open types



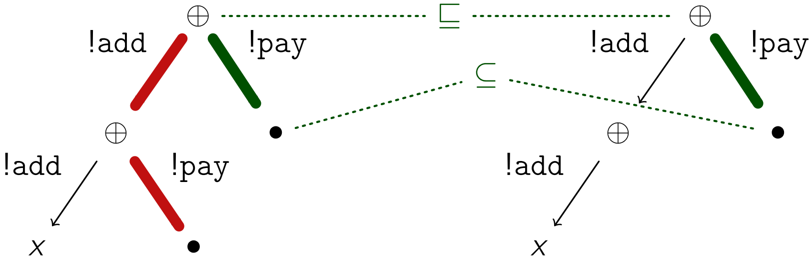
- no coinductive reasoning allowed!

Convergence and open types



- no coinductive reasoning allowed!

Convergence and open types



- no coinductive reasoning allowed!

Axioms for convergence

[c-end]
$$\mathsf{end} \sqsubseteq_X \mathsf{end}$$

[c-var]
$$x \sqsubseteq_{X \cup \{x\}} x$$

[c-rec]
$$\frac{T \sqsubseteq_{X \cup \{x\}} S}{\mu x.T \sqsubseteq_X \mu x.S}$$

[c-input]
$$\frac{\forall i \in I : T_i \sqsubseteq_X S_i}{\sum_{i \in I} ?\mathsf{a}_i.T_i \sqsubseteq_X \sum_{i \in I} ?\mathsf{a}_i.S_i}$$

[c-output 1]
$$\frac{\forall i \in I : T_i \sqsubseteq_X S_i}{\bigoplus_{i \in I} !\mathsf{a}_i.T_i \sqsubseteq_X \bigoplus_{i \in I} !\mathsf{a}_i.S_i}$$

[c-output 2]
$$\frac{\exists k \in I : T_k \sqsubseteq_\emptyset S_k}{\bigoplus_{i \in I \cup J} !\mathsf{a}_i.T_i \sqsubseteq_X \bigoplus_{i \in I} !\mathsf{a}_i.S_i}$$

Lemma

$T \sqsubseteq_{\{x\}} S$ *iff* $\mu x.T \sqsubseteq_\emptyset \mu x.S$

Axioms for fair subtyping

[f-end]

$$\text{end} \leqslant_{\mathsf{F}} \text{end}$$

[f-var]

$$x \leqslant_{\mathsf{F}} x$$

[f-rec]

$$\frac{T \leqslant_{\mathsf{F}} S \qquad T \sqsubseteq_{\{x\}} S}{\mu x.T \leqslant_{\mathsf{F}} \mu x.S}$$

[f-input]

$$\frac{\forall i \in I : T_i \leqslant_{\mathsf{F}} S_i}{\sum_{i \in I} ?\mathsf{a}_i.T_i \leqslant_{\mathsf{F}} \sum_{i \in I} ?\mathsf{a}_i.S_i}$$

[f-output]

$$\frac{\forall i \in I : T_i \leqslant_{\mathsf{F}} S_i}{\bigoplus_{i \in I \cup J} !\mathsf{a}_i.T_i \leqslant_{\mathsf{F}} \bigoplus_{i \in I} !\mathsf{a}_i.S_i}$$

Theorem (correctness & completeness)

- $T \leqslant_{\mathsf{F}} S$ *implies* $T \leqslant S$
- $T \leqslant S$ *implies* $T \approx T' \leqslant_{\mathsf{F}} S' \approx S$ *for some* T' *and* S'

Deciding fair subtyping

$$\text{[c-output 1]}\ \frac{\forall i \in I : T_i \sqsubseteq_{X} S_i}{\bigoplus_{i\in I} !\mathsf{a}_i.T_i \sqsubseteq_{X} \bigoplus_{i\in I} !\mathsf{a}_i.S_i}$$

$$\text{[c-output 2]}\ \frac{\exists k \in I : T_k \sqsubseteq_{\emptyset} S_k}{\bigoplus_{i\in I\cup J} !\mathsf{a}_i.T_i \sqsubseteq_{X} \bigoplus_{i\in I} !\mathsf{a}_i.S_i}$$

Theorem (algorithm)

- *there exists a syntax-directed axiomatization of convergence*
- *fair subtyping can be decided in $O(n^4)$ (subtyping in $O(n^2)$)*

Wrap up

- coarsest liveness-preserving subtyping for session types
- complete (co)inductive and axiomatic characterizations
- polynomial decision algorithm

Related work

- Cleaveland, Natarajan, **Divergence and fair testing**, ICALP 1995
- Rensink, Vogler, **Fair testing**, Inf. & Comp., 2007

☹ no complete axiomatization

☹ trace equivalence

☹ exponential algorithm

Ongoing and future work

Seller's dream

$$\text{if } \vdash_{\texttt{pay}} P\text{, then } P \text{ eventually pays}$$

Teacher's nightmare

$$\mu x.?\mathit{homework}.(!\mathit{FAIL}.x \oplus !\mathit{PASS})$$