

Semantic Subtyping for Session Types

Luca Padovani

Dipartimento di Informatica, Università di Torino

BTW'11

Semantic subtyping in a nutshell

- Frisch, Castagna, Benzaken, **Semantic Subtyping**, 2008

$$t \leq s \stackrel{\text{def}}{\iff} \llbracket t \rrbracket \subseteq \llbracket s \rrbracket$$

+ Intuition

$$\llbracket t \wedge s \rrbracket = \llbracket t \rrbracket \cap \llbracket s \rrbracket \qquad \llbracket t \vee s \rrbracket = \llbracket t \rrbracket \cup \llbracket s \rrbracket$$

+ Expressiveness

$$\llbracket \neg t \rrbracket = \mathcal{V} \setminus \llbracket t \rrbracket$$

+ Precision

$$t \not\leq s \quad \text{implies} \quad v \in \llbracket t \rrbracket \setminus \llbracket s \rrbracket$$

Subtyping for session types

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

end \leq_U end

$$\frac{T_i \leq_U S_i \quad (i \in I)}{\sum_{i \in I} ?a_i.T_i \leq_U \sum_{i \in I \cup J} ?a_i.S_i}$$

$$\frac{T_i \leq_U S_i \quad (i \in I)}{\bigoplus_{i \in I \cup J} !a_i.T_i \leq_U \bigoplus_{i \in I} !a_i.S_i}$$

$T \leq_U S$ means...

- it is safe to use a channel of type T where a channel of type S is expected, or...
- it is safe to use a process that behaves as S where a process that behaves as T is expected

Subtyping for session types

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

end \leq_U end

$$\frac{T_i \leq_U S_i \quad (i \in I)}{\sum_{i \in I} p?a_i.T_i \leq_U \sum_{i \in I \cup J} p?a_i.S_i}$$

$$\frac{T_i \leq_U S_i \quad (i \in I)}{\bigoplus_{i \in I \cup J} p!a_i.T_i \leq_U \bigoplus_{i \in I} p!a_i.S_i}$$

$T \leq_U S$ means...

- it is safe to use a channel of type T where a channel of type S is expected, or...
- it is safe to use a process that behaves as S where a process that behaves as T is expected

Example: multi-party session

$$\begin{array}{c} q!a \\ \curvearrowright \\ \oplus \end{array} \xrightarrow{q!b} \oplus \xrightarrow{r!c} \text{end} \quad \begin{array}{c} p?a \\ \curvearrowright \\ + \end{array} \xrightarrow{p?b} \text{end} \quad + \xrightarrow{p?c} \text{end}$$

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

Is this session “OK”?

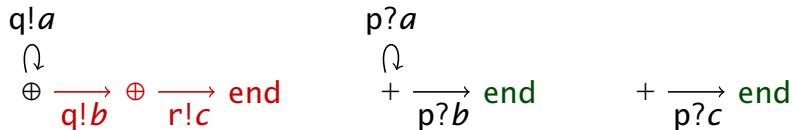
Example: multi-party session

$$\begin{array}{c} q!a \\ \curvearrowright \\ \oplus \end{array} \xrightarrow{q!b} \oplus \xrightarrow{r!c} \text{end} \quad \begin{array}{c} p?a \\ \curvearrowright \\ + \end{array} \xrightarrow{p?b} \text{end} \quad + \xrightarrow{p?c} \text{end}$$

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

Is this session “OK”? Yes, under a **fairness** assumption

Example: multi-party session (and subtyping)



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

Example: multi-party session (and subtyping)

$$q!a$$
$$\Downarrow$$
$$\oplus$$
$$p?a$$
$$\Downarrow$$
$$+ \xrightarrow{p?b} \text{end}$$
$$+ \xrightarrow{p?c} \text{end}$$

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

Is this session is “OK”?

How to fix subtyping

Definition (OK session)

- $p_1 : T_1 \mid \dots \mid p_n : T_n$ **OK** if
 - $p_1 : T_1 \mid \dots \mid p_n : T_n \Rightarrow p_1 : T'_1 \mid \dots \mid p_n : T'_n$ implies
 - $p_1 : T'_1 \mid \dots \mid p_n : T'_n \Rightarrow p_1 : \text{end} \mid \dots \mid p_n : \text{end}$

Definition (semantic subtyping)

- $\llbracket T \rrbracket = \{M \mid (p : T \mid M) \text{ is OK}\}$
- $T \leq S$ iff $\llbracket T \rrbracket \subseteq \llbracket S \rrbracket$

How to fix subtyping

Definition (OK session)

- $p_1 : T_1 \mid \dots \mid p_n : T_n$ **OK** if
 $p_1 : T_1 \mid \dots \mid p_n : T_n \Rightarrow p_1 : T'_1 \mid \dots \mid p_n : T'_n$ implies
 $p_1 : T'_1 \mid \dots \mid p_n : T'_n \Rightarrow p_1 : \text{end} \mid \dots \mid p_n : \text{end}$

Definition (semantic subtyping)

- $\llbracket T \rrbracket = \{M \mid (p : T \mid M) \text{ is OK}\}$
- $T \leq S$ iff $\llbracket T \rrbracket \subseteq \llbracket S \rrbracket$

How to fix subtyping

Definition (OK session)

- $p_1 : T_1 \mid \dots \mid p_n : T_n$ **OK** if
 - $p_1 : T_1 \mid \dots \mid p_n : T_n \Rightarrow p_1 : T'_1 \mid \dots \mid p_n : T'_n$ implies
 - $p_1 : T'_1 \mid \dots \mid p_n : T'_n \Rightarrow p_1 : \mathbf{end} \mid \dots \mid p_n : \mathbf{end}$

Definition (semantic subtyping)

- $\llbracket T \rrbracket = \{M \mid (p : T \mid M) \text{ is OK}\}$
- $T \leq S$ iff $\llbracket T \rrbracket \subseteq \llbracket S \rrbracket$

How to fix subtyping

Definition (OK session)

- $p_1 : T_1 \mid \dots \mid p_n : T_n$ **OK** if
 $p_1 : T_1 \mid \dots \mid p_n : T_n \Rightarrow p_1 : T'_1 \mid \dots \mid p_n : T'_n$ implies
 $p_1 : T'_1 \mid \dots \mid p_n : T'_n \Rightarrow p_1 : \mathbf{end} \mid \dots \mid p_n : \mathbf{end}$

Definition (semantic subtyping)

- $\llbracket T \rrbracket = \{M \mid (p : T \mid M) \text{ is } \mathbf{OK}\}$
- $T \leq S$ iff $\llbracket T \rrbracket \subseteq \llbracket S \rrbracket$

Dilemma

\leq_U versus \leq

- \leq_U is intuitive but unsound
- \leq is sound but obscure

(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
- ...

Fair testing

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

\leq_U and \leq are incomparable

$$\begin{aligned} T &= p!a.T \\ S &= q?b.S \end{aligned}$$

$$\begin{aligned} T &\leq S \\ S &\leq T \end{aligned}$$

$$\begin{aligned} T &\not\leq_U S \\ S &\not\leq_U T \end{aligned}$$

\leq_U and \leq are incomparable

$$\begin{aligned} T &= p!a.T \\ S &= q?b.S \end{aligned}$$

$$\begin{aligned} T &\leq S \\ S &\leq T \end{aligned}$$

$$\begin{aligned} T &\not\leq_U S \\ S &\not\leq_U T \end{aligned}$$

not viable $\text{fail} \leq T \leq S \leq \dots$

$\leq \sqsubset \leq_U$

viable

A normal form for session types

T is in **normal form** if either

- $T = \text{fail}$, or
- $\text{end} \in \text{trees}(S)$ for every $S \in \text{trees}(T)$

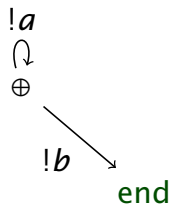
Proposition

For every T there exists $S \preceq T$ in nf

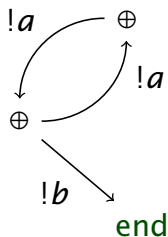
Theorem

Let $T, S \neq \text{fail}$ be in nf. Then $T \leq S$ implies $T \leq_U S$

Experiment 1



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



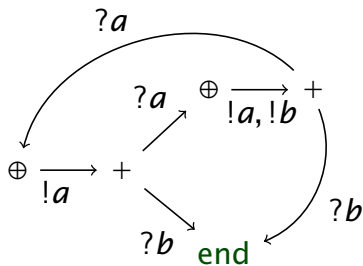
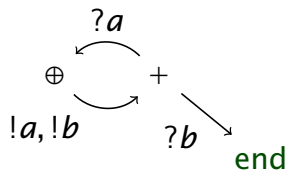
$$S = !a.!a.S \oplus !b.end$$

Is there a context R such that

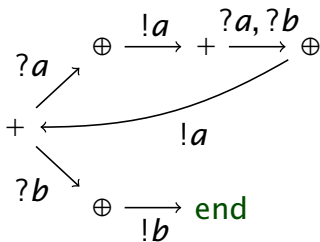
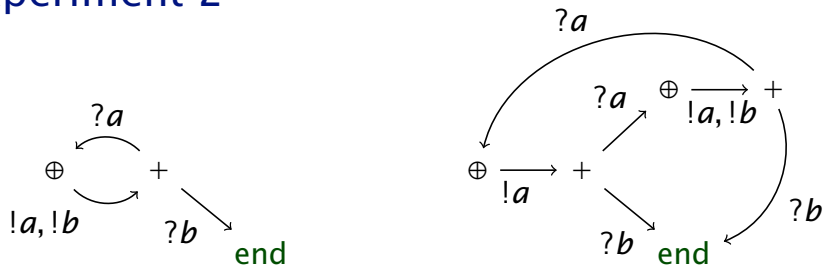
- $R | T$ is **OK**
- $R | S \Rightarrow end | end$

?

Experiment 2



Experiment 2



Rule of thumb

If

- $!a.T$ does not occur in a loop

or

- $!a.T$ occurs in a loop ℓ of p , and
- there exists an exit path in ℓ that starts from a \oplus node,

then

- $!a.T$ can be safely pruned

Rationale

- no context can rely on the eventual observation of $!a$ from p because p can **autonomously** exit ℓ

Behavioral difference

Theorem

Let T, S be in nf and $T \leq_U S$.

Then $T - S$ viable iff $R \mid T$ **OK** and $R \mid S \Rightarrow \text{end} \mid \text{end}$ for some R

$$\text{end} - \text{end} = \text{fail}$$

$$\sum_{i \in I} p?a_i.T_i - \sum_{i \in I \cup J} p?a_i.S_i = \sum_{i \in I} p?a_i.(T_i - S_i)$$

$$\bigoplus_{i \in I \cup J} p!a_i.T_i - \bigoplus_{i \in I} p!a_i.S_i = \bigoplus_{i \in I} p!a_i.(T_i - S_i) \oplus \bigoplus_{j \in J} p!a_j.T_j$$

Fair subtyping, at last

fail $\leq_A T$ end \leq_A end

$$\frac{T_i \leq_A S_i \quad (i \in I)}{\sum_{i \in I} p?a_i.T_i \leq_A \sum_{i \in I \cup J} p?a_i.S_i}$$

$$\frac{T_i \leq_A S_i \quad (i \in I) \quad \text{nf}(T - S) = \text{fail}}{T = \bigoplus_{i \in I \cup J} p!a_i.T_i \leq_A \bigoplus_{i \in I} p!a_i.S_i = S}$$

Theorem

$T \leq S$ iff $\text{nf}(T) \leq_A \text{nf}(S)$

Fair subtyping, at last

fail $\leq_A T$ **end** \leq_A **end**

$$\frac{T_i \leq_A S_i \quad (i \in I)}{\sum_{i \in I} p?a_i.T_i \leq_A \sum_{i \in I \cup J} p?a_i.S_i}$$

$$\frac{T_i \leq_A S_i \quad (i \in I) \quad \text{nf}(T - S) = \text{fail}}{T = \bigoplus_{i \in I \cup J} p!a_i.T_i \leq_A \bigoplus_{i \in I} p!a_i.S_i = S}$$

Theorem

$T \leq S$ iff $\text{nf}(T) \leq_A \text{nf}(S)$

Fair subtyping, at last

fail $\leq_A T$ **end** $\leq_A \text{end}$

$$\frac{T_i \leq_A S_i \quad (i \in I)}{\sum_{i \in I} p?a_i.T_i \leq_A \sum_{i \in I \cup J} p?a_i.S_i}$$

$$\frac{T_i \leq_A S_i \quad (i \in I) \quad \text{nf}(T - S) = \text{fail}}{T = \bigoplus_{i \in I \cup J} p!a_i.T_i \leq_A \bigoplus_{i \in I} p!a_i.S_i = S}$$

Theorem

$T \leq S$ iff $\text{nf}(T) \leq_A \text{nf}(S)$

Fair subtyping, at last

fail $\leq_A T$ **end** $\leq_A \text{end}$

$$\frac{T_i \leq_A S_i \quad (i \in I)}{\sum_{i \in I} p?a_i.T_i \leq_A \sum_{i \in I \cup J} p?a_i.S_i}$$

$$\frac{T_i \leq_A S_i \quad (i \in I) \quad \text{nf}(T - S) = \text{fail}}{T = \bigoplus_{i \in I \cup J} p!a_i.T_i \leq_A \bigoplus_{i \in I} p!a_i.S_i = S}$$

Theorem

$T \leq S$ iff $\text{nf}(T) \leq_A \text{nf}(S)$

Fair testing vs fair subtyping

Fair testing

- Cleaveland, Natarajan, **Divergence and fair testing**, 1995
- Rensink, Vogler, **Fair testing**, 2007
- denotational (= obscure) characterization
- no complete deduction system
- exponential

Fair subtyping

- + operational (= hopefully less obscure) characterization
(and maybe it can be further simplified)
- + complete deduction system
- + polynomial

More on semantic subtyping

- Padovani, **Session Types = Intersection Types + Union Types**, ITRS 2010

$$\begin{aligned} !a.T \oplus !b.S &\iff !a.T \wedge !b.S \\ ?a.T + ?b.S &\iff ?a.T \vee ?b.S \end{aligned}$$

$$?a.T \vee ?a.S \leqslant ?a.(T \vee S)$$

More on fair subtyping

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

Future work: fair type checking

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

$$\frac{\frac{u : T \vdash P}{u : !a.T \vdash u!a.P} \text{ (T-Output)} \quad T \leq !a.T}{u : T \vdash P} \text{ (T-Narrow)}$$

thank you