

Two Notions of Sub-behaviour for Session-based Client/Server Systems

Franco Barbanera¹ and Ugo de'Liguoro²

¹Dipartimento di Matematica e Informatica, Università di Catania

²Dipartimento di Informatica, Università di Torino

PPDP'10 - July 27 2010, Hagenberg

Overview

Session

A *session* is a logic unit, collecting and structuring messages exchanged among a determined set of agents, sharing a private channel to prevent interference by third parties.

- *Session types* have been introduced to formalise *two-sided* sessions in type systems for the π -calculus

We set up a behavioural semantic investigation of session types using the notion of **contract**.

- *Contracts* are a process algebraic formalism to describe the behaviour of services in a client/server scenario

Session Types (Honda, Vasconcelos, Kubo)

Session types = regular trees of ordinary types
of (polyadic) π -calculus

If $\Gamma \vdash P$ is derivable and

$$\Gamma(x) = \mu X. \text{ ?}(\mathbf{Int}).\&\langle \ell_0 : ![\mathbf{Bool}]\text{end}, \\ \ell_1 : \oplus\langle \ell_2 : \text{end}, \\ \ell_3 : X \rangle \rangle$$

then channel x is used in P to carry the following “session”:

- ① input an integer
- ② on receiving the message ℓ_0 send a boolean then stop
- ③ on receiving ℓ_1 either issue ℓ_2 then stop, or issue ℓ_3 and start over the whole session

Session Types (Honda, Vasconcelos, Kubo)

The syntax:

T	::=	Int Bool ... S	ground/session type
S	::=	end	ended session
		$?(T)S$	input of type T , then S
		$![T]S$	output of type T , then S
		$\&\langle \ell_i : S_i \mid i \in I \rangle$	branching (I finite)
		$\oplus\langle \ell_i : S_i \mid i \in I \rangle$	selection (I finite)
		X	variable
		$\mu X. S$	recursion (S not a variable)

If T is restricted to ground types, these are *first order* session types; they are *higher-order* otherwise.

Session Types (Honda, Vasconcelos, Kubo)

The “duality” relation over session types:

$$\begin{array}{lcl}
 \overline{\text{end}} & = & \text{end} \\
 \overline{?(T)S} & = & ![T]\overline{S} \\
 \overline{\&\langle \ell_i : S_i \mid i \in I \rangle} & = & \oplus \langle \ell_i : \overline{S}_i \mid i \in I \rangle \\
 \overline{\oplus \langle \ell_i : S_i \mid i \in I \rangle} & = & \&\langle \ell_i : \overline{S}_i \mid i \in I \rangle \\
 \overline{X} & = & X \\
 \overline{\mu X. S} & = & \mu X. \overline{S}
 \end{array}$$

The following rule is at the hearth of *error freeness* property within a typeable session:

$$\frac{\Delta, x : S \vdash P \quad \Delta, x : \overline{S} \vdash Q}{\Delta \vdash (\nu x)(P|Q)}$$

Subtyping Session Types (Gay-Hole)

Subtyping intuition

$A <: B$ if and only if any channel that satisfies the stricter “protocol” A also satisfies the protocol B

The $A <: B$ relation has been axiomatized by Gay and Hole.

They proved it *operationally* sound by showing that the *narrowing* rule:

$$\frac{\Delta, x : B \vdash P \quad A <: B}{\Delta, x : A \vdash P}$$

doesn't break subject reduction.

Note that subsumption rule is just the dual of *subsumption* rule of the λ -calculus with subtyping.

Coinductive Axiomatization of FO-Subtyping

A coinductive reformulation: let $\Gamma = \{A_1 <: B_1, \dots, A_k <: B_k\}$, then we derive judgements of the form $\Gamma \vdash A <: B$ by the rules:

$$\frac{}{\Gamma \vdash \mu X.A <: A\{\mu X.A/X\}} \quad \frac{}{\Gamma \vdash A\{\mu X.A/X\} <: \mu X.A}$$

$$\frac{\Gamma, \&_{i \in I} \langle \ell_i : A_i \rangle <: \&_{j \in J} \langle \ell_j : B_j \rangle \vdash A_i <: B_i \quad \forall i \in I \quad I \subseteq J}{\Gamma \vdash \&_{i \in I} \langle \ell_i : A_i \rangle <: \&_{j \in J} \langle \ell_j : B_j \rangle}$$

$$\frac{\Gamma, \oplus_{i \in I} \langle \ell_i : A_i \rangle <: \oplus_{j \in J} \langle \ell_j : B_j \rangle \vdash A_j <: B_j \quad \forall j \in J \quad I \supseteq J}{\Gamma \vdash \oplus_{i \in I} \langle \ell_i : A_i \rangle <: \oplus_{j \in J} \langle \ell_j : B_j \rangle}$$

Behavioural semantics of session types

Problem

Is there a semantic characterization of session subtyping?

Answer: behavioural semantics

- provide a formal definition of protocols as **behaviours**
- give a concept of **sub-behaviour**
- interpret session types as behaviours

We understand behaviours as a suitable kind of processes, for which we choose **contracts**

Contracts (Castagna, Laneve, Padovani)

- **contracts** are abstract specifications of web-services (and of client queries)
- central is the **compliance** relation among a client query and a server contract:

ρ **complies** with τ ($\rho \dashv \tau$, ρ is a client for σ)



every request from ρ is satisfied by σ

- compliance induces a **subcontract** relation:

σ is a **subcontract** of τ ($\sigma \preceq \tau$) \Leftrightarrow every client of σ is such of τ

Contracts (Castagna, Laneve, Padovani)

Web *contracts* are parallel-free CCS terms (without τ) generated by the grammar:

$$\sigma ::= \mathbf{1} \mid \alpha.\sigma \mid \sigma + \rho \mid \sigma \oplus \rho \mid x \mid \text{rec } x.\sigma$$

where $\alpha \in \mathcal{N} \cup \overline{\mathcal{N}}$.

Semantics is defined by the LTS:

- $\alpha.\sigma \xrightarrow{\alpha} \sigma$
- $\sigma \xrightarrow{\alpha} \sigma' \Rightarrow \sigma + \rho \xrightarrow{\alpha} \sigma', \rho + \sigma \xrightarrow{\alpha} \sigma'$
- $\sigma \oplus \rho \longrightarrow \sigma, \sigma \oplus \rho \longrightarrow \rho$
- $\text{rec } x.\sigma \longrightarrow \sigma\{\text{rec } x.\sigma/x\}$

Example

The contract of a ballot service might be:

$$\text{rec } x.\text{Login}.\left(\overline{\text{Wrong}}.x \oplus \overline{0k}.\left(\text{VoteA}.\left(\text{Va1}+\text{Va2}\right)+\text{VoteB}.\left(\text{Vb1}+\text{Vb2}\right)\right)\right)$$

Example

The contract of a ballot service might be:

```
rec x.Login.(\overline{Wrong}.x \oplus \overline{Ok}.(VoteA.(Va1+Va2)+VoteB.(Vb1+Vb2)))
```

meaning:

- wait for a Login action
- acknowledge the (in)correctness of login

Example

The contract of a ballot service might be:

```
rec x.Login.(\overline{Wrong}.x \oplus \overline{Ok}.(VoteA.(Va1+Va2)+VoteB.(Vb1+Vb2)))
```

meaning:

- wait for a Login action
- acknowledge the (in)correctness of login
- in the negative restart

Example

The contract of a ballot service might be:

$$\text{rec } x.\text{Login}.\overline{(\text{Wrong}.x \oplus \overline{0k}.\text{VoteA}.\text{(Va1 + Va2)} + \text{VoteB}.\text{(Vb1 + Vb2))})}$$

meaning:

- wait for a Login action
- acknowledge the (in)correctness of login
- in the negative restart
- in the positive prompt for voting either A or B
- then offer the possibility for voting for a ticket

Example

The contract of a ballot service might be:

$$\text{rec } x.\text{Login}.\overline{(\text{Wrong}.x \oplus \overline{0}\text{k}.\text{VoteA}.\text{(Va1+Va2)+VoteB}.\text{(Vb1 + Vb2))})}$$

meaning:

- wait for a Login action
- acknowledge the (in)correctness of login
- in the negative restart
- in the positive prompt for voting either A or B
- then offer the possibility for voting for a ticket

Session Behaviours as Contracts interpreting Session Types

Consider the mapping from (first order) session types to contracts:

$$\llbracket X \rrbracket = x$$

$$\llbracket \text{end} \rrbracket = \mathbf{1} \qquad \llbracket \mu X. A \rrbracket = \text{rec } x. \llbracket A \rrbracket$$

$$\llbracket ?(\gamma)A \rrbracket = \gamma. \llbracket A \rrbracket \qquad \llbracket ![\gamma] A \rrbracket = \bar{\gamma}. \llbracket A \rrbracket$$

$$\llbracket \&\langle l_i : B_i \mid i \in I \rangle \rrbracket = \sum_{i \in I} l_i. \llbracket B_i \rrbracket$$

$$\llbracket \oplus\langle l_i : B_i \mid i \in I \rangle \rrbracket = \bigoplus_{i \in I} \bar{l}_i. \llbracket B_i \rrbracket$$

The image of the $\llbracket \cdot \rrbracket$ map is a subset of the set of contracts.

Session Behaviours: the grammar

\mathcal{S} (*Session Behaviours*) are the closed expressions among those defined by the grammar:

σ	::=	1	
		$a_1.\sigma_1 + \cdots + a_n.\sigma_n$	external choice, a_i distinct
		$\bar{a}_1.\sigma_1 \oplus \cdots \oplus \bar{a}_n.\sigma_n$	internal choice, \bar{a}_i distinct
		x	variable
		$\text{rec } x.\sigma$	recursion, σ not a variable

Contracts describe the overall behaviour of a (client)server.

Session Behaviors describe the possible interactions of a process over a channel.

Compliance and Orthogonality

Extend the reduction relation to pairs of session-behaviours $\rho \parallel \sigma$:

$$\frac{\rho \xrightarrow{\alpha} \rho' \quad \sigma \xrightarrow{\bar{\alpha}} \sigma'}{\rho \parallel \sigma \longrightarrow \rho' \parallel \sigma'} \quad \frac{\rho \longrightarrow \rho'}{\rho \parallel \sigma \longrightarrow \rho' \parallel \sigma} \quad \frac{\sigma \longrightarrow \sigma'}{\rho \parallel \sigma \longrightarrow \rho \parallel \sigma'}$$

Compliance: the *client* ρ *complies* with the *server* σ , $\rho \dashv \sigma$ if

$$\forall \rho', \sigma' \quad \rho \parallel \sigma \xrightarrow{*} \rho' \parallel \sigma' \not\rightarrow \Rightarrow \rho' = \mathbf{1}$$

i.e. *any request of the client is eventually satisfied by the server.*

Orthogonality:

$$\rho \perp \sigma \Leftrightarrow \rho \dashv \sigma \ \& \ \sigma \dashv \rho$$

Examples

$\bar{a} \oplus \bar{b} \dashv a + b + c$ because:

$$\begin{array}{l} \bar{a} \oplus \bar{b} \| a + b + c \longrightarrow \bar{a} \| a + b + c \longrightarrow \mathbf{1} \| \mathbf{1} \\ \quad \searrow \\ \quad \quad \bar{b} \| a + b + c \longrightarrow \mathbf{1} \| \mathbf{1} \end{array}$$

and also $a + b + c \dashv \bar{a} \oplus \bar{b}$ hence $\bar{a} \oplus \bar{b} \perp a + b + c$.

But $\bar{a} \oplus \bar{b} \oplus \bar{c} \not\vdash a + b$ (and $a + b \not\vdash \bar{a} \oplus \bar{b} \oplus \bar{c}$) since:

$$\bar{a} \oplus \bar{b} \oplus \bar{c} \| a + b \longrightarrow \bar{c} \| a + b \not\rightarrow$$

Note that $\text{rec } x.a.x \dashv \text{rec } x.\bar{a}.x$ (without reaching $\mathbf{1} \| \dots$) since:

$$\begin{array}{l} \text{rec } x.a.x \| \text{rec } x.\bar{a}.x \xrightarrow{2} a.\text{rec } x.a.x \| \bar{a}.\text{rec } x.\bar{a}.x \\ \longrightarrow \text{rec } x.a.x \| \text{rec } x.\bar{a}.x \longrightarrow \dots \end{array}$$

Client/Server Sub-Behaviours

For $\sigma, \rho \in \mathcal{S}$, let

$$\text{Client}(\sigma) = \{\rho \in \mathcal{S} \mid \rho \dashv \sigma\}, \quad \text{Server}(\rho) = \{\sigma \in \mathcal{S} \mid \rho \dashv \sigma\}$$

Then define the relations:

- 1 $\sigma \preceq_s \sigma'$ if and only if $\text{Client}(\sigma) \subseteq \text{Client}(\sigma')$;
- 2 $\rho \preceq_c \rho'$ if and only if $\text{Server}(\rho) \subseteq \text{Server}(\rho')$.

In words: $\sigma \preceq_s \sigma'$ if the server σ' has a larger set of clients than σ , and similarly for $\rho \preceq_c \rho'$.

Note. Our \preceq_s is essentially the subcontract relation by Castagna et alii.

Duality in \mathcal{S}

Let us extend the $\bar{\cdot}$ operation to all (also open) behaviours:

- $\bar{\mathbf{1}} = \mathbf{1}$
- $\overline{a.\sigma} = \bar{a}.\bar{\sigma}$ and $\overline{\bar{a}.\sigma} = a.\bar{\sigma}$
- $\overline{\sigma + \tau} = \bar{\sigma} \oplus \bar{\tau}$
- $\overline{\sigma \oplus \tau} = \bar{\sigma} + \bar{\tau}$
- $\bar{x} = x$
- $\overline{\text{rec } x.\sigma} = \text{rec } x.\bar{\sigma}$

If $\sigma \in \mathcal{S}$ then $\bar{\sigma} \in \mathcal{S}$, and $\overline{\bar{\sigma}} = \sigma$. Moreover:

$$\sigma = \llbracket A \rrbracket \quad \text{if and only if} \quad \bar{\sigma} = \llbracket \bar{A} \rrbracket$$

Duality in \mathcal{S}

A relation exists between the syntactic operator $\bar{\cdot}$ and the server/client preorders:

Proposition. Let $\tau \in \mathcal{S}$:

- 1 $\bar{\tau}$ is the minimum server among those of τ :

$$\forall \sigma \in \text{Server}(\tau). \bar{\tau} \preceq_s \sigma$$

- 2 $\bar{\tau}$ is the minimum client among those of τ :

$$\forall \rho \in \text{Client}(\tau). \bar{\tau} \preceq_c \rho$$

This does not hold outside of \mathcal{S} :

- $\bar{a} \oplus \bar{a}.\bar{b} \not\preceq a + a.b$
- the minimum of $\text{Client}(a + a.b)$ is actually \bar{a}
- $a + a.b \not\preceq \bar{a} \oplus \bar{a}.\bar{b}$
- the minimum of $\text{Server}(a + a.b)$ is $\bar{a}.\bar{b}$
- $\text{Server}(a.\bar{b} + a.\bar{c}) = \emptyset$

Behavioural Subtyping

Let $A^\perp = \{\sigma \in \mathcal{S} \mid \exists \tau \in A. \sigma \perp \tau\}$ and $\sigma^\perp = \{\sigma\}^\perp$:

$$\sigma \preceq: \tau \iff \sigma^\perp \subseteq \tau^\perp$$

Theorem

Behavioural subtyping is the intersection of both client and server-subbehaviour relations:

$$\preceq: = \preceq_c \cap \preceq_s$$

It follows that for any $\sigma, \tau \in \mathcal{S}$, $\bar{\sigma}$ is minimal in σ^\perp w.r.t. $\preceq:$ and

$$\sigma \preceq: \tau \text{ if and only if } \bar{\tau} \preceq: \bar{\sigma}$$

matching with the fact that $A <: B \iff \bar{B} <: \bar{A}$.

Higher-Order LTS

Higher-order Behaviours add input/output of behaviors to prefixes:

$$\sigma, \tau ::= \dots \mid ?\sigma^p.\tau \mid !\sigma^p.\tau$$

where $p \in \{s, c\}$.

The higher-order LTS:

$$\frac{\sigma \xrightarrow{?\rho_2^p} \sigma' \quad \tau \xrightarrow{!\rho_1^p} \tau' \quad \rho_1 \preceq_p \rho_2}{\sigma \parallel \tau \longrightarrow \sigma' \parallel \tau'} \quad \frac{\sigma \xrightarrow{!\rho_1^p} \sigma' \quad \tau \xrightarrow{?\rho_2^p} \tau' \quad \rho_1 \preceq_p \rho_2}{\sigma \parallel \tau \longrightarrow \sigma' \parallel \tau'}$$

Note the use of \preceq_s, \preceq_c in the LTS rules.

The syntactical duality extends as:

$$\overline{?\sigma^p.\tau} = !\sigma^p.\bar{\tau}, \quad \overline{!\sigma^p.\tau} = ?\sigma^p.\bar{\tau}$$

Interpreting Higher-Order Sessions

Higher-order session may send and receive session types:

$$A, B, ::= \dots \mid ?(A^p)B \mid ![A^p]B \quad \text{for } p = c, s$$

By considering higher-order behaviours we can extend the interpretation map to higher order session types straightforwardly:

$$\llbracket ?(A^p)B \rrbracket = ?\llbracket A \rrbracket^p \llbracket B \rrbracket, \quad \llbracket ![A^p]B \rrbracket = !\llbracket A \rrbracket^p \llbracket B \rrbracket$$

Note. We have studied asymmetric session-types, with polarized channels to record either client or server role, elsewhere: see [Barbanera-Capocchi-de'Liguoro, Proc. of FSEN'09].

Subtyping Higher-Order Sessions

We decorate the sent/received session by a polarity:

$$A, B, ::= \dots \mid ?(A^p)B \mid ![A^p]B \text{ for } p = c, s.$$

Then consider the (coinductive versions of) the Gay-Hole rules:

$$\frac{\Gamma, ?(A^p)B <: ?(C^p)D \vdash A <: C, B <: D}{\Gamma \vdash ?(A^p)B <: ?(C^p)D}$$

$$\frac{\Gamma, ![A^p]B <: ![C^p]D \vdash C <: A, B <: D}{\Gamma \vdash ![A^p]B <: ![C^p]D}$$

Fact $A <: B$ (according to Gay-Hole) if and only if $\vdash A <: B$

The Soundness Theorem

Main Theorem

Define:

- 1 $\models A <: B$ iff $\llbracket A \rrbracket \preceq \llbracket B \rrbracket$
- 2 $\models \Gamma$ iff $\models C <: D$ for all $C <: D \in \Gamma$
- 3 $\Gamma \models A <: B$ iff $\models \Gamma$ implies $\models A <: B$

then

$$\Gamma \vdash A <: B \Rightarrow \Gamma \models A <: B$$

Conjecture. Completeness holds, hence known decision algorithms for session subtyping are useful also for the subcontract relation in case of session behaviours.

Final Remarks

Results and conjectures:

- we have proposed an interpretation of session types into behaviours which is sound w.r.t. Gay-Hole subtyping
- we conjecture that the interpretation is actually complete
- when restricting to \mathcal{S} , there is no theoretical loss w.r.t. the full set of contracts **in the case of two-ended sessions**

Further work:

- things are different when considering multiparty sessions and fairness concepts are involved
- the power of higher-order LTS in giving semantics to the typed π -calculus deserves further attention