Sub-typing and sub-behaviour relations

Franco Barbanera[1] and Ugo de'Liguoro[2]

[1]Dipartimento di Matematica e Informatica, Università di Catania
[2]Dipartimento di Informatica, Università di Torino

WSBT - 20th April 2011, Lisboa

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

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

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

where the T in $?(T)$, $![T]$ has to be closed (a restriction w.r.t. [HVK] and [GH] session types).

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:

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

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

Coinductive Axiomatization of FO-Subtyping

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

$$\frac{\Gamma \vdash A\{\mu X.A/X\} \leq_p B}{\Gamma \vdash \mu X.A \leq_p B} \qquad \frac{\Gamma \vdash B \leq_p A\{\mu X.A/X\}}{\Gamma \vdash B \leq_p \mu X.A}$$

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

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

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:

$$\rho \text{ complies with } \tau \quad (\rho \dashv \tau, \quad \rho \text{ is a client for } \sigma)$$

$$\Updownarrow$$

$$\text{every request from } \rho \text{ is satisfied by } \sigma$$

- 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:

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

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

Semantics is defined by the LTS:

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

Example

The contract of a ballot service might be:

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

Example

The contract of a ballot service might be:

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

meaning:

- wait for a `Login` action

Example

The contract of a ballot service might be:

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

meaning:

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

Example

The contract of a ballot service might be:

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

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:

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

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`

Example

The contract of a ballot service might be:

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

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:

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

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:

$$
\begin{array}{rclrcl}
[\![X]\!] & = & x & & & \\
[\![\mathsf{end}]\!] & = & \mathbf{1} & [\![\mu X.\, A]\!] & = & \mathsf{rec}\, x.\, [\![A]\!] \\
[\![\, ?(\gamma) A]\!] & = & \gamma.[\![A]\!] & [\![\, ![\gamma] A]\!] & = & \overline{\gamma}.[\![A]\!]
\end{array}
$$

$$
\begin{array}{rcl}
[\![\&\langle \ell_i : B_i \mid i \in I \rangle]\!] & = & \sum_{i \in I} \ell_i.[\![B_i]\!] \\
[\![\oplus\langle \ell_i : B_i \mid i \in I \rangle]\!] & = & \bigoplus_{i \in I} \overline{\ell}_i.[\![B_i]\!]
\end{array}
$$

The image of the $[\![\cdot]\!]$ map is a subset of the set of contracts.

Session Behaviours: the grammar

Session Behaviours in $\mathcal{S}$ are the closed expressions defined by the grammar:

$$
\begin{array}{rcll}
\sigma & ::= & \mathbf{1} & \\
& | & a_1.\sigma_1 + \cdots + a_n.\sigma_n & \text{external choice, } a_i \text{ distinct} \\
& | & \overline{a}_1.\sigma_1 \oplus \cdots \oplus \overline{a}_n.\sigma_n & \text{internal choice, } \overline{a}_i \text{ distinct} \\
& | & x & \text{variable} \\
& | & \mathsf{rec}\, x.\sigma & \text{recursion, } \sigma \text{ not a variable}
\end{array}
$$

Contracts describe the overall behaviour of a client or a 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 $\rho \| \sigma$:

$$\frac{\rho \xrightarrow{\alpha} \rho' \quad \sigma \xrightarrow{\overline{\alpha}} \sigma'}{\rho \| \sigma \longrightarrow \rho' \| \sigma'} \qquad \frac{\rho \longrightarrow \rho'}{\rho \| \sigma \longrightarrow \rho' \| \sigma} \qquad \frac{\sigma \longrightarrow \sigma'}{\rho \| \sigma \longrightarrow \rho \| \sigma'}$$

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

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

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

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

$$\begin{array}{cccccc}
(\overline{a} \oplus \overline{b}) \| (a + b + c) & \longrightarrow & \overline{a} \| (a + b + c) & \longrightarrow & \mathbf{1} \| \mathbf{1} \\
& \searrow & & & \\
& & \overline{b} \| (a + b + c) & \longrightarrow & \mathbf{1} \| \mathbf{1}
\end{array}$$

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

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

$$(\overline{a} \oplus \overline{b} \oplus \overline{c}) \| (a + b) \longrightarrow \overline{c} \| (a + b) \not\longrightarrow$$

Note that $\mathrm{rec}\, x.a.x \dashv \mathrm{rec}\, x.\overline{a}.x$ (without reaching $\mathbf{1} \| \cdots$) since:

$$\begin{array}{rcl}
\mathrm{rec}\, x.a.x \| \mathrm{rec}\, x.\overline{a}.x & \xrightarrow{2} & a.\mathrm{rec}\, x.a.x \| \overline{a}.\mathrm{rec}\, x.\overline{a}.x \\
& \longrightarrow & \mathrm{rec}\, x.a.x \| \mathrm{rec}\, x.\overline{a}.x \longrightarrow \cdots
\end{array}$$

Client/Server Sub-Behaviours

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

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

Then define the relations:

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

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 $\overline{\cdot}$ operation to all (also open) behaviours:

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

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

$$\sigma = [\![A]\!] \quad \text{if and only if} \quad \overline{\sigma} = [\![\overline{A}]\!]$$

Duality in $\mathcal{S}$

Relating the syntactic operator $\bar{\cdot}$ to the server/client preorders:

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

1. $\overline{\tau}$ is the minimum server among those of τ:

   $$\forall \sigma \in \text{Server}(\tau).\ \ \overline{\tau} \preceq_s \sigma \ \ (\text{i.e. } \text{Client}(\overline{\tau}) \subseteq \text{Client}(\sigma))$$

2. $\overline{\tau}$ is the minimum client among those of τ:

   $$\forall \rho \in \text{Client}(\tau).\ \ \overline{\tau} \preceq_c \rho \ \ (\text{i.e. } \text{Server}(\overline{\tau}) \subseteq \text{Server}(\rho))$$

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

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

Behavioural Subtyping

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

$$\sigma \preceq: \tau \quad \stackrel{\Delta}{\Leftrightarrow} \quad \sigma^{\perp} \subseteq \tau^{\perp}$$

Theorem

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

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

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

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

matching with the fact that $A <: B \Leftrightarrow \overline{B} <: \overline{A}$.

Higher-Order LTS

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

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

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

The higher-order LTS:

$$?\rho^p.\sigma \xrightarrow{?\rho^p} \sigma \qquad\qquad !\rho^p.\sigma \xrightarrow{!\rho^p} \sigma$$

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

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

The syntactical duality extends as:

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

Interpreting Higher-Order Sessions

Higher-order session may send and receive session types:

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

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

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

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

Subtyping Higher-Order Sessions

We decorate the sent/received session by a polarity:

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

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 $\vdash A <: B$

Results

Main Theorem

Define:

1. $\models A <: B$ iff $[\![A]\!] \preceq: [\![B]\!]$
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 (soundness)

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

Completeness also holds:

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

Final Remarks

Results:

- we have proposed an interpretation of session types into behaviours which is sound w.r.t. Gay-Hole subtyping
- we also have that the interpretation is 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