The *must* preorder revisited

An algebraic theory for Web services contracts

Cosimo Laneve[1]   Luca Padovani[2]

[1]University of Bologna

[2]University of Urbino

CONCUR 2007

Web services in a nutshell

- distributed processes
- communicating through standard Web protocols (tcp, http, soap)
- exchanging data in platform-neutral format (xml)
- dynamically linked
- with machine-understandable descriptions

Technologies for Web services

Interface descriptions

- `WSDL` 1.1 (W3C note, 2001)
- `WSDL` 2.0 (W3C recommendation, 2007)

Behavioral descriptions

- `WSCL` 1.0 (W3C note, 2002)
- `WSCI` 1.0 (W3C note, 2002)
- `WS – BPEL` 2.0 (OASIS standard, 2007)

  "Enabling users to describe business process activities as Web services and define how they can be connected to accomplish specific tasks"

Web services yellow pages (*registries*)

- `UDDI` 3.0.2 (OASIS standard, 2004)

  "Defining a standard method for enterprises to dynamically discover and invoke Web services"

Technologies for Web services

Interface descriptions

- `WSDL` 1.1 (W3C note, 2001)
- `WSDL` 2.0 (W3C recommendation, 2007)

Behavioral descriptions

- `WSCL` 1.0 (W3C note, 2002)
- `WSCI` 1.0 (W3C note, 2002)
- `WS – BPEL` 2.0 (OASIS standard, 2007)

  "Enabling users to describe business process activities as Web services and define how they can be connected to accomplish specific tasks"

Web services yellow pages (*registries*)

- `UDDI` 3.0.2 (OASIS standard, 2004)

  "Defining a standard method for enterprises to dynamically discover and invoke Web services"

Technologies for Web services

Interface descriptions

- `WSDL` 1.1 (W3C note, 2001)
- `WSDL` 2.0 (W3C recommendation, 2007)

Behavioral descriptions

- `WSCL` 1.0 (W3C note, 2002)
- `WSCI` 1.0 (W3C note, 2002)
- `WS` – `BPEL` 2.0 (OASIS standard, 2007)

  "Enabling users to describe business process activities as Web services and define how they can be connected to accomplish specific tasks"

Web services yellow pages (*registries*)

- `UDDI` 3.0.2 (OASIS standard, 2004)

  "Defining a standard method for enterprises to dynamically discover and invoke Web services"

Discovering Web services

Search key

- name
- industrial classification
- location
- ...
- **behavioral type!**

Problem

We need a *formal* notion of behavioral equivalence which

- preserves client satisfaction
- is abstract (based on the described, observable behavior)

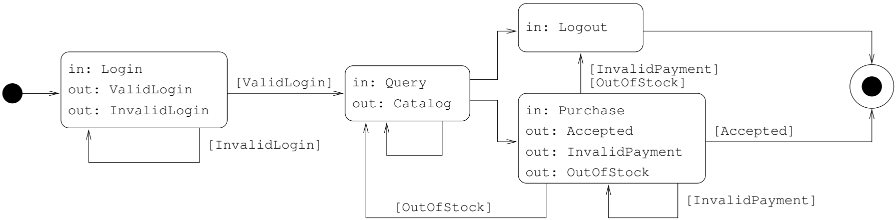
Plan

Synthesize *contracts* from Web service descriptions, give contracts a formal semantics, use contracts for searching (and possibly more...)

Summary

1. understand what contracts look like
2. define client satisfaction (*compliance*)
3. define contract equivalence (*subcontract*)
4. see how to query a registry (*duality*)
5. study the properties enjoyed by the equivalence w.r.t. common Web service scenarios (*choreographies*)

What's in a contract?



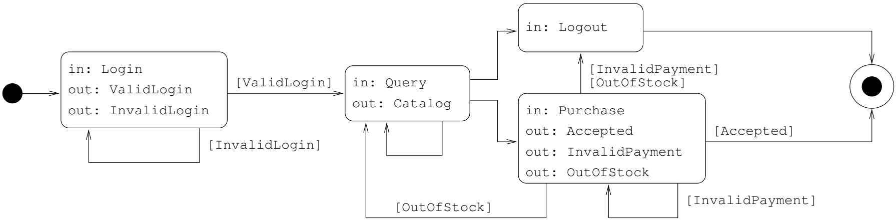
- the behavior of the service at Login is

$$\sigma = \mathtt{Login}.(\overline{\mathtt{InvalidLogin}}.\sigma \oplus \overline{\mathtt{ValidLogin}}.\sigma')$$

- the behavior of the service at Query is

$$\tau = \mathtt{Query}.\overline{\mathtt{Catalog}}.(\tau + \mathtt{Logout} + \mathtt{Purchase}.\tau')$$

What's in a contract?



- the behavior of the service at $\texttt{Login}$ is

$$\sigma = \texttt{Login}.(\overline{\texttt{InvalidLogin}}.\sigma \oplus \overline{\texttt{ValidLogin}}.\sigma')$$

- the behavior of the service at $\texttt{Query}$ is

$$\tau = \texttt{Query}.\overline{\texttt{Catalog}}.(\tau + \texttt{Logout} + \texttt{Purchase}.\tau')$$

A language for contracts – ccs without τ's

Contracts are pairs $I : \sigma$

- the *interface* I is a finite set of *names*
- the *behavior* σ is defined by the grammar ($\alpha = a, \overline{a}$):

$$
\begin{array}{lll}
\sigma ::= & & \\
& 0 & \text{(end of connection)} \\
& \alpha.\sigma & \text{(action prefix)} \\
& \sigma + \sigma & \text{(external choice)} \\
& \sigma \oplus \sigma & \text{(internal choice)} \\
& x & \text{(variable)} \\
& \text{rec } x.\sigma & \text{(recursion)}
\end{array}
$$

Example

$$
\begin{array}{l}
\text{rec } x.\texttt{Login}.(\overline{\texttt{InvalidLogin}}.x \oplus \overline{\texttt{ValidLogin}}.\text{rec } y. \\
\quad \texttt{Query}.\overline{\texttt{Catalog}}.(y + \texttt{Logout} + \text{rec } z.\texttt{Purchase}. \\
\qquad \overline{\texttt{Accepted}} \oplus \overline{\texttt{InvalidPayment}}.(z + \texttt{Logout}) \oplus \overline{\texttt{OutOfStock}}.(y + \texttt{Logout})))
\end{array}
$$

Behavior transition relation

$$\alpha.\sigma \xrightarrow{\alpha} \sigma \qquad \frac{\sigma \xrightarrow{\alpha} \sigma'}{\sigma + \tau \xrightarrow{\alpha} \sigma'}$$

$$\sigma \oplus \tau \longrightarrow \sigma \qquad \frac{\sigma \longrightarrow \sigma'}{\sigma + \tau \longrightarrow \sigma' + \tau} \qquad \mathsf{rec}\ x.\sigma \longrightarrow \sigma\{^{\mathsf{rec}\ x.\sigma}/_{x}\}$$

Standard notation

$\Longrightarrow$, $\quad \sigma \stackrel{\alpha}{\Longrightarrow} \sigma'$, $\quad \sigma\uparrow$, $\quad \sigma\downarrow$, $\quad \mathtt{init}(\sigma) = \{\alpha \mid \sigma \stackrel{\alpha}{\Longrightarrow}\}$

Remark

Same transition relation as the one of ccs without τ's

- $a + (b \oplus c) \longrightarrow a + b$
- $(a + b) \oplus (a + c) \longrightarrow a + b$

Compliance = graceful termination

Client/service interaction

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

Compliance (e indicates client's satisfaction)

ρ is *compliant with* σ ($\rho \dashv \sigma$) if $\rho \mid \sigma \Longrightarrow \rho' \mid \sigma'$ implies

1. if $\rho' \mid \sigma' \not\longrightarrow$, then $\{\mathsf{e}\} \subseteq \mathtt{init}(\rho')$
2. if $\sigma' \uparrow$, then $\{\mathsf{e}\} = \mathtt{init}(\rho')$

Examples

- $a.\mathsf{e} + b.\mathsf{e} \dashv \overline{a} \oplus \overline{b}$ and $a.\mathsf{e} + b.\mathsf{e} \dashv \overline{a}$
- $a.\mathsf{e} \oplus b.\mathsf{e} \not\dashv \overline{a} \oplus \overline{b}$
- $\mathsf{e} + \overline{a}.\mathsf{e} \dashv 0$ and $\mathsf{e} + \overline{a}.\mathsf{e} \not\dashv {}^{\cdot}$ $\quad$ $({}^{\cdot} = \mathsf{rec}\ x.x)$

Compliance = graceful termination

Client/service interaction

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

Compliance (e indicates client's satisfaction)

ρ is *compliant with* σ ($\rho \dashv \sigma$) if $\rho \mid \sigma \Longrightarrow \rho' \mid \sigma'$ implies

1. if $\rho' \mid \sigma' \not\longrightarrow$, then $\{\mathsf{e}\} \subseteq \mathtt{init}(\rho')$
2. if $\sigma' \uparrow$, then $\{\mathsf{e}\} = \mathtt{init}(\rho')$

Examples

- $a.\mathsf{e} + b.\mathsf{e} \dashv \overline{a} \oplus \overline{b}$ and $a.\mathsf{e} + b.\mathsf{e} \dashv \overline{a}$
- $a.\mathsf{e} \oplus b.\mathsf{e} \not\dashv \overline{a} \oplus \overline{b}$
- $\mathsf{e} + \overline{a}.\mathsf{e} \dashv 0$ and $\mathsf{e} + \overline{a}.\mathsf{e} \not\dashv \Omega$ ($\Omega = \mathrm{rec}\ x.x$)

Compliance = graceful termination

Client/service interaction

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

Compliance (e indicates client's satisfaction)

ρ is *compliant with* σ ($\rho \dashv \sigma$) if $\rho \mid \sigma \Longrightarrow \rho' \mid \sigma'$ implies

1. if $\rho' \mid \sigma' \not\longrightarrow$, then $\{\mathsf{e}\} \subseteq \mathtt{init}(\rho')$
2. if $\sigma'\uparrow$, then $\{\mathsf{e}\} = \mathtt{init}(\rho')$

Examples

- $a.\mathsf{e} + b.\mathsf{e} \dashv \overline{a} \oplus \overline{b}$ and $a.\mathsf{e} + b.\mathsf{e} \dashv \overline{a}$
- $a.\mathsf{e} \oplus b.\mathsf{e} \not\dashv \overline{a} \oplus \overline{b}$
- $\mathsf{e} + \overline{a}.\mathsf{e} \dashv 0$ and $\mathsf{e} + \overline{a}.\mathsf{e} \not\dashv \cdot$ $\quad$ ($\cdot = \mathrm{rec}\ x.x$)

Querying the registry with the right **key**

Searching services with compliance

$$\texttt{query}(K : \rho) = \{ I : \sigma \mid K \subseteq I \ \text{ and } \ \rho \dashv \sigma \}$$

- effective but not efficient
- still don't have equivalence for contracts

Idea: subcontract relation

$$[\![I : \sigma]\!] = \{ K : \rho \mid K \subseteq I \ \text{ and } \ \rho \dashv \sigma \}$$

$$I : \sigma \preceq J : \tau \iff [\![I : \sigma]\!] \subseteq [\![J : \tau]\!]$$

1. compute $\rho^\perp$ such that $\rho \dashv \rho^\perp$
2. $\texttt{query}(K : \rho) = \{ I : \sigma \mid K : \rho^\perp \preceq I : \sigma \}$ (cached!)
3. make sure $K : \rho^\perp$ is the *principal dual contract*

Querying the registry with the right **key**

Searching services with compliance

$$\texttt{query}(K : \rho) = \{I : \sigma \mid K \subseteq I \ \text{ and } \ \rho \dashv \sigma\}$$

- effective but not efficient
- still don't have equivalence for contracts

Idea: subcontract relation

$$[\![I : \sigma]\!] = \{K : \rho \mid K \subseteq I \ \text{ and } \ \rho \dashv \sigma\}$$

$$I : \sigma \preceq J : \tau \iff [\![I : \sigma]\!] \subseteq [\![J : \tau]\!]$$

1. compute $\rho^\perp$ such that $\rho \dashv \rho^\perp$
2. $\texttt{query}(K : \rho) = \{I : \sigma \mid K : \rho^\perp \preceq I : \sigma\}$ (cached!)
3. make sure $K : \rho^\perp$ is the *principal dual contract*

Querying the registry with the right **key**

Searching services with compliance

$$\mathtt{query}(K : \rho) = \{ I : \sigma \mid K \subseteq I \ \text{ and } \ \rho \dashv \sigma \}$$

- effective but not efficient
- still don't have equivalence for contracts

Idea: subcontract relation

$$[\![I : \sigma]\!] = \{ K : \rho \mid K \subseteq I \ \text{ and } \ \rho \dashv \sigma \}$$

$$I : \sigma \preceq J : \tau \iff [\![I : \sigma]\!] \subseteq [\![J : \tau]\!]$$

1. compute $\rho^\perp$ such that $\rho \dashv \rho^\perp$
2. $\mathtt{query}(K : \rho) = \{ I : \sigma \mid K : \rho^\perp \preceq I : \sigma \}$ (cached!)
3. make sure $K : \rho^\perp$ is the *principal dual contract*

Subcontract relation: examples

- $I : {}^{\cdot} \preceq I : \sigma$
- $\{a, b\} : \overline{a} \oplus \overline{b} \preceq \{a, b\} : \overline{a}$
- $\{a, b\} : \overline{a} \oplus \overline{b} \preceq \{a, b\} : \overline{a} + \overline{b}$
- $\{a\} : \overline{a} \preceq \{a, b\} : \overline{a} + \overline{b}$ (width extension)
  - $\mathtt{Query} + \mathtt{Logout} + \mathtt{Purchase} \preceq$ $\mathtt{Query} + \mathtt{Logout} + \mathtt{Purchase} + \mathtt{SaveForLater}$
- $\{a\} : \overline{a} \preceq \{a, b\} : \overline{a}.\overline{b}$ (depth extension)
  - $\mathtt{Purchase}.\overline{\mathtt{Accepted}} \preceq \mathtt{Purchase}.\overline{\mathtt{Accepted}}.\overline{\mathtt{Invoice}}$
- $\{a, b\} : \overline{a} \not\preceq \{a, b\} : \overline{a} + \overline{b}$ (because of $\{a, b\} : a.\mathsf{e} + b.a.\mathsf{e}$)

Subcontract relation: examples

- $I : \mathbf{\cdot} \preceq I : \sigma$
- $\{a, b\} : \overline{a} \oplus \overline{b} \preceq \{a, b\} : \overline{a}$
- $\{a, b\} : \overline{a} \oplus \overline{b} \preceq \{a, b\} : \overline{a} + \overline{b}$
- $\{a\} : \overline{a} \preceq \{a, b\} : \overline{a} + \overline{b}$ (width extension)
  - $\mathtt{Query} + \mathtt{Logout} + \mathtt{Purchase} \preceq$
    $\mathtt{Query} + \mathtt{Logout} + \mathtt{Purchase} + \mathtt{SaveForLater}$
- $\{a\} : \overline{a} \preceq \{a, b\} : \overline{a}.\overline{b}$ (depth extension)
  - $\mathtt{Purchase}.\overline{\mathtt{Accepted}} \preceq \mathtt{Purchase}.\overline{\mathtt{Accepted}}.\overline{\mathtt{Invoice}}$
- $\{a, b\} : \overline{a} \not\preceq \{a, b\} : \overline{a} + \overline{b}$ (because of $\{a, b\} : a.e + b.a.e$)

Subcontract relation: examples

- $I : \dot{} \preceq I : \sigma$
- $\{a, b\} : \overline{a} \oplus \overline{b} \preceq \{a, b\} : \overline{a}$
- $\{a, b\} : \overline{a} \oplus \overline{b} \preceq \{a, b\} : \overline{a} + \overline{b}$
- $\{a\} : \overline{a} \preceq \{a, b\} : \overline{a} + \overline{b}$ (width extension)
  - $\texttt{Query} + \texttt{Logout} + \texttt{Purchase} \preceq$
    $\texttt{Query} + \texttt{Logout} + \texttt{Purchase} + \texttt{SaveForLater}$
- $\{a\} : \overline{a} \preceq \{a, b\} : \overline{a}.\overline{b}$ (depth extension)
  - $\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$
- $\{a, b\} : \overline{a} \not\preceq \{a, b\} : \overline{a} + \overline{b}$ (because of $\{a, b\} : a.\mathsf{e} + b.a.\mathsf{e}$)

Subcontract relation: déjà vu?

Contracts for WS's
being compliant
$\preceq$

Testing framework
"passing a test"
$\sqsubseteq_{\mathtt{must}}$

Clients
$e + \overline{a}.b.e \not\dashv a$
$e \oplus e \dashv \cdot$

Tests
$a \ \mathtt{must}\ e + \overline{a}.b.e$
$\cdot \ \mathtt{m\!\!/ust}\ e \oplus e$

Services
$\{a\} : a \preceq \{a, b\} : a + b$
$\{a\} : a \preceq \{a, b\} : a.b$

Processes
$a \not\sqsubseteq_{\mathtt{must}} a + b$
$a \not\sqsubseteq_{\mathtt{must}} a.b$

Theorem
$I : \sigma \preceq I : \tau$ *if and only if* $\sigma \sqsubseteq_{\mathtt{must}} \tau$

Subcontract relation: déjà vu?

Contracts for WS's

being compliant

$\preceq$

Testing framework

"passing a test"

$\sqsubseteq_{\texttt{must}}$

Clients

$e + \overline{a}.b.e \not\dashv a$

$e \oplus e \dashv \cdot$

Tests

$a \; \texttt{must} \; e + \overline{a}.b.e$

$\cdot \; \cancel{\texttt{must}} \; e \oplus e$

Services

$\{a\} : a \preceq \{a, b\} : a + b$

$\{a\} : a \preceq \{a, b\} : a.b$

Processes

$a \not\sqsubseteq_{\texttt{must}} a + b$

$a \not\sqsubseteq_{\texttt{must}} a.b$

Theorem

$I : \sigma \preceq I : \tau$ if and only if $\sigma \sqsubseteq_{\texttt{must}} \tau$

Subcontract relation: déjà vu?

Contracts for WS's

being compliant

$\preceq$

Clients

$e + \overline{a}.b.e \not\dashv a$

$e \oplus e \dashv \cdot$

Services

$\{a\} : a \preceq \{a, b\} : a + b$

$\{a\} : a \preceq \{a, b\} : a.b$

Testing framework

"passing a test"

$\sqsubseteq_{\mathtt{must}}$

Tests

$a \;\mathtt{must}\; e + \overline{a}.b.e$

$\cdot \;\cancel{\mathtt{must}}\; e \oplus e$

Processes

$a \not\sqsubseteq_{\mathtt{must}} a + b$

$a \not\sqsubseteq_{\mathtt{must}} a.b$

Theorem

$I : \sigma \preceq I : \tau$ if and only if $\sigma \sqsubseteq_{\mathtt{must}} \tau$

Subcontract relation: déjà vu?

Contracts for WS's

being compliant

$\preceq$

Testing framework

"passing a test"

$\sqsubseteq_{\mathtt{must}}$

Clients

$\mathsf{e} + \overline{a}.b.\mathsf{e} \not\dashv a$

$\mathsf{e} \oplus \mathsf{e} \dashv \cdot$

Tests

$a \; \mathtt{must} \; \mathsf{e} + \overline{a}.b.\mathsf{e}$

$\cdot \; \not{\mathtt{must}} \; \mathsf{e} \oplus \mathsf{e}$

Services

$\{a\} : a \preceq \{a, b\} : a + b$

$\{a\} : a \preceq \{a, b\} : a.b$

Processes

$a \not\sqsubseteq_{\mathtt{must}} a + b$

$a \not\sqsubseteq_{\mathtt{must}} a.b$

Theorem

$I : \sigma \preceq I : \tau$ *if and only if* $\sigma \sqsubseteq_{\mathtt{must}} \tau$

Principal dual contracts

Problem: given a (client) contract $K : \rho$ compute the least ($\preceq$) contract $K : \rho^\perp$ such that $\rho \dashv \rho^\perp$

Attempt 1: complement actions and swap choices

- $(a.\mathsf{e} + b.\mathsf{e})^\perp = \overline{a} \oplus \overline{b}$
- swapping does not preserve equivalence
  $\{a, b, c\} : a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e} \simeq \{a, b, c\} : a.(\overline{b}.\mathsf{e} \oplus \overline{c}.\mathsf{e})$, but
  $\{a, b, c\} : \overline{a}.b \oplus \overline{a}.c \not\simeq \{a, b, c\} : \overline{a}.(b + c)$

Attempt 2: normalize, then complement and swap

- $(a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e})^\perp = \overline{a}.(b + c)$
- not enough for principality
  $a.\mathsf{e} \dashv \overline{a} + a$ but $\{a\} : \overline{a} \not\preceq \{a\} : \overline{a} + a$

Principal dual contracts

Problem: given a (client) contract $K : \rho$ compute the least ($\preceq$) contract $K : \rho^\perp$ such that $\rho \dashv \rho^\perp$

Attempt 1: complement actions and swap choices

- $(a.\mathsf{e} + b.\mathsf{e})^\perp = \overline{a} \oplus \overline{b}$
- swapping does not preserve equivalence
  $\{a, b, c\} : a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e} \simeq \{a, b, c\} : a.(\overline{b}.\mathsf{e} \oplus \overline{c}.\mathsf{e})$, but
  $\{a, b, c\} : \overline{a}.b \oplus \overline{a}.c \not\simeq \{a, b, c\} : \overline{a}.(b + c)$

Attempt 2: normalize, then complement and swap

- $(a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e})^\perp = \overline{a}.(b + c)$
- not enough for principality
  $a.\mathsf{e} \dashv \overline{a} + a$ but $\{a\} : \overline{a} \not\preceq \{a\} : \overline{a} + a$

Principal dual contracts

Problem: given a (client) contract $K : \rho$ compute the least ($\preceq$) contract $K : \rho^\perp$ such that $\rho \dashv \rho^\perp$

Attempt 1: complement actions and swap choices

- $(a.\mathsf{e} + b.\mathsf{e})^\perp = \overline{a} \oplus \overline{b}$
- swapping does not preserve equivalence
  $\{a, b, c\} : a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e} \simeq \{a, b, c\} : a.(\overline{b}.\mathsf{e} \oplus \overline{c}.\mathsf{e})$, but
  $\{a, b, c\} : \overline{a}.b \oplus \overline{a}.c \not\simeq \{a, b, c\} : \overline{a}.(b + c)$

Attempt 2: normalize, then complement and swap

- $(a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e})^\perp = \overline{a}.(b + c)$
- not enough for principality
  $a.\mathsf{e} \dashv \overline{a} + a$ but $\{a\} : \overline{a} \not\preceq \{a\} : \overline{a} + a$

Principal dual contracts

Problem: given a (client) contract $K : \rho$ compute the least ($\preceq$) contract $K : \rho^\perp$ such that $\rho \dashv \rho^\perp$

Attempt 1: complement actions and swap choices

- $(a.\mathsf{e} + b.\mathsf{e})^\perp = \overline{a} \oplus \overline{b}$
- swapping does not preserve equivalence
  $\{a, b, c\} : a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e} \simeq \{a, b, c\} : a.(\overline{b}.\mathsf{e} \oplus \overline{c}.\mathsf{e})$, but
  $\{a, b, c\} : \overline{a}.b \oplus \overline{a}.c \not\simeq \{a, b, c\} : \overline{a}.(b + c)$

Attempt 2: normalize, then complement and swap

- $(a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e})^\perp = \overline{a}.(b + c)$
- not enough for principality
  $a.\mathsf{e} \dashv \overline{a} + a$ but $\{a\} : \overline{a} \not\preceq \{a\} : \overline{a} + a$

Principal dual contracts

Problem: given a (client) contract $K : \rho$ compute the least ($\preceq$) contract $K : \rho^\perp$ such that $\rho \dashv \rho^\perp$

Attempt 1: complement actions and swap choices

- $(a.\mathsf{e} + b.\mathsf{e})^\perp = \overline{a} \oplus \overline{b}$
- swapping does not preserve equivalence
  $\{a, b, c\} : a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e} \simeq \{a, b, c\} : a.(\overline{b}.\mathsf{e} \oplus \overline{c}.\mathsf{e})$, but
  $\{a, b, c\} : \overline{a}.b \oplus \overline{a}.c \not\simeq \{a, b, c\} : \overline{a}.(b + c)$

Attempt 2: normalize, then complement and swap

- $(a.\overline{b}.\mathsf{e} + a.\overline{c}.\mathsf{e})^\perp = \overline{a}.(b + c)$
- not enough for principality
  $a.\mathsf{e} \dashv \overline{a} + a$ but $\{a\} : \overline{a} \not\preceq \{a\} : \overline{a} + a$

Principal dual contracts: the definition at last

Ready set

$$\sigma \Downarrow \mathsf{r} \quad \text{iff} \quad \sigma \Longrightarrow \sigma' \text{ and } \mathsf{r} = \mathtt{init}(\sigma')$$

The dual operator is relative to the interface K and works for "canonical" clients only

$$\mathtt{dual}(K : \rho) = \begin{cases} \cdot, & \text{if } \mathtt{init}(\rho) = \{\mathsf{e}\} \\ \sum_{\rho \Downarrow \mathsf{r}, \mathsf{r} \setminus \{\mathsf{e}\} \neq \emptyset} \Big(\underbrace{0 \oplus}_{\text{if } \mathsf{e} \in \mathsf{r}} \bigoplus_{\alpha \in \mathsf{r} \setminus \{\mathsf{e}\}} \overline{\alpha}.\mathtt{dual}(K : \bigoplus_{\rho \Longrightarrow \xrightarrow{\alpha} \rho'} \rho') \Big) & \\ \underbrace{+ \Big(0 \oplus \bigoplus_{\alpha \in (K \cup \overline{K}) \setminus \mathtt{init}(\rho)} \overline{\alpha}.\cdot \Big)}_{\text{if } (K \cup \overline{K}) \setminus \mathtt{init}(\rho) \neq \emptyset}, & \text{otherwise} \end{cases}$$

where $\overline{K} = \{\overline{a} \mid a \in K\}$

Principal dual contracts: phew!

Examples

- $\mathtt{dual}(\{a\} : a.\mathrm{e}) \;=\; \overline{a}.^{\cdot} + (0 \oplus a.^{\cdot})$
  $\simeq\; \overline{a}.^{\cdot} \oplus (\overline{a}.^{\cdot} + a.^{\cdot}) \preceq \overline{a} + a$
- $\mathtt{dual}(\{a\} : a.\mathrm{e} \oplus \mathrm{e}) \;=\;$ ...the same...
- $\mathtt{dual}(\{a\} : a.\mathrm{e} + \mathrm{e}) \;=\; (0 \oplus \overline{a}.^{\cdot}) + (0 \oplus a.^{\cdot})$
  $\simeq\; 0 \oplus a.^{\cdot} \oplus \overline{a}.^{\cdot}$
- $\mathtt{dual}(\{a\} : {}^{\cdot}) \;=\;$ ...exercise...
- $\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) \;=\; \overline{a}.\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) + (0 \oplus a.^{\cdot})$
  $\simeq\; \mathrm{rec}\ x.(\overline{a}.x \oplus (\overline{a}.x + a.^{\cdot}))$

Theorem

Let $K : \rho$ be a (client) contract and $I : \sigma$ be a (service) contract. Then

1. $\rho \dashv \mathtt{dual}(K : \rho)$
2. *if $\rho \dashv \sigma$ and $K \subseteq I$, then $K : \mathtt{dual}(K : \rho) \preceq I : \sigma$*

Principal dual contracts: phew!

Examples

- $\mathtt{dual}(\{a\} : a.\mathsf{e}) \quad = \quad \overline{a}.^{\cdot} + (0 \oplus a.^{\cdot})$
  $\quad \simeq \quad \overline{a}.^{\cdot} \oplus (\overline{a}.^{\cdot} + a.^{\cdot}) \preceq \overline{a} + a$
- $\mathtt{dual}(\{a\} : a.\mathsf{e} \oplus \mathsf{e}) \quad = \quad$ ...the same...
- $\mathtt{dual}(\{a\} : a.\mathsf{e} + \mathsf{e}) \quad = \quad (0 \oplus \overline{a}.^{\cdot}) + (0 \oplus a.^{\cdot})$
  $\quad \simeq \quad 0 \oplus a.^{\cdot} \oplus \overline{a}.^{\cdot}$
- $\mathtt{dual}(\{a\} : {}^{\cdot}) \quad = \quad$ ...exercise...
- $\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) \quad = \quad \overline{a}.\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) + (0 \oplus a.^{\cdot})$
  $\quad \simeq \quad \mathrm{rec}\ x.(\overline{a}.x \oplus (\overline{a}.x + a.^{\cdot}))$

Theorem

Let $K : \rho$ be a (client) contract and $I : \sigma$ be a (service) contract. Then

1. $\rho \dashv \mathtt{dual}(K : \rho)$
2. *if $\rho \dashv \sigma$ and $K \subseteq I$, then $K : \mathtt{dual}(K : \rho) \preceq I : \sigma$*

Principal dual contracts: phew!

Examples

- $\mathtt{dual}(\{a\} : a.\mathsf{e}) \;=\; \bar{a}.^{\cdot} + (0 \oplus a.^{\cdot})$
  $\simeq\; \bar{a}.^{\cdot} \oplus (\bar{a}.^{\cdot} + a.^{\cdot}) \preceq \bar{a} + a$
- $\mathtt{dual}(\{a\} : a.\mathsf{e} \oplus \mathsf{e}) \;=\;$ …the same…
- $\mathtt{dual}(\{a\} : a.\mathsf{e} + \mathsf{e}) \;=\; (0 \oplus \bar{a}.^{\cdot}) + (0 \oplus a.^{\cdot})$
  $\simeq\; 0 \oplus a.^{\cdot} \oplus \bar{a}.^{\cdot}$
- $\mathtt{dual}(\{a\} : {}^{\cdot}) \;=\;$ …exercise…
- $\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) \;=\; \bar{a}.\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) + (0 \oplus a.^{\cdot})$
  $\simeq\; \mathrm{rec}\ x.(\bar{a}.x \oplus (\bar{a}.x + a.^{\cdot}))$

Theorem

Let $K : \rho$ be a (client) contract and $I : \sigma$ be a (service) contract. Then

1. $\rho \dashv \mathtt{dual}(K : \rho)$
2. *if $\rho \dashv \sigma$ and $K \subseteq I$, then $K : \mathtt{dual}(K : \rho) \preceq I : \sigma$*

Principal dual contracts: phew!

Examples

- $\mathtt{dual}(\{a\} : a.\mathrm{e}) = \bar{a}.^{\cdot} + (0 \oplus a.^{\cdot})$
  $\simeq \bar{a}.^{\cdot} \oplus (\bar{a}.^{\cdot} + a.^{\cdot}) \preceq \bar{a} + a$
- $\mathtt{dual}(\{a\} : a.\mathrm{e} \oplus \mathrm{e}) = \ldots$ the same$\ldots$
- $\mathtt{dual}(\{a\} : a.\mathrm{e} + \mathrm{e}) = (0 \oplus \bar{a}.^{\cdot}) + (0 \oplus a.^{\cdot})$
  $\simeq 0 \oplus a.^{\cdot} \oplus \bar{a}.^{\cdot}$
- $\mathtt{dual}(\{a\} : {}^{\cdot}) = \ldots$ exercise$\ldots$
- $\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) = \bar{a}.\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) + (0 \oplus a.^{\cdot})$
  $\simeq \mathrm{rec}\ x.(\bar{a}.x \oplus (\bar{a}.x + a.^{\cdot}))$

Theorem

Let $K : \rho$ be a (client) contract and $I : \sigma$ be a (service) contract. Then

1. $\rho \dashv \mathtt{dual}(K : \rho)$
2. *if $\rho \dashv \sigma$ and $K \subseteq I$, then $K : \mathtt{dual}(K : \rho) \preceq I : \sigma$*

Principal dual contracts: phew!

Examples

- $\mathtt{dual}(\{a\} : a.\mathsf{e})$ $\quad = \quad \overline{a}.^{\cdot} + (0 \oplus a.^{\cdot})$
  $\quad \simeq \quad \overline{a}.^{\cdot} \oplus (\overline{a}.^{\cdot} + a.^{\cdot}) \preceq \overline{a} + a$
- $\mathtt{dual}(\{a\} : a.\mathsf{e} \oplus \mathsf{e})$ $\quad = \quad \ldots$ the same$\ldots$
- $\mathtt{dual}(\{a\} : a.\mathsf{e} + \mathsf{e})$ $\quad = \quad (0 \oplus \overline{a}.^{\cdot}) + (0 \oplus a.^{\cdot})$
  $\quad \simeq \quad 0 \oplus a.^{\cdot} \oplus \overline{a}.^{\cdot}$
- $\mathtt{dual}(\{a\} : ^{\cdot})$ $\quad = \quad \ldots$ exercise$\ldots$
- $\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x)$ $\quad = \quad \overline{a}.\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) + (0 \oplus a.^{\cdot})$
  $\quad \simeq \quad \mathrm{rec}\ x.(\overline{a}.x \oplus (\overline{a}.x + a.^{\cdot}))$

Theorem

Let $K : \rho$ be a (client) contract and $I : \sigma$ be a (service) contract. Then

1. $\rho \dashv \mathtt{dual}(K : \rho)$
2. *if $\rho \dashv \sigma$ and $K \subseteq I$, then $K : \mathtt{dual}(K : \rho) \preceq I : \sigma$*

Principal dual contracts: phew!

Examples

- $\mathtt{dual}(\{a\} : a.\mathrm{e})$ $= \overline{a}.\dot{} + (0 \oplus a.\dot{})$
  $\simeq \overline{a}.\dot{} \oplus (\overline{a}.\dot{} + a.\dot{}) \preceq \overline{a} + a$
- $\mathtt{dual}(\{a\} : a.\mathrm{e} \oplus \mathrm{e})$ $=$ ...the same...
- $\mathtt{dual}(\{a\} : a.\mathrm{e} + \mathrm{e})$ $= (0 \oplus \overline{a}.\dot{}) + (0 \oplus a.\dot{})$
  $\simeq 0 \oplus a.\dot{} \oplus \overline{a}.\dot{}$
- $\mathtt{dual}(\{a\} : \dot{})$ $=$ ...exercise...
- $\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x)$ $= \overline{a}.\mathtt{dual}(\{a\} : \mathrm{rec}\ x.a.x) + (0 \oplus a.\dot{})$
  $\simeq \mathrm{rec}\ x.(\overline{a}.x \oplus (\overline{a}.x + a.\dot{}))$

Theorem

Let $K : \rho$ be a (client) contract and $I : \sigma$ be a (service) contract. Then

1. $\rho \dashv \mathtt{dual}(K : \rho)$
2. *if* $\rho \dashv \sigma$ *and* $K \subseteq I$, *then* $K : \mathtt{dual}(K : \rho) \preceq I : \sigma$

From services to choreographies

Web services and parallelism

- if $I : \sigma \preceq J : \tau$, the service $J : \tau$ can be used in place of $I : \sigma$
- what happens if $I : \sigma$ is part of a larger system?
- larger system = choreography
- a choreography description specifies which communications occur in the (parallel) composition of n services
- a choreography description can be projected to a term

$$\Sigma = (I_1 : \sigma_1 \mid \cdots \mid I_n : \sigma_n) \setminus L$$

From services to choreographies

Web services and parallelism

- if $I : \sigma \preceq J : \tau$, the service $J : \tau$ can be used in place of $I : \sigma$
- what happens if $I : \sigma$ is part of a larger system?
- larger system = choreography
- a choreography description specifies which communications occur in the (parallel) composition of n services
- a choreography description can be projected to a term

$$\Sigma = (I_1 : \sigma_1 \mid \cdots \mid I_n : \sigma_n) \setminus L$$

Lifting $\preceq$ to choreographies

Choreography refinement

$$\Sigma = (I_1 : \sigma_1 \mid \cdots \mid I_n : \sigma_n) \setminus L$$
$$\Sigma' = (J_1 : \tau_1 \mid \cdots \mid J_n : \tau_n) \setminus L'$$

$$\boxed{I_k : \sigma_k \preceq J_k : \tau_k}$$

Under which conditions can Σ' be safely used in place of Σ?

1. $\bigcup_{k \in 1..n} I_k \setminus L = \bigcup_{k \in 1..n} J_k \setminus L'$
2. for every $1 \leq i, j \leq n$ with $i \neq j$ we have that $L' \cap (\mathtt{names}(\tau_i) \setminus \mathtt{names}(\sigma_i)) \cap (\mathtt{names}(\tau_j) \setminus \mathtt{names}(\sigma_j)) = \emptyset$

Theorem

If $K : \rho$ is compliant with Σ and Σ' is a refinement of Σ, then $K : \rho$ is also compliant with Σ'

Lifting $\preceq$ to choreographies

Choreography refinement

$$\Sigma = (I_1 : \sigma_1 \mid \cdots \mid I_n : \sigma_n) \setminus L$$
$$\Sigma' = (J_1 : \tau_1 \mid \cdots \mid J_n : \tau_n) \setminus L' \qquad \boxed{I_k : \sigma_k \preceq J_k : \tau_k}$$

Under which conditions can Σ' be safely used in place of Σ?

1. $\bigcup_{k \in 1..n} I_k \setminus L = \bigcup_{k \in 1..n} J_k \setminus L'$
2. for every $1 \leq i, j \leq n$ with $i \neq j$ we have that $L' \cap (\mathtt{names}(\tau_i) \setminus \mathtt{names}(\sigma_i)) \cap (\mathtt{names}(\tau_j) \setminus \mathtt{names}(\sigma_j)) = \emptyset$

Theorem

If $K : \rho$ is compliant with Σ and Σ' is a refinement of Σ, then $K : \rho$ is also compliant with Σ'

Lifting $\preceq$ to choreographies

Choreography refinement

$$\Sigma = (I_1 : \sigma_1 \mid \cdots \mid I_n : \sigma_n) \setminus L$$
$$\Sigma' = (J_1 : \tau_1 \mid \cdots \mid J_n : \tau_n) \setminus L' \qquad \boxed{I_k : \sigma_k \preceq J_k : \tau_k}$$

Under which conditions can Σ' be safely used in place of Σ?

1. $\bigcup_{k \in 1..n} I_k \setminus L = \bigcup_{k \in 1..n} J_k \setminus L'$
2. for every $1 \leq i, j \leq n$ with $i \neq j$ we have that $L' \cap (\mathtt{names}(\tau_i) \setminus \mathtt{names}(\sigma_i)) \cap (\mathtt{names}(\tau_j) \setminus \mathtt{names}(\sigma_j)) = \emptyset$

Theorem

If $K : \rho$ is compliant with Σ and Σ' is a refinement of Σ, then $K : \rho$ is also compliant with Σ'

Wrap-up

- theory for *searching* and *reasoning about* services by their contracts (= behavioral types)
- $\preceq$ gives safe substitution of services
- $\preceq$ can speed up querying Web service registries
- $\preceq$ behaves well in choreographies

With respect to $\sqsubseteq_{\mathtt{must}}$

- $\dashv$ has a (more) practical justification
- interfaces make $\preceq$ more general than $\sqsubseteq_{\mathtt{must}}$ (width and depth extensions *are* possible)
- interfaces permit the computation of *finite principal dual contracts*

Wrap-up

- theory for *searching* and *reasoning about* services by their contracts (= behavioral types)
- $\preceq$ gives safe substitution of services
- $\preceq$ can speed up querying Web service registries
- $\preceq$ behaves well in choreographies

With respect to $\sqsubseteq_{\texttt{must}}$

- $\dashv$ has a (more) practical justification
- interfaces make $\preceq$ more general than $\sqsubseteq_{\texttt{must}}$ (width and depth extensions *are* possible)
- interfaces permit the computation of *finite principal dual contracts*

Related work

Foundations

- acceptance trees (Hennessy)
- testing processes (De Nicola, Hennessy)
- ccs without τ's (De Nicola, Hennessy)

Contracts aka...

- session types (Gay, Hole, Vasconcelos)
- interface automata (De Alfaro)

Variations on the theme

- "*Performace-oriented comparison of Web services via client-specific testing preorders*", FMOODS'07 (Bernardo, Padovani)
- "*A theory of contracts for Web services*", POPL'08 (Castagna, Gesbert, Padovani)

What next?

Languages

- checking and inferring contracts
- progress guarantees for choreographies

Implementations

- from contracts to session types (and back)
- asynchrony

Extensions

- names and *higher-order Web services* (`WSDL` 2.0)

Subcontract relation: alternative characterization

Coinductive subcontract relation

If $(I : \sigma, J : \tau) \in \mathcal{R}$, then $I \subseteq J$ and whenever $\sigma\downarrow$ then

1. $\tau\downarrow$, and
2. $\tau \Downarrow \mathsf{r}$ implies $\sigma \Downarrow \mathsf{s}$ and $\mathsf{s} \subseteq \mathsf{r}$, and
3. $\alpha \in I \cup \overline{I}$ and $\tau \stackrel{\alpha}{\Longrightarrow} \tau'$ implies $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma_1, \ldots, \sigma \stackrel{\alpha}{\Longrightarrow} \sigma_n$ and $(I : \bigoplus_{1 \le i \le n} \sigma_i, J : \tau') \in \mathcal{R}$

Watch for condition 3

- $\{a, b, c\} : a.\overline{b} + a.\overline{c} \simeq \{a, b, c\} : a.(\overline{b} \oplus \overline{c})$
- no single a-derivative of $a.\overline{b} + a.\overline{c}$ is smaller than $\overline{b} \oplus \overline{c}$
- we take the internal choice of *all* the a-derivatives: $\overline{b} \oplus \overline{c}$

Theorem

$\preceq$ *is the largest coinductive subcontract relation*

Subcontract relation: alternative characterization

Coinductive subcontract relation

If $(I : \sigma, J : \tau) \in \mathcal{R}$, then $I \subseteq J$ and whenever $\sigma\downarrow$ then

1. $\tau\downarrow$, and
2. $\tau \Downarrow \mathsf{r}$ implies $\sigma \Downarrow \mathsf{s}$ and $\mathsf{s} \subseteq \mathsf{r}$, and
3. $\alpha \in I \cup \overline{I}$ and $\tau \stackrel{\alpha}{\Longrightarrow} \tau'$ implies $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma_1, \ldots, \sigma \stackrel{\alpha}{\Longrightarrow} \sigma_n$ and $(I : \bigoplus_{1 \leq i \leq n} \sigma_i, J : \tau') \in \mathcal{R}$

Watch for condition 3

- $\{a, b, c\} : a.\overline{b} + a.\overline{c} \simeq \{a, b, c\} : a.(\overline{b} \oplus \overline{c})$
- no single a-derivative of $a.\overline{b} + a.\overline{c}$ is smaller than $\overline{b} \oplus \overline{c}$
- we take the internal choice of *all* the a-derivatives: $\overline{b} \oplus \overline{c}$

Theorem

$\preceq$ *is the largest coinductive subcontract relation*

Subcontract relation: alternative characterization

Coinductive subcontract relation

If $(I : \sigma, J : \tau) \in \mathcal{R}$, then $I \subseteq J$ and whenever $\sigma\downarrow$ then

1. $\tau\downarrow$, and
2. $\tau \Downarrow \mathsf{r}$ implies $\sigma \Downarrow \mathsf{s}$ and $\mathsf{s} \subseteq \mathsf{r}$, and
3. $\alpha \in I \cup \overline{I}$ and $\tau \stackrel{\alpha}{\Longrightarrow} \tau'$ implies $\sigma \stackrel{\alpha}{\Longrightarrow} \sigma_1, \ldots, \sigma \stackrel{\alpha}{\Longrightarrow} \sigma_n$ and $(I : \bigoplus_{1 \leq i \leq n} \sigma_i, J : \tau') \in \mathcal{R}$

Watch for condition 3

- $\{a, b, c\} : a.\overline{b} + a.\overline{c} \simeq \{a, b, c\} : a.(\overline{b} \oplus \overline{c})$
- no single a-derivative of $a.\overline{b} + a.\overline{c}$ is smaller than $\overline{b} \oplus \overline{c}$
- we take the internal choice of *all* the a-derivatives: $\overline{b} \oplus \overline{c}$

Theorem

$\preceq$ *is the largest coinductive subcontract relation*

An alternative compliance relation

ρ is *compliant with* σ ($\rho \dashv \sigma$) if $\rho \mid \sigma \Longrightarrow \rho' \mid \sigma'$ implies

1. if $\rho' \mid \sigma' \not\longrightarrow$, then $\{\mathtt{e}\} \subseteq \mathtt{init}(\rho')$
2. if $\sigma'\uparrow$, then $\{\mathtt{e}\} = \mathtt{init}(\rho')$

Practical implications

- a client cannot *try* to perform an action: $\mathtt{e} + a.\mathtt{e} \not\dashv 0$
- among the actions proposed by the client, at least one must succeed

Theoretical implications

- 0 and ˙ are indistinguishable
- depth extensions are possible without interfaces: $\{a\} : 0 \preceq \{a\} : a$
- weaker connection with $\sqsubseteq_{\mathtt{must}}$, weaker precongruence properties

An alternative compliance relation

ρ is *compliant with* σ ($\rho \dashv \sigma$) if $\rho \mid \sigma \Longrightarrow \rho' \mid \sigma'$ implies

1. if $\rho' \mid \sigma' \not\longrightarrow$, then $\{\mathtt{e}\} = \mathtt{init}(\rho')$
2. if $\sigma'\!\uparrow$, then $\{\mathtt{e}\} = \mathtt{init}(\rho')$

Practical implications

- a client cannot *try* to perform an action: $\mathtt{e} + a.\mathtt{e} \not\dashv 0$
- among the actions proposed by the client, at least one must succeed

Theoretical implications

- 0 and ˙ are indistinguishable
- depth extensions are possible without interfaces: $\{a\} : 0 \preceq \{a\} : a$
- weaker connection with $\sqsubseteq_{\mathtt{must}}$, weaker precongruence properties

The *must* preorder

- $\sigma_0 \mid \rho_0 \longrightarrow \sigma_1 \mid \rho_1 \longrightarrow \cdots$ is *maximal* if either it is infinite or the last term $\sigma_n \mid \rho_n$ is such that $\sigma_n \mid \rho_n \not\longrightarrow$

- $\sigma\ \mathtt{must}\ \rho$ if, for every maximal $\sigma \mid \rho = \sigma_0 \mid \rho_0 \longrightarrow \sigma_1 \mid \rho_1 \longrightarrow \cdots$, there exists $n \geq 0$ such that $\rho_n \xrightarrow{\mathsf{e}}$

- $\sigma \sqsubseteq_{\mathtt{must}} \tau$ if and only if, for every ρ, $\sigma\ \mathtt{must}\ \rho$ implies $\tau\ \mathtt{must}\ \rho$

Lifting $\dashv$ to choreographies

Notation

- $\Sigma[i \mapsto J : \rho]$ is Σ where the i-th participant replaced by $J : \rho$
- write Σ_L when we want to recall the private names of the choreography

Transition relation of choreographies

$$\frac{\sigma \xrightarrow{\alpha} \sigma' \qquad \mathtt{names}(\alpha) \notin L}{\Sigma_L[i \mapsto I : \sigma] \xrightarrow{\alpha} \Sigma_L[i \mapsto I : \sigma']} \qquad \frac{\sigma \longrightarrow \sigma'}{\Sigma_L[i \mapsto I : \sigma] \longrightarrow \Sigma_L[i \mapsto I : \sigma']}$$

$$\frac{i \neq j \quad \sigma \xrightarrow{\alpha} \sigma' \quad \rho \xrightarrow{\overline{\alpha}} \rho' \quad \mathtt{names}(\alpha) \in L}{\Sigma_L[i \mapsto I : \sigma][j \mapsto J : \rho] \longrightarrow \Sigma_L[i \mapsto I : \sigma'][j \mapsto J : \rho']}$$

Compliance w.r.t. a choreography

$\rho \dashv \Sigma$ (overload $\longrightarrow$)

(Missing) parallelism

- we restrict to *finite-state* conversations
- we may need to *overestimate* the contract of clients
- we may need to *underestimate* the contract of services
- we can use the *expansion law* for describing finite (sub)processes

$$
\begin{array}{rcl}
a \mid b & \simeq & a.b + b.a \\
\overline{a} \mid a.b & \simeq & (\overline{a}.a.b + a.(\overline{a} \mid b) + b) \oplus b \\
(\overline{a} \mid a.b) \setminus a & \simeq & b
\end{array}
$$