A Formal Account of Contracts for Web Services

Samuele Carpineti, Giuseppe Castagna, Cosimo Laneve, Luca Padovani

University of Bologna, University of Urbino, École Normale Supérieure de Paris

8 september 2006

Summary

- Contracts and technologies for Web Services
- A language of contracts
- Subcontract relation and contract compliance
- Contract synthesis and process compliance
- Contract compliance $\Rightarrow$ process compliance
- Concluding remarks

Reasoning about compatibility of behavior

Why is it important to formalize the contract of a client or of a service?

Use:

- dynamic discovery
- dynamic composition
- type checking
- debugging
- automatic code generation
- run-time analysis

Focus:

- communication between two parties (no choreography)

Reasoning about compatibility of behavior

Why is it important to formalize the contract of a client or of a service?

Use:

- dynamic discovery
- dynamic composition
- type checking
- debugging
- automatic code generation
- run-time analysis

Focus:

- communication between two parties (no choreography)

Reasoning about compatibility of behavior

Why is it important to formalize the contract of a client or of a service?

Use:

- dynamic discovery
- dynamic composition
- type checking
- debugging
- automatic code generation
- run-time analysis

Focus:

- communication between two parties (no choreography)

Contracts in WSDL

Focus on the static interface:

- Interface = set of operations
- Operation = name + message exchange pattern (MEP)

```
<operation name="A"
    pattern="http://www.w3.org/2006/01/wsdl/in-only">
  <input messageLabel="In"/>
</operation>

<operation name="B"
    pattern="http://www.w3.org/2006/01/wsdl/robust-in-only">
  <input messageLabel="In"/>
  <outfault messageLabel="Fault"/>
</operation>
```

Contracts in WSCL

Focus on the dynamic interface:

- Conversation = Interactions + Transitions
- Interaction = Types of exchanged messages

+ distinction between internal and external choice
+ possibly cyclic patterns

Contracts in WSCL

Focus on the dynamic interface:

- Conversation = Interactions + Transitions
- Interaction = Types of exchanged messages

+ distinction between internal and external choice
+ possibly cyclic patterns

Contracts in WSCL

Focus on the dynamic interface:

- Conversation = Interactions + Transitions
- Interaction = Types of exchanged messages

+ distinction between internal and external choice
+ possibly cyclic patterns

Encoding MEPs into contracts

```
<operation name="A"
    pattern="http://www.w3.org/2006/01/wsdl/in-only">
  <input messageLabel="In"/>
</operation>

<operation name="B"
    pattern="http://www.w3.org/2006/01/wsdl/robust-in-only">
  <input messageLabel="In"/>
  <outfault messageLabel="Fault"/>
</operation>
```

$$
\begin{aligned}
\mathtt{A} &\stackrel{\text{def}}{=} \mathtt{In}.\overline{\mathtt{End}} \\
\mathtt{B} &\stackrel{\text{def}}{=} \mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}})
\end{aligned}
$$

Encoding WSCL into contracts

$\texttt{Login}.(\overline{\texttt{InvalidLogin}}.\overline{\texttt{End}} \oplus \overline{\texttt{ValidLogin}}.\texttt{Query}.\overline{\texttt{Catalog}}.($
$\quad \texttt{Logout}.\overline{\texttt{End}} + \texttt{Purchase}.($
$\quad\quad \overline{\texttt{Accepted}}.\overline{\texttt{End}} \oplus \overline{\texttt{InvalidPayment}}.\overline{\texttt{End}} \oplus \overline{\texttt{OutOfStock}}.\overline{\texttt{End}})))$

Encoding WSCL into contracts

$$\begin{aligned}&\texttt{Login}.(\overline{\texttt{InvalidLogin}}.\overline{\texttt{End}} \oplus \overline{\texttt{ValidLogin}}.\texttt{Query}.\overline{\texttt{Catalog}}.(\\&\quad \texttt{Logout}.\overline{\texttt{End}} + \texttt{Purchase}.(\\&\qquad \overline{\texttt{Accepted}}.\overline{\texttt{End}} \oplus \overline{\texttt{InvalidPayment}}.\overline{\texttt{End}} \oplus \overline{\texttt{OutOfStock}}.\overline{\texttt{End}})))\end{aligned}$$

Encoding WSCL into contracts

$$\begin{aligned}&\mathtt{Login}.(\overline{\mathtt{InvalidLogin}}.\mathtt{End} \oplus \overline{\mathtt{ValidLogin}}.\mathtt{Query}.\overline{\mathtt{Catalog}}.(\\&\quad \mathtt{Logout}.\mathtt{End} + \mathtt{Purchase}.(\\&\qquad \overline{\mathtt{Accepted}}.\mathtt{End} \oplus \overline{\mathtt{InvalidPayment}}.\mathtt{End} \oplus \overline{\mathtt{OutOfStock}}.\mathtt{End})))\end{aligned}$$

Encoding WSCL into contracts

$$
\begin{aligned}
&\mathtt{Login}.(\overline{\mathtt{InvalidLogin}}.\overline{\mathtt{End}} \oplus \overline{\mathtt{ValidLogin}}.\mathtt{Query}.\overline{\mathtt{Catalog}}.(\\
&\quad \mathtt{Logout}.\overline{\mathtt{End}} + \mathtt{Purchase}.(\\
&\qquad \overline{\mathtt{Accepted}}.\overline{\mathtt{End}} \oplus \overline{\mathtt{InvalidPayment}}.\overline{\mathtt{End}} \oplus \overline{\mathtt{OutOfStock}}.\overline{\mathtt{End}})))
\end{aligned}
$$

A formal contract language

contracts	σ	$::=$		
			$\mathbf{0}$	(*void*)
			$\alpha.\sigma$	(*action prefix*)
			$\sigma + \sigma$	(*external choice*)
			$\sigma \oplus \sigma$	(*internal choice*)
actions	α	$::=$		
			a	(*name*)
			$\overline{a}$	(*co-name*)

Names represent types, operations, ...

c.f. De Nicola, Hennessy, "CCS without τ's"

A formal contract language

contracts	σ	$::=$		
			$\mathbf{0}$	(*void*)
			$\alpha.\sigma$	(*action prefix*)
			$\sigma + \sigma$	(*external choice*)
			$\sigma \oplus \sigma$	(*internal choice*)
actions	α	$::=$		
			a	(*name*)
			$\overline{a}$	(*co-name*)

Names represent types, operations, ...

c.f. De Nicola, Hennessy, "CCS without τ's"

A formal contract language

contracts	σ	$::=$		
			$\mathbf{0}$	(*void*)
			$\alpha.\sigma$	(*action prefix*)
			$\sigma + \sigma$	(*external choice*)
			$\sigma \oplus \sigma$	(*internal choice*)
actions	α	$::=$		
			a	(*name*)
			$\overline{a}$	(*co-name*)

Names represent types, operations, ...

c.f. De Nicola, Hennessy, "CCS without τ's"

Comparing contracts: the subcontract relation $\preceq$

σ is a subcontract of σ' if σ' is *more deterministic* than σ

$$a \oplus b \preceq a + b \qquad\qquad a \oplus b \preceq a$$

$$\mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}}) \preceq \mathtt{In}.\overline{\mathtt{End}}$$

(*c.f. must pre-order*)

σ is a subcontract of σ' if σ' has *more interacting capabilities* than σ

$$a \preceq a.b \qquad\qquad a \preceq a + b \qquad\qquad \mathbf{0} \preceq \sigma$$

$$\mathtt{Logout} + \mathtt{Purchase} \preceq \mathtt{Logout} + \mathtt{Purchase} + \mathtt{BuyLater}$$

($\preceq$ *is different from testing, must, may,* ...)

Comparing contracts: the subcontract relation $\preceq$

σ is a subcontract of σ' if σ' is *more deterministic* than σ

$$a \oplus b \preceq a + b \qquad\qquad a \oplus b \preceq a$$

$$\mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}}) \preceq \mathtt{In}.\overline{\mathtt{End}}$$

(*c.f. must pre-order*)

σ is a subcontract of σ' if σ' has *more interacting capabilities* than σ

$$a \preceq a.b \qquad\qquad a \preceq a + b \qquad\qquad \mathbf{0} \preceq \sigma$$

$$\mathtt{Logout} + \mathtt{Purchase} \preceq \mathtt{Logout} + \mathtt{Purchase} + \mathtt{BuyLater}$$

($\preceq$ *is different from testing, must, may, ...*)

Comparing contracts: the subcontract relation $\preceq$

σ is a subcontract of σ' if σ' is *more deterministic* than σ

$$a \oplus b \preceq a + b \qquad a \oplus b \preceq a$$

$$\mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}}) \preceq \mathtt{In}.\overline{\mathtt{End}}$$

(*c.f. must pre-order*)

σ is a subcontract of σ' if σ' has *more interacting capabilities* than σ

$$a \preceq a.b \qquad a \preceq a + b \qquad \mathbf{0} \preceq \sigma$$

$$\mathtt{Logout} + \mathtt{Purchase} \preceq \mathtt{Logout} + \mathtt{Purchase} + \mathtt{BuyLater}$$

($\preceq$ *is different from testing, must, may, ...*)

Comparing contracts: the subcontract relation $\preceq$

σ is a subcontract of σ' if σ' is *more deterministic* than σ

$$a \oplus b \preceq a + b \qquad\qquad a \oplus b \preceq a$$

$$\mathtt{In}.(\overline{\mathtt{End}} \oplus \overline{\mathtt{Fault}}.\overline{\mathtt{End}}) \preceq \mathtt{In}.\overline{\mathtt{End}}$$

(*c.f. must pre-order*)

σ is a subcontract of σ' if σ' has *more interacting capabilities* than σ

$$a \preceq a.b \qquad\qquad a \preceq a + b \qquad\qquad \mathbf{0} \preceq \sigma$$

$$\mathtt{Logout} + \mathtt{Purchase} \preceq \mathtt{Logout} + \mathtt{Purchase} + \mathtt{BuyLater}$$

($\preceq$ *is different from testing, must, may, ...*)

Summary of the technical part

1. define contract transition and ready sets
2. define subcontract $\preceq$ and contract compliance $\ll$
3. synthesize contracts out of processes
4. define process compliance as "successful interaction"
5. prove that contract compliance implies process compliance

Contracts: transition relation

Interacting party's point of view:

$$a.b + a.c \overset{a}{\longmapsto} b \oplus c$$

$$\alpha.\sigma \overset{\alpha}{\longmapsto} \sigma$$

$$\frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\longmapsto} \sigma_2'}{\sigma_1 + \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1' \oplus \sigma_2'} \qquad \frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\not\longmapsto}}{\sigma_1 + \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1'}$$

$$\frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\longmapsto} \sigma_2'}{\sigma_1 \oplus \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1' \oplus \sigma_2'} \qquad \frac{\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \overset{\alpha}{\not\longmapsto}}{\sigma_1 \oplus \sigma_2 \overset{\alpha}{\longmapsto} \sigma_1'}$$

Contracts: transition relation

Interacting party's point of view:

$$a.b + a.c \stackrel{a}{\longmapsto} b \oplus c$$

$$\alpha.\sigma \stackrel{\alpha}{\longmapsto} \sigma$$

$$\frac{\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_2'}{\sigma_1 + \sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_1' \oplus \sigma_2'} \qquad \frac{\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \stackrel{\alpha}{\not\longmapsto}}{\sigma_1 + \sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_1'}$$

$$\frac{\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_2'}{\sigma_1 \oplus \sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_1' \oplus \sigma_2'} \qquad \frac{\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1' \quad \sigma_2 \stackrel{\alpha}{\not\longmapsto}}{\sigma_1 \oplus \sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_1'}$$

Contracts: ready sets

$\sigma \Downarrow \mathrm{R}$: the service can be in a state where the actions in R are allowed

$$
\begin{array}{ll}
\mathbf{0} \Downarrow \emptyset & \\
\alpha.\sigma \Downarrow \{\alpha\} & \\
(\sigma + \sigma') \Downarrow \mathrm{R} \cup \mathrm{R}' & \text{if } \sigma \Downarrow \mathrm{R} \text{ and } \sigma' \Downarrow \mathrm{R}' \\
(\sigma \oplus \sigma') \Downarrow \mathrm{R} & \text{if either } \sigma \Downarrow \mathrm{R} \text{ or } \sigma' \Downarrow \mathrm{R}
\end{array}
$$

Example of nondeterministic contract/service:

$$a \oplus b \Downarrow \{a\} \qquad a \oplus b \Downarrow \{b\}$$

Example of deterministic contract/service:

$$a + b \Downarrow \{a, b\}$$

Contracts: ready sets

$\sigma \Downarrow \mathrm{R}$: the service can be in a state where the actions in R are allowed

$$
\begin{array}{ll}
\mathbf{0} \Downarrow \emptyset & \\
\alpha.\sigma \Downarrow \{\alpha\} & \\
(\sigma + \sigma') \Downarrow \mathrm{R} \cup \mathrm{R}' & \text{if } \sigma \Downarrow \mathrm{R} \text{ and } \sigma' \Downarrow \mathrm{R}' \\
(\sigma \oplus \sigma') \Downarrow \mathrm{R} & \text{if either } \sigma \Downarrow \mathrm{R} \text{ or } \sigma' \Downarrow \mathrm{R}
\end{array}
$$

Example of nondeterministic contract/service:

$$a \oplus b \Downarrow \{a\} \qquad a \oplus b \Downarrow \{b\}$$

Example of deterministic contract/service:

$$a + b \Downarrow \{a, b\}$$

Contracts: ready sets

$\sigma \Downarrow \mathrm{R}$: the service can be in a state where the actions in R are allowed

$$
\begin{array}{ll}
\mathbf{0} \Downarrow \emptyset & \\
\alpha.\sigma \Downarrow \{\alpha\} & \\
(\sigma + \sigma') \Downarrow \mathrm{R} \cup \mathrm{R}' & \text{if } \sigma \Downarrow \mathrm{R} \text{ and } \sigma' \Downarrow \mathrm{R}' \\
(\sigma \oplus \sigma') \Downarrow \mathrm{R} & \text{if either } \sigma \Downarrow \mathrm{R} \text{ or } \sigma' \Downarrow \mathrm{R}
\end{array}
$$

Example of nondeterministic contract/service:

$$a \oplus b \Downarrow \{a\} \qquad\qquad a \oplus b \Downarrow \{b\}$$

Example of deterministic contract/service:

$$a + b \Downarrow \{a, b\}$$

Subcontract relation

$\preceq$ is the largest relation such that $\sigma_1 \preceq \sigma_2$ implies:

1. if $\sigma_2 \Downarrow \mathrm{R}_2$ then $\sigma_1 \Downarrow \mathrm{R}_1$ with $\mathrm{R}_1 \subseteq \mathrm{R}_2$
2. if $\sigma_1 \xmapsto{\alpha} \sigma_1'$ and $\sigma_2 \xmapsto{\alpha} \sigma_2'$ then $\sigma_1' \preceq \sigma_2'$

Key:

1. σ_2 has no more internal states than σ_1 has:

$$a \oplus b \preceq a \qquad\qquad a \oplus b \preceq b$$

   and they all allow more capabilities than those in σ_1:

$$a \oplus b \preceq a + b \qquad\qquad a \preceq a + b$$

2. if σ_1 and σ_2 share a common action, the continuations are in the subcontract relation:

$$\mathbf{0} \preceq \sigma \qquad\qquad a.b \preceq a.b + c$$

Subcontract relation

$\preceq$ is the largest relation such that $\sigma_1 \preceq \sigma_2$ implies:

1. if $\sigma_2 \Downarrow \mathrm{R}_2$ then $\sigma_1 \Downarrow \mathrm{R}_1$ with $\mathrm{R}_1 \subseteq \mathrm{R}_2$
2. if $\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1'$ and $\sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_2'$ then $\sigma_1' \preceq \sigma_2'$

Key:

1. σ_2 has no more internal states than σ_1 has:

$$a \oplus b \preceq a \qquad a \oplus b \preceq b$$

and they all allow more capabilities than those in σ_1:

$$a \oplus b \preceq a + b \qquad a \preceq a + b$$

2. if σ_1 and σ_2 share a common action, the continuations are in the subcontract relation:

$$\mathbf{0} \preceq \sigma \qquad a.b \preceq a.b + c$$

Subcontract relation

$\preceq$ is the largest relation such that $\sigma_1 \preceq \sigma_2$ implies:

1. if $\sigma_2 \Downarrow \mathrm{R}_2$ then $\sigma_1 \Downarrow \mathrm{R}_1$ with $\mathrm{R}_1 \subseteq \mathrm{R}_2$
2. if $\sigma_1 \stackrel{\alpha}{\longmapsto} \sigma_1'$ and $\sigma_2 \stackrel{\alpha}{\longmapsto} \sigma_2'$ then $\sigma_1' \preceq \sigma_2'$

Key:

1. σ_2 has no more internal states than σ_1 has:

$$a \oplus b \preceq a \qquad\qquad a \oplus b \preceq b$$

   and they all allow more capabilities than those in σ_1:

$$a \oplus b \preceq a + b \qquad\qquad a \preceq a + b$$

2. if σ_1 and σ_2 share a common action, the continuations are in the subcontract relation:

$$\mathbf{0} \preceq \sigma \qquad\qquad a.b \preceq a.b + c$$

Subcontract relation

$\preceq$ is the largest relation such that $\sigma_1 \preceq \sigma_2$ implies:

1. if $\sigma_2 \Downarrow \mathrm{R}_2$ then $\sigma_1 \Downarrow \mathrm{R}_1$ with $\mathrm{R}_1 \subseteq \mathrm{R}_2$
2. if $\sigma_1 \overset{\alpha}{\longmapsto} \sigma_1'$ and $\sigma_2 \overset{\alpha}{\longmapsto} \sigma_2'$ then $\sigma_1' \preceq \sigma_2'$

Key:

1. σ_2 has no more internal states than σ_1 has:

$$a \oplus b \preceq a \qquad\qquad a \oplus b \preceq b$$

   and they all allow more capabilities than those in σ_1:

$$a \oplus b \preceq a + b \qquad\qquad a \preceq a + b$$

2. if σ_1 and σ_2 share a common action, the continuations are in the subcontract relation:

$$\mathbf{0} \preceq \sigma \qquad\qquad a.b \preceq a.b + c$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$
\begin{array}{rcll}
a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\
a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\
a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\
a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) &
\end{array}
$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \quad \simeq \quad \bigoplus_{\sigma \Downarrow \mathrm{R}} \quad \sum_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \quad \stackrel{\text{def}}{=} \quad \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \quad \stackrel{\text{def}}{=} \quad \overline{\sigma} \preceq \sigma'$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$
\begin{array}{rcll}
a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\
a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\
a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\
a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) &
\end{array}
$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \quad \simeq \quad \bigoplus_{\sigma \Downarrow \mathrm{R}} \quad \sum_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \quad \stackrel{\mathrm{def}}{=} \quad \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \quad \stackrel{\mathrm{def}}{=} \quad \overline{\sigma} \preceq \sigma'$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$\begin{array}{rcll} a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\ a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\ a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\ a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) & \end{array}$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \quad \simeq \quad \bigoplus_{\sigma \Downarrow \mathrm{R}} \quad \sum_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \quad \stackrel{\text{def}}{=} \quad \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \quad \stackrel{\text{def}}{=} \quad \overline{\sigma} \preceq \sigma'$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$\begin{array}{rcll} a+b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\ a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\ a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\ a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) & \end{array}$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \simeq \bigoplus_{\sigma \Downarrow \mathrm{R}} \sum_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \stackrel{\mathrm{def}}{=} \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \stackrel{\mathrm{def}}{=} \overline{\sigma} \preceq \sigma'$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$
\begin{array}{rcll}
a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\
a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\
a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\
a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) &
\end{array}
$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \simeq \bigoplus_{\sigma \Downarrow \mathrm{R}} \sum_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \stackrel{\mathrm{def}}{=} \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \stackrel{\alpha}{\longmapsto} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \stackrel{\mathrm{def}}{=} \overline{\sigma} \preceq \sigma'$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$\begin{array}{rcll} a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\ a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\ a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\ a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) & \end{array}$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \simeq \bigoplus_{\sigma \Downarrow \mathrm{R}} \sum_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \stackrel{\text{def}}{=} \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \stackrel{\text{def}}{=} \overline{\sigma} \preceq \sigma'$$

Client/service duality and contract compliance

If a client has contract σ, what is the "cheapest" service that interacts successfully with σ?

$$
\begin{array}{rcll}
a + b & \Rightarrow & \overline{a} \oplus \overline{b} & \text{also } \overline{a}\ldots \\
a \oplus b & \Rightarrow & \overline{a} + \overline{b} & \\
a.b + a.c & \Rightarrow & \overline{a}.\overline{b} \oplus \overline{a}.\overline{c} & \text{NO!} \\
a.b + a.c & \Rightarrow & \overline{a}.(\overline{b} + \overline{c}) &
\end{array}
$$

The dual contract of σ is defined on σ's normal form:

$$\sigma \simeq \bigoplus_{\sigma \Downarrow \mathrm{R}} \sum_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \alpha.\sigma'$$

$$\overline{\sigma} \stackrel{\text{def}}{=} \sum_{\sigma \Downarrow \mathrm{R}, \mathrm{R} \neq \emptyset} \bigoplus_{\sigma \xmapsto{\alpha} \sigma', \alpha \in \mathrm{R}} \overline{\alpha}.\overline{\sigma'}$$

Contract compliance:

$$\sigma \ll \sigma' \stackrel{\text{def}}{=} \overline{\sigma} \preceq \sigma'$$

Simple processes: finite CCS without choice

Syntax:

$$P \ ::= \quad \mathbf{0} \quad | \quad a.P \quad | \quad \overline{a}.P \quad | \quad P \setminus a \quad | \quad P \mid P$$

Transition relation:

(IN)
$$a.P \xrightarrow{a} P$$

(OUT)
$$\overline{a}.P \xrightarrow{\overline{a}} P$$

(RES)
$$\frac{P \xrightarrow{\mu} Q \qquad \mu \notin \{a, \overline{a}\}}{P \setminus a \xrightarrow{\mu} Q \setminus a}$$

(PAR)
$$\frac{P \xrightarrow{\mu} Q}{P \mid R \xrightarrow{\mu} Q \mid R}$$

(COM)
$$\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\overline{\alpha}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Synthesizing contracts from processes

The type system:

$$\mathbf{0} \vdash \mathbf{0} \qquad \frac{P \vdash \sigma}{\alpha.P \vdash \alpha.\sigma} \qquad \frac{P \vdash \sigma}{P \setminus a \vdash \sigma \setminus a} \qquad \frac{P \vdash \sigma \quad Q \vdash \sigma'}{P \mid Q \vdash \sigma \mid \sigma'}$$

The $\setminus$ meta-operator behaves like the axioms for $\setminus$ in the aziomatization of must/testing pre-orders:

$$\begin{array}{rcll} a.\sigma \setminus a & = & \mathbf{0} & \\ b.\sigma \setminus a & = & b.(\sigma \setminus b) & \qquad a \neq b \end{array}$$

The $\mid$ meta-operator is just the expansion law (in the testing equivalence):

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

Synthesizing contracts from processes

The type system:

$$\mathbf{0} \vdash \mathbf{0} \qquad \frac{P \vdash \sigma}{\alpha.P \vdash \alpha.\sigma} \qquad \frac{P \vdash \sigma}{P \setminus a \vdash \sigma \setminus a} \qquad \frac{P \vdash \sigma \quad Q \vdash \sigma'}{P \mid Q \vdash \sigma \mid \sigma'}$$

The $\setminus$ meta-operator behaves like the axioms for $\setminus$ in the aziomatization of must/testing pre-orders:

$$\begin{array}{rcll} a.\sigma \setminus a & = & \mathbf{0} & \\ b.\sigma \setminus a & = & b.(\sigma \setminus b) & \quad a \neq b \end{array}$$

The $\mid$ meta-operator is just the expansion law (in the testing equivalence):

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

Synthesizing contracts from processes

The type system:

$$\mathbf{0} \vdash \mathbf{0} \qquad \frac{P \vdash \sigma}{\alpha.P \vdash \alpha.\sigma} \qquad \frac{P \vdash \sigma}{P \setminus a \vdash \sigma \setminus a} \qquad \frac{P \vdash \sigma \quad Q \vdash \sigma'}{P \mid Q \vdash \sigma \mid \sigma'}$$

The $\setminus$ meta-operator behaves like the axioms for $\setminus$ in the aziomatization of must/testing pre-orders:

$$\begin{array}{rcll} a.\sigma \setminus a & = & \mathbf{0} & \\ b.\sigma \setminus a & = & b.(\sigma \setminus b) & \quad a \neq b \end{array}$$

The $\mid$ meta-operator is just the expansion law (in the testing equivalence):

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

The completion property

How do we characterize a "successful interaction" of a system $P \parallel Q$?

System transition:

- if $P \xrightarrow{\tau} P'$ then $P \parallel Q \longrightarrow P' \parallel Q$;
- if $Q \xrightarrow{\tau} Q'$ then $P \parallel Q \longrightarrow P \parallel Q'$;
- if $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

P complies with Q, noted $P \ll Q$, if either

1. $P \stackrel{\alpha}{\not\longrightarrow}$, or
2. $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \ll Q'$

Theorem. If $P \vdash \sigma$, $Q \vdash \sigma'$, and $\sigma \ll \sigma'$ then $P \ll Q$

The completion property

How do we characterize a "successful interaction" of a system $P \parallel Q$?

System transition:

- if $P \xrightarrow{\tau} P'$ then $P \parallel Q \longrightarrow P' \parallel Q$;
- if $Q \xrightarrow{\tau} Q'$ then $P \parallel Q \longrightarrow P \parallel Q'$;
- if $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

P complies with Q, noted $P \ll Q$, if either

1. $P \stackrel{\alpha}{\not\longrightarrow}$, or
2. $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \ll Q'$

Theorem. If $P \vdash \sigma$, $Q \vdash \sigma'$, and $\sigma \ll \sigma'$ then $P \ll Q$

The completion property

How do we characterize a "successful interaction" of a system $P \parallel Q$?

System transition:

- if $P \xrightarrow{\tau} P'$ then $P \parallel Q \longrightarrow P' \parallel Q$;
- if $Q \xrightarrow{\tau} Q'$ then $P \parallel Q \longrightarrow P \parallel Q'$;
- if $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

P complies with Q, noted $P \ll Q$, if either

1. $P \stackrel{\alpha}{\not\longrightarrow}$, or
2. $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \ll Q'$

Theorem. If $P \vdash \sigma$, $Q \vdash \sigma'$, and $\sigma \ll \sigma'$ then $P \ll Q$

The completion property

How do we characterize a "successful interaction" of a system $P \parallel Q$?

System transition:

- if $P \xrightarrow{\tau} P'$ then $P \parallel Q \longrightarrow P' \parallel Q$;
- if $Q \xrightarrow{\tau} Q'$ then $P \parallel Q \longrightarrow P \parallel Q'$;
- if $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

P complies with Q, noted $P \ll Q$, if either

1. $P \stackrel{\alpha}{\not\longrightarrow}$, or
2. $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \ll Q'$

Theorem. If $P \vdash \sigma$, $Q \vdash \sigma'$, and $\sigma \ll \sigma'$ then $P \ll Q$

Open issues

- is $\preceq$ the right compatibility relation? It is *not* a pre-congruence w.r.t. $|$
  $\preceq$ is good for searching, not for typing (subsumption)
- $\ll$ is sufficient but not necessary:

  $$P \equiv x \mid \overline{x} \qquad Q \equiv \mathbf{0} \qquad P \ll Q \quad \text{however} \quad (x.\overline{x} + \overline{x}.x) \oplus \mathbf{0} \not\ll \mathbf{0}$$

  Is $x \mid \overline{x}$ "valid"?
- experiment the effectiveness of contracts (PiDuce)

Open issues

- is $\preceq$ the right compatibility relation? It is *not* a pre-congruence w.r.t. $|$
  $\preceq$ is good for searching, not for typing (subsumption)
- $\ll$ is sufficient but not necessary:

  $$P \equiv x \mid \overline{x} \qquad Q \equiv \mathbf{0} \qquad P \ll Q \quad \text{however} \quad (x.\overline{x} + \overline{x}.x) \oplus \mathbf{0} \not\ll \mathbf{0}$$

  Is $x \mid \overline{x}$ "valid"?
- experiment the effectiveness of contracts (PiDuce)

Open issues

- is $\preceq$ the right compatibility relation? It is *not* a pre-congruence w.r.t. $|$
  $\preceq$ is good for searching, not for typing (subsumption)
- $\ll$ is sufficient but not necessary:

$$P \equiv x \mid \overline{x} \qquad Q \equiv \mathbf{0} \qquad P \ll Q \quad \text{however} \quad (x.\overline{x} + \overline{x}.x) \oplus \mathbf{0} \not\ll \mathbf{0}$$

  Is $x \mid \overline{x}$ “valid”?
- experiment the effectiveness of contracts (PiDuce)

Open issues

- is $\preceq$ the right compatibility relation? It is *not* a pre-congruence w.r.t. $|$
  $\preceq$ is good for searching, not for typing (subsumption)
- $\ll$ is sufficient but not necessary:

$$P \equiv x \mid \overline{x} \qquad Q \equiv \mathbf{0} \qquad P \ll Q \quad \text{however} \quad (x.\overline{x} + \overline{x}.x) \oplus \mathbf{0} \not\ll \mathbf{0}$$

  Is $x \mid \overline{x}$ "valid"?
- experiment the effectiveness of contracts (PiDuce)

Future work

- Recursive contracts

$$\mu x.\sigma$$

How do we infer contracts from processes? Syntactic restrictions over processes or regular approximations?

- Name passing:

$$a(x).\overline{x}$$

- Adapting $\preceq$ to asynchronous communication
- Relationship with linear logic and set-theoretic interpretation of contracts
- Contract isomorphisms and automatic generation of adapters:

$$a.b \iff b.a$$

Future work

- Recursive contracts

$$\mu x.\sigma$$

How do we infer contracts from processes? Syntactic restrictions over processes or regular approximations?

- Name passing:

$$a(x).\overline{x}$$

- Adapting $\preceq$ to asynchronous communication
- Relationship with linear logic and set-theoretic interpretation of contracts
- Contract isomorphisms and automatic generation of adapters:

$$a.b \iff b.a$$

Future work

- Recursive contracts

$$\mu x.\sigma$$

  How do we infer contracts from processes? Syntactic restrictions over processes or regular approximations?
- Name passing:

$$a(x).\overline{x}$$

- Adapting $\preceq$ to asynchronous communication
- Relationship with linear logic and set-theoretic interpretation of contracts
- Contract isomorphisms and automatic generation of adapters:

$$a.b \iff b.a$$

Future work

- Recursive contracts

$$\mu x.\sigma$$

  How do we infer contracts from processes? Syntactic restrictions over processes or regular approximations?

- Name passing:

$$a(x).\overline{x}$$

- Adapting $\preceq$ to asynchronous communication
- Relationship with linear logic and set-theoretic interpretation of contracts
- Contract isomorphisms and automatic generation of adapters:

$$a.b \iff b.a$$

Future work

- Recursive contracts

$$\mu x.\sigma$$

  How do we infer contracts from processes? Syntactic restrictions over processes or regular approximations?

- Name passing:

$$a(x).\overline{x}$$

- Adapting $\preceq$ to asynchronous communication
- Relationship with linear logic and set-theoretic interpretation of contracts
- Contract isomorphisms and automatic generation of adapters:

$$a.b \iff b.a$$