

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)

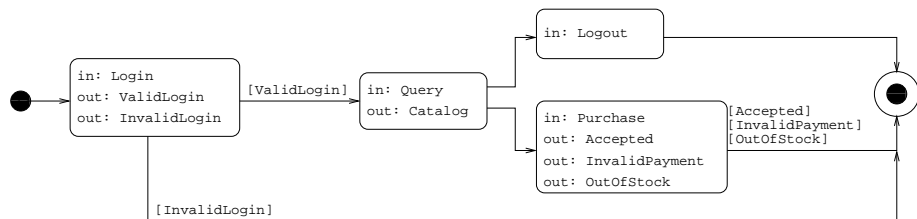
```
<operation name="A"  
  pattern="http://www.w3.org/2006/01/wsdl/in-only">  
  <input messageLabel="In"/>  
</operation>
```

```
<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



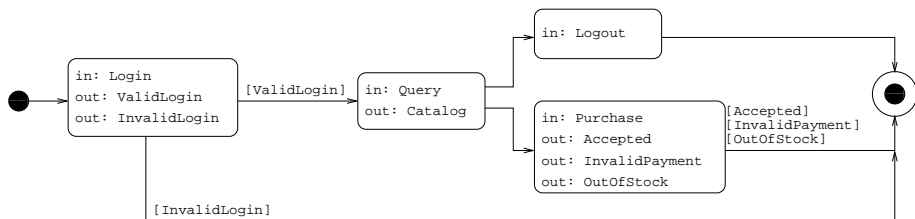
+ distinction between internal and external choice

+ possibly cyclic patterns

Contracts in WSCL

Focus on the dynamic interface:

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



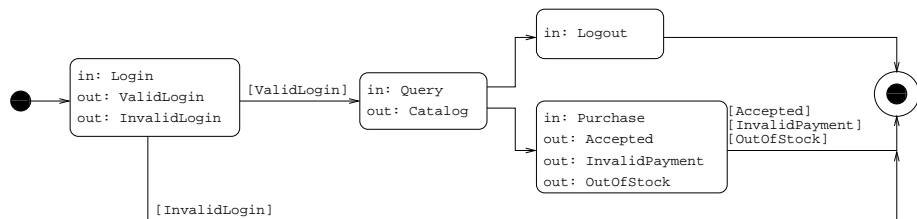
+ distinction between internal and external choice

+ 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

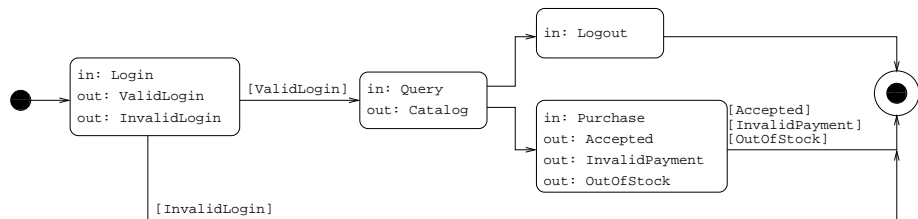
Encoding MEPs into contracts

```
<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>
```

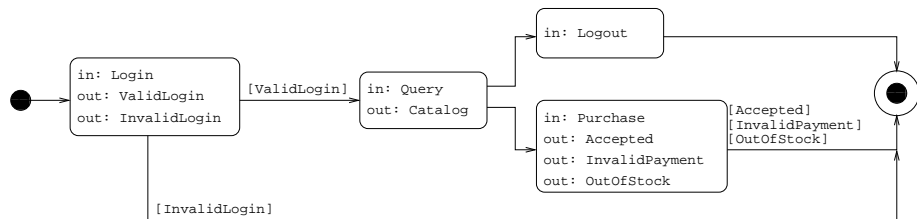
$$\begin{array}{l} A \stackrel{\text{def}}{=} \text{In}.\overline{\text{End}} \\ B \stackrel{\text{def}}{=} \text{In}.\overline{(\text{End} \oplus \text{Fault}.\overline{\text{End}})} \end{array}$$

Encoding WSCL into contracts



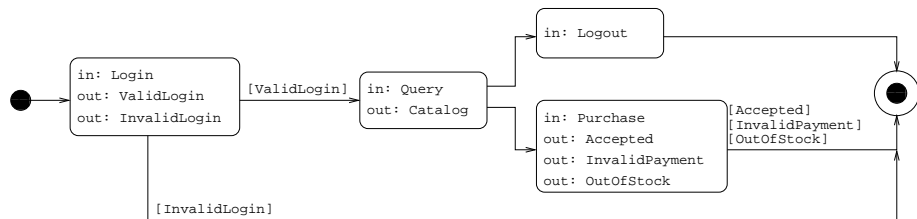
$\text{Login.}(\overline{\text{InvalidLogin.End}} \oplus \overline{\text{ValidLogin}} \text{Query.} \overline{\text{Catalog.}}(\text{Logout.End} + \text{Purchase.}(\overline{\text{Accepted.End}} \oplus \overline{\text{InvalidPayment.End}} \oplus \overline{\text{OutOfStock.End}})))$

Encoding WSCL into contracts



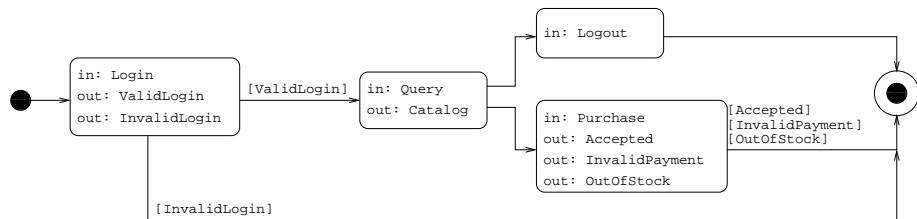
$\text{Login.}(\overline{\text{InvalidLogin.End}} \oplus \overline{\text{ValidLogin.Query.Catalog.}}(\text{Logout.End} + \text{Purchase.}(\overline{\text{Accepted.End}} \oplus \overline{\text{InvalidPayment.End}} \oplus \overline{\text{OutOfStock.End}})))$

Encoding WSCL into contracts



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

Encoding WSCL into contracts


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

A formal contract language

contracts $\sigma ::=$

- $\mathbf{0}$ (*void*)
- $\alpha.\sigma$ (*action prefix*)
- $\sigma + \sigma$ (*external choice*)
- $\sigma \oplus \sigma$ (*internal choice*)

actions $\alpha ::=$

- a (*name*)
- \bar{a} (*co-name*)

Names represent *types, operations, ...*

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

A formal contract language

contracts $\sigma ::=$

- $\mathbf{0}$ (*void*)
- $\alpha.\sigma$ (*action prefix*)
- $\sigma + \sigma$ (*external choice*)
- $\sigma \oplus \sigma$ (*internal choice*)

actions $\alpha ::=$

- a (*name*)
- \bar{a} (*co-name*)

Names represent **types**, **operations**, ...

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

A formal contract language

contracts $\sigma ::=$

- $\mathbf{0}$ (*void*)
- $\alpha.\sigma$ (*action prefix*)
- $\sigma + \sigma$ (*external choice*)
- $\sigma \oplus \sigma$ (*internal choice*)

actions $\alpha ::=$

- a (*name*)
- \bar{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

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

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

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

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

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

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

(\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

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

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

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

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

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

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

(\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

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

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

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

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

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

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

(\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

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

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

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

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

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

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

(\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:

$$a.b + a.c \xrightarrow{a} b \oplus c$$

$$\alpha.\sigma \xrightarrow{\alpha} \sigma$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 + \sigma_2 \xrightarrow{\alpha} \sigma'_1 \oplus \sigma'_2}$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 + \sigma_2 \xrightarrow{\alpha} \sigma'_1}$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 \oplus \sigma_2 \xrightarrow{\alpha} \sigma'_1 \oplus \sigma'_2}$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 \oplus \sigma_2 \xrightarrow{\alpha} \sigma'_1}$$

Contracts: transition relation

Interacting party's point of view:

$$a.b + a.c \xrightarrow{a} b \oplus c$$

$$\alpha.\sigma \xrightarrow{\alpha} \sigma$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 + \sigma_2 \xrightarrow{\alpha} \sigma'_1 \oplus \sigma'_2}$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 + \sigma_2 \xrightarrow{\alpha} \sigma'_1}$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 \oplus \sigma_2 \xrightarrow{\alpha} \sigma'_1 \oplus \sigma'_2}$$

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \xrightarrow{\alpha} \sigma'_2}{\sigma_1 \oplus \sigma_2 \xrightarrow{\alpha} \sigma'_1}$$

Contracts: ready sets

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

$$\mathbf{0} \Downarrow \emptyset$$

$$\alpha.\sigma \Downarrow \{\alpha\}$$

$$(\sigma + \sigma') \Downarrow R \cup R' \quad \text{if } \sigma \Downarrow R \text{ and } \sigma' \Downarrow R'$$

$$(\sigma \oplus \sigma') \Downarrow R \quad \text{if either } \sigma \Downarrow R \text{ or } \sigma' \Downarrow R$$

Example of nondeterministic contract/service:

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

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

Example of deterministic contract/service:

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

Contracts: ready sets

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

$$\mathbf{0} \Downarrow \emptyset$$

$$\alpha.\sigma \Downarrow \{\alpha\}$$

$$(\sigma + \sigma') \Downarrow R \cup R' \quad \text{if } \sigma \Downarrow R \text{ and } \sigma' \Downarrow R'$$

$$(\sigma \oplus \sigma') \Downarrow R \quad \text{if either } \sigma \Downarrow R \text{ or } \sigma' \Downarrow R$$

Example of nondeterministic contract/service:

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

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

Example of deterministic contract/service:

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

Contracts: ready sets

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

$$\mathbf{0} \Downarrow \emptyset$$

$$\alpha.\sigma \Downarrow \{\alpha\}$$

$$(\sigma + \sigma') \Downarrow R \cup R' \quad \text{if } \sigma \Downarrow R \text{ and } \sigma' \Downarrow R'$$

$$(\sigma \oplus \sigma') \Downarrow R \quad \text{if either } \sigma \Downarrow R \text{ or } \sigma' \Downarrow R$$

Example of nondeterministic contract/service:

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

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

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 $\sigma_2 \Downarrow R_2$ then $\sigma_1 \Downarrow R_1$ with $R_1 \subseteq R_2$
- 2 if $\sigma_1 \xrightarrow{\alpha} \sigma'_1$ and $\sigma_2 \xrightarrow{\alpha} \sigma'_2$ then $\sigma'_1 \preceq \sigma'_2$

Key:

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

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

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:

$$0 \preceq \sigma \qquad a.b \preceq a.b + c$$

Subcontract relation

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

- 1 if $\sigma_2 \Downarrow R_2$ then $\sigma_1 \Downarrow R_1$ with $R_1 \subseteq R_2$
- 2 if $\sigma_1 \xrightarrow{\alpha} \sigma'_1$ and $\sigma_2 \xrightarrow{\alpha} \sigma'_2$ then $\sigma'_1 \preceq \sigma'_2$

Key:

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

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

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:

$$0 \preceq \sigma \qquad a.b \preceq a.b + c$$

Subcontract relation

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

- 1 if $\sigma_2 \Downarrow R_2$ then $\sigma_1 \Downarrow R_1$ with $R_1 \subseteq R_2$
- 2 if $\sigma_1 \xrightarrow{\alpha} \sigma'_1$ and $\sigma_2 \xrightarrow{\alpha} \sigma'_2$ then $\sigma'_1 \preceq \sigma'_2$

Key:

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

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

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:

$$0 \preceq \sigma \qquad a.b \preceq a.b + c$$

Subcontract relation

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

- 1 if $\sigma_2 \Downarrow R_2$ then $\sigma_1 \Downarrow R_1$ with $R_1 \subseteq R_2$
- 2 if $\sigma_1 \xrightarrow{\alpha} \sigma'_1$ and $\sigma_2 \xrightarrow{\alpha} \sigma'_2$ then $\sigma'_1 \preceq \sigma'_2$

Key:

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

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

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:

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

Client/service duality and contract compliance

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

$$\begin{aligned}a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\a \oplus b &\Rightarrow \bar{a} + \bar{b} \\a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)}\end{aligned}$$

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

$$\begin{aligned}\sigma &\simeq \bigoplus_{\sigma \downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}'\end{aligned}$$

Contract compliance:

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

Client/service duality and contract compliance

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

$$\begin{aligned} a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\ a \oplus b &\Rightarrow \bar{a} + \bar{b} \\ a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\ a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)} \end{aligned}$$

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

$$\begin{aligned} \sigma &\simeq \bigoplus_{\sigma \downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}' \end{aligned}$$

Contract compliance:

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

Client/service duality and contract compliance

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

$$\begin{aligned} a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\ a \oplus b &\Rightarrow \bar{a} + \bar{b} \\ a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\ a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)} \end{aligned}$$

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

$$\begin{aligned} \sigma &\simeq \bigoplus_{\sigma \downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}' \end{aligned}$$

Contract compliance:

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

Client/service duality and contract compliance

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

$$\begin{aligned} a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\ a \oplus b &\Rightarrow \bar{a} + \bar{b} \\ a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\ a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)} \end{aligned}$$

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

$$\begin{aligned} \sigma &\simeq \bigoplus_{\sigma \downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}' \end{aligned}$$

Contract compliance:

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

Client/service duality and contract compliance

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

$$\begin{aligned} a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\ a \oplus b &\Rightarrow \bar{a} + \bar{b} \\ a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\ a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)} \end{aligned}$$

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

$$\begin{aligned} \sigma &\simeq \bigoplus_{\sigma \downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}' \end{aligned}$$

Contract compliance:

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

Client/service duality and contract compliance

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

$$\begin{aligned} a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\ a \oplus b &\Rightarrow \bar{a} + \bar{b} \\ a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\ a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)} \end{aligned}$$

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

$$\begin{aligned} \sigma &\simeq \bigoplus_{\sigma \downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}' \end{aligned}$$

Contract compliance:

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

Client/service duality and contract compliance

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

$$\begin{aligned} a + b &\Rightarrow \bar{a} \oplus \bar{b} && \text{also } \bar{a} \dots \\ a \oplus b &\Rightarrow \bar{a} + \bar{b} \\ a.b + a.c &\Rightarrow \bar{a}.\bar{b} \oplus \bar{a}.\bar{c} && \text{NO!} \\ a.b + a.c &\Rightarrow \bar{a}.\overline{(b + c)} \end{aligned}$$

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

$$\begin{aligned} \sigma &\simeq \bigoplus_{\sigma \Downarrow R} \sum_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \alpha.\sigma' \\ \bar{\sigma} &\stackrel{\text{def}}{=} \sum_{\sigma \Downarrow R, R \neq \emptyset} \bigoplus_{\sigma \xrightarrow{\alpha} \sigma', \alpha \in R} \bar{\alpha}.\bar{\sigma}' \end{aligned}$$

Contract compliance:

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

Simple processes: finite CCS without choice

Syntax:

$$P ::= \mathbf{0} \mid a.P \mid \bar{a}.P \mid P \setminus a \mid P \mid P$$

Transition relation:

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

$$\text{(OUT)} \quad \bar{a}.P \xrightarrow{\bar{a}} P$$

$$\text{(RES)} \quad \frac{P \xrightarrow{\mu} Q \quad \mu \notin \{a, \bar{a}\}}{P \setminus a \xrightarrow{\mu} Q \setminus a}$$

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

$$\text{(COM)} \quad \frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\bar{\alpha}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Synthesizing contracts from processes

The **type system**:

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

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

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

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

$$\begin{aligned} a | b &= a.b + b.a \\ a | \bar{a}.b &= (a.\bar{a}.b + \bar{a}.(a | b) + b) \oplus b \end{aligned}$$

Synthesizing contracts from processes

The **type system**:

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

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

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

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

$$\begin{aligned} a | b &= a.b + b.a \\ a | \bar{a}.b &= (a.\bar{a}.b + \bar{a}.(a | b) + b) \oplus b \end{aligned}$$

Synthesizing contracts from processes

The **type system**:

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

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

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

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

$$\begin{aligned} a | b &= a.b + b.a \\ a | \bar{a}.b &= (a.\bar{a}.b + \bar{a}.(a | b) + b) \oplus b \end{aligned}$$

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 $Q \xrightarrow{\bar{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

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

- 1 $P \xrightarrow{\alpha}$, 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 $Q \xrightarrow{\bar{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

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

- 1 $P \xrightarrow{\alpha}$, 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 $Q \xrightarrow{\bar{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

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

- 1 $P \not\xrightarrow{\alpha}$, 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 $Q \xrightarrow{\bar{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$.

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

- 1 $P \not\xrightarrow{\alpha}$, 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:

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

Is $x \mid \bar{x}$ “valid”?

- experiment the effectiveness of contracts (PiDuce)

- 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:

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

Is $x \mid \bar{x}$ “valid”?

- experiment the effectiveness of contracts (PiDuce)

- 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:

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

Is $x \mid \bar{x}$ “valid”?

- experiment the effectiveness of contracts (PiDuce)

- 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:

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

Is $x \mid \bar{x}$ “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:

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

- 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:

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

- 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:

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

- 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:

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

- 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:

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

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

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