

A Theory of Contracts for Web Services

Giuseppe Castagna, Nils Gesbert, Luca Padovani

Université Paris 7, Université Paris Sud, Università di Urbino

January 20, 2007

Summary

Part I

- Contracts and technologies for Web Services
- A language of contracts
- Desirable properties of the subcontract relation

Part II

- Contract semantics
- Searching for a subcontract relation

Part III

- Filters as explicit coercions
- The subcontract relation
- Contracts and processes

Future and ongoing work

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?

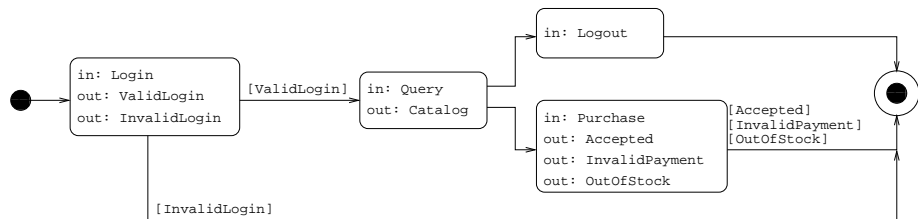
Use

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

Focus

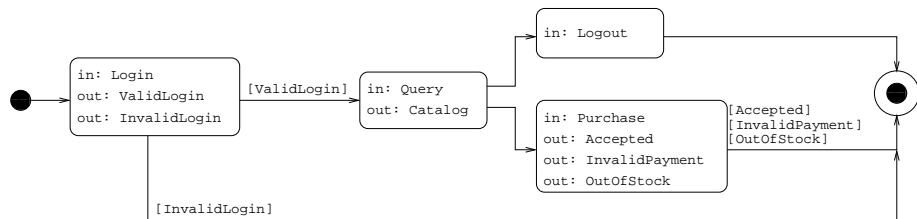
- communication between two parties (no choreography)

Contracts in WSCL



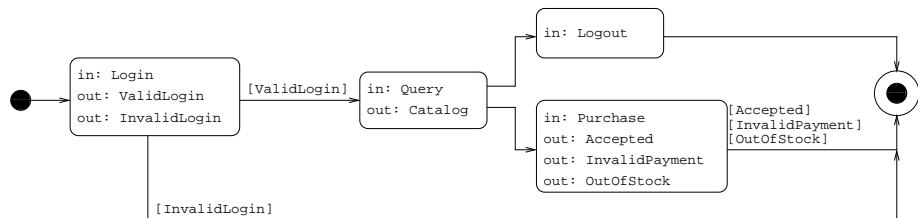
$\text{Login.}(\overline{\text{InvalidLogin.End}} \oplus \overline{\text{ValidLogin}} \text{Query.} \overline{\text{Catalog.}}(\text{Logout.End} + \text{Purchase.}(\overline{\text{Accepted.End}} \oplus \overline{\text{InvalidPayment.End}} \oplus \overline{\text{OutOfStock.End}})))$

Contracts in WSCL



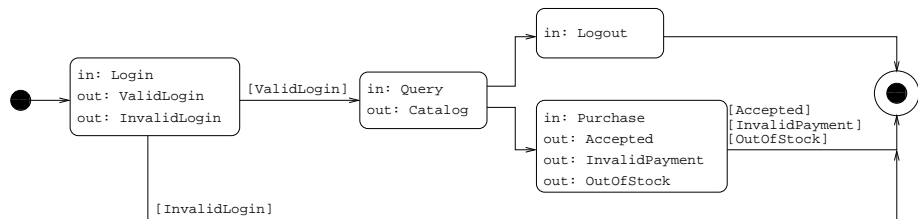
$\text{Login.}(\overline{\text{InvalidLogin.End}} \oplus \overline{\text{ValidLogin.Query.Catalog.}}(\text{Logout.End} + \text{Purchase.}(\overline{\text{Accepted.End}} \oplus \overline{\text{InvalidPayment.End}} \oplus \overline{\text{OutOfStock.End}})))$

Contracts in WSCL



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

Contracts in WSCL



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

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 (*receive*)
- \bar{a} (*send*)

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

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 (*receive*)
- \bar{a} (*send*)

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

Desirable properties of the **subcontract** relation \preceq

Intuition: a client that is “happy” with a service σ should be equally “happy” with any service σ' such that “does more”: $\sigma \preceq \sigma'$

- *reduce* (internal) nondeterminism

$$a \oplus b \preceq a$$

$$\overline{\text{Accepted}} \oplus \overline{\text{InvalidPayment}} \preceq \overline{\text{Accepted}}$$

- *increase* provided capabilities

$$a \preceq a.b$$

$$a \preceq a + b$$

$$0 \preceq \sigma$$

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

- \preceq should be a precongruence w.r.t. $+$, \oplus , prefix, and possibly more operators (service substitution)

Desirable properties of the **subcontract** relation \preceq

Intuition: a client that is “happy” with a service σ should be equally “happy” with any service σ' such that “does more”: $\sigma \preceq \sigma'$

- *reduce* (internal) nondeterminism

$$a \oplus b \preceq a$$

$$\overline{\text{Accepted}} \oplus \overline{\text{InvalidPayment}} \preceq \overline{\text{Accepted}}$$

- *increase* provided capabilities

$$a \preceq a.b$$

$$a \preceq a + b$$

$$0 \preceq \sigma$$

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

- \preceq should be a precongrence w.r.t. $+$, \oplus , prefix, and possibly more operators (service substitution)

Desirable properties of the **subcontract** relation \preceq

Intuition: a client that is “happy” with a service σ should be equally “happy” with any service σ' such that “does more”: $\sigma \preceq \sigma'$

- *reduce* (internal) nondeterminism

$$a \oplus b \preceq a$$

$$\overline{\text{Accepted}} \oplus \overline{\text{InvalidPayment}} \preceq \overline{\text{Accepted}}$$

- *increase* provided capabilities

$$a \preceq a.b$$

$$a \preceq a + b$$

$$\mathbf{0} \preceq \sigma$$

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

- \preceq should be a precongurence w.r.t. $+$, \oplus , prefix, and possibly more operators (service substitution)

Desirable properties of the **subcontract** relation \preceq

Intuition: a client that is “happy” with a service σ should be equally “happy” with any service σ' such that “does more”: $\sigma \preceq \sigma'$

- *reduce* (internal) nondeterminism

$$a \oplus b \preceq a$$

$$\overline{\text{Accepted}} \oplus \overline{\text{InvalidPayment}} \preceq \overline{\text{Accepted}}$$

- *increase* provided capabilities

$$a \preceq a.b$$

$$a \preceq a + b$$

$$\mathbf{0} \preceq \sigma$$

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

- \preceq should be a precongruence w.r.t. $+$, \oplus , prefix, and possibly more operators (service substitution)

Contract semantics: transition relation

Interacting party's point of view

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

Transition rules

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

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

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

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

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

Contract semantics: transition relation

Interacting party's point of view

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

Transition rules

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

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

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \not\xrightarrow{\alpha}}{\sigma_1 + \sigma_2 \xrightarrow{\alpha} \sigma'_1}$$

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

$$\frac{\sigma_1 \xrightarrow{\alpha} \sigma'_1 \quad \sigma_2 \not\xrightarrow{\alpha}}{\sigma_1 \oplus \sigma_2 \xrightarrow{\alpha} \sigma'_1}$$

Contract semantics: 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 internal choice

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

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

Example of external choice

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

Contract semantics: 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 internal choice

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

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

Example of external choice

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

Contract semantics: 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 internal choice

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

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

Example of external choice

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

“Happiness” is (graceful) termination

Strong compliance

$\sigma_c \dashv_S \sigma_s$ implies that

- 1 for all $R_c \neq \emptyset$ and R_s such that $\sigma_c \Downarrow R_c$ and $\sigma_s \Downarrow R_s$, we have $\text{co}(R_c) \cap R_s \neq \emptyset$, and
- 2 for all α , $\sigma_c \xrightarrow{\bar{\alpha}} \sigma'_c$ and $\sigma_s \xrightarrow{\alpha} \sigma'_s$ implies $\sigma'_c \dashv_S \sigma'_s$.

Read: whenever the client has not terminated, interaction with the service should be guaranteed

Examples

- $\bar{a} + \bar{b} \dashv_S a \oplus b$
- $\bar{a} \not\vdash_S a \oplus b$
- $\bar{a} + \bar{b} \dashv_S a + b$
- $\bar{a}.\bar{b} + \bar{a}.\bar{c} \not\vdash_S a.b + a.c$ (exercise)

“Happiness” is (graceful) termination

Strong compliance

$\sigma_c \dashv_S \sigma_s$ implies that

- 1 for all $R_c \neq \emptyset$ and R_s such that $\sigma_c \Downarrow R_c$ and $\sigma_s \Downarrow R_s$, we have $\text{co}(R_c) \cap R_s \neq \emptyset$, and
- 2 for all α , $\sigma_c \xrightarrow{\bar{\alpha}} \sigma'_c$ and $\sigma_s \xrightarrow{\alpha} \sigma'_s$ implies $\sigma'_c \dashv_S \sigma'_s$.

Read: whenever the client has not terminated, interaction with the service should be guaranteed

Examples

- $\bar{a} + \bar{b} \dashv_S a \oplus b$
- $\bar{a} \not\vdash_S a \oplus b$
- $\bar{a} + \bar{b} \dashv_S a + b$
- $\bar{a}.\bar{b} + \bar{a}.\bar{c} \not\vdash_S a.b + a.c$ (exercise)

“Happiness” is (graceful) termination

Strong compliance

$\sigma_c \dashv_S \sigma_s$ implies that

- 1 for all $R_c \neq \emptyset$ and R_s such that $\sigma_c \Downarrow R_c$ and $\sigma_s \Downarrow R_s$, we have $\text{co}(R_c) \cap R_s \neq \emptyset$, and
- 2 for all α , $\sigma_c \xrightarrow{\bar{\alpha}} \sigma'_c$ and $\sigma_s \xrightarrow{\alpha} \sigma'_s$ implies $\sigma'_c \dashv_S \sigma'_s$.

Read: whenever the client has not terminated, interaction with the service should be guaranteed

Examples

- $\bar{a} + \bar{b} \dashv_S a \oplus b$
- $\bar{a} \not\vdash_S a \oplus b$
- $\bar{a} + \bar{b} \dashv_S a + b$
- $\bar{a}.\bar{b} + \bar{a}.\bar{c} \not\vdash_S a.b + a.c$ (exercise)

“Happiness” is (graceful) termination

Strong compliance

$\sigma_c \dashv_S \sigma_s$ implies that

- 1 for all $R_c \neq \emptyset$ and R_s such that $\sigma_c \Downarrow R_c$ and $\sigma_s \Downarrow R_s$, we have $\text{co}(R_c) \cap R_s \neq \emptyset$, and
- 2 for all α , $\sigma_c \xrightarrow{\bar{\alpha}} \sigma'_c$ and $\sigma_s \xrightarrow{\alpha} \sigma'_s$ implies $\sigma'_c \dashv_S \sigma'_s$.

Read: whenever the client has not terminated, interaction with the service should be guaranteed

Examples

- $\bar{a} + \bar{b} \dashv_S a \oplus b$
- $\bar{a} \not\vdash_S a \oplus b$
- $\bar{a} + \bar{b} \dashv_S a + b$
- $\bar{a}.\bar{b} + \bar{a}.\bar{c} \not\vdash_S a.b + a.c$ (exercise)

“Happiness” is (graceful) termination

Strong compliance

$\sigma_c \dashv_S \sigma_s$ implies that

- 1 for all $R_c \neq \emptyset$ and R_s such that $\sigma_c \Downarrow R_c$ and $\sigma_s \Downarrow R_s$, we have $\text{co}(R_c) \cap R_s \neq \emptyset$, and
- 2 for all α , $\sigma_c \xrightarrow{\bar{\alpha}} \sigma'_c$ and $\sigma_s \xrightarrow{\alpha} \sigma'_s$ implies $\sigma'_c \dashv_S \sigma'_s$.

Read: whenever the client has not terminated, interaction with the service should be guaranteed

Examples

- $\bar{a} + \bar{b} \dashv_S a \oplus b$
- $\bar{a} \not\vdash_S a \oplus b$
- $\bar{a} + \bar{b} \dashv_S a + b$
- $\bar{a}.\bar{b} + \bar{a}.\bar{c} \not\vdash_S a.b + a.c$ (exercise)

Subcontract relation I: “bibliographic” approach

Is there a well-known preorder that can be used as the subcontract relation? The *must preorder* \sqsubseteq !

$\sigma_1 \sqsubseteq \sigma_2$ implies that

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

- $a \oplus b \sqsubseteq a$
- well-known
- precongruence w.r.t. $+$, \oplus
- $a \not\sqsubseteq a + b$
- $0 \not\sqsubseteq \sigma$

Subcontract relation I: “bibliographic” approach

Is there a well-known preorder that can be used as the subcontract relation? The *must preorder* \sqsubseteq !

$\sigma_1 \sqsubseteq \sigma_2$ implies that

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

● $a \oplus b \sqsubseteq a$

● well-known

● precongruence w.r.t. $+$, \oplus

● $a \not\sqsubseteq a + b$

● $0 \not\sqsubseteq \sigma$

Subcontract relation I: “bibliographic” approach

Is there a well-known preorder that can be used as the subcontract relation? The *must preorder* \sqsubseteq !

$\sigma_1 \sqsubseteq \sigma_2$ implies that

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

● $a \oplus b \sqsubseteq a$

● well-known

● precongruence w.r.t. $+$, \oplus

● $a \not\sqsubseteq a + b$

● $0 \not\sqsubseteq \sigma$

Subcontract relation I: “bibliographic” approach

Is there a well-known preorder that can be used as the subcontract relation? The *must preorder* \sqsubseteq !

$\sigma_1 \sqsubseteq \sigma_2$ implies that

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

● $a \oplus b \sqsubseteq a$

● well-known

● precongruence w.r.t. $+$, \oplus

● $a \not\sqsubseteq a + b$

● $\mathbf{0} \not\sqsubseteq \sigma$

Subcontract relation II: semantic approach

The “semantics” of a contract is the set of clients that are happy with it

$$\sigma_1 \preceq_{\text{strong}} \sigma_2 \stackrel{\text{def}}{\iff} \sigma_c \dashv_S \sigma_1 \Rightarrow \sigma_c \dashv_S \sigma_2$$

- semantic subtyping (union, intersection, negation \Rightarrow powerful search operators)
- $0 \preceq_{\text{strong}} \sigma$
- $a \not\preceq_{\text{strong}} a + b$
- not a precongruence w.r.t. $+$

Subcontract relation II: semantic approach

The “semantics” of a contract is the set of clients that are happy with it

$$\sigma_1 \preceq_{\text{strong}} \sigma_2 \stackrel{\text{def}}{\iff} \sigma_c \dashv_S \sigma_1 \Rightarrow \sigma_c \dashv_S \sigma_2$$

- semantic subtyping (union, intersection, negation \Rightarrow powerful search operators)
- $\mathbf{0} \preceq_{\text{strong}} \sigma$
- $a \not\preceq_{\text{strong}} a + b$
- not a precongruence w.r.t. $+$

Subcontract relation II: semantic approach

The “semantics” of a contract is the set of clients that are happy with it

$$\sigma_1 \preceq_{\text{strong}} \sigma_2 \stackrel{\text{def}}{\iff} \sigma_c \dashv_S \sigma_1 \Rightarrow \sigma_c \dashv_S \sigma_2$$

- semantic subtyping (union, intersection, negation \Rightarrow powerful search operators)
- $\mathbf{0} \preceq_{\text{strong}} \sigma$
- $a \not\preceq_{\text{strong}} a + b$
- not a precongruence w.r.t. $+$

Subcontract relation III: syntactic approach

If a client C has contract σ , what is the “cheapest” contract that a service should expose to make C happy?

$$\overline{a \oplus b} = \bar{a} + \bar{b} \quad \text{and} \quad \overline{a + b} = \bar{a} \oplus \bar{b}$$

$$\sigma_1 \times \sigma_2 \stackrel{\text{def}}{\iff} \bar{\sigma}_1 \dashv_S \sigma_2$$

● $a \times a \oplus b$ ● $a \times a + b$ ● $0 \times \sigma$

● \times is not transitive:

$$a \oplus b.\bar{c} \times a \quad \text{and} \quad a \times a + b.\bar{d}$$

but

$$a \oplus b.\bar{c} \not\times a + b.\bar{d}$$

Subcontract relation III: syntactic approach

If a client C has contract σ , what is the “cheapest” contract that a service should expose to make C happy?

$$\overline{a \oplus b} = \bar{a} + \bar{b} \quad \text{and} \quad \overline{a + b} = \bar{a} \oplus \bar{b}$$

$$\sigma_1 \times \sigma_2 \stackrel{\text{def}}{\iff} \bar{\sigma}_1 \dashv_S \sigma_2$$

● $a \times a \oplus b$

● $a \times a + b$

● $0 \times \sigma$

● \times is **not transitive**:

$$a \oplus b.\bar{c} \times a \quad \text{and} \quad a \times a + b.\bar{d}$$

but

$$a \oplus b.\bar{c} \not\times a + b.\bar{d}$$

Subcontract relation III: syntactic approach

If a client C has contract σ , what is the “cheapest” contract that a service should expose to make C happy?

$$\overline{a \oplus b} = \bar{a} + \bar{b} \quad \text{and} \quad \overline{a + b} = \bar{a} \oplus \bar{b}$$

$$\sigma_1 \times \sigma_2 \stackrel{\text{def}}{\iff} \bar{\sigma}_1 \dashv_S \sigma_2$$

● $a \times a \oplus b$

● $a \times a + b$

● $0 \times \sigma$

● \times is not transitive:

$$a \oplus b.\bar{c} \times a \quad \text{and} \quad a \times a + b.\bar{d}$$

but

$$a \oplus b.\bar{c} \not\times a + b.\bar{d}$$

Subcontract relation III: syntactic approach

If a client C has contract σ , what is the “cheapest” contract that a service should expose to make C happy?

$$\overline{a \oplus b} = \bar{a} + \bar{b} \quad \text{and} \quad \overline{a + b} = \bar{a} \oplus \bar{b}$$

$$\sigma_1 \times \sigma_2 \stackrel{\text{def}}{\iff} \bar{\sigma}_1 \dashv_S \sigma_2$$

● $a \times a \oplus b$ ● $a \times a + b$ ● $\mathbf{0} \times \sigma$

● \times is not transitive:

$$a \oplus b.\bar{c} \times a \quad \text{and} \quad a \times a + b.\bar{d}$$

but

$$a \oplus b.\bar{c} \not\times a + b.\bar{d}$$

Subcontract relation III: syntactic approach

If a client C has contract σ , what is the “cheapest” contract that a service should expose to make C happy?

$$\overline{a \oplus b} = \bar{a} + \bar{b} \quad \text{and} \quad \overline{a + b} = \bar{a} \oplus \bar{b}$$

$$\sigma_1 \times \sigma_2 \stackrel{\text{def}}{\iff} \bar{\sigma}_1 \dashv_S \sigma_2$$

● $a \times a \oplus b$ ● $a \times a + b$ ● $\mathbf{0} \times \sigma$

● \times is **not transitive**:

$$a \oplus b.\bar{c} \times a \quad \text{and} \quad a \times a + b.\bar{d}$$

but

$$a \oplus b.\bar{c} \not\times a + b.\bar{d}$$

The reasons of failure

$$a \oplus b.\bar{c} \not\sim a + b.\bar{d}$$

Imagine a client that is compliant with the service on the left...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a \oplus b.\bar{c}$$

... and silently replace the service on the left with the one on the right...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a + b.\bar{d}$$

If $\sigma \preceq \sigma'$ and C is compliant with σ , then C ...

- is
- can be made

... compliant with σ' too

The reasons of failure

$$a \oplus b.\bar{c} \not\sim a + b.\bar{d}$$

Imagine a client that is compliant with the service on the left...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a \oplus b.\bar{c}$$

... and silently replace the service on the left with the one on the right...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a + b.\bar{d}$$

If $\sigma \preceq \sigma'$ and C is compliant with σ , then C ...

- is
- can be made

... compliant with σ' too

The reasons of failure

$$a \oplus b.\bar{c} \not\approx a + b.\bar{d}$$

Imagine a client that is compliant with the service on the left...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a \oplus b.\bar{c}$$

... and silently replace the service on the left with the one on the right...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a + b.\bar{d}$$

If $\sigma \preceq \sigma'$ and C is compliant with σ , then C ...

- is
- can be made

... compliant with σ' too

The reasons of failure

$$a \oplus b.\bar{c} \not\leq a + b.\bar{d}$$

Imagine a client that is compliant with the service on the left...

$$\bar{a} + \bar{b}.c \quad || \quad a \oplus b.\bar{c}$$

... and silently replace the service on the left with the one on the right...

$$\bar{a} + \bar{b}.c \quad || \quad a + b.\bar{d}$$

If $\sigma \preceq \sigma'$ and C is compliant with σ , then C ...

- is
- can be made

... compliant with σ' too

The reasons of failure

$$a \oplus b.\bar{c} \not\leq a + b.\bar{d}$$

Imagine a client that is compliant with the service on the left...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a \oplus b.\bar{c}$$

... and silently replace the service on the left with the one on the right...

$$\bar{a} + \bar{b}.c \quad \parallel \quad a + b.\bar{d}$$

If $\sigma \preceq \sigma'$ and C is compliant with σ , then C ...

- is
- can be made

... compliant with σ' too

Subcontract relation IV

$\sigma_1 \preceq \sigma_2$ iff σ_2 can look like σ_1 by hiding some actions

- we can only act upon external choices (“system” non-determinism)
- we cannot affect internal choices (by definition)

$\sigma_1 \preceq \sigma_2$ implies that

- 1 for all R such that $\sigma_2 \Downarrow R$ there exists $S_R \subseteq R$ such that $\sigma_1 \Downarrow S_R$ and
- 2 for all $\alpha \in S_R$ we have $\sigma_1(\alpha) \preceq \sigma_2(\alpha)$.

We hide the actions in $R \setminus S_R$

$$a \oplus b.\bar{c} \preceq a \preceq a + b.\bar{d}$$

Proposition

\preceq is the transitive closure of \bowtie

Subcontract relation IV

$\sigma_1 \preceq \sigma_2$ iff σ_2 can look like σ_1 by hiding some actions

- we can only act upon external choices (“system” non-determinism)
- we cannot affect internal choices (by definition)

$\sigma_1 \preceq \sigma_2$ implies that

- 1 for all R such that $\sigma_2 \Downarrow R$ there exists $S_R \subseteq R$ such that $\sigma_1 \Downarrow S_R$ and
- 2 for all $\alpha \in S_R$ we have $\sigma_1(\alpha) \preceq \sigma_2(\alpha)$.

We hide the actions in $R \setminus S_R$

$$a \oplus b.\bar{c} \preceq a \preceq a + b.\bar{d}$$

Proposition

\preceq is the transitive closure of \bowtie

Subcontract relation IV

$\sigma_1 \preceq \sigma_2$ iff σ_2 can look like σ_1 by hiding some actions

- we can only act upon external choices (“system” non-determinism)
- we cannot affect internal choices (by definition)

$\sigma_1 \preceq \sigma_2$ implies that

- 1 for all R such that $\sigma_2 \Downarrow R$ there exists $S_R \subseteq R$ such that $\sigma_1 \Downarrow S_R$ and
- 2 for all $\alpha \in S_R$ we have $\sigma_1(\alpha) \preceq \sigma_2(\alpha)$.

We hide the actions in $R \setminus S_R$

$$a \oplus b.\bar{c} \preceq a \preceq a + b.\bar{d}$$

Proposition

\preceq is the transitive closure of \bowtie

Subcontract relation IV

$\sigma_1 \preceq \sigma_2$ iff σ_2 can look like σ_1 by hiding some actions

- we can only act upon external choices (“system” non-determinism)
- we cannot affect internal choices (by definition)

$\sigma_1 \preceq \sigma_2$ implies that

- 1 for all R such that $\sigma_2 \Downarrow R$ there exists $S_R \subseteq R$ such that $\sigma_1 \Downarrow S_R$ and
- 2 for all $\alpha \in S_R$ we have $\sigma_1(\alpha) \preceq \sigma_2(\alpha)$.

We hide the actions in $R \setminus S_R$

$$a \oplus b.\bar{c} \preceq a \preceq a + b.\bar{d}$$

Proposition

\preceq is the transitive closure of \bowtie

Filters as explicit coercions

$$\mathbf{filters} \quad f ::= \prod_{\alpha \in A} \alpha.f_{\alpha}$$

Transition relation of filters

$$\prod_{\alpha \in A} \alpha.f_{\alpha} \xrightarrow{\beta} f_{\beta} \quad \text{if } \beta \in A$$

Contract coercion through a filter

$$\begin{aligned} f(\mathbf{0}) &= \mathbf{0} \\ f(\alpha.\sigma) &= \mathbf{0} && \text{if } f \xrightarrow{\alpha} \\ f(\alpha.\sigma) &= \alpha.f'(\sigma) && \text{if } f \xrightarrow{\alpha} f' \\ f(\sigma_1 + \sigma_2) &= f(\sigma_1) + f(\sigma_2) \\ f(\sigma_1 \oplus \sigma_2) &= f(\sigma_1) \oplus f(\sigma_2) \end{aligned}$$

Proposition

$$\sigma_1 \preceq \sigma_2 \iff \sigma_1 \sqsubseteq f(\sigma_2) \text{ for some filter } f$$

Filters as explicit coercions

$$\mathbf{filters} \quad f ::= \prod_{\alpha \in A} \alpha.f_{\alpha}$$

Transition relation of filters

$$\prod_{\alpha \in A} \alpha.f_{\alpha} \xrightarrow{\beta} f_{\beta} \quad \text{if } \beta \in A$$

Contract coercion through a filter

$$\begin{aligned} f(\mathbf{0}) &= \mathbf{0} \\ f(\alpha.\sigma) &= \mathbf{0} && \text{if } f \xrightarrow{\alpha} \\ f(\alpha.\sigma) &= \alpha.f'(\sigma) && \text{if } f \xrightarrow{\alpha} f' \\ f(\sigma_1 + \sigma_2) &= f(\sigma_1) + f(\sigma_2) \\ f(\sigma_1 \oplus \sigma_2) &= f(\sigma_1) \oplus f(\sigma_2) \end{aligned}$$

Proposition

$$\sigma_1 \preceq \sigma_2 \iff \sigma_1 \sqsubseteq f(\sigma_2) \text{ for some filter } f$$

Filters as explicit coercions

$$\mathbf{filters} \quad f ::= \prod_{\alpha \in A} \alpha.f_{\alpha}$$

Transition relation of filters

$$\prod_{\alpha \in A} \alpha.f_{\alpha} \xrightarrow{\beta} f_{\beta} \quad \text{if } \beta \in A$$

Contract coercion through a filter

$$\begin{aligned} f(\mathbf{0}) &= \mathbf{0} \\ f(\alpha.\sigma) &= \mathbf{0} && \text{if } f \not\xrightarrow{\alpha} \\ f(\alpha.\sigma) &= \alpha.f'(\sigma) && \text{if } f \xrightarrow{\alpha} f' \\ f(\sigma_1 + \sigma_2) &= f(\sigma_1) + f(\sigma_2) \\ f(\sigma_1 \oplus \sigma_2) &= f(\sigma_1) \oplus f(\sigma_2) \end{aligned}$$

Proposition

$$\sigma_1 \preceq \sigma_2 \iff \sigma_1 \sqsubseteq f(\sigma_2) \text{ for some filter } f$$

Filters as explicit coercions

$$\mathbf{filters} \quad f ::= \prod_{\alpha \in A} \alpha.f_{\alpha}$$

Transition relation of filters

$$\prod_{\alpha \in A} \alpha.f_{\alpha} \xrightarrow{\beta} f_{\beta} \quad \text{if } \beta \in A$$

Contract coercion through a filter

$$\begin{aligned} f(\mathbf{0}) &= \mathbf{0} \\ f(\alpha.\sigma) &= \mathbf{0} && \text{if } f \xrightarrow{\alpha} \\ f(\alpha.\sigma) &= \alpha.f'(\sigma) && \text{if } f \xrightarrow{\alpha} f' \\ f(\sigma_1 + \sigma_2) &= f(\sigma_1) + f(\sigma_2) \\ f(\sigma_1 \oplus \sigma_2) &= f(\sigma_1) \oplus f(\sigma_2) \end{aligned}$$

Proposition

$$\sigma_1 \preceq \sigma_2 \iff \sigma_1 \sqsubseteq f(\sigma_2) \text{ for some filter } f$$

Contracts and processes

Language of processes

$$P \xrightarrow{\mu} P'$$

μ is either a visible action or τ

Type system

$$\vdash P : \sigma$$

“Subject reduction”

If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and

- 1 if $\sigma = \mathbf{0}$, then $\mu = \tau$ and $\sigma' = \mathbf{0}$
- 2 if $\mu = \tau$, then $\sigma \Longrightarrow \sigma'$ (e.g. $a \oplus b \Longrightarrow a$)
- 3 if $\mu = \alpha$, then $\sigma \xrightarrow{\alpha} \Longrightarrow \sigma'$

Contracts and processes

Language of processes

$$P \xrightarrow{\mu} P'$$

μ is either a visible action or τ

Type system

$$\vdash P : \sigma$$

“Subject reduction”

If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and

- 1 if $\sigma = \mathbf{0}$, then $\mu = \tau$ and $\sigma' = \mathbf{0}$
- 2 if $\mu = \tau$, then $\sigma \Longrightarrow \sigma'$ (e.g. $a \oplus b \Longrightarrow a$)
- 3 if $\mu = \alpha$, then $\sigma \xrightarrow{\alpha} \Longrightarrow \sigma'$

Contracts and processes

Language of processes

$$P \xrightarrow{\mu} P'$$

μ is either a visible action or τ

Type system

$$\vdash P : \sigma$$

“Subject reduction”

If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and

- 1 if $\sigma = \mathbf{0}$, then $\mu = \tau$ and $\sigma' = \mathbf{0}$
- 2 if $\mu = \tau$, then $\sigma \Longrightarrow \sigma'$ (e.g. $a \oplus b \Longrightarrow a$)
- 3 if $\mu = \alpha$, then $\sigma \xrightarrow{\alpha} \Longrightarrow \sigma'$

Strong compliance for processes

System = client || service

Interaction

- 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 $P \parallel Q \longrightarrow P' \parallel Q'$

Strong compliance for processes

$P \dashv_S Q$ if

- 1 $P \not\xrightarrow{\mu}$ for every μ , or
- 2 $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \dashv_S Q'$

Theorem

If $\vdash P : \sigma$ and $\vdash Q : \sigma'$ and $\sigma \dashv_S \sigma'$ then $P \dashv_S Q$

Strong compliance for processes

System = client || service

Interaction

- 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 $P \parallel Q \longrightarrow P' \parallel Q'$

Strong compliance for processes

$P \dashv_S Q$ if

- 1 $P \not\xrightarrow{\mu}$ for every μ , or
- 2 $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \dashv_S Q'$

Theorem

If $\vdash P : \sigma$ and $\vdash Q : \sigma'$ and $\sigma \dashv_S \sigma'$ then $P \dashv_S Q$

Strong compliance for processes

System = client \parallel service

Interaction

- 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 $P \parallel Q \longrightarrow P' \parallel Q'$

Strong compliance for processes

$P \dashv_S Q$ if

- 1 $P \xrightarrow{\mu} \text{for every } \mu$, or
- 2 $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \dashv_S Q'$

Theorem

If $\vdash P : \sigma$ and $\vdash Q : \sigma'$ and $\sigma \dashv_S \sigma'$ then $P \dashv_S Q$

Strong compliance for processes

System = client || service

Interaction

- 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 $P \parallel Q \longrightarrow P' \parallel Q'$

Strong compliance for processes

$P \dashv_S Q$ if

- 1 $P \xrightarrow{\mu} _$ for every μ , or
- 2 $P \parallel Q \longrightarrow P' \parallel Q'$ and $P' \dashv_S Q'$

Theorem

If $\vdash P : \sigma$ and $\vdash Q : \sigma'$ and $\sigma \dashv_S \sigma'$ then $P \dashv_S Q$

Filtering processes

We add filters to the process language: $f[P]$

Transition rules for filters

$$\frac{P \xrightarrow{\alpha} P' \quad f \vdash \alpha \rightarrow f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

Proposition

“Subject reduction” still holds

Corollary

If $\vdash P : \sigma_c$, $\vdash Q : \sigma_s$, and $\sigma_c \dashv_S f(\sigma_s)$, then $P \dashv_S f[Q]$

Filtering processes

We add filters to the process language: $f[P]$

Transition rules for filters

$$\frac{P \xrightarrow{\alpha} P' \quad f \vdash^{\alpha} f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

Proposition

“Subject reduction” still holds

Corollary

If $\vdash P : \sigma_c$, $\vdash Q : \sigma_s$, and $\sigma_c \dashv_S f(\sigma_s)$, then $P \dashv_S f[Q]$

Filtering processes

We add filters to the process language: $f[P]$

Transition rules for filters

$$\frac{P \xrightarrow{\alpha} P' \quad f \vdash^{\alpha} f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

Proposition

“Subject reduction” still holds

Corollary

If $\vdash P : \sigma_c$, $\vdash Q : \sigma_s$, and $\sigma_c \dashv_S f(\sigma_s)$, then $P \dashv_S f[Q]$

Filtering processes

We add filters to the process language: $f[P]$

Transition rules for filters

$$\frac{P \xrightarrow{\alpha} P' \quad f \vdash^{\alpha} f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

Proposition

“Subject reduction” still holds

Corollary

If $\vdash P : \sigma_c$, $\vdash Q : \sigma_s$, and $\sigma_c \dashv_S f(\sigma_s)$, then $P \dashv_S f[Q]$

Concluding remarks

What we have done

A theory of contracts (reasoning on the behavior of processes)

- contracts = (simple) behavioral types
- notions of compliance for contracts/processes
- filters = behavioral coercions that enlarge the space of compliant contracts/processes
- language-independent results (up to “subject reduction”)

What has been done

Contract inference (Carpinetti, Castagna, Laneve, and Padovani 2006)

Session types (Gay and Hole 2005):

- language restrictions (+ matched by \oplus and viceversa) make \preceq almost trivial
- no explicit coercions

Concluding remarks

What we have done

A theory of contracts (reasoning on the behavior of processes)

- contracts = (simple) behavioral types
- notions of compliance for contracts/processes
- filters = behavioral coercions that enlarge the space of compliant contracts/processes
- language-independent results (up to “subject reduction”)

What has been done

Contract inference (Carpineti, Castagna, Laneve, and Padovani 2006)

Session types (Gay and Hole 2005):

- language restrictions (+ matched by \oplus and viceversa) make \preceq almost trivial
- no explicit coercions

What remains to do

- *axiomatization*: define a sound/complete proof system so that

$$\vdash f : \sigma_1 \leq \sigma_2 \iff \sigma_1 \preceq \sigma_2$$

and f is a *canonical filter*

- exploring *expressivity of filters* (filters can replace restriction $P \setminus a$)
- generalize filters to *(iso)morphisms*

$$\vdash f : a.b \leq b.a$$

- *name passing*

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

- recursive contracts, asynchronous communication, ...