A theory of contracts for Web services

Giuseppe Castagna, Nils Gesbert, Luca Padovani

Université Paris 7, University of Glasgow, Università di Urbino

POPL 2008, San Francisco

Presentation outline

Web service discovery

- Registries (UDDI...)
- Contain descriptions (WSDL...)
- Several possible search keys:
  - Name
  - Keywords
  - ...
  - Behaviour (available interaction patterns)

Web service discovery

- Registries (UDDI...)
- Contain descriptions (WSDL...)
- Several possible search keys:
  - Name
  - Keywords
  - ...
  - Behaviour (available interaction patterns)

A formal contract language

actions/coactions $\alpha ::= a \mid \overline{a}$

contrats $\sigma ::=$

0	*(termination)*
$\alpha.\sigma$	*(actionprefix)*
$\sigma + \sigma$	*(external choice)*
$\sigma \oplus \sigma$	*(internal choice)*

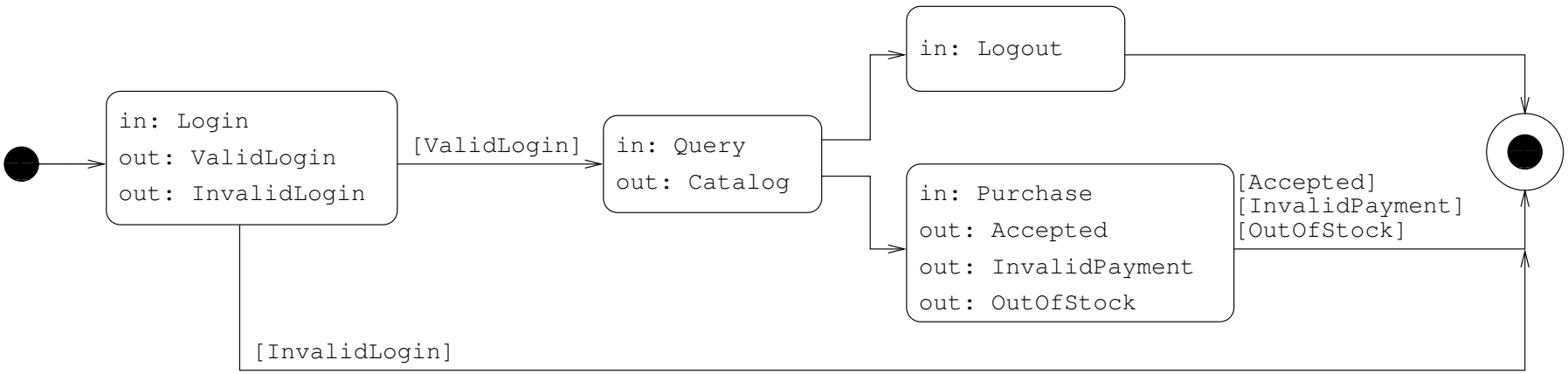
A formal contract language

actions/coactions $\alpha ::= a \mid \overline{a}$

contrats $\sigma ::=$

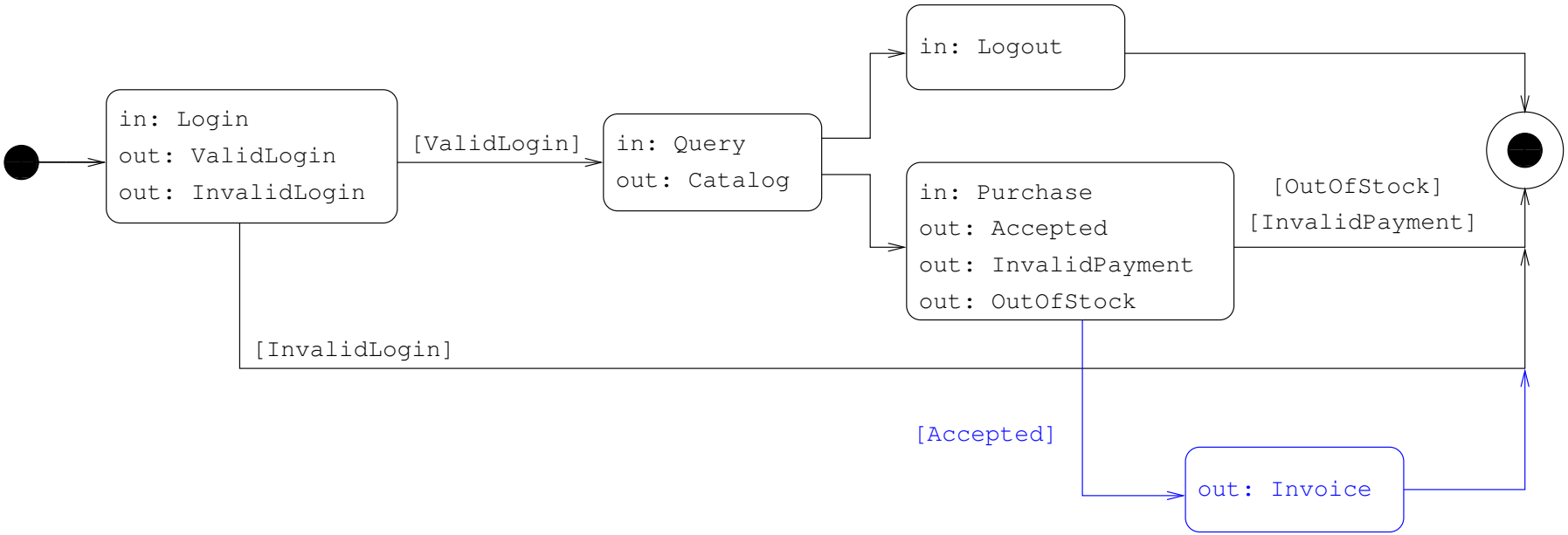
$\mathbf{0}$	(*termination*)
$\alpha.\sigma$	(*actionprefix*)
$\sigma + \sigma$	(*external choice*)
$\sigma \oplus \sigma$	(*internal choice*)

Example



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

Example



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

Find a service suitable for a given client

Two possible ways:

- Directly, by defining a contract for the client
- Starting from a known suitable service

Find a service suitable for a given client

Two possible ways:

- Directly, by defining a contract for the client

  $\rightarrow$compliance relation

- Starting from a known suitable service

Find a service suitable for a given client

Two possible ways:

- Directly, by defining a contract for the client

→compliance relation

- Starting from a known suitable service

→subcontract relation

Relating client and service: **compliance**

- Clients too are described by contracts
- A special action ω represents client success

Definition of compliance

ω must be available whenever no more interaction is possible

Relating client and service: **compliance**

- Clients too are described by contracts
- A special action ω represents client success

Definition of compliance

ω must be available whenever no more interaction is possible

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. be more deterministic:

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. offer more choices:

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout+Purchase} \preceq \texttt{Logout+Purchase+SaveForLater}$$

3. offer longer interaction patterns:

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. **be more deterministic:**

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. offer more choices:

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout+Purchase} \preceq \texttt{Logout+Purchase+SaveForLater}$$

3. offer longer interaction patterns:

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. **be more deterministic:**

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. offer more choices:

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout}+\texttt{Purchase} \preceq \texttt{Logout}+\texttt{Purchase}+\texttt{SaveForLater}$$

3. offer longer interaction patterns:

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. **be more deterministic:**

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. **offer more choices:**

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout}+\texttt{Purchase} \preceq \texttt{Logout}+\texttt{Purchase}+\texttt{SaveForLater}$$

3. **offer longer interaction patterns:**

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. **be more deterministic:**

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. **offer more choices:**

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout}+\texttt{Purchase} \preceq \texttt{Logout}+\texttt{Purchase}+\texttt{SaveForLater}$$

3. offer longer interaction patterns:

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. **be more deterministic:**

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. **offer more choices:**

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout}+\texttt{Purchase} \preceq \texttt{Logout}+\texttt{Purchase}+\texttt{SaveForLater}$$

3. **offer longer interaction patterns:**

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Relating two services: **subcontract**

Informal definition

$\sigma \preceq \tau$: "a client designed for a σ service will work with a τ one."

τ may:

1. **be more deterministic:**

$$\overline{a} \oplus \overline{b}.c \preceq \overline{a}$$

$$\overline{\texttt{InvalidLogin}} \oplus \overline{\texttt{ValidLogin}} \preceq \overline{\texttt{ValidLogin}}$$

2. **offer more choices:**

$$\overline{a} \preceq \overline{a} + \overline{b}.d$$

$$\texttt{Logout}+\texttt{Purchase} \preceq \texttt{Logout}+\texttt{Purchase}+\texttt{SaveForLater}$$

3. **offer longer interaction patterns:**

$$\overline{a} \preceq \overline{a}.\overline{b}.d$$

$$\texttt{Purchase}.\overline{\texttt{Accepted}} \preceq \texttt{Purchase}.\overline{\texttt{Accepted}}.\overline{\texttt{Invoice}}$$

Transitivity problem!

- So we want:

$$\overline{a} \oplus \overline{b}.c \overset{1}{\preceq} \overline{a} \overset{2}{\preceq} \overline{a} + \overline{b}.d$$

- but with the client contract $a.\omega + b.\overline{c}.\omega$ we have:

$$a.\omega + b.\overline{c}.\omega \dashv \overline{a} \oplus \overline{b}.c \qquad a.\omega + b.\overline{c}.\omega \not\dashv \overline{a} + \overline{b}.d$$

- Replace a service of contract $\overline{a} \oplus \overline{b}.c$ with a service of contract $\overline{a} + \overline{b}.d$?

- possible using explicit coercions (filtering out foreign actions)

Transitivity problem!

- So we want:

$$\overline{a} \oplus \overline{b}.c \quad \underset{1}{\preceq} \quad \overline{a} \quad \underset{2}{\preceq} \quad \overline{a} + \overline{b}.d$$

- but with the client contract $a.\omega + b.\overline{c}.\omega$ we have:

$$a.\omega + b.\overline{c}.\omega \dashv \overline{a} \oplus \overline{b}.c \qquad a.\omega + b.\overline{c}.\omega \not\dashv \overline{a} + \overline{b}.d$$

- Replace a service of contract $\overline{a} \oplus \overline{b}.c$ with a service of contract $\overline{a} + \overline{b}.d$?

- possible using explicit coercions (filtering out foreign actions)

Transitivity problem!

- So we want:

$$\overline{a} \oplus \overline{b}.c \overset{1}{\preceq} \overline{a} \overset{2}{\preceq} \overline{a} + \overline{b}.d$$

- but with the client contract $a.\omega + b.\overline{c}.\omega$ we have:

$$a.\omega + b.\overline{c}.\omega \dashv \overline{a} \oplus \overline{b}.c \qquad a.\omega + b.\overline{c}.\omega \not\dashv \overline{a} + \overline{b}.d$$

- Replace a service of contract $\overline{a} \oplus \overline{b}.c$ with a service of contract $\overline{a} + \overline{b}.d$?

- possible using explicit coercions (filtering out foreign actions)

Transitivity problem!

- So we want:

$$\overline{a} \oplus \overline{b}.c \underset{\sim}{\overset{1}{\preceq}} \overline{a} \underset{\sim}{\overset{2}{\preceq}} \overline{a} + \overline{b}.d$$

- but with the client contract $a.\omega + b.\overline{c}.\omega$ we have:

$$a.\omega + b.\overline{c}.\omega \dashv \overline{a} \oplus \overline{b}.c \qquad a.\omega + b.\overline{c}.\omega \not\dashv \overline{a} + \overline{b}.d$$

- Replace a service of contract $\overline{a} \oplus \overline{b}.c$ with a service of contract $\overline{a} + \overline{b}.d$?

- possible using explicit coercions (filtering out foreign actions)

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter
  (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter
  (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Gluing compliance and subcontracting: **Filters**

Filters are operators that, at each step, let only pass a subset of the available actions.

Property

if $\sigma \preceq \tau$ then for some filter f: $\quad \rho \dashv \sigma \iff \rho \dashv f(\tau)$

Filters are "proofs" of subcontracting

$$f : \sigma \preceq \tau$$

$$f : \sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \iff \quad \rho \dashv f(\tau)$$

- deduction system for subcontracting: $f : \sigma \preceq \tau$,
- algebraic theory for filters
- existence and effectiveness of a most general filter
  (via cut-elimination, yields subcontracting coherence)
- subcontracting decidability

Application to languages

Our contracts work for any language as long as they come equipped with:

1. A labelled transition system

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

μ is either a visible action or an invisible τ action

2. A type system

$$\vdash P : \sigma$$

σ is a contract

3. The latter abstracts the former.

Application to languages

Our contracts work for any language as long as they come equipped with:

1. **A labelled transition system**

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

μ is either a visible action or an invisible τ action

2. A type system

$$\vdash P : \sigma$$

σ is a contract

3. The latter abstracts the former.

Application to languages

Our contracts work for any language as long as they come equipped with:

1. **A labelled transition system**

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

μ is either a visible action or an invisible τ action

2. **A type system**

$$\vdash P : \sigma$$

σ is a contract

3. The latter abstracts the former.

Application to languages

Our contracts work for any language as long as they come equipped with:

1. **A labelled transition system**

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

μ is either a visible action or an invisible τ action

2. **A type system**

$$\vdash P : \sigma$$

σ is a contract

3. **The latter abstracts the former.**

Process compliance and filtering

Compliance *for processes*

- Client P is compliant with server Q if: $P \parallel Q \longrightarrow\!\!\!\!\!\!\not\quad$ implies $P \xrightarrow{\omega}$

Add filters to the language: $f[P]$

- Transition relation
- Typing rules

Theorem (Process filtering)

If $\vdash P : \rho$ *and* $\vdash Q : \sigma$ *and* $f : \rho \dashv f(\sigma)$, *then* $P \dashv f[Q]$

Process compliance and filtering

Compliance *for processes*

- Client P is compliant with server Q if: $P \parallel Q \longrightarrow\!\!\!\!\!\!\not\quad$ implies $P \xrightarrow{\omega}$

Add filters to the language: $f[P]$

- Transition relation
- Typing rules

Theorem (Process filtering)

If $\vdash P : \rho$ *and* $\vdash Q : \sigma$ *and* $f : \rho \dashv f(\sigma)$, *then* $P \dashv f[Q]$

Process compliance and filtering

Compliance *for processes*

- Client P is compliant with server Q if: $P \parallel Q \longrightarrow\!\!\!\!\!\!\not\quad$ implies $P \xrightarrow{\omega}$

Add filters to the language: $f[P]$

- Transition relation
- Typing rules

Theorem (Process filtering)

If $\vdash P : \rho$ *and* $\vdash Q : \sigma$ *and* $f : \rho \dashv f(\sigma)$, *then* $P \dashv f[Q]$

Process compliance and filtering

Compliance *for processes*

- Client P is compliant with server Q if: $P \parallel Q \longrightarrow\!\!\!\!\!\not\;\;$ implies $P \xrightarrow{\omega}$

Add filters to the language: $f[P]$

- Transition relation
- Typing rules

Theorem (Process filtering)

If $\vdash P : \rho$ *and* $\vdash Q : \sigma$ *and* $f : \rho \dashv f(\sigma)$, *then* $P \dashv f[Q]$

Future work

- Recursion (done!)
- Messages with content, higher order
- Asymmetric choices
- Relations with other formalisms

Future work

- Recursion (done!)
- Messages with content, higher order
- Asymmetric choices
- Relations with other formalisms

Future work

- Recursion (done!)
- Messages with content, higher order
- Asymmetric choices
- Relations with other formalisms

Future work

- Recursion (done!)
- Messages with content, higher order
- Asymmetric choices
- Relations with other formalisms

Future work

- Recursion (done!)
- Messages with content, higher order
- Asymmetric choices
- Relations with other formalisms

Axiomatisation of the weak subcontract relation

(Must)
$$l_\sigma \vee l_\tau : \sigma \oplus \tau \preceq \sigma$$

(DepthExt)
$$\mathbf{0} : \mathbf{0} \preceq \sigma$$

(Prefix)
$$\frac{f : \sigma \preceq \tau}{\alpha.f : \alpha.\sigma \preceq \alpha.\tau}$$

(Weakening)
$$\frac{f : \sigma \preceq \tau \quad g \wedge l_\tau \leqslant f}{f \vee g : \sigma \preceq \tau}$$

(Transitivity)
$$\frac{f : \sigma \preceq \sigma' \quad g : \sigma' \preceq \sigma''}{f \wedge g : \sigma \preceq \sigma''}$$

(IntChoice)
$$\frac{f : \sigma \preceq \sigma' \quad f : \tau \preceq \tau'}{f : \sigma \oplus \tau \preceq \sigma' \oplus \tau'}$$

(ExtChoice)
$$\frac{f : \sigma \preceq \sigma' \quad f : \tau \preceq \tau'}{f : \sigma + \tau \preceq \sigma' + \tau'}$$