Contrats pour les services web

Giuseppe Castagna, Nils Gesbert, Luca Padovani

Université Paris 7, Université Paris Sud, Università di Urbino

ACI MD/Tralala (XML TRAnsformation LAnguages : Logic and Applications)

Grand colloque STIC 2007

Plan de la présentation

Recherche de services web

- Annuaires (UDDI...)
- Contiennent des descriptions (WSDL...)
- Recherche selon différents critères :
  - Nom
  - Mots-clefs
  - ...
  - Comportement (séquences d'interactions avec le client)

Recherche de services web

- Annuaires (UDDI...)
- Contiennent des descriptions (WSDL...)
- Recherche selon différents critères :
  - Nom
  - Mots-clefs
  - ...
  - Comportement (séquences d'interactions avec le client)

Un langage formel de contrats

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

contrats $\sigma ::=$

0	*(terminaison)*
$\alpha.\sigma$	*(action)*
$\sigma + \sigma$	*(choix externe)*
$\sigma \oplus \sigma$	*(choix interne)*

Un langage formel de contrats

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

contrats $\sigma ::=$

$\mathbf{0}$	(*terminaison*)
$\alpha.\sigma$	(*action*)
$\sigma + \sigma$	(*choix externe*)
$\sigma \oplus \sigma$	(*choix interne*)

Sémantique des contrats

Les contrats sont caractérisés par :

1. Une relation de transition $\longrightarrow$ actions que le service peut faire

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

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

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

2. Des “ready sets” $\longrightarrow$ états dans lesquels le service peut se trouver

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

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

$$(\sigma + \sigma') \Downarrow R \cup R' \qquad \text{si } \sigma \Downarrow R \text{ et } \sigma' \Downarrow R'$$

$$(\sigma \oplus \sigma') \Downarrow R \qquad \text{si } \sigma \Downarrow R \text{ ou } \sigma' \Downarrow R$$

Sémantique des contrats

Les contrats sont caractérisés par :

1. Une relation de transition → actions que le service peut faire

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

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

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

2. Des "ready sets" → états dans lesquels le service peut se trouver

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

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

$$(\sigma + \sigma') \Downarrow R \cup R' \qquad \text{si } \sigma \Downarrow R \text{ et } \sigma' \Downarrow R'$$

$$(\sigma \oplus \sigma') \Downarrow R \qquad \text{si } \sigma \Downarrow R \text{ ou } \sigma' \Downarrow R$$

Sémantique des contrats

Les contrats sont caractérisés par :

1. Une relation de transition → actions que le service peut faire

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

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

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

2. Des "ready sets" → états dans lesquels le service peut se trouver

$$\begin{aligned}
&\mathbf{0} \Downarrow \emptyset \\
&\alpha.\sigma \Downarrow \{\alpha\} \\
&(\sigma + \sigma') \Downarrow R \cup R' && \text{si } \sigma \Downarrow R \text{ et } \sigma' \Downarrow R' \\
&(\sigma \oplus \sigma') \Downarrow R && \text{si } \sigma \Downarrow R \text{ ou } \sigma' \Downarrow R
\end{aligned}$$

Trouver un service convenant à un client

Deux approches possibles :

- Directement, en utilisant un contrat client
- À partir d'un service connu comme convenant au client

Trouver un service convenant à un client

Deux approches possibles :

- Directement, en utilisant un contrat client

  $\rightarrow$relation de concordance

- À partir d'un service connu comme convenant au client

Trouver un service convenant à un client

Deux approches possibles :

- Directement, en utilisant un contrat client

  $\rightarrow$relation de concordance

- À partir d'un service connu comme convenant au client

  $\rightarrow$relation de sous-contrat

Concordance entre client et serveur

- Les clients sont eux aussi décrits par des contrats
- Une action spéciale ω représente le succès du client
- ω doit être disponible lorsque l'interaction est terminée

Définition formelle : $\rho \dashv \sigma$ si :

1. $\rho \Downarrow R$, $\sigma \Downarrow S$ et $\overline{R} \cap S = \varnothing$ impliquent $\omega \in R$
2. $\rho \stackrel{\overline{\alpha}}{\longmapsto} \rho'$ et $\sigma \stackrel{\alpha}{\longmapsto} \sigma'$ impliquent $\rho' \dashv \sigma'$

Concordance entre client et serveur

- Les clients sont eux aussi décrits par des contrats
- Une action spéciale ω représente le succès du client
- ω doit être disponible lorsque l'interaction est terminée

Définition formelle : $\rho \dashv \sigma$ si :

1. $\rho \Downarrow R$, $\sigma \Downarrow S$ et $\overline{R} \cap S = \varnothing$ impliquent $\omega \in R$
2. $\rho \stackrel{\overline{\alpha}}{\longmapsto} \rho'$ et $\sigma \stackrel{\alpha}{\longmapsto} \sigma'$ impliquent $\rho' \dashv \sigma'$

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: “un contrat conçu pour σ fonctionnera pour τ.”
τ peut :

1. être plus déterministe :

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

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

2. proposer des choix supplémentaires :

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

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

3. proposer des séquences d’interaction plus longues :

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

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

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: "un contrat conçu pour σ fonctionnera pour τ."
τ peut :

1. **être plus déterministe :**

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

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

2. **proposer des choix supplémentaires :**

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

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

3. **proposer des séquences d'interaction plus longues :**

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

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

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: "un contrat conçu pour σ fonctionnera pour τ."
τ peut :

1. **être plus déterministe :**

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

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

2. proposer des choix supplémentaires :

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

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

3. proposer des séquences d'interaction plus longues :

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

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

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: "un contrat conçu pour σ fonctionnera pour τ."
τ peut :

1. **être plus déterministe :**

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

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

2. **proposer des choix supplémentaires :**

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

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

3. **proposer des séquences d'interaction plus longues :**

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

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

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: "un contrat conçu pour σ fonctionnera pour τ."
τ peut :

1. **être plus déterministe :**

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

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

2. **proposer des choix supplémentaires :**

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

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

3. proposer des séquences d'interaction plus longues :

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

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

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: "un contrat conçu pour σ fonctionnera pour τ."
τ peut :

1. **être plus déterministe :**

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

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

2. **proposer des choix supplémentaires :**

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

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

3. **proposer des séquences d'interaction plus longues :**

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

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

Relation de sous-contrat entre deux services

On veut définir $\sigma \preceq \tau$: "un contrat conçu pour σ fonctionnera pour τ."
τ peut :

1. **être plus déterministe :**

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

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

2. **proposer des choix supplémentaires :**

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

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

3. **proposer des séquences d'interaction plus longues :**

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

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

Problème de transitivité !

- On veut donc :

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

- mais pour un client $a.\omega + b.\overline{c}.\omega$ on a :

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

- Remplacer un serveur de contrat $\overline{a} \oplus \overline{b}.c$ par un serveur de contrat $\overline{a} + \overline{b}.d$?

- possible par coercion explicite (filtrage des actions nuisibles)

Problème de transitivité !

- On veut donc :

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

- mais pour un client $a.w + b.\overline{c}.w$ on a :

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

- Remplacer un serveur de contrat $\overline{a} \oplus \overline{b}.c$ par un serveur de contrat $\overline{a} + \overline{b}.d$?

- possible par coercion explicite (filtrage des actions nuisibles)

Problème de transitivité !

- On veut donc :

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

- mais pour un client $a.w + b.\overline{c}.w$ on a :

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

- Remplacer un serveur de contrat $\overline{a} \oplus \overline{b}.c$ par un serveur de contrat $\overline{a} + \overline{b}.d$?

- possible par coercion explicite (filtrage des actions nuisibles)

Problème de transitivité !

- On veut donc :

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

- mais pour un client $a.w + b.\overline{c}.w$ on a :

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

- Remplacer un serveur de contrat $\overline{a} \oplus \overline{b}.c$ par un serveur de contrat $\overline{a} + \overline{b}.d$?

- possible par coercion explicite (filtrage des actions nuisibles)

Filtres

filtres $\qquad f ::= \coprod_{\alpha \in A} \alpha.f_\alpha$

Relation de transition sur les filtres

$$\coprod_{\alpha \in A} \alpha.f_\alpha \xrightarrow{\beta} f_\beta \qquad \text{if } \beta \in A$$

Coercion d'un contrat par un filtre

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

Propriété

$$\sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \Longleftrightarrow \quad \rho \dashv f(\tau) \quad \text{pour un certain filtre } f$$

Filtres

filtres $\qquad f ::= \coprod_{\alpha \in A} \alpha.f_\alpha$

Relation de transition sur les filtres

$$\coprod_{\alpha \in A} \alpha.f_\alpha \stackrel{\beta}{\longmapsto} f_\beta \qquad \text{if } \beta \in A$$

Coercion d'un contrat par un filtre

$$\begin{array}{rcll} f(\mathbf{0}) & = & \mathbf{0} & \\ f(\alpha.\sigma) & = & \mathbf{0} & \text{si } f \stackrel{\alpha}{\not\longmapsto} \\ f(\alpha.\sigma) & = & \alpha.f'(\sigma) & \text{si } f \stackrel{\alpha}{\longmapsto} f' \\ f(\sigma_1 + \sigma_2) & = & f(\sigma_1) + f(\sigma_2) & \\ f(\sigma_1 \oplus \sigma_2) & = & f(\sigma_1) \oplus f(\sigma_2) & \end{array}$$

Propriété

$$\sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \Longleftrightarrow \quad \rho \dashv f(\tau) \quad \text{pour un certain filtre } f$$

Filtres

filtres $\qquad f ::= \coprod_{\alpha \in A} \alpha.f_\alpha$

Relation de transition sur les filtres

$$\coprod_{\alpha \in A} \alpha.f_\alpha \stackrel{\beta}{\longmapsto} f_\beta \qquad \text{if } \beta \in A$$

Coercion d'un contrat par un filtre

$$\begin{array}{rcll} f(\mathbf{0}) & = & \mathbf{0} & \\ f(\alpha.\sigma) & = & \mathbf{0} & \text{si } f \stackrel{\alpha}{\not\longmapsto} \\ f(\alpha.\sigma) & = & \alpha.f'(\sigma) & \text{si } f \stackrel{\alpha}{\longmapsto} f' \\ f(\sigma_1 + \sigma_2) & = & f(\sigma_1) + f(\sigma_2) & \\ f(\sigma_1 \oplus \sigma_2) & = & f(\sigma_1) \oplus f(\sigma_2) & \end{array}$$

Propriété

$\sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \Longleftrightarrow \quad \rho \dashv f(\tau)$ pour un certain filtre f

Filtres

filtres $\qquad f ::= \coprod_{\alpha \in A} \alpha.f_\alpha$

Relation de transition sur les filtres

$$\coprod_{\alpha \in A} \alpha.f_\alpha \xrightarrow{\beta} f_\beta \qquad \text{if } \beta \in A$$

Coercion d'un contrat par un filtre

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

Propriété

$$\sigma \preceq \tau \quad \wedge \quad \rho \dashv \sigma \quad \Longleftrightarrow \quad \rho \dashv f(\tau) \quad \text{pour un certain filtre } f$$

Résultats

Filtres comme “preuves” de la relation de sous-contrat

$$f : \sigma \preceq \tau \text{ si } \rho \dashv \sigma \Rightarrow \rho \dashv \tau$$

- axiomatisation de la relation : $f : \sigma \preceq \tau$
- existence et calculabilité d'un “meilleur” filtre
- décidabilité de la relation

Résultats

Filtres comme “preuves” de la relation de sous-contrat

$$f : \sigma \preceq \tau \text{ si } \rho \dashv \sigma \Rightarrow \rho \dashv \tau$$

- axiomatisation de la relation : $f : \sigma \preceq \tau$
- existence et calculabilité d'un “meilleur” filtre
- décidabilité de la relation

Résultats

Filtres comme “preuves” de la relation de sous-contrat

$$f : \sigma \preceq \tau \text{ si } \rho \dashv \sigma \Rightarrow \rho \dashv \tau$$

- axiomatisation de la relation : $f : \sigma \preceq \tau$
- existence et calculabilité d'un “meilleur” filtre
- décidabilité de la relation

Résultats

Filtres comme “preuves” de la relation de sous-contrat

$$f : \sigma \preceq \tau \text{ si } \rho \dashv \sigma \Rightarrow \rho \dashv \tau$$

- axiomatisation de la relation : $f : \sigma \preceq \tau$
- existence et calculabilité d'un “meilleur” filtre
- décidabilité de la relation

Résultats

Filtres comme “preuves” de la relation de sous-contrat

$$f : \sigma \preceq \tau \text{ si } \rho \dashv \sigma \Rightarrow \rho \dashv \tau$$

- axiomatisation de la relation : $f : \sigma \preceq \tau$
- existence et calculabilité d'un “meilleur” filtre
- décidabilité de la relation

Perspectives

- Récursion
- Messages avec contenu, ordre supérieur
- Choix asymétriques
- Relations avec d'autres formalismes

Perspectives

- Récursion
- Messages avec contenu, ordre supérieur
- Choix asymétriques
- Relations avec d'autres formalismes

Perspectives

- Récursion
- Messages avec contenu, ordre supérieur
- Choix asymétriques
- Relations avec d'autres formalismes

Perspectives

- Récursion
- Messages avec contenu, ordre supérieur
- Choix asymétriques
- Relations avec d'autres formalismes

Perspectives

- Récursion
- Messages avec contenu, ordre supérieur
- Choix asymétriques
- Relations avec d'autres formalismes

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

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 :**
   1. If $\vdash P : \sigma$ and $\sigma \xrightarrow{\alpha}$, then $P \xrightarrow{\alpha}$
   2. If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and
      - if $\mu = \tau$, then $\sigma \sqsubseteq \sigma'$
      - if $\mu = \alpha$, then $\sigma \xrightarrow{\alpha} \sqsubseteq \sigma'$

$\sqsubseteq$ measures non-determinism

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 :
   1. If $\vdash P : \sigma$ and $\sigma \xrightarrow{\alpha}$, then $P \xrightarrow{\alpha}$
   2. If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and
      - if $\mu = \tau$, then $\sigma \sqsubseteq \sigma'$
      - if $\mu = \alpha$, then $\sigma \xrightarrow{\alpha} \sqsubseteq \sigma'$

$\sqsubseteq$ measures non-determinism

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 :
   1. If $\vdash P : \sigma$ and $\sigma \xrightarrow{\alpha}$, then $P \xrightarrow{\alpha}$
   2. If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and
      - if $\mu = \tau$, then $\sigma \sqsubseteq \sigma'$
      - if $\mu = \alpha$, then $\sigma \xrightarrow{\alpha} \sqsubseteq \sigma'$

$\sqsubseteq$ measures non-determinism

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 :**
   1. If $\vdash P : \sigma$ and $\sigma \overset{\alpha}{\longmapsto}$, then $P \xrightarrow{\alpha}$
   2. If $\vdash P : \sigma$ and $P \xrightarrow{\mu} P'$ then $\vdash P' : \sigma'$ and
      - if $\mu = \tau$, then $\sigma \sqsubseteq \sigma'$
      - if $\mu = \alpha$, then $\sigma \overset{\alpha}{\longmapsto} \sqsubseteq \sigma'$

$\sqsubseteq$ measures non-determinism

Process compliance

Compliance *for processes*

- If $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$ (plus τ-moves)
- Client P complies with server Q (noted $P \dashv Q$) if
  - $P \parallel Q \not\longrightarrow$ and $P \xrightarrow{\omega}$ or
  - $P \parallel Q \longrightarrow P' \parallel Q'$ implies $P' \dashv Q'$

Theorem (Process compliance)

If $\vdash P : p$ and $\vdash Q : \sigma$ and $p \dashv \sigma$, then $P \dashv Q$

Process compliance

Compliance *for processes*

- If $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$ (plus τ-moves)
- Client P complies with server Q (noted $P \dashv Q$) if
  - $P \parallel Q \not\longrightarrow$ and $P \xrightarrow{\omega}$ or
  - $P \parallel Q \longrightarrow P' \parallel Q'$ implies $P' \dashv Q'$

Theorem (Process compliance)

If $\vdash P : p$ and $\vdash Q : \sigma$ and $p \dashv \sigma$, then $P \dashv Q$

Process compliance

Compliance *for processes*

- If $P \xrightarrow{\alpha} P'$ and $Q \xrightarrow{\overline{\alpha}} Q'$ then $P \parallel Q \longrightarrow P' \parallel Q'$ (plus τ-moves)
- Client P complies with server Q (noted $P \dashv Q$) if
  - $P \parallel Q \not\longrightarrow$ and $P \xrightarrow{\omega}$ or
  - $P \parallel Q \longrightarrow P' \parallel Q'$ implies $P' \dashv Q'$

Theorem (Process compliance)

If $\vdash P : \rho$ *and* $\vdash Q : \sigma$ *and* $\rho \dashv \sigma$, *then* $P \dashv Q$

Process Filtering

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

Transition rules for filters

$$\text{(Filter1)}\quad \frac{P \xrightarrow{\alpha} P' \qquad f \overset{\alpha}{\longmapsto} f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \text{(Filter2)}\quad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\text{(T-Filter)}\quad \frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

"Subject reduction" still holds

Process Filtering

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

Transition rules for filters

(Filter1)
$$\frac{P \xrightarrow{\alpha} P' \qquad f \overset{\alpha}{\longmapsto} f'}{f[P] \xrightarrow{\alpha} f'[P']}$$

(Filter2)
$$\frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

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

"Subject reduction" still holds

Theorem (Process filtering)

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

Process Filtering

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

Transition rules for filters

$$\text{(Filter1)}\quad \frac{P \xrightarrow{\alpha} P' \qquad f \overset{\alpha}{\longmapsto} f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \text{(Filter2)}\quad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\text{(T-Filter)}\quad \frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

"Subject reduction" still holds

Theorem (Process filtering)

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

Process Filtering

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

Transition rules for filters

$$\text{(Filter1)}\quad \frac{P \xrightarrow{\alpha} P' \qquad f \overset{\alpha}{\longmapsto} f'}{f[P] \xrightarrow{\alpha} f'[P']} \qquad \text{(Filter2)}\quad \frac{P \xrightarrow{\tau} P'}{f[P] \xrightarrow{\tau} f[P']}$$

Typing rules for filters

$$\text{(T-Filter)}\quad \frac{\vdash P : \sigma}{\vdash f[P] : f(\sigma)}$$

"Subject reduction" still holds

Theorem (Process filtering)

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