JoCaml et le join-calcul

Samuel Mimram

samuel.mimram@ens-lyon.fr

ÉNS Lyon

Motivations

- le π-calcul semble être un bon outil : modélise la concurrence

Motivations

- le π-calcul semble être un bon outil : modélise la concurrence
- il est riche : λ-calcul, calculs orientés objets ou impératifs implémentables

Motivations

- le π-calcul semble être un bon outil : modélise la concurrence
- il est riche : λ-calcul, calculs orientés objets ou impératifs implémentables
- mais. . .

Motivations

- le π-calcul semble être un bon outil : modélise la concurrence
- il est riche : λ-calcul, calculs orientés objets ou impératifs implémentables
- mais. . . *discipline de programmation peu naturelle*
  - la séquence des calculs qu'un processus doit faire est prédéfinie, on préfèrerait : « si les résultats de tel et tel calcul sont disponibles alors je peux faire tel calcul »

$$u(x).\texttt{let}\ y = f(x)\ \texttt{in}\ v\langle y\rangle \,|w\langle 2 \times y\rangle$$

Motivations

- le π-calcul semble être un bon outil : modélise la concurrence
- il est riche : λ-calcul, calculs orientés objets ou impératifs implémentables
- mais. . . *difficile à implémenter / peu sécurisé*
  - la séquence des calculs qu'un processus doit faire est prédéfinie, on préfèrerait : « si les résultats de tel et tel calcul sont disponibles alors je peux faire tel calcul »
  - les canaux peuvent être globaux $\rightsquigarrow$ implémentation (broadcast) ? sécurité (process in the middle) ?

Motivations

- le π-calcul semble être un bon outil : modélise la concurrence
- il est riche : λ-calcul, calculs orientés objets ou impératifs implémentables
- mais. . .
  - la séquence des calculs qu'un processus doit faire est prédéfinie, on préfèrerait : « si les résultats de tel et tel calcul sont disponibles alors je peux faire tel calcul »
  - les canaux peuvent être globaux $\rightsquigarrow$ implémentation (broadcast) ? sécurité (process in the middle) ?
  - on aimerait des primitives agréables pour la migration de code

Le join-calcul

La syntaxe des processus est :

$$P \quad ::= \quad x\langle u\rangle \quad | \quad P_1|P_2 \quad | \quad \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$$

- $x\langle u\rangle$: émission

Le join-calcul

La syntaxe des processus est :

$$P \quad ::= \quad x\langle u\rangle \quad | \quad P_1|P_2 \quad | \quad \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$$

- $x\langle u\rangle$: émission
- $P_1|P_2$: activité parallèle

Le join-calcul

La syntaxe des processus est :

$$P \quad ::= \quad x\langle u\rangle \quad | \quad P_1|P_2 \quad | \quad \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$$

- $x\langle u\rangle$: émission
- $P_1|P_2$: activité parallèle
- $\texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$: nouveau

Le join-calcul

La syntaxe des processus est :

$$P \quad ::= \quad x\langle u\rangle \quad | \quad P_1|P_2 \quad | \quad \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$$

- $x\langle u\rangle$: émission
- $P_1|P_2$: activité parallèle
- $\texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$: nouveau
  - $x\langle u\rangle\,|y\langle v\rangle \triangleright P_1$: *join-pattern*

    réactions définies dynamiquement

Le join-calcul

La syntaxe des processus est :

$$P \quad ::= \quad x\langle u\rangle \quad | \quad P_1|P_2 \quad | \quad \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$$

- $x\langle u\rangle$: émission
- $P_1|P_2$: activité parallèle
- $\texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$: nouveau
  - $x\langle u\rangle\,|y\langle v\rangle \triangleright P_1$: *join-pattern* (lie u et v dans P_1)

    réactions définies dynamiquement
  - $\texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$: lie x et y dans P_2

    remplace restriction ($\rightarrow$ sécu), réplication, réception

Exemples de processus

- $\texttt{def}\ x\,\langle u\rangle \triangleright y\,\langle u\rangle\ \texttt{in}\ P$

Exemples de processus

- $\texttt{def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $\texttt{def}\ y\langle u\rangle \triangleright x\langle u\rangle\ \texttt{in def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$

Exemples de processus

- $\texttt{def}\ x\,\langle u\rangle \triangleright y\,\langle u\rangle\ \texttt{in}\ P$
- $\texttt{def}\ y\,\langle u\rangle \triangleright x\,\langle u\rangle\ \texttt{in def}\ x\,\langle u\rangle \triangleright y\,\langle u\rangle\ \texttt{in}\ P$
- $x_1\,\langle u\rangle\,|x_2\,\langle v\rangle \triangleright x\,\langle u, v\rangle$ encodage

Exemples de processus

- $\texttt{def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $\texttt{def}\ y\langle u\rangle \triangleright x\langle u\rangle\ \texttt{in def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $x_1\langle u\rangle \,|\, x_2\langle v\rangle \triangleright x\langle u, v\rangle$ encodage
- $\texttt{def}\ x\langle v\rangle \,|\, y\langle \kappa\rangle \triangleright \kappa\langle v\rangle\ \texttt{in}\ P$

Exemples de processus

- $\texttt{def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $\texttt{def}\ y\langle u\rangle \triangleright x\langle u\rangle\ \texttt{in def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $x_1\langle u\rangle \,|\, x_2\langle v\rangle \triangleright x\langle u, v\rangle$ encodage
- $\texttt{def}\ x\langle v\rangle \,|\, y\langle \kappa\rangle \triangleright \kappa\langle v\rangle\ \texttt{in}\ P$
- $\texttt{def}\ s\langle\rangle \triangleright P \wedge s\langle\rangle \triangleright Q\ \texttt{in}\ s\langle\rangle$

Exemples de processus

- $\texttt{def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $\texttt{def}\ y\langle u\rangle \triangleright x\langle u\rangle\ \texttt{in def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $x_1\langle u\rangle \,|x_2\langle v\rangle \triangleright x\langle u, v\rangle$ encodage
- $\texttt{def}\ x\langle v\rangle \,|y\langle \kappa\rangle \triangleright \kappa\langle v\rangle\ \texttt{in}\ P$
- $\texttt{def}\ s\langle\rangle \triangleright P \wedge s\langle\rangle \triangleright Q\ \texttt{in}\ s\langle\rangle$
- $\texttt{def}\ once\langle\rangle \,|y\langle v\rangle \triangleright x\langle v\rangle\ \texttt{in}\ y\langle 1\rangle \,|y\langle 2\rangle \,|y\langle 3\rangle \,|once\langle\rangle$

Exemples de processus

- $\texttt{def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $\texttt{def}\ y\langle u\rangle \triangleright x\langle u\rangle\ \texttt{in def}\ x\langle u\rangle \triangleright y\langle u\rangle\ \texttt{in}\ P$
- $x_1\langle u\rangle \,|x_2\langle v\rangle \triangleright x\langle u, v\rangle$ encodage
- $\texttt{def}\ x\langle v\rangle \,|y\langle \kappa\rangle \triangleright \kappa\langle v\rangle\ \texttt{in}\ P$
- $\texttt{def}\ s\langle\rangle \triangleright P \wedge s\langle\rangle \triangleright Q\ \texttt{in}\ s\langle\rangle$
- $\texttt{def}\ once\langle\rangle \,|y\langle v\rangle \triangleright x\langle v\rangle\ \texttt{in}\ y\langle 1\rangle \,|y\langle 2\rangle \,|y\langle 3\rangle \,|once\langle\rangle$
- $\texttt{def}\ loop\langle\rangle \triangleright P|loop\langle\rangle\ \texttt{in}\ loop\langle\rangle \,|Q$

Variables reçues, définies, libres

$$P \quad ::= \quad x\langle u\rangle \quad | \quad P_1|P_2 \quad | \quad \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle \triangleright P_1\ \texttt{in}\ P_2$$

$$\begin{aligned}
\mathrm{rv}(x\langle\tilde{v}\rangle) &\stackrel{\text{déf}}{=} \{u \in \tilde{v}\} \\
\mathrm{rv}(J|J') &\stackrel{\text{déf}}{=} \mathrm{rv}(J) \uplus \mathrm{rv}(J')
\end{aligned}$$

$$\begin{aligned}
\mathrm{dv}(x\langle\tilde{v}\rangle) &\stackrel{\text{déf}}{=} \{x\} \\
\mathrm{dv}(J|J') &\stackrel{\text{déf}}{=} \mathrm{dv}(J) \cup \mathrm{dv}(J') \\
\mathrm{dv}(J \triangleright P) &\stackrel{\text{déf}}{=} \mathrm{dv}(J)
\end{aligned}$$

$$\begin{aligned}
\mathrm{fv}(x\langle\tilde{v}\rangle) &\stackrel{\text{déf}}{=} \{x\} \cup \{u \in \tilde{v}\} \\
\mathrm{fv}(\texttt{def}\ D\ \texttt{in}\ P) &\stackrel{\text{déf}}{=} (\mathrm{fv}(P) \cup \mathrm{fv}(D)) - \mathrm{dv}(D) \\
\mathrm{fv}(P|P') &\stackrel{\text{déf}}{=} \mathrm{fv}(P) \cup \mathrm{fv}(P') \\
\mathrm{fv}(J \triangleright P) &\stackrel{\text{déf}}{=} \mathrm{dv}(J) \cup (\mathrm{fv}(P) - \mathrm{rv}(J))
\end{aligned}$$

La congruence structurelle $\equiv$

Plus petite relation d'équivalence telle que :

$$
\begin{array}{rcl}
P|Q & \equiv & Q|P \\
(P|Q)|R & \equiv & P|(Q|R) \\
P|(\texttt{def}\ D\ \texttt{in}\ Q) & \equiv & \texttt{def}\ D\ \texttt{in}\ P|Q \\
\texttt{def}\ D\ \texttt{in}\ \texttt{def}\ D'\ \texttt{in}\ P & \equiv & \texttt{def}\ D'\ \texttt{in}\ \texttt{def}\ D\ \texttt{in}\ P
\end{array}
$$

si $P \equiv_\alpha Q$ alors $P \equiv Q$

si $P \equiv Q$ alors $P|R \equiv Q|R$

si $R \equiv S$ et $P \equiv Q$ alors $\texttt{def}\ J \triangleright R\ \texttt{in}\ P \equiv \texttt{def}\ J \triangleright S\ \texttt{in}\ Q$

avec $\mathrm{dv}(D)$ et $\mathrm{dv}(D')$ *frais*

La congruence structurelle $\equiv$

Plus petite relation d'équivalence telle que :

$$\begin{aligned} P|Q &\equiv Q|P \\ (P|Q)|R &\equiv P|(Q|R) \\ P|(\texttt{def}\ D\ \texttt{in}\ Q) &\equiv \texttt{def}\ D\ \texttt{in}\ P|Q \\ \texttt{def}\ D\ \texttt{in}\ \texttt{def}\ D'\ \texttt{in}\ P &\equiv \texttt{def}\ D'\ \texttt{in}\ \texttt{def}\ D\ \texttt{in}\ P \end{aligned}$$

si $P \equiv_\alpha Q$ alors $P \equiv Q$

si $P \equiv Q$ alors $P|R \equiv Q|R$

si $R \equiv S$ et $P \equiv Q$ alors $\texttt{def}\ J \triangleright R\ \texttt{in}\ P \equiv \texttt{def}\ J \triangleright S\ \texttt{in}\ Q$

avec $\mathrm{dv}(D)$ et $\mathrm{dv}(D')$ *frais* :

$$x\langle y\rangle\,|\texttt{def}\ x\langle u\rangle\,|y\langle v\rangle\ \texttt{in}\ x\langle v\rangle \not\equiv \texttt{def}\ x\langle u\rangle\,|y\langle v\rangle\ \texttt{in}\ x\langle v\rangle\,|x\langle y\rangle$$

Un LTS

$\xrightarrow{\delta}$ avec $\delta \in \{D\} \cup \{\tau\}$; plus petite relation telle que :

- pour tout $D = x\langle u\rangle \,|y\langle v\rangle \triangleright R$,

$$x\langle s\rangle \,|y\langle t\rangle \xrightarrow{D} R[{}^{s}/_{u}, {}^{t}/_{v}]$$

$\xrightarrow{\tau}$: réduction

Un LTS

$\xrightarrow{\delta}$ avec $\delta \in \{D\} \cup \{\tau\}$; plus petite relation telle que :

- pour tout $D = x\langle u\rangle \,|y\langle v\rangle \triangleright R$,

$$x\langle s\rangle \,|y\langle t\rangle \xrightarrow{D} R[{}^{s}/_{u}, {}^{t}/_{v}]$$

- pour toute transition $P \xrightarrow{\delta} P'$,

$$\begin{array}{rcll} P|Q & \xrightarrow{\delta} & P'|Q & \\ \texttt{def}\ D\ \texttt{in}\ P & \xrightarrow{\delta} & \texttt{def}\ D\ \texttt{in}\ P' & \text{si } \mathrm{fv}(D) \cap \mathrm{dv}(\delta) = \emptyset \\ \texttt{def}\ \delta\ \texttt{in}\ P & \xrightarrow{\tau} & \texttt{def}\ \delta\ \texttt{in}\ P' & \text{si } \delta \neq \tau \\ Q & \xrightarrow{\delta} & Q' & \text{si } P \equiv Q \text{ et } P' \equiv Q' \end{array}$$

$\xrightarrow{\tau}$: réduction

La CHAM (solutions)

CHAM = . . . , métaphore
vision opérationnelle

Solutions : $\mathcal{R} \vdash \mathcal{M}$.

- processus :

$$P \quad ::= \quad x\langle \tilde{v} \rangle \quad | \quad \texttt{def}\ D\ \texttt{in}\ P \quad | \quad P|P$$

La CHAM (solutions)

CHAM = . . . , métaphore
vision opérationnelle

Solutions : $\mathcal{R} \vdash \mathcal{M}$.

- processus :

$$P \quad ::= \quad x\langle \tilde{v} \rangle \quad | \quad \texttt{def}\ D\ \texttt{in}\ P \quad | \quad P|P$$

- définition :

$$D \quad ::= \quad J \triangleright P \quad | \quad D \wedge D$$

La CHAM (solutions)

CHAM = . . . , métaphore
vision opérationnelle

Solutions : $\mathcal{R} \vdash \mathcal{M}$.

- processus :

$$P \quad ::= \quad x\langle\tilde{v}\rangle \quad | \quad \texttt{def}\ D\ \texttt{in}\ P \quad | \quad P|P$$

- définition :

$$D \quad ::= \quad J \triangleright P \quad | \quad D \wedge D$$

- join-pattern :

$$J \quad ::= \quad x\langle\tilde{v}\rangle \quad | \quad J|J$$

La CHAM (règles)

$$
\begin{array}{lrcl}
\text{(str-join)} & \vdash P|Q & \rightleftharpoons & \vdash P, Q \\
\text{(str-and)} & D \wedge E \vdash & \rightleftharpoons & D, E \vdash \\
\text{(str-def)} & \vdash \texttt{def}\ D\ \texttt{in}\ P & \rightleftharpoons & D\sigma_{\mathrm{dv}} \vdash P\sigma_{\mathrm{dv}} \\
\\
& J \triangleright P \vdash J\sigma_{\mathrm{rv}} & \rightarrow & J \triangleright P \vdash P\sigma_{\mathrm{rv}}
\end{array}
$$

explication des σ

chauffage / refroidissement : catalyseur (pt faible ?)

La CHAM (règles)

$$\begin{array}{lllllll}
\text{(str-join)} & & \vdash & P|Q & \rightleftharpoons & \vdash & P, Q \\
\text{(str-and)} & D \wedge E & \vdash & & \rightleftharpoons & D, E & \vdash \\
\text{(str-def)} & & \vdash & \texttt{def}\ D\ \texttt{in}\ P & \rightleftharpoons & D\sigma_{\mathrm{dv}} & \vdash\ P\sigma_{\mathrm{dv}} \\
\\
& J \triangleright P & \vdash & J\sigma_{\mathrm{rv}} & \rightarrow & J \triangleright P & \vdash\ P\sigma_{\mathrm{rv}}
\end{array}$$

explication des σ

chauffage / refroidissement : catalyseur (pt faible ?)

Correction :

- $P \equiv Q$ ssi $\vdash P \rightleftharpoons^* \vdash Q$
- $P \xrightarrow{\tau} Q$ ssi $\vdash P \rightarrow \vdash Q$

Observabilité

- barbelé asynchrone en sortie seulement :

$$P \Downarrow_x \quad \stackrel{\text{déf}}{=} \quad x \in \mathrm{fv}(P) \wedge \exists \tilde{v}, \mathcal{R}, \mathcal{M}, \emptyset \vdash P \rightarrow^* \mathcal{R} \vdash \mathcal{M}, x\,\langle \tilde{v} \rangle$$

x nom libre, teste la capa d'émettre, $\rightarrow^*$ pour $(\rightarrow \cup \rightleftharpoons)^*$

Observabilité

- barbelé asynchrone en sortie seulement :

$$P \Downarrow_x \quad \stackrel{\text{déf}}{=} \quad x \in \mathrm{fv}(P) \wedge \exists \tilde{v}, \mathcal{R}, \mathcal{M}, \emptyset \vdash P \rightarrow^* \mathcal{R} \vdash \mathcal{M}, x\,\langle \tilde{v} \rangle$$

x nom libre, teste la capa d'émettre, $\rightarrow^*$ pour $(\rightarrow \cup \rightleftharpoons)^*$

- congruence observationnelle : raffinement de $\Downarrow_x$ $i.e.$

1. $\forall x \in \mathcal{N}$, si $P \Downarrow_x$ alors $Q \Downarrow_x$
2. si $P \rightarrow^* P'$ alors $\exists Q', Q \rightarrow^* Q'$ et $P' \approx Q'$
3. $\forall D, \mathtt{def}\ D\ \mathtt{in}\ P \approx \mathtt{def}\ D\ \mathtt{in}\ Q$
4. $\forall R, R|P \approx R|Q$

Observabilité (exemples)

- si $\mathrm{fv}(P) = \emptyset$ alors $P \approx 0$

Observabilité (exemples)

- si $\mathrm{fv}(P) = \emptyset$ alors $P \approx 0$
- si $P \equiv Q$ alors $P \approx Q$

Observabilité (exemples)

- si $\mathrm{fv}(P) = \emptyset$ alors $P \approx 0$
- si $P \equiv Q$ alors $P \approx Q$
- $x\,\langle u \rangle \not\approx y\,\langle u \rangle$

Observabilité (exemples)

- si $\mathrm{fv}(P) = \emptyset$ alors $P \approx 0$
- si $P \equiv Q$ alors $P \approx Q$
- $x\,\langle u \rangle \not\approx y\,\langle u \rangle$
- $\texttt{def}\ z\,\langle t \rangle \triangleright t\,\langle u \rangle\ \texttt{in}\ z\,\langle x \rangle \not\approx \texttt{def}\ z\,\langle t \rangle \triangleright t\,\langle u \rangle\ \texttt{in}\ z\,\langle y \rangle$

Observabilité (exemples)

- si $\mathrm{fv}(P) = \emptyset$ alors $P \approx 0$
- si $P \equiv Q$ alors $P \approx Q$
- $x\,\langle u \rangle \not\approx y\,\langle u \rangle$
- $\texttt{def}\ z\,\langle t \rangle \triangleright t\,\langle u \rangle\ \texttt{in}\ z\,\langle x \rangle \not\approx \texttt{def}\ z\,\langle t \rangle \triangleright t\,\langle u \rangle\ \texttt{in}\ z\,\langle y \rangle$
- $z\,\langle x \rangle \not\approx z\,\langle y \rangle$

Observabilité (exemples)

- si $\mathrm{fv}(P) = \emptyset$ alors $P \approx 0$
- si $P \equiv Q$ alors $P \approx Q$
- $x\,\langle u \rangle \not\approx y\,\langle u \rangle$
- $\texttt{def}\ z\,\langle t \rangle \triangleright t\,\langle u \rangle\ \texttt{in}\ z\,\langle x \rangle \not\approx \texttt{def}\ z\,\langle t \rangle \triangleright t\,\langle u \rangle\ \texttt{in}\ z\,\langle y \rangle$
- $z\,\langle x \rangle \not\approx z\,\langle y \rangle$
- $\texttt{def}\ u\,\langle z \rangle \triangleright v\,\langle z \rangle\ \texttt{in}\ x\,\langle u \rangle \approx x\,\langle v \rangle$

Encodage du λ-calcul

- appel par nom : explication

$$
\begin{aligned}
[\![x]\!]_v &\stackrel{\text{déf}}{=} x\,\langle v\rangle \\
[\![\lambda x.T]\!]_v &\stackrel{\text{déf}}{=} \texttt{def}\ \kappa\,\langle x, w\rangle \triangleright [\![T]\!]_w\ \texttt{in}\ v\,\langle\kappa\rangle \\
[\![TU]\!]_v &\stackrel{\text{déf}}{=} \texttt{def}\ x\,\langle u\rangle \triangleright [\![U]\!]_u\ \texttt{in def}\ w\,\langle\kappa\rangle \triangleright \kappa\,\langle x, v\rangle\ \texttt{in}\ [\![T]\!]_w
\end{aligned}
$$

ss-ensb déterministe du join

Encodage du λ-calcul

- appel par nom : explication

$$
\begin{aligned}
[\![x]\!]_v &\stackrel{\text{déf}}{=} x\,\langle v\rangle \\
[\![\lambda x.T]\!]_v &\stackrel{\text{déf}}{=} \texttt{def}\ \kappa\,\langle x, w\rangle \triangleright [\![T]\!]_w\ \texttt{in}\ v\,\langle\kappa\rangle \\
[\![TU]\!]_v &\stackrel{\text{déf}}{=} \texttt{def}\ x\,\langle u\rangle \triangleright [\![U]\!]_u\ \texttt{in def}\ w\,\langle\kappa\rangle \triangleright \kappa\,\langle x, v\rangle\ \texttt{in}\ [\![T]\!]_w
\end{aligned}
$$

ss-ensb déterministe du join

- appel par valeur en parallèle :

$$
\begin{aligned}
[\![x]\!]_v &\stackrel{\text{déf}}{=} v\,\langle x\rangle \\
[\![\lambda x.T]\!]_v &\stackrel{\text{déf}}{=} \texttt{def}\ \kappa\,\langle x, w\rangle \triangleright [\![T]\!]_w\ \texttt{in}\ v\,\langle\kappa\rangle \\
[\![TU]\!]_v &\stackrel{\text{déf}}{=} \texttt{def}\ \tau\,\langle\kappa\rangle\,|u\,\langle w\rangle \triangleright \kappa\,\langle w, v\rangle\ \texttt{in}\ [\![T]\!]_t|[\![U]\!]_u
\end{aligned}
$$

confluent

Comparer le join-calcul et le π-calcul

- encodage totalement abstrait : si $\mathcal{P}_1$ et $\mathcal{P}_2$ sont deux calculs processus avec respectivement $\approx_1$ et $\approx_2$ comme équivalence sur leurs processus alors $\mathcal{P}_2$ est dit *plus expressif* que $\mathcal{P}_1$ s'il exsite un *encodage totalement abstrait* $[\![\ ]\!]_{1\to 2}$ de $\mathcal{P}_1$ dans $\mathcal{P}_2$ c'est-à-dire que pour tous processus P, Q de $\mathcal{P}_1$ l'on ait :

$$P \approx_1 Q \quad \Leftrightarrow \quad [\![P]\!]_{1\to 2} \approx_2 [\![Q]\!]_{1\to 2}$$

Comparer le join-calcul et le π-calcul

- encodage totalement abstrait : si $\mathcal{P}_1$ et $\mathcal{P}_2$ sont deux calculs processus avec respectivement $\approx_1$ et $\approx_2$ comme équivalence sur leurs processus alors $\mathcal{P}_2$ est dit *plus expressif* que $\mathcal{P}_1$ s'il exsite un *encodage totalement abstrait* $[\![\ ]\!]_{1\to 2}$ de $\mathcal{P}_1$ dans $\mathcal{P}_2$ c'est-à-dire que pour tous processus P, Q de $\mathcal{P}_1$ l'on ait :

$$P \approx_1 Q \quad \Leftrightarrow \quad [\![P]\!]_{1\to 2} \approx_2 [\![Q]\!]_{1\to 2}$$

- $\approx_\pi$: congruence barbelée asynchrone dont les barbes sont les émissions sur les canaux libres on prendra $\approx$ sur join

Comparer le join-calcul et le π-calcul

- encodage totalement abstrait : si $\mathcal{P}_1$ et $\mathcal{P}_2$ sont deux calculs processus avec respectivement $\approx_1$ et $\approx_2$ comme équivalence sur leurs processus alors $\mathcal{P}_2$ est dit *plus expressif* que $\mathcal{P}_1$ s'il exsite un *encodage totalement abstrait* $[\![\ ]\!]_{1\to 2}$ de $\mathcal{P}_1$ dans $\mathcal{P}_2$ c'est-à-dire que pour tous processus P, Q de $\mathcal{P}_1$ l'on ait :

$$P \approx_1 Q \quad \Leftrightarrow \quad [\![P]\!]_{1\to 2} \approx_2 [\![Q]\!]_{1\to 2}$$

- $\approx_\pi$: congruence barbelée asynchrone dont les barbes sont les émissions sur les canaux libres on prendra $\approx$ sur join
- on ne peut pas observer la réception d'un message : $x(u).\overline{x}u \approx_\pi 0$.

Comparer le join-calcul et le π-calcul

- encodage totalement abstrait : si $\mathcal{P}_1$ et $\mathcal{P}_2$ sont deux calculs processus avec respectivement $\approx_1$ et $\approx_2$ comme équivalence sur leurs processus alors $\mathcal{P}_2$ est dit *plus expressif* que $\mathcal{P}_1$ s'il exsite un *encodage totalement abstrait* $[\![\ ]\!]_{1\to 2}$ de $\mathcal{P}_1$ dans $\mathcal{P}_2$ c'est-à-dire que pour tous processus P, Q de $\mathcal{P}_1$ l'on ait :

$$P \approx_1 Q \quad \Leftrightarrow \quad [\![P]\!]_{1\to 2} \approx_2 [\![Q]\!]_{1\to 2}$$

- $\approx_\pi$: congruence barbelée asynchrone dont les barbes sont les émissions sur les canaux libres on prendra $\approx$ sur join
- équateur : $M^{\pi}_{x,y} \quad \stackrel{\text{déf}}{=} \quad !x(u).\overline{y}u|!y(v).\overline{x}v$

Comparer le join-calcul et le π-calcul

- encodage totalement abstrait : si $\mathcal{P}_1$ et $\mathcal{P}_2$ sont deux calculs processus avec respectivement $\approx_1$ et $\approx_2$ comme équivalence sur leurs processus alors $\mathcal{P}_2$ est dit *plus expressif* que $\mathcal{P}_1$ s'il exsite un *encodage totalement abstrait* $[\![\ ]\!]_{1\to 2}$ de $\mathcal{P}_1$ dans $\mathcal{P}_2$ c'est-à-dire que pour tous processus P, Q de $\mathcal{P}_1$ l'on ait :

$$P \approx_1 Q \quad \Leftrightarrow \quad [\![P]\!]_{1\to 2} \approx_2 [\![Q]\!]_{1\to 2}$$

- $\approx_\pi$: congruence barbelée asynchrone dont les barbes sont les émissions sur les canaux libres on prendra $\approx$ sur join
- équateur : $M^\pi_{x,y} \stackrel{\text{déf}}{=} !x(u).\overline{y}u|!y(v).\overline{x}v$

  Si $Q[^x/_y] \approx_\pi R[^x/_y]$ alors $M^\pi_{x,y}|Q \approx_\pi M^\pi_{x,y}|R$.

Encodage du π-calcul

- encodage naïf : pas fully abstract

$$
\begin{aligned}
[\![P|Q]\!]_\pi &\overset{\text{déf}}{=} [\![P]\!]_\pi \,|\, [\![Q]\!]_\pi \\
[\![\nu x.P]\!]_\pi &\overset{\text{déf}}{=} \texttt{def}\ x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \,|\, x_{\mathrm{i}}\langle\kappa\rangle \triangleright \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle\ \texttt{in}\ [\![P]\!]_\pi \\
[\![\overline{x}v]\!]_\pi &\overset{\text{déf}}{=} x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \\
[\![x(v).P]\!]_\pi &\overset{\text{déf}}{=} \texttt{def}\ \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright [\![P]\!]_\pi\ \texttt{in}\ x_{\mathrm{i}}\langle\kappa\rangle \\
[\![!x(v).P]\!]_\pi &\overset{\text{déf}}{=} \texttt{def}\ \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright x_{\mathrm{i}}\langle\kappa\rangle \,|\, [\![P]\!]_\pi\ \texttt{in}\ x_{\mathrm{i}}\langle\kappa\rangle
\end{aligned}
$$

Encodage du π-calcul

- encodage naïf : pas fully abstract

$$
\begin{aligned}
[\![P|Q]\!]_\pi &\stackrel{\text{déf}}{=} [\![P]\!]_\pi \,|\, [\![Q]\!]_\pi \\
[\![\nu x.P]\!]_\pi &\stackrel{\text{déf}}{=} \texttt{def}\ x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \,|\, x_{\mathrm{i}}\langle \kappa\rangle \triangleright \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle\ \texttt{in}\ [\![P]\!]_\pi \\
[\![\overline{x}v]\!]_\pi &\stackrel{\text{déf}}{=} x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \\
[\![x(v).P]\!]_\pi &\stackrel{\text{déf}}{=} \texttt{def}\ \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright [\![P]\!]_\pi\ \texttt{in}\ x_{\mathrm{i}}\langle \kappa\rangle \\
[\![!x(v).P]\!]_\pi &\stackrel{\text{déf}}{=} \texttt{def}\ \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright x_{\mathrm{i}}\langle \kappa\rangle \,|\, [\![P]\!]_\pi\ \texttt{in}\ x_{\mathrm{i}}\langle \kappa\rangle
\end{aligned}
$$

- pare-feus :

$$
\begin{aligned}
\mathcal{P}_x[\,] &\stackrel{\text{déf}}{=} \texttt{def}\ x_{\mathrm{l}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \,|\, x_{\mathrm{i}}\langle \kappa\rangle \triangleright \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle\ \texttt{in} \\
&\qquad \texttt{def}\ x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright p\langle v_{\mathrm{o}}, v_{\mathrm{i}}, x_{\mathrm{l}}\rangle\ \texttt{in}\ [\,] \\
\mathcal{E}_x[\,] &\stackrel{\text{déf}}{=} \mathcal{P}_x\,[x_{\mathrm{e}}\langle x_{\mathrm{o}}, x_{\mathrm{i}}\rangle \,|\, [\,]] \\
\mathcal{M}[\,] &\stackrel{\text{déf}}{=} \texttt{def}\ p\langle x_{\mathrm{o}}, x_{\mathrm{i}}, \kappa\rangle \triangleright \mathcal{P}_y[\kappa\langle y_{\mathrm{o}}, y_{\mathrm{i}}\rangle \,|\, [\![M^{\pi}_{x,y}]\!]]\ \texttt{in}\ [\,]
\end{aligned}
$$

Encodage du π-calcul

- encodage naïf : pas fully abstract

$$
\begin{aligned}
[\![P|Q]\!]_\pi &\overset{\text{déf}}{=} [\![P]\!]_\pi \,|\, [\![Q]\!]_\pi \\
[\![\nu x.P]\!]_\pi &\overset{\text{déf}}{=} \texttt{def}\ x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \,|\, x_{\mathrm{i}}\langle\kappa\rangle \triangleright \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle\ \texttt{in}\ [\![P]\!]_\pi \\
[\![\overline{x}v]\!]_\pi &\overset{\text{déf}}{=} x_{\mathrm{o}}\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \\
[\![x(v).P]\!]_\pi &\overset{\text{déf}}{=} \texttt{def}\ \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright [\![P]\!]_\pi\ \texttt{in}\ x_{\mathrm{i}}\langle\kappa\rangle \\
[\![!x(v).P]\!]_\pi &\overset{\text{déf}}{=} \texttt{def}\ \kappa\langle v_{\mathrm{o}}, v_{\mathrm{i}}\rangle \triangleright x_{\mathrm{i}}\langle\kappa\rangle \,|\, [\![P]\!]_\pi\ \texttt{in}\ x_{\mathrm{i}}\langle\kappa\rangle
\end{aligned}
$$

- $P \approx_\pi Q$ si et seulement si $\mathcal{E}[[\![P]\!]_\pi] \approx \mathcal{E}[[\![Q]\!]_\pi]$ avec $\mathcal{E}[[\![P]\!]_\pi] \overset{\text{déf}}{=} \mathcal{M}[\mathcal{E}_{x_1}[\ldots\mathcal{E}_{x_i}[\ldots\mathcal{E}_{x_k}[[\![P]\!]_\pi]\ldots]\ldots]]$ où $\{x_i\} = \mathrm{fv}(P)$.

Encodage dans le π-calcul

- encodage naïf : plus simple

$$
\begin{aligned}
[\![Q|R]\!]_{\mathrm{j}} &\stackrel{\text{déf}}{=} [\![Q]\!]_{\mathrm{j}} | [\![R]\!]_{\mathrm{j}} \\
[\![x\,\langle v\rangle]\!]_{\mathrm{j}} &\stackrel{\text{déf}}{=} \overline{x}v \\
[\![\texttt{def}\ x\,\langle u\rangle\,|y\,\langle v\rangle \triangleright Q\ \texttt{in}\ R_{\mathrm{j}}]\!]_{\mathrm{j}} &\stackrel{\text{déf}}{=} \nu xy.\,(!x(u).y(v).[\![Q]\!]_{\mathrm{j}} | [\![R]\!]_{\mathrm{j}})
\end{aligned}
$$

Encodage dans le π-calcul

- encodage naïf : plus simple

$$
\begin{array}{rcl}
[\![Q|R]\!]_{\mathrm{j}} & \stackrel{\text{déf}}{=} & [\![Q]\!]_{\mathrm{j}}|[\![R]\!]_{\mathrm{j}} \\
[\![x\,\langle v\rangle]\!]_{\mathrm{j}} & \stackrel{\text{déf}}{=} & \overline{x}v \\
[\![\mathtt{def}\ x\,\langle u\rangle\,|y\,\langle v\rangle \triangleright Q\ \mathtt{in}\ R_{\mathrm{j}}]\!]_{\mathrm{j}} & \stackrel{\text{déf}}{=} & \nu xy.\,(!x(u).y(v).[\![Q]\!]_{\mathrm{j}}|[\![R]\!]_{\mathrm{j}})
\end{array}
$$

- pare-feu :

$$
\begin{array}{rcl}
R_{xy} & \stackrel{\text{déf}}{=} & !x(v).\nu v_{\mathrm{e}}.(\overline{r}v_{\mathrm{e}}v|\overline{y}v_{\mathrm{e}}) \\
\mathcal{R}[\,] & \stackrel{\text{déf}}{=} & \nu r.!r(x,x_{\mathrm{e}}).R_{xx_{\mathrm{e}}}|[\,] \\
\mathcal{E}^{\pi}_{x}[\,] & \stackrel{\text{déf}}{=} & \nu x.(R_{xx_{\mathrm{e}}}|[\,])
\end{array}
$$

Encodage dans le π-calcul

- encodage naïf : plus simple

$$
\begin{array}{rcl}
[\![Q|R]\!]_{\mathrm{j}} & \stackrel{\text{déf}}{=} & [\![Q]\!]_{\mathrm{j}}|[\![R]\!]_{\mathrm{j}} \\
[\![x\,\langle v\rangle]\!]_{\mathrm{j}} & \stackrel{\text{déf}}{=} & \overline{x}v \\
[\![\mathtt{def}\ x\,\langle u\rangle\,|y\,\langle v\rangle \triangleright Q\ \mathtt{in}\ R_{\mathrm{j}}]\!]_{\mathrm{j}} & \stackrel{\text{déf}}{=} & \nu xy.\,(!x(u).y(v).[\![Q]\!]_{\mathrm{j}}|[\![R]\!]_{\mathrm{j}})
\end{array}
$$

- pare-feu :

$$
\begin{array}{rcl}
R_{xy} & \stackrel{\text{déf}}{=} & !x(v).\nu v_{\mathrm{e}}.(\overline{r}v_{\mathrm{e}}v|\overline{y}v_{\mathrm{e}}) \\
\mathcal{R}[\,] & \stackrel{\text{déf}}{=} & \nu r.!r(x,x_{\mathrm{e}}).R_{xx_{\mathrm{e}}}|[\,] \\
\mathcal{E}^{\pi}_{x}[\,] & \stackrel{\text{déf}}{=} & \nu x.(R_{xx_{\mathrm{e}}}|[\,])
\end{array}
$$

- $Q \approx R$ si et seulement si $\mathcal{E}^{\pi}[[\![Q]\!]_{\mathrm{j}}] \approx_{\pi} \mathcal{E}^{\pi}[[\![R]\!]_{\mathrm{j}}]$.

JoCaml

http://pauillac.inria.fr/jocaml/

Démonstration

Et maintenant, regardez la console. . .