Typage d'une version décorée du π-calcul

Julien Gabet
sous la direction d'Emmanuel Beffara

Institut de Mathématiques de Marseille

Mars-Juin 2018

INSTITUT
de MATHÉMATIQUES
de MARSEILLE

Introduction

1. Définitions
   - Le π-calcul
   - Une version annotée du π-calcul

2. Exécution du calcul annoté

3. Typage du calcul décoré avec MLL
   - Le système de types
   - L'élimination de la coupure

Conclusion

- Processus mobiles : objets en interaction sur un réseau, actions non déterministes.
- Modèle habituel : π-calcul.
- Garantir le comportement d'un objet du modèle : typage.
- Notre approche : annoter les termes pour guider l'exécution.

Le π-calcul

- Syntaxe :

$$
\begin{array}{rll}
P, Q ::= & 0 & \text{processus inactif} \\
& P \big| Q & \text{mise en parallèle} \\
& u(t).P \;\; ; \;\; \bar{u}(v).P & \text{réception/envoi de nom} \\
& (\nu u)P & \text{privatisation de nom}
\end{array}
$$

Le π-calcul

- Syntaxe :

$$\begin{array}{rll} P, Q ::= & 0 & \text{processus inactif} \\ & P \mid Q & \text{mise en parallèle} \\ & u(t).P \ ; \ \bar{u}(v).P & \text{réception/envoi de nom} \\ & (\nu u)P & \text{privatisation de nom} \end{array}$$

- $|$ est commutatif, associatif et admet 0 pour élément neutre.

Le π-calcul

- Syntaxe :

$$
\begin{array}{rll}
P, Q ::= & 0 & \text{processus inactif} \\
& P \mid Q & \text{mise en parallèle} \\
& u(t).P \quad ; \quad \bar{u}(v).P & \text{réception/envoi de nom} \\
& (\nu u)P & \text{privatisation de nom}
\end{array}
$$

- $\mid$ est commutatif, associatif et admet 0 pour élément neutre.
- Règle de réduction :

$$u(t).P \mid \bar{u}(v).Q \to P[v/t] \mid Q$$

Cette réduction se fait en contexte

Exemple

Un terme simple pour voir le système de réduction :

$$\Big(u(t).P \,\Big|\, Q\Big) \,\Big|\, \bar{u}(v).R$$

Exemple

Un terme simple pour voir le système de réduction :

$$\Big(u(t).P \Big| Q\Big) \Big| \bar{u}(v).R \equiv Q \Big| \Big(u(t).P \Big| \bar{u}(v).R\Big)$$

Exemple

Un terme simple pour voir le système de réduction :

$$\begin{aligned}\Big(u(t).P\Big|Q\Big)\Big|\bar{u}(v).R \equiv & Q\Big|\Big(u(t).P\Big|\bar{u}(v).R\Big)\\ \rightarrow & Q\Big|\Big(P[v/t]\Big|R\Big)\end{aligned}$$

Exemple

Un exemple non confluent :

$$\begin{array}{ccc} \Big(u(t).P \,\Big|\, \bar{u}(v).Q\Big) \,\Big|\, u(t).S & \equiv & u(t).P \,\Big|\, \Big(\bar{u}(v).Q \,\Big|\, u(t).S\Big) \\ \downarrow & & \downarrow \\ \Big(P[v/t] \,\Big|\, Q\Big) \,\Big|\, u(t).S & & u(t).P \,\Big|\, \Big(Q \,\Big|\, S[v/t]\Big) \end{array}$$

Exemple

Annotons l'exemple non confluent :

$$\begin{array}{ccc}
\left(u_x(t).P \|_x \bar{u}_x(v).Q\right) \|_y u_y(t).S & \not\equiv & u_x(t).P \|_x \left(\bar{u}_y(v).Q \|_y u_y(t).S\right) \\
\downarrow & & \downarrow \\
\left(P[v/t] \|_x Q\right) \|_y u_y(t).S & & u_x(t).P \|_x \left(Q \|_y S[v/t]\right)
\end{array}$$

Les termes annotés

Les termes annotés sont définis comme suit :

$$
\begin{aligned}
P, Q ::= & \; x \leftrightarrow y \;\; ; \;\; 0_x \;\; ; && \text{termes de base} \\
& P \big|_x Q \;\; ; \;\; P \big|\big|_x Q \;\; ; && \text{parallèle et synchronisation} \\
& u_x(t).P \;\; ; \;\; \bar{u}_x(v).P \;\; ; && \text{préfixes d'action} \\
& \epsilon_x.P \;\; ; \;\; \lambda_x y.P \;\; ; && \text{préfixes de planification} \\
& (\nu u)P && \text{privatisation de nom}
\end{aligned}
$$

où $x \in fv(P) \cap fv(Q)$ pour les règles $|$ et $||$,
$x \notin fv(P)$ pour la règle ϵ, et $x, y \in fv(P)$ pour les autres règles.

La réduction

- Règles de réduction :

$$\begin{array}{ll} \epsilon_x.P||_x 0_x \to P & (P|_x Q)||_x \lambda_x y.R \to P||_x(Q[y/x]||_y R) \\ P||_x x \leftrightarrow y \to P[y/x] & \bar{u}_x(v).P||_x u_x(t).Q \to P||_x Q[v/t] \end{array}$$

La réduction

- Règles de réduction :

$$\epsilon_x.P||_x 0_x \to P \qquad (P|_x Q)||_x \lambda_x y.R \to P||_x(Q[y/x]||_y R)$$
$$P||_x x \leftrightarrow y \to P[y/x] \quad \bar{u}_x(v).P||_x u_x(t).Q \to P||_x Q[v/t]$$

- Règles de remontée du $||$:
  Si le contexte l'exige, on note P^y si le terme P contient y.
  $(P^y|_x Q)||_y R \succ (P||_y R)|_x Q \qquad (P|_x Q^y)||_y R \succ P|_x(Q||_y R)$

La réduction

Il faut une congruence pour faire confluer $\succ$:

$$(P^y|_x Q)|_y R \equiv (P^x|_y R)|_x Q \qquad P|_x(Q|_y R^x) \equiv Q|_y(P|_x R^y)$$
$$(P|_x Q^y)|_y R \equiv P|_x(Q^x|_y R)$$

Ces règles font commuter les relations selon des variables distinctes.

Proposition

- $\rightarrow$ *est fortement confluent*
- $\succ$ *est confluent à* $\equiv$ *près.*

Confluence de $\rightarrow$ et $\succ$

Pour $\rightarrow$, seule la règle $x \leftrightarrow y$ interagit avec elle-même :

$$
\begin{tikzcd}
& x \leftrightarrow y \|_x x \leftrightarrow z \arrow[dl] \arrow[dr] & \\
(x \leftrightarrow z)[y/x] \arrow[d, equal] & & (x \leftrightarrow y)[z/x] \arrow[d, equal] \\
y \leftrightarrow z & = & z \leftrightarrow y
\end{tikzcd}
$$

Confluence de $\to$ et $\succ$

Pour $\succ$, on regarde un cas (les autres sont similaires) :

$$
\begin{array}{ccc}
 & (P^y|_x Q)||_y (R^y|_z S) & \\
\swarrow_{\succ\, left} & & \searrow_{\succ\, right} \\
\Big(P^y||_y (R^y|_z S)\Big)\Big|_x Q & & \Big((P^y|_x Q)||_y R^y\Big)\Big|_z S \\
\curlyvee & & \curlyvee \\
\Big((P^y||_y R^y)|_z S\Big)\Big|_x Q & \equiv & \Big((P^y||_y R^y)|_x Q\Big)\Big|_z S
\end{array}
$$

Les règles de typage

$$\frac{}{0_x \vdash x : 1}\ \text{nop} \qquad \frac{P \vdash \Gamma \quad x \notin \Gamma}{\epsilon_x.P \vdash \Gamma, x : \bot}\ \text{bot}$$

$$\frac{}{x \leftrightarrow y \vdash x : E^\perp, y : E}\ \text{ax} \qquad \frac{P \vdash \Gamma, x : E, y : F}{\lambda_x y.P \vdash \Gamma, x : E ⅋ F}\ \text{lam}$$

$$\frac{P \vdash \Gamma, x : E \quad Q \vdash \Delta, x : F}{P|_x Q \vdash \Gamma, \Delta, x : E \otimes F}\ \text{para} \qquad \frac{P \vdash \Gamma, x : E \quad Q \vdash \Delta, x : E^\perp}{P||_x Q \vdash \Gamma, \Delta}\ \text{cut}$$

$$\frac{P \vdash \Gamma, x : A[v/t]^\perp}{\bar{u}_x(v).P \vdash \Gamma, x : \exists_u t.A^\perp}\ \text{in} \qquad \frac{P \vdash \Gamma, x : A \quad t \notin \Gamma}{u_x(t).P \vdash \Gamma, x : \forall_u t.A}\ \text{out}$$

Propriétés des relations typées

Proposition

Les relations $\rightarrow$, $\succ$ et $\equiv$ préservent le type,
ie. si $P \vdash \Gamma$ et $P \alpha P'$, alors $P' \vdash \Gamma$. (pour α l'une de ces relations)

• La preuve est standard, on regarde chaque cas de réécriture avec les règles de typage correspondantes.

Propriétés des relations typées

Proposition

Les relations $\rightarrow$, $\succ$ et $\equiv$ préservent le type,
ie. si $P \vdash \Gamma$ et $P \alpha P'$, alors $P' \vdash \Gamma$. *(pour α l'une de ces relations)*

• La preuve est standard, on regarde chaque cas de réécriture avec les règles de typage correspondantes.

Proposition

Les relations $\rightarrow$ et $\succ$ terminent, ie. n'admettent pas de chaîne de réécriture infinie.

• La preuve est également assez standard, pour un bon choix de mesure de terminaison. Une mesure qui fonctionne est la somme des tailles des sous-termes immédiats de la coupure/synchronization.

Élimination de la coupure

On ajoute des règles pour passer et agir sous les préfixes :

$$\epsilon_x.P||_y Q \gtrsim \epsilon_x.(P||_y Q) \qquad u_x(t).P||_y Q \gtrsim u_x(t).(P||_y Q)$$
$$\lambda_x y.P||_z Q \gtrsim \lambda_x y.(P||_z Q) \qquad \bar{u}_x(v).P||_y Q \gtrsim \bar{u}_x(v).(P||_y Q)$$

$$P|_x\alpha_y.Q \cong \alpha_y.(P|_x Q) \quad \alpha\cdot \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\}, y, t \notin fv(P)$$
$$\alpha_x.\beta_y.P \cong \beta_y.\alpha_x.P \qquad \alpha., \beta. \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\}, x \neq y$$

Et $\rightarrow$ est étendue en $\rightsquigarrow$ qui agit sous les préfixes.

Élimination de la coupure

On ajoute des règles pour passer et agir sous les préfixes :

$$\epsilon_x.P||_y Q \gtrsim \epsilon_x.(P||_y Q) \qquad u_x(t).P||_y Q \gtrsim u_x(t).(P||_y Q)$$
$$\lambda_x y.P||_z Q \gtrsim \lambda_x y.(P||_z Q) \qquad \bar{u}_x(v).P||_y Q \gtrsim \bar{u}_x(v).(P||_y Q)$$

$$P|_x \alpha_y.Q \cong \alpha_y.(P|_x Q) \quad \alpha\cdot \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\}, y, t \notin fv(P)$$
$$\alpha_x.\beta_y.P \cong \beta_y.\alpha_x.P \qquad \alpha., \beta. \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\}, x \neq y$$

Et $\rightarrow$ est étendue en $\rightsquigarrow$ qui agit sous les préfixes.

- Ces nouvelles règles préservent le type, $\rightsquigarrow$ et $\gtrsim$ terminent et confluent comme avant.

Élimination de la coupure

On ajoute des règles pour passer et agir sous les préfixes :

$$\epsilon_x.P||_y Q \gtrsim \epsilon_x.(P||_y Q) \qquad u_x(t).P||_y Q \gtrsim u_x(t).(P||_y Q)$$
$$\lambda_x y.P||_z Q \gtrsim \lambda_x y.(P||_z Q) \qquad \bar{u}_x(v).P||_y Q \gtrsim \bar{u}_x(v).(P||_y Q)$$

$$P|_x\alpha_y.Q \cong \alpha_y.(P|_x Q) \quad \alpha\cdot \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\}, y, t \notin fv(P)$$
$$\alpha_x.\beta_y.P \cong \beta_y.\alpha_x.P \qquad \alpha., \beta. \in \{\epsilon., \lambda.z, u.(t), \bar{u}.\langle v\rangle\}, x \neq y$$

Et $\rightarrow$ est étendue en $\rightsquigarrow$ qui agit sous les préfixes.

- Ces nouvelles règles préservent le type, $\rightsquigarrow$ et $\gtrsim$ terminent et confluent comme avant.
- Ces règles éliminent la coupure.

Conclusion

Les aspects techniques non abordés :

- effacement des annotations,
- actions internes (non observable depuis l'extérieur),
- la confluence de $\rightarrow$ après quotient par $\equiv$,
- les limitations techniques de cette approche.

Les aspects à rechercher encore :

- partie additive de π et de LL,
- réplication de π et modalités de LL,
- annoter un terme à partir d'une réduction connue ?