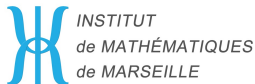


Typage d'une version décorée du π -calcul

Julien Gabet
sous la direction d'Emmanuel Beffara

Institut de Mathématiques de Marseille

Mars-Juin 2018



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 :

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

Le π -calcul

► Syntaxe :

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

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

Le π -calcul

► Syntaxe :

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

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

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

Cette réduction se fait en contexte.

Exemple

Un terme simple pour voir le système de réduction :

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

Exemple

Un terme simple pour voir le système de réduction :

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

Exemple

Un terme simple pour voir le système de réduction :

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

Exemple

Un exemple non confluent :

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

Exemple

Annotons l'exemple non confluent :

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

Les termes annotés

Les termes annotés sont définis comme suit :

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

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 :

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

La réduction

► Règles de réduction :

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

► 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 :

$$\begin{aligned}(P^y|_x Q)|_y R &\equiv (P^x|_y R)|_x Q & P|_x(Q|_y R^x) &\equiv Q|_y(P|_x R^y) \\ (P|_x Q^y)|_y R &\equiv P|_x(Q^x|_y R)\end{aligned}$$

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 :

$$\begin{array}{ccc}
 & x \leftrightarrow y \parallel_x x \leftrightarrow z & \\
 \swarrow & & \searrow \\
 (x \leftrightarrow z)[y/x] & & (x \leftrightarrow y)[z/x] \\
 \parallel & & \parallel \\
 y \leftrightarrow z & = & z \leftrightarrow y
 \end{array}$$

Confluence de \rightarrow et \succ

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

$$(P^y|_x Q) ||_y (R^y|_z S)$$

 \succ_{left}
 \succ_{right}

$$\left(P^y ||_y (R^y|_z S) \right) |_x Q$$

$$\left((P^y|_x Q) ||_y R^y \right) |_z S$$

 Υ
 Υ

$$\left((P^y ||_y R^y) |_z S \right) |_x Q$$

 \equiv

$$\left((P^y ||_y R^y) |_x Q \right) |_z S$$

Des termes dont la réduction se bloque

- ▶ une action contre un autre préfixe :

$$\epsilon_x.0_y ||_x u_x(t).0_x$$

- ▶ deux actions qui s'entremêlent :

$$\bar{u}_x a.v_x(t).P ||_x \bar{v}_x b.u_x(t').Q$$

Les règles de typage

$$\frac{}{0_x \vdash x : 1} \text{ nop}$$

$$\frac{P \vdash \Gamma \quad x \notin \Gamma}{\epsilon_x.P \vdash \Gamma, x : \perp} \text{ bot}$$

$$\frac{}{x \leftrightarrow y \vdash x : E^\perp, y : E} \text{ ax}$$

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

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

$$\frac{P \vdash \Gamma, x : E, y : F}{\lambda_x y. P \vdash \Gamma, x : E \wp F} \text{ lam}$$

$$\frac{P \vdash \Gamma, x : A[v/t]^\perp}{\bar{u}_x v. P \vdash \Gamma, x : \exists_u t. A^\perp} \text{ in}$$

$$\frac{P \vdash \Gamma, x : A \quad t \notin \Gamma}{u_x(t). P \vdash \Gamma, x : \forall_u t. A} \text{ out}$$

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/synchronisation.

Élimination de la coupure

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

$$\begin{array}{ll}
 \epsilon_x.P||_y Q \rightsquigarrow \epsilon_x.(P||_y Q) & u_x(t).P||_y Q \rightsquigarrow u_x(t).(P||_y Q) \\
 \lambda_x y.P||_z Q \rightsquigarrow \lambda_x y.(P||_z Q) & \bar{u}_x v.P||_y Q \rightsquigarrow \bar{u}_x v.(P||_y Q)
 \end{array}$$

$$\begin{array}{ll}
 P|_x \alpha_y.Q \cong \alpha_y.(P|_x Q) & \alpha \in \{\epsilon., \lambda.z, u.(t), \bar{u}.v\}, y, t \notin fv(P) \\
 \alpha_x.\beta_y.P \cong \beta_y.\alpha_x.P & \alpha., \beta. \in \{\epsilon., \lambda.z, u.(t), \bar{u}.v\}, x \neq y
 \end{array}$$

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 :

$$\begin{array}{ll} \epsilon_x.P||_yQ \rightsquigarrow \epsilon_x.(P||_yQ) & u_x(t).P||_yQ \rightsquigarrow u_x(t).(P||_yQ) \\ \lambda_xy.P||_zQ \rightsquigarrow \lambda_xy.(P||_zQ) & \bar{u}_xv.P||_yQ \rightsquigarrow \bar{u}_xv.(P||_yQ) \end{array}$$

$$\begin{array}{ll} P|_x\alpha_y.Q \cong \alpha_y.(P|_xQ) & \alpha \cdot \in \{\epsilon., \lambda.z, u.(t), \bar{u}.v\}, y, t \notin fv(P) \\ \alpha_x.\beta_y.P \cong \beta_y.\alpha_x.P & \alpha., \beta. \in \{\epsilon., \lambda.z, u.(t), \bar{u}.v\}, x \neq y \end{array}$$

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

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

Élimination de la coupure

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

$$\begin{aligned}
 \epsilon_x.P||_yQ &\rightsquigarrow \epsilon_x.(P||_yQ) & u_x(t).P||_yQ &\rightsquigarrow u_x(t).(P||_yQ) \\
 \lambda_xy.P||_zQ &\rightsquigarrow \lambda_xy.(P||_zQ) & \bar{u}_xv.P||_yQ &\rightsquigarrow \bar{u}_xv.(P||_yQ)
 \end{aligned}$$

$$\begin{aligned}
 P|_x\alpha_y.Q &\cong \alpha_y.(P|_xQ) & \alpha. &\in \{\epsilon., \lambda.z, u.(t), \bar{u}.v\}, y, t \notin fv(P) \\
 \alpha_x.\beta_y.P &\cong \beta_y.\alpha_x.P & \alpha., \beta. &\in \{\epsilon., \lambda.z, u.(t), \bar{u}.v\}, x \neq y
 \end{aligned}$$

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

- Ces nouvelles règles préservent le type, \rightsquigarrow et \rightsquigarrow terminent et confluents 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,
- ▶ réplification de π et modalités de LL,
- ▶ annoter un terme à partir d'une réduction connue?
- ▶ place du système annoté parmi les systèmes de types existants