

HOCore en Coq : résumé

Aurèle Barrière

28 janvier 2016

Table des matières

1	Introduction à HOCore	1
1.1	Pi-calcul	1
1.2	Pi-calcul d'ordre supérieur : HOPi	2
1.3	HOCore	2
1.4	Réductions	2
2	Équivalence décidable	2
2.1	alpha-conversion	2
2.2	Indices de De Bruijn	3
3	Bissimilarités	3
4	Formalisation en Coq	3
4.1	axiomatisation	3
4.2	Correction de preuves	3

1 Introduction à HOCore

1.1 Pi-calcul

Le π -calcul est un langage formel utilisé pour décrire, en particulier, les exécutions distribuées de processus. Sa syntaxe, très simple, décrit simplement l'exécution en parallèle.

En π -calcul, on manipule des processus, qui peuvent s'exécuter séquentiellement ou parallèlement et terminer ou non. Des canaux sont également disponibles pour la réception et l'émission de messages ou de variables.

Le π -calcul utilise donc la grammaire suivante :

$P = 0$	fin du processus
$!P$	répéter le processus
$P P$	lancer les deux processus en parallèle
$x(y).P$	lire un message sur le canal x pour remplacer y , puis lancer P
$\bar{x}(y).P$	envoyer le message y sur le canal x , puis lancer P
$(\nu x)P$	réserver le nom x pour le processus P

Il s'agit d'un calcul Turing Complet.

1.2 Pi-calcul d'ordre supérieur : HOPi

Pour l'ordre supérieur, on se permet de communiquer par les canaux aussi bien des noms (variables) que des processus.

Dans la grammaire proposée plus haut, x et y peuvent donc désigner des processus.

1.3 HOCore

Il s'agit d'une restriction qui conserve le caractère Turing Complet du π -calcul d'ordre supérieur.

La grammaire utilisée est la suivante :

$$\begin{aligned}
 P = & 0 \\
 & x \\
 & P||P \\
 & x(y).P \\
 & \bar{x}(P)
 \end{aligned}$$

1.4 Réductions

Lorsqu'un processus attend un message sur un canal et qu'en parallèle, un autre processus émet un message sur ce même canal, on remplace toutes les instances de la variable.

2 Équivalence décidable

2.1 alpha-conversion

Le nom donné aux variables n'importe pas dans la sémantique d'un processus, mais pose un problème pour l'équivalence de processus.

2.2 Indices de De Bruijn

Il existe un moyen de se débarrasser des noms de variables pour éviter tout problème lié à l’alpha réduction.

3 Bissimilarités

4 Formalisation en Coq

Un des principaux travaux de l’équipe de recherche a été de formaliser HOCore en Coq (l’assistant de preuve).

4.1 Axiomatisation

On traduit la grammaire de HOCore ainsi :

4.2 Expression des transitions

HOCore utilise un système de transition labelées (LTS). Pour la formalisation en Coq, cela pose le problème du nom des variables “bound”.

4.3 Correction de preuves

Dans l’article [12], des preuves étaient fausses.