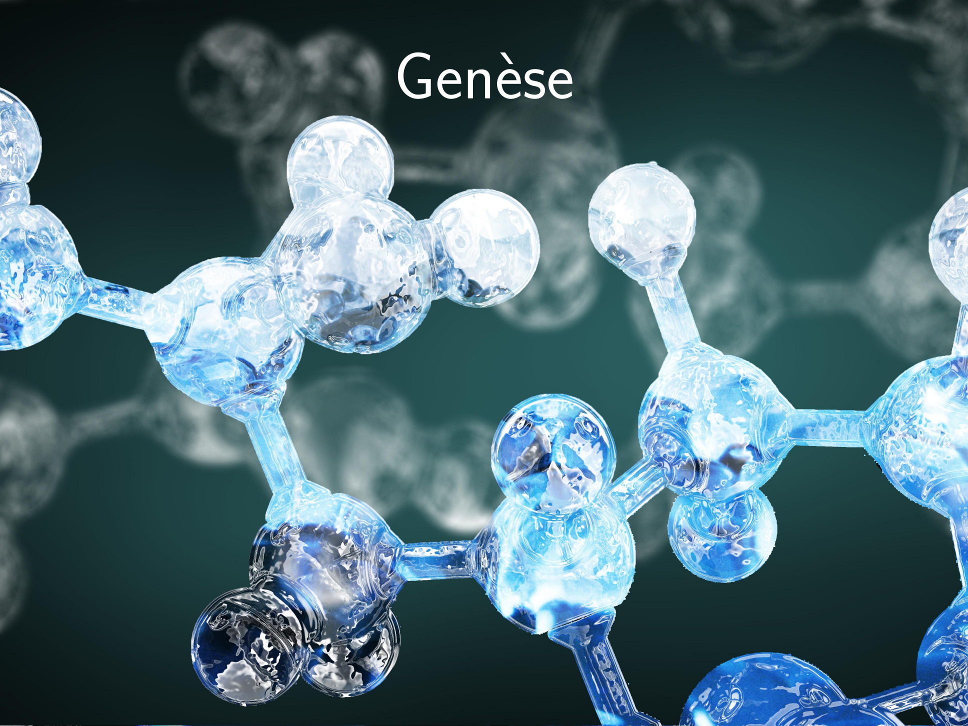
De la KAM avec un processus d'ordre supérieur

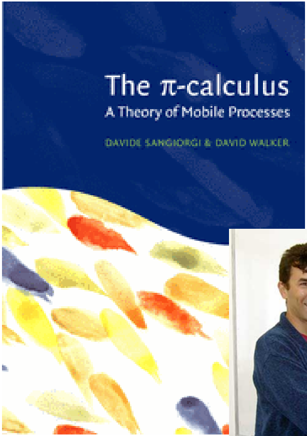
Damien Pous, Alan Schmitt

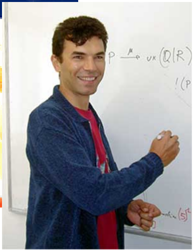
10 janvier 2013

Genèse



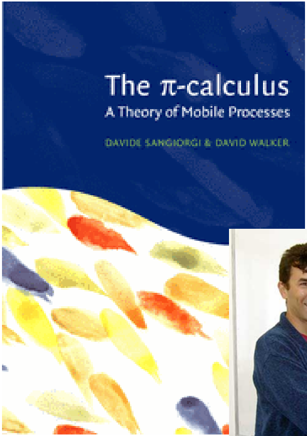
De HOπ à HOcore

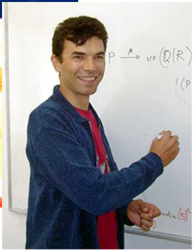




$$
\begin{aligned}
P ::= \; & \\
\mid \; & a(x).P \\
\mid \; & x \\
\mid \; & \overline{a}\langle P \rangle \\
\mid \; & P \parallel P \\
\mid \; & \mathbf{0} \\
\mid \; & \nu a.P
\end{aligned}
$$

De HOπ à HOcore





$$
\begin{aligned}
P ::= \; & \\
\mid \; & a(x).P \\
\mid \; & x \\
\mid \; & \overline{a}\langle P \rangle \\
\mid \; & P \parallel P \\
\mid \; & \mathbf{0} \\
\mid \; & \nu a.P
\end{aligned}
$$

$$P ::= a(x).P \;\Big|\; x \;\Big|\; \overline{a}\langle P\rangle \;\Big|\; P \parallel P \;\Big|\; \mathbf{0}$$

$$\frac{}{\overline{a}\langle P\rangle \parallel a(x).Q \longrightarrow [P \,/\, x]Q}$$

$$P ::= a(x).P \;\Big|\; x \;\Big|\; \overline{a}\langle P\rangle \;\Big|\; P \parallel P \;\Big|\; \mathbf{0}$$

$$\frac{}{P \parallel \mathbf{0} \equiv P} \qquad \frac{}{P \parallel Q \equiv Q \parallel P}$$

$$\frac{}{P \parallel (Q \parallel R) \equiv (P \parallel Q) \parallel R}$$

$$\frac{P \equiv Q \qquad Q \longrightarrow Q' \qquad Q' \equiv P'}{P \longrightarrow P'}$$

$$\frac{P \longrightarrow P'}{P \parallel Q \longrightarrow P' \parallel Q}$$

HOcore: un calcul de processus fondamental

L'équivalence contextuelle est décidable

P et Q sont équivalents ($P \sim Q$) si et seulement si:

- pour tout contexte C, $C[P] \sim C[Q]$;
- si $P \rightarrow P'$ alors $\exists Q'.Q \rightarrow Q'$ et $P' \sim Q'$ (et inversement);
- P émet un message sur a, noté $P \downarrow_a$, ssi $Q \downarrow_a$.

C'est un calcul Turing Complet

On encode les machines de Minsky

Ceci n'est pas un paradoxe

On ne peut pas décider si un processus termine, mais on peut décider si deux processus font la même chose.

- Équivalence forte
- Toute communication peut être interceptée

$$\left.\begin{array}{l}\overline{a}\langle \mathbf{0}\rangle \parallel \overline{b}\langle \mathbf{0}\rangle \parallel a(x).(b(y).x) \\ \overline{a}\langle \mathbf{0}\rangle \parallel \overline{b}\langle \mathbf{0}\rangle \parallel a(x).(b(y).y)\end{array}\right. \text{ distingués par } \overline{a}\langle \overline{c}\langle \mathbf{0}\rangle\rangle \parallel \cdot$$

Axiomatisation de l'équivalence

Congruence générée par les règles

$$\frac{P \equiv Q}{P \sim Q} \qquad \frac{}{\prod_1^{k+1} a(x).P \sim a(x).\left(P \parallel \prod_1^{k} a(x).P\right)}$$

λ en

1
H
1.008

8
O
15.999

27
Co
58.933
re

Le problème de la soupe

- Application dans le λ calcul structurelle
- Nombre d'applications non borné

$$(\lambda x.(xx)(xx))@(\lambda x.(xx)(xx)) \rightarrow$$
$$((\lambda x.(xx)(xx))@(\lambda x.(xx)(xx)))((\lambda x.(xx)(xx))@(\lambda x.(xx)(xx))) \rightarrow$$
$$\ldots$$

- Pas de structure dans HOcore, émissions et réceptions liées par leurs noms

$\Longrightarrow$ nombre non borné de redex + un nom différent par redex + nombre de noms borné = problème

Spécifier la stratégie d'évaluation

1. Utiliser une transformation CPS pour spécifier le redex actif
2. Traduire en HOcore

$$\begin{aligned}
[\![\lambda x.M]\!] &= c(k).k \parallel \overline{c}\langle a(x).\,[\![M]\!]\rangle \\
[\![x]\!] &= x \\
[\![MN]\!] &= c(k).\,[\![M]\!] \parallel \overline{c}\langle c(v).v \parallel \overline{a}\langle[\![N]\!]\rangle \parallel \overline{c}\langle k\rangle\rangle
\end{aligned}$$

Apologie de la KAM

$$C ::= M \star \pi$$

$$M ::= x \mid MN \mid \lambda x.M$$

$$\pi ::= M :: \pi \mid []$$

$$MN \star \pi \mapsto M \star N :: \pi \qquad (\textsc{Push})$$

$$\lambda x.M \star N :: \pi \mapsto [N / x]M \star \pi \qquad (\textsc{Grab})$$

KAM = pile + substitution

$$MN \star \pi \mapsto M \star N :: \pi \qquad (\textsc{Push})$$

$$\lambda x.M \star N :: \pi \mapsto [N \,/\, x]M \star \pi \qquad (\textsc{Grab})$$

Encoder la pile

$[]$	processus arbitraire $\overline{b}\langle \mathbf{0} \rangle$
tete :: *queue*	un message transportant la tête, un autre la queue
$1 :: 2 :: 3 :: []$	$\overline{a}\langle 1 \rangle \parallel \overline{c}\langle \overline{a}\langle 2 \rangle \parallel \overline{c}\langle \overline{a}\langle 3 \rangle \parallel \overline{c}\langle \overline{b}\langle \mathbf{0} \rangle \rangle \rangle \rangle$

La KAM avec un processus d'ordre supérieur

$$
\begin{aligned}
[\![[]]\!] &\triangleq \overline{b}\langle \mathbf{0} \rangle \\
[\![M :: \pi]\!] &\triangleq \overline{a}\langle [\![M]\!] \rangle \parallel \overline{c}\langle [\![\pi]\!] \rangle \\
[\![MN]\!] &\triangleq c(s).([\![M]\!] \parallel \overline{c}\langle \overline{a}\langle [\![N]\!] \rangle \parallel \overline{c}\langle s \rangle \rangle) \\
[\![\lambda x.M]\!] &\triangleq c(s).(a(x).\,[\![M]\!]) \parallel s \\
[\![x]\!] &\triangleq x \\
[\![M \star \pi]\!] &\triangleq [\![M]\!] \parallel \overline{c}\langle [\![\pi]\!] \rangle
\end{aligned}
$$

$$
\begin{aligned}
MN \star \pi &\mapsto M \star N :: \pi \\
\lambda x.M \star N :: \pi &\mapsto [N \,/\, x]M \star \pi
\end{aligned}
$$

Pour toute KAM encodée, le contrôle est offert

$$\mathsf{cc} \star M :: \pi \mapsto M \star k_\pi :: \pi \qquad (\textsc{CallCC})$$

$$k_\pi \star M :: \pi' \mapsto M \star \pi \qquad (\textsc{Restore})$$

$$K(P) \triangleq c(s_0).(s_0 \parallel a(u).c(\_).(u \parallel \overline{c}\langle P \rangle))$$

$$[\![\mathsf{cc}]\!] \triangleq c(s_0).(s_0 \parallel c(s).a(u).(u \parallel \overline{c}\langle \overline{a}\langle K(s) \rangle \parallel \overline{c}\langle s \rangle \rangle))$$

$$[\![k_\pi]\!] \triangleq K([\![\pi]\!])$$

Couvrez ce canal que je ne saurais voir

L'équivalence contextuelle avec canaux $\mathcal{N}$ privés

$(P \sim_{\mathcal{N}} Q)$ si et seulement si:

- pour tout contexte C n'utilisant pas $\mathcal{N}$, $C[P] \sim_{\mathcal{N}} C[Q]$;
- si $P \to P'$, alors $\exists Q'.Q \to Q'$ et $P' \sim_{\mathcal{N}} Q'$ (et inversement);
- $\forall a \notin \mathcal{N}$, $P \downarrow_a$ ssi $Q \downarrow_a$.

Théorème

L'équivalence contextuelle avec deux canaux privés est indécidable

Preuve

Pour tout λ terme M, on a
$[\![M]\!] \sim_{\{a,c\}} \overline{a}\langle a(x).(\overline{a}\langle x\rangle \parallel x)\rangle \parallel a(x).(\overline{a}\langle x\rangle \parallel x)$ ssi M diverge.

Et avec un seul nom ?

Émission synchrone

$$\overline{a}\langle P\rangle.Q \parallel a(x).R \longrightarrow Q \parallel [P\,/\,x]R$$

Encodage

$$\begin{aligned}
[\![[]]\!] &\triangleq \overline{b}\langle \mathbf{0}\rangle \\
[\![M :: \pi]\!] &\triangleq \overline{a}\langle [\![M]\!]\rangle.\overline{a}\langle [\![\pi]\!]\rangle \\
[\![MN]\!] &\triangleq a(s).([\![M]\!] \parallel \overline{a}\langle \overline{a}\langle [\![N]\!]\rangle.\overline{a}\langle s\rangle\rangle) \\
[\![\lambda x.M]\!] &\triangleq a(s).(a(x).\,[\![M]\!] \parallel s) \\
[\![x]\!] &\triangleq x \\
[\![M \star \pi]\!] &\triangleq [\![M]\!] \parallel \overline{a}\langle [\![\pi]\!]\rangle
\end{aligned}$$

Vers une Abstraction Intégrale

Plus de contrôle, et au delà

- Encodage d'opérateurs de contrôle délimités
  - KAM avec *shift* et *reset*
  - utilisation d'une pile supplémentaire

$$
\begin{aligned}
(t_1 t_2, E, F) &\rightarrow (t_1, t_2 :: E, F)\\
(\lambda x.t_1, t_2 :: E, F) &\rightarrow ([t_2 \,/\, x]t_1, E, F)\\
(\lambda x.t_1, [], E :: F) &\rightarrow (\lambda x.t1, E, F)\\
(\mathtt{shift}, t :: E, F) &\rightarrow (t, k_E :: [], F)\\
(k_E, t :: E', F) &\rightarrow (t, E, E' :: F)\\
(< t >, E, F) &\rightarrow (t, [], E :: F)
\end{aligned}
$$

Abstraction intégrale

$$M \sim_\lambda N \iff [\![M]\!] \sim [\![N]\!]$$

Abstraction intégrale

$$M \sim_\lambda N \iff [\![M]\!] \sim [\![N]\!]$$

Équivalence de λ termes

$M \sim_\lambda N$ ssi

$$M \star [] \Uparrow \iff N \star [] \Uparrow$$

$$\left.\begin{array}{r} M \star [] \rightarrow^* \lambda x.M' \star [] \\ N \star [] \rightarrow^* \lambda x.N' \star [] \end{array}\right\} \implies \forall L, \lambda x.M' \star L :: [] \sim_\lambda \lambda x.N' \star L :: []$$

Abstraction intégrale

$$M \sim_\lambda N \iff [\![M]\!] \sim [\![N]\!]$$

Équivalence de λ termes

$M \sim_\lambda N$ ssi

$$M \star [] \Uparrow \iff N \star [] \Uparrow$$

$$\left.\begin{array}{r} M \star [] \to^* \lambda x.M' \star [] \\ N \star [] \to^* \lambda x.N' \star [] \end{array}\right\} \implies \forall L, \lambda x.M' \star L :: [] \sim_\lambda \lambda x.N' \star L :: []$$

Équivalence contextuelle faible

$(P \approx_{\mathcal{N}} Q)$ si et seulement si:

- pour tout contexte C n'utilisant pas $\mathcal{N}$, $C[P] \approx_{\mathcal{N}} C[Q]$;
- si $P \to^* P'$, alors $\exists Q'.Q \to^* Q'$ et $P' \approx_{\mathcal{N}} Q'$ (et inversement);
- $\forall a \notin \mathcal{N}$, $P \to^* \downarrow_a$ ssi $Q \to^* \downarrow_a$.

Si on ne cache rien

$$\begin{aligned}
[\![[]]\!] &\triangleq b(x).x \\
[\![M :: \pi]\!] &\triangleq \overline{a}\langle [\![M]\!] \rangle \parallel \overline{c}\langle [\![\pi]\!] \rangle \\
[\![MN]\!] &\triangleq c(s).([\![M]\!] \parallel \overline{c}\langle \overline{a}\langle [\![N]\!] \rangle \parallel \overline{c}\langle s \rangle \rangle) \\
[\![\lambda x.M]\!] &\triangleq c(s).(a(x).\,[\![M]\!]) \parallel s \\
[\![x]\!] &\triangleq x \\
[\![M \star \pi]\!] &\triangleq [\![M]\!] \parallel \overline{c}\langle [\![\pi]\!] \rangle
\end{aligned}$$

Conjecture 1

$[\![M]\!] \approx [\![N]\!] \implies M \sim_\lambda N$

Idée

Relancer le processus avec le contexte $\cdot \parallel \overline{b}\langle [\![L :: []]\!] \rangle$.

Si on cache a et c

On ne peut plus relancer avec le contexte $\cdot \parallel \overline{b}\langle [\![L :: []]\!] \rangle$.

Rappel

$(P \approx_{\mathcal{N}} Q)$ si et seulement si:

- pour tout contexte C n'utilisant pas $\mathcal{N}$, $C[P] \approx_{\mathcal{N}} C[Q]$;
- si $P \rightarrow^* P'$, alors $\exists Q'.Q \rightarrow^* Q'$ et $P' \approx_{\mathcal{N}} Q'$ (et inversement);
- $\forall a \notin \mathcal{N}$, $P \rightarrow^* \downarrow_a$ ssi $Q \rightarrow^* \downarrow_a$.

Alternative

Faire traduire le λ-terme par notre processus: $\cdot \parallel \overline{b}\langle "L" \rangle$

$$[\![[]]\!] \triangleq b(x).\mathtt{Trad} \parallel x$$

Si on cache a et c

On ne peut plus relancer avec le contexte $\cdot \parallel \overline{b}\langle [\![L :: []]\!] \rangle$.

Rappel

$(P \approx_{\mathcal{N}} Q)$ si et seulement si:

- pour tout contexte C n'utilisant pas $\mathcal{N}$, $C[P] \approx_{\mathcal{N}} C[Q]$;
- si $P \rightarrow^* P'$, alors $\exists Q'. Q \rightarrow^* Q'$ et $P' \approx_{\mathcal{N}} Q'$ (et inversement);
- $\forall a \notin \mathcal{N}$, $P \rightarrow^* \downarrow_a$ ssi $Q \rightarrow^* \downarrow_a$.

Alternative

Faire traduire le λ-terme par notre processus: $\cdot \parallel \overline{b}\langle "L" \rangle$

$$[\![[]]\!] \triangleq b(x).\mathtt{Trad} \parallel x$$

et les lieurs?

Ma KAM

This is my rifle. There are many others like it, but this one is mine.

$$C ::= (M, \pi) \star \pi \qquad E ::= (M, E) :: E \;\Big|\; []$$

$$M ::= k \;\Big|\; MN \;\Big|\; \lambda.M \qquad \pi ::= (M, E) :: \pi \;\Big|\; []$$

$$\begin{aligned}
(MN, E) \star \pi &\mapsto (M, E) \star (N, E) :: \pi \\
(\lambda.M, E) \star (N, E') :: \pi &\mapsto (M, (N, E') :: E) \star \pi \\
(0, (N, E') :: E) \star \pi &\mapsto (N, E') \star \pi \\
(k+1, (N, E') :: E) \star \pi &\mapsto (k, E) \star \pi
\end{aligned}$$

Merci Olivier !

Beauty is in the eye of the beholder

- La beauté des choses simples

Foundational Calculi for Programming Languages

[To appear in the *CRC Handbook of Computer Science and Engineering*]

Benjamin C. Pierce*

December 22, 1995

- HOcore permet d'élégants encodages
- Vers des sémantiques non entrelacées

$$\frac{P \to P' \qquad Q \to Q'}{P \parallel Q \to P' \parallel Q'}$$

Conjecture $P \sim Q \iff P \equiv Q$