

Propriétés du $\lambda\sigma$ -calcul

Questions

- ❖ Le $\lambda\sigma$ -calcul implémente-t-il correctement le λ -calcul?
- ❖ Le $\lambda\sigma$ -calcul est-il confluent? (Faciliterait les implémentations...)
- ❖ Propriétés de normalisation forte / faible?
- ❖ Qu'en est-il du sous-calcul σ ? Confluent? Terminant?

Le sous-calcul σ des substitutions

(β) $(\lambda M)N \rightarrow M[N \cdot \text{id}]$

(σ)

. $1[N \cdot S] \rightarrow N$

$\uparrow \circ (N \cdot S) \rightarrow S$

. $(MN)[S] \rightarrow M[S](N[S])$

$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$

. $(\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$

. $M[S][S'] \rightarrow M[S \circ S']$

$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$

.

$\text{id} \circ S \rightarrow S$

. $M[\text{id}] \rightarrow M$

$S \circ \text{id} \rightarrow S$

.

- ❖ **Prop.** σ est localement confluent.
- ❖ On a tout fait pour!
- ❖ Preuve: il y a des programmes implémentant Knuth-Bendix pour vérifier ça... (ne le faites pas à la main!)

Le sous-calcul σ des substitutions

- ❖ σ est localement confluent.
- ❖ Nous allons montrer que σ **termine**.
- ❖ Ceci nécessitera quelques efforts...
- ❖ Une fois ceci fait, le lemme de Newman nous dira que tout $\lambda\sigma$ -terme a une **σ -forme normale unique**.

(β) $(\lambda M)N \rightarrow M[N \cdot \text{id}]$

(σ)

. $1[N \cdot S] \rightarrow N$

$\uparrow \circ (N \cdot S) \rightarrow S$

. $(MN)[S] \rightarrow M[S](N[S])$

$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$

. $(\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$

. $M[S][S'] \rightarrow M[S \circ S']$

$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$

.

$\text{id} \circ S \rightarrow S$

. $M[\text{id}] \rightarrow M$

$S \circ \text{id} \rightarrow S$

.

Pour ça, je vais vous parler d'une technique de preuve de terminaison très pratique:
les **recursive path orderings (rpo)**;
ici le **lexicographic path ordering (lpo)**.

Terminaison des systèmes de réécriture

- ❖ Systèmes de règles $\ell \rightarrow r$
entre expressions de la logique du premier ordre
- ❖ Comme σ
- ❖ Comme $\rightarrow_{\mathbb{N}}$
- ❖ Comme encore
bien d'autres

(β) $(\lambda M)N \rightarrow M[N \cdot \text{id}]$

(σ)

. $1[N \cdot S] \rightarrow N$

$\uparrow \circ (N \cdot S) \rightarrow S$

. $(MN)[S] \rightarrow M[S](N[S])$

$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$

. $(\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$

. $M[S][S'] \rightarrow M[S \circ S']$

$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$

.

$\text{id} \circ S \rightarrow S$

$$D_x(x) \rightarrow 1$$

$$D_x(a) \rightarrow 0$$

$$D_x(M + N) \rightarrow D_x(M) + D_x(N)$$

$$D_x(M \times N) \rightarrow D_x(M) \times N + M \times D_x(N)$$

$$D_x(M - N) \rightarrow D_x(M) - D_x(N)$$

$$D_x(-M) \rightarrow -D_x(M)$$

$$D_x(M/N) \rightarrow \left(D_x(M) \times N - M \times D_x(N) \right) / N^2$$

$$D_x(\log(M)) \rightarrow D_x(M)/M$$

$$\begin{array}{l} \perp \\ s \approx t \\ s \\ S(s+t) \\ 0 \\ s*t+s \end{array}$$

La technique des ordres bien fondés

- ❖ Trouver un ordre strict **bien fondé** $>$ sur les expressions tel que:
 - pour toute règle $\ell \rightarrow r$, $\ell > r$
 - passage au contexte:
si $u > v$ alors $f(\dots, u, \dots) > f(\dots, v, \dots)$
- ❖ Si $u \rightarrow v$, alors $u > v$.
- ❖ Donc pas de réduction infinie:
si $u = u_0 \rightarrow u_1 \rightarrow \dots \rightarrow u_n \rightarrow^\infty \dots$ alors $u = u_0 > u_1 > \dots > u_n > \dots$
- ❖ Pour $\rightarrow_{\mathbb{N}}$, on avait défini $e > e'$ ssi $[e] > [e']$

= sans chaîne infinie décroissante

$$u_0 > u_1 > \dots > u_n > \dots$$

$(0 \approx S)$	$0 \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$\perp$
$(S \approx S)$	$S(s) \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
$(+0)$	$s+0$	$\rightarrow_{\mathbb{N}}$	s
$(+S)$	$s+S(t)$	$\rightarrow_{\mathbb{N}}$	$S(s+t)$
$(*0)$	$s*0$	$\rightarrow_{\mathbb{N}}$	0
$(*S)$	$s*S(t)$	$\rightarrow_{\mathbb{N}}$	$s*t+s$

Ordre lexicographique

- ❖ Soit $>, >'$ deux ordres stricts.

On pose $(u, u') (> \times >')_{\text{lex}} (v, v')$ ssi

— $u > v$ ou

— $u = v$ et $u' > v'$

- ❖ **Prop.** Si $>$ et $>'$ bien fondés, alors $(> \times >')_{\text{lex}}$ aussi.

- ❖ *Preuve* (esquisse). Si chaîne infinie décroissante, alors à partir d'un certain rang: les u sont égaux;
puis il n'y plus qu'un nb. fini d'étapes où v décroît. $\square$

Extension lexicographique

- ❖ Soit $>$ ordre strict.

On pose $(u_1, \dots, u_m) >_{\text{lex}} (v_1, \dots, v_n)$ ssi

— $m > n$ ou

— $m = n$ et $u_1 = v_1$ et ... et $u_{i-1} = v_{i-1}$ et $u_i > v_i$

pour un certain $i, 1 \leq i \leq n$

- ❖ **Prop.** Si $>$ bien fondé, alors $>_{\text{lex}}$ aussi.

- ❖ *Preuve* (esquisse). Si chaîne infinie décroissante, alors à partir d'un certain rang: les m sont égaux;
puis les u_1 ; ...; puis les u_m . $\square$

Le lpo

- ❖ On se fixe un ordre strict $>$ (la **précédence**)

- ❖ On définit $>^{\text{lpo}}$ sur les expressions closes par les règles:

$$(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n))$$

$$(\exists i) s_i \geq^{\text{lpo}} t$$

$$s >^{\text{lpo}} t$$

$[\geq^{\text{lpo}}$ abrège ($>^{\text{lpo}}$ ou $=$)]

Permet notamment de conclure

$$f(s_1, \dots, s_m) >^{\text{lpo}} s_i$$

$$(\forall j) s >^{\text{lpo}} t_j$$

$$s >^{\text{lpo}} t$$

$[\gg \text{ abrège } (> \times (>^{\text{lpo}})_{\text{lex}})_{\text{lex}}]$

C'est-à-dire:

$f > g$ ou bien

$(f = g \text{ et: } m > n \text{ ou}$

$m = n \text{ et: } s_1 >^{\text{lpo}} t_1 \text{ ou}$

$s_1 = t_1 \text{ et: } s_2 >^{\text{lpo}} t_2 \text{ ou}$

etc.)

Ça, c'est pour éviter que les t_j ne grossissent trop. Par ex., si $f > g$:

$$s = f(s_1) \gg g(s)$$

mais $s \rightarrow g(s) \rightarrow g(g(s)) \rightarrow \dots$

réduction infinie

$$\begin{aligned} &= f(s_1, \dots, s_m) \\ &= g(t_1, \dots, t_n) \end{aligned}$$

Propriétés élémentaires du lpo

- ❖ $>^{\text{lpo}}$ est un ordre strict
(et même total si $>$ est total)

$$\frac{(\exists i) s_i >^{\text{lpo}} t}{s >^{\text{lpo}} t} \quad \frac{(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n)) \quad (\forall j) s >^{\text{lpo}} t_j}{s >^{\text{lpo}} t}$$

- ❖ ... qui passe au contexte
- ❖ ... qui a la propriété de sous-terme:
si t est un **sous-terme strict** de s (en notation $s \triangleright t$),
alors $s >^{\text{lpo}} t$
- ❖ ... et bien fondé si $>$ est bien fondé
(preuve: transparents suivants, après l'exemple)

Lpo: exemple 1

❖ Comment montrer que $\rightarrow_{\mathbb{N}}$ termine sans se fatiguer

$(0 \approx S)$	$0 \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$\perp$
$(S \approx S)$	$S(s) \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
$(+0)$	$s+0$	$\rightarrow_{\mathbb{N}}$	s
$(+S)$	$s+S(t)$	$\rightarrow_{\mathbb{N}}$	$S(s+t)$
$(*0)$	$s*0$	$\rightarrow_{\mathbb{N}}$	0
$(*S)$	$s*S(t)$	$\rightarrow_{\mathbb{N}}$	$s*t+s$

❖ Précédence: $\approx > \perp \quad * > + > S$

$(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n))$	
$(\exists i) \ s_i \geq_{\text{lpo}} t$	$(\forall j) \ s >_{\text{lpo}} t_j$
$s >_{\text{lpo}} t$	$s >_{\text{lpo}} t$

$$\frac{?}{s+S(t) >_{\text{lpo}} S(s+t)}$$

Lpo: exemple 1

❖ Comment montrer que $\rightarrow_{\mathbb{N}}$ termine sans se fatiguer

❖ Précédence: $\approx > \perp \quad * > + > S$

$(0 \approx S)$	$0 \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$\perp$
$(S \approx S)$	$S(s) \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
$(+0)$	$s+0$	$\rightarrow_{\mathbb{N}}$	s
$(+S)$	$s+S(t)$	$\rightarrow_{\mathbb{N}}$	$S(s+t)$
$(*0)$	$s*0$	$\rightarrow_{\mathbb{N}}$	0
$(*S)$	$s*S(t)$	$\rightarrow_{\mathbb{N}}$	$s*t+s$

$(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n))$	
$(\exists i) s_i \geq_{\text{lpo}} t$	$(\forall j) s >_{\text{lpo}} t_j$
$s >_{\text{lpo}} t$	$s >_{\text{lpo}} t$

?
$s+S(t) >_{\text{lpo}} s+t$
$s+S(t) >_{\text{lpo}} S(s+t)$

Règle de droite (car $+ > S$)

Lpo: exemple 1

- ❖ Comment montrer que $\rightarrow_{\mathbb{N}}$ termine sans se fatiguer
- ❖ Précédence: $\approx > \perp \quad * > + > S$

$(0 \approx S)$	$0 \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$\perp$
$(S \approx S)$	$S(s) \approx S(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
$(+0)$	$s+0$	$\rightarrow_{\mathbb{N}}$	s
$(+S)$	$s+S(t)$	$\rightarrow_{\mathbb{N}}$	$S(s+t)$
$(*0)$	$s*0$	$\rightarrow_{\mathbb{N}}$	0
$(*S)$	$s*S(t)$	$\rightarrow_{\mathbb{N}}$	$s*t+s$

$(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n))$	
$(\exists i) s_i \geq_{\text{lpo}} t$	$(\forall j) s >_{\text{lpo}} t_j$
$s >_{\text{lpo}} t$	$s >_{\text{lpo}} t$

$$\begin{array}{c}
 \frac{}{S(t) >_{\text{lpo}} t} \\
 \frac{s+S(t) >_{\text{lpo}} s \quad s+S(t) >_{\text{lpo}} t}{s+S(t) >_{\text{lpo}} s+t} \\
 \frac{s+S(t) >_{\text{lpo}} s+t}{s+S(t) >_{\text{lpo}} S(s+t)}
 \end{array}$$

Sous-terme! (ou bien règle de gauche)

Règle de droite (ici $+=+$, $s=s$ et on vérifie $S(t) >_{\text{lpo}} t$)

Règle de droite (car $+ > S$)

Je vous laisse vérifier les autres règles (exercice!)

Lpo: exemple 2

❖ Plus compliqué?

❖ Pas du tout,
c'est totalement
mécanique

❖ Précédence:

$D_x > 1, 0, +, \times, -, /, ^$

$$\begin{aligned}
 D_x(x) &\rightarrow 1 \\
 D_x(a) &\rightarrow 0 \\
 D_x(M + N) &\rightarrow D_x(M) + D_x(N) \\
 D_x(M \times N) &\rightarrow D_x(M) \times N + M \times D_x(N) \\
 D_x(M - N) &\rightarrow D_x(M) - D_x(N) \\
 D_x(-M) &\rightarrow -D_x(M) \\
 D_x(M/N) &\rightarrow (D_x(M) \times N - M \times D_x(N)) / N^2 \\
 D_x(\log(M)) &\rightarrow D_x(M) / M
 \end{aligned}$$

$$\frac{(\exists i) s_i \geq_{\text{lpo}} t}{s \geq_{\text{lpo}} t} \quad \frac{(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n)) \quad (\forall j) s \geq_{\text{lpo}} t_j}{s \geq_{\text{lpo}} t}$$

Le lpo est bien fondé

Soit: Acc la partie bien fondée de $>^{\text{lpo}}$
 $= \{s \mid \text{pas de chaîne infinie décroissante pour } >^{\text{lpo}} \text{ partant de } s\}$

$\text{Acc} = \{f(s_1, \dots, s_m) \mid \text{tout } s_i \text{ est dans Acc}\}$

❖ Supposons $\text{Acc} - \text{Acc}$ non vide

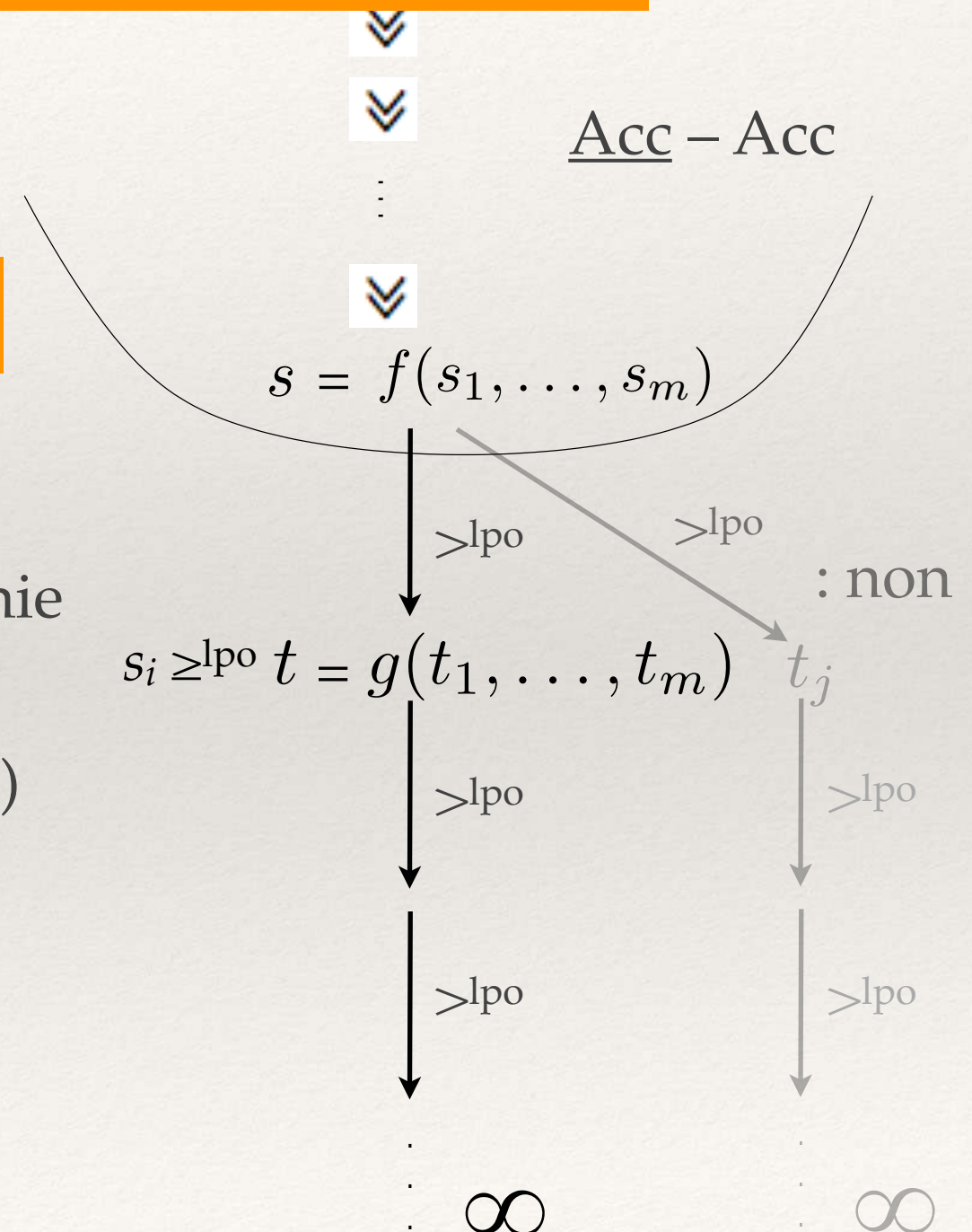
- ❖ $\gg$ est bien fondé sur $\text{Acc} - \text{Acc}$
 (car produit lexico.): soit s $\gg$ -minimal dedans
- ❖ s n'est pas dans Acc : démarre une chaîne infinie
- ❖ parmi ces chaînes, on en choisit une où le 2ème élément t est $\triangleright$ -minimal ($\triangleright$ =sous-terme)

• Cas 1

$$\frac{(\exists i) s_i \geq^{\text{lpo}} t}{s >^{\text{lpo}} t}$$

comme $s \in \text{Acc}$, $s_i \in \text{Acc}$: impossible.

❖ Supposons $>$ bien fondé



Le lpo est bien fondé

Soit: Acc la partie bien fondée de $>_{\text{lpo}}$
 $= \{s \mid \text{pas de chaîne infinie décroissante pour } >_{\text{lpo}} \text{ partant de } s\}$

$\text{Acc} = \{f(s_1, \dots, s_m) \mid \text{tout } s_i \text{ est dans Acc}\}$

❖ Supposons $\text{Acc} - \text{Acc}$ non vide

- ❖ $\gg$ est bien fondé sur $\text{Acc} - \text{Acc}$
 (car produit lexico.): soit s $\gg$ -minimal dedans
- ❖ s n'est pas dans Acc : démarre une chaîne infinie
- ❖ parmi ces chaînes, on en choisit une où le 2ème élément t est $\triangleright$ -minimal ($\triangleright = \text{sous}$)

• Cas 2:

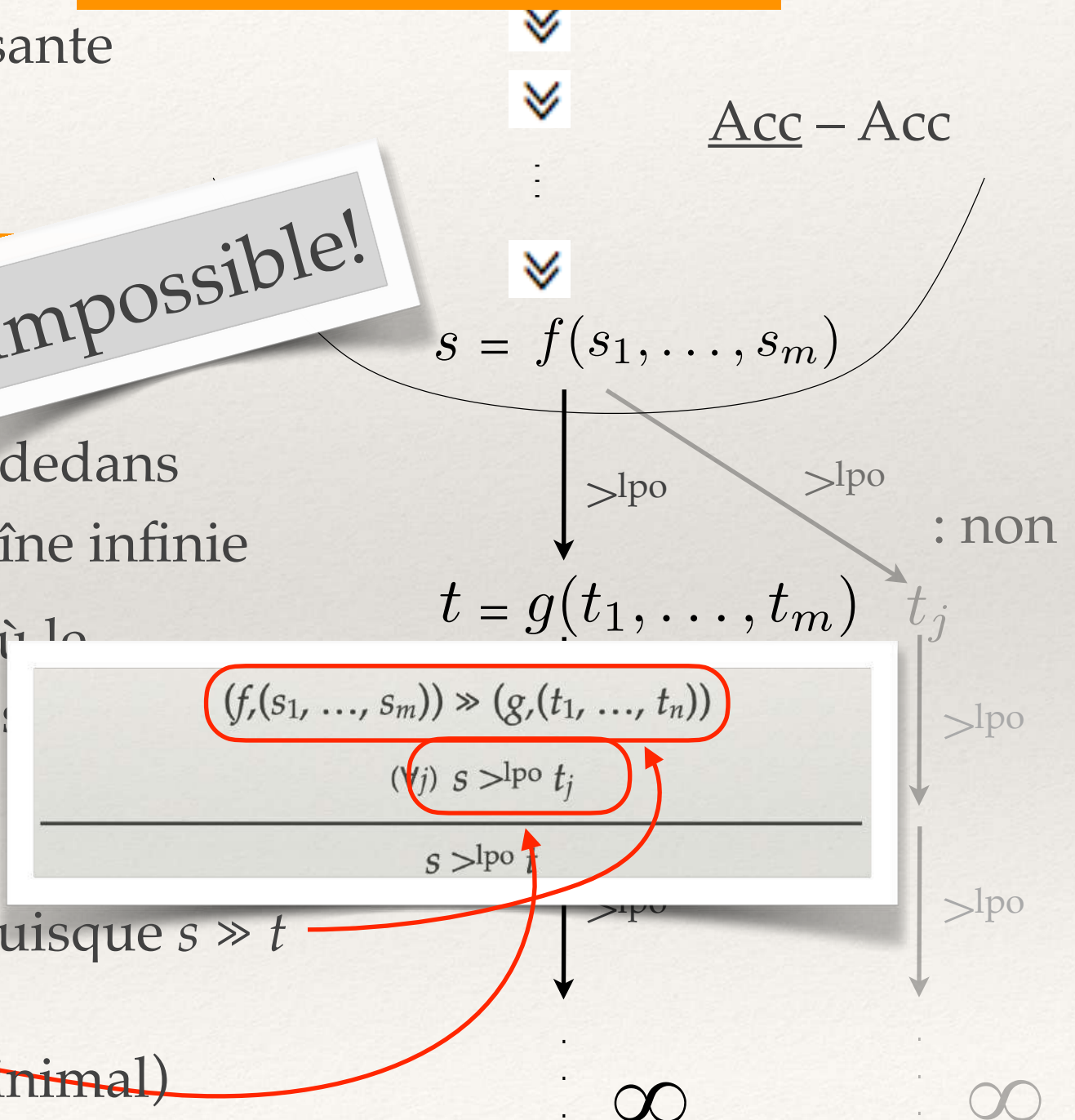
$t \notin \text{Acc}$, sinon t serait dans $\text{Acc} - \text{Acc}$,
 ... contredisant la $\gg$ -minimalité de s , puisque $s \gg t$

Donc il existe un $t_j \notin \text{Acc}$

Mais $s >_{\text{lpo}} t_j$: on a dit non! (t pas $\triangleright$ -minimal)

❖ Supposons $>$ bien fondé

Impossible!



Le lpo est bien fondé

Soit: Acc la partie bien fondée de $>^{\text{lpo}}$
 $= \{s \mid \text{pas de chaîne infinie décroissante}$
 $\text{pour } >^{\text{lpo}} \text{ partant de } s\}$

$\underline{\text{Acc}} = \{f(s_1, \dots, s_m) \mid \text{tout } s_i \text{ est dans } \text{Acc}\}$

❖ Supposons $>$ bien fondé

❖ Donc $\underline{\text{Acc}}$ – Acc est vide.

❖ Autrement dit: pour tout $s = f(s_1, \dots, s_m)$,
si tous les s_i sont dans Acc (i.e., si $s \in \underline{\text{Acc}}$)
alors s est dans Acc .

❖ Donc, par récurrence sur (la taille de) s :

❖ **Thm.** Toute expression s est dans Acc .

❖ I.e., $>^{\text{lpo}}$ est bien fondé (si $>$ est bien fondé).

Et pour σ ?

❖ Ça ne fonctionne pas

❖ Quelle que soit la
précédence choisie,

vous aurez
des problèmes

$(\lambda M)[S] \rightarrow$



<https://www.win.tue.nl/~hzantema/hznaw.jpg>

$(\beta) (\lambda M)N \rightarrow M[N \cdot \text{id}]$

(σ)

$\cdot 1[N \cdot S] \rightarrow N$

$\uparrow \circ (N \cdot S) \rightarrow S$

$\cdot (MN)[S] \rightarrow M[S](N[S])$

$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$

$\cdot (\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$

$\cdot M[S][S'] \rightarrow M[S \circ S']$

$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$

$\cdot$

$\text{id} \circ S \rightarrow S$

$\cdot M[\text{id}] \rightarrow M$

$S \circ \text{id} \rightarrow S$

$\cdot$

$(\exists i) s_i \geq_{\text{lpo}} t$

$(f, (s_1, \dots, s_m)) \gg (g, (t_1, \dots, t_n))$

$s \geq_{\text{lpo}} t$

$(\forall j) s \geq_{\text{lpo}} t_j$

$s \geq_{\text{lpo}} t$

❖ En fait, la terminaison de σ
est un problème **difficile**.

❖ Nous verrons une preuve
due à Hans Zantema.

1/3: simplification

- ❖ On traduit les $\lambda\sigma$ -termes en remplaçant $[]$ par $\circ$, $@$ par $\cdot$.

$$(\beta) (\lambda M)N \rightarrow M[N \cdot \text{id}]$$

(σ)

$$\cdot 1[N \cdot S] \rightarrow N$$

$$\uparrow \circ (N \cdot S) \rightarrow S$$

$$\cdot (MN)[S] \rightarrow M[S](N[S])$$

$$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$$

$$\cdot (\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$$

$$\cdot M[S][S'] \rightarrow M[S \circ S']$$

$$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$$

$$\cdot M[\text{id}] \rightarrow M$$

$$\text{id} \circ S \rightarrow S$$

$$\cdot S \circ \text{id} \rightarrow S$$

$$S \circ \text{id} \rightarrow S$$

(σ')

- ❖ $1 \circ (N \cdot S) \rightarrow N$ $\uparrow \circ (N \cdot S) \rightarrow S$
 $(N \cdot S) \circ S' \rightarrow (N \circ S') \cdot (S \circ S')$
 $(\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow)))$
 $(M \circ S) \circ S' \rightarrow M \circ (S \circ S')$
 $\text{id} \circ S \rightarrow S$ $S \circ \text{id} \rightarrow S$
 $1 \cdot \uparrow \rightarrow \text{id}$ $(1 \circ S) \cdot (\uparrow \circ S) \rightarrow S$

- ❖ **Fait.** Si (σ') termine, alors (σ) termine.

2/3: le pb. $(\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow)))$

- ❖ Le « **semantic labeling** » de Hans Zantema:
- ❖ on va étiqueter chaque opérateur $\circ$ par un entier:
 - qui est la valeur sémantique de quelque chose
 - qui reste invariant (ou décroît) le long des réductions
- ❖ Ici, $\ell(M)$ mesure la profondeur max. de λ sous lesquels S serait poussée dans les σ -réductions partant de $M \circ S$



<https://www.win.tue.nl/~hzantema/hznaw.jpg>

si $M \rightarrow N$ en (σ') ,
alors $\ell(M) \geq \ell(N)$

$$\begin{aligned}\ell(x) &\hat{=} 0 \\ \ell(\uparrow) &\hat{=} 0 \\ \ell(M \cdot N) &\hat{=} \max(\ell(M), \ell(N)) \\ \ell(id) &\hat{=} 0\end{aligned}$$

$$\begin{aligned}\ell(1) &\hat{=} 0 \\ \ell(\lambda M) &\hat{=} \ell(M) + 1 \\ \ell(M \circ N) &\hat{=} \ell(M) + \ell(N)\end{aligned}$$

2/3: le pb. $(\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow)))$

- ❖ On étiquette $\circ$ dans $M \circ S$ par $\ell(M)+\ell(S)$

$$\begin{array}{ll} \ell(x) \hat{=} 0 & \ell(1) \hat{=} 0 \\ \ell(\uparrow) \hat{=} 0 & \ell(\lambda M) \hat{=} \ell(M) + 1 \\ \ell(M \cdot N) \hat{=} \max(\ell(M), \ell(N)) & \\ \ell(id) \hat{=} 0 & \ell(M \circ N) \hat{=} \ell(M) + \ell(N) \end{array}$$

		(σ')
$1 \circ (N \cdot S) \rightarrow N$	$\uparrow \circ (N \cdot S) \rightarrow S$	
$(N \cdot S) \circ S' \rightarrow (N \circ S') \cdot (S \circ S')$		
$(\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow)))$		
$(M \circ S) \circ S' \rightarrow M \circ (S \circ S')$		
$id \circ S \rightarrow S$	$S \circ id \rightarrow S$	
$1 \cdot \uparrow \rightarrow id$	$(1 \circ S) \cdot (\uparrow \circ S) \rightarrow S$	

- ❖ La règle problématique devient:

$$(\lambda M) \circ_{m+1+s} S \rightarrow \lambda(M \circ_{m+s} (1 \cdot (S \circ_s \uparrow)))$$

où $m=\ell(M)$, $s=\ell(S)$

On a maintenant une infinité
d'opérateurs $\circ_k$,
mais si on les ordonne selon k ,

$$\circ_{m+1+s} > \circ_{m+s}, \circ_s !$$

2/3: le pb. $(\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow)))$

- ❖ On étiquette $\circ$ dans $M \circ S$ par $\ell(M)+\ell(S)$

$$\begin{array}{ll}
 \ell(x) \hat{=} 0 & \ell(1) \hat{=} 0 \\
 \ell(\uparrow) \hat{=} 0 & \ell(\lambda M) \hat{=} \ell(M) + 1 \\
 \ell(M \cdot N) \hat{=} \max(\ell(M), \ell(N)) & \\
 \ell(id) \hat{=} 0 & \ell(M \circ N) \hat{=} \ell(M) + \ell(N)
 \end{array}$$

(σ')

$$\begin{array}{ll}
 1 \circ (N \cdot S) \rightarrow N & \uparrow \circ (N \cdot S) \rightarrow S \\
 (N \cdot S) \circ S' \rightarrow (N \circ S') \cdot (S \circ S') & \\
 (\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow))) & \\
 (M \circ S) \circ S' \rightarrow M \circ (S \circ S') & \\
 id \circ S \rightarrow S & S \circ id \rightarrow S \\
 1 \cdot \uparrow \rightarrow id & (1 \circ S) \cdot (\uparrow \circ S) \rightarrow S
 \end{array}$$

(σ'')

- ❖

$$\begin{array}{ll}
 1 \circ_{\max(n,s)} (N \cdot S) \rightarrow N & \uparrow \circ_{\max(n,s)} (N \cdot S) \rightarrow S \\
 (N \cdot S) \circ_{\max(n,s)+s'} S' \rightarrow (N \circ_{n+s'} S') \cdot (S \circ_{s+s'} S') & \\
 (\lambda M) \circ_{m+1+s} S \rightarrow \lambda(M \circ_{m+s} (1 \cdot (S \circ_s \uparrow))) & \\
 (M \circ_{m+s} S) \circ_{m+s+s'} S' \rightarrow M \circ_{m+s+s'} (S \circ_{s+s'} S') & \\
 id \circ_s S \rightarrow S & S \circ_s id \rightarrow S \\
 1 \cdot \uparrow \rightarrow id & (1 \circ_s S) \cdot (\uparrow \circ_s S) \rightarrow S
 \end{array}$$

2/3: le pb. $(\lambda M) \circ S \rightarrow \lambda(M \circ (1 \cdot (S \circ \uparrow)))$

- ❖ **Fait.** Si $M \rightarrow N$ en (σ')
alors la version étiquetée $\llbracket M \rrbracket$ de M
se réécrit en celle, $\llbracket N \rrbracket$, de N en (σ')
- ❖ Donc si (σ'') termine,
alors (σ') , donc aussi (σ) termine.

Faux!

$(1 \circ (N \cdot S)) \circ S' \rightarrow N \circ S'$ mais
 $(1 \circ_{\max(n,s)} (\llbracket N \rrbracket \cdot \llbracket S \rrbracket)) \circ_{\max(n,s)+s'} \llbracket S' \rrbracket$
 $\rightarrow \llbracket N \rrbracket \circ_{n+s'} \llbracket S' \rrbracket$???

$$1 \cdot \uparrow \rightarrow \text{id}$$

$$(1 \circ S) \cdot (\uparrow \circ S) \rightarrow S$$

ajouter: $M \circ_m S \rightarrow M \circ_n S$ si $m > n$

- ❖ $1 \circ_{\max(n,s)} (N \cdot S) \rightarrow N$ $\uparrow \circ_{\max(n,s)} (S \cdot S') \rightarrow S'$
 $(N \cdot S) \circ_{\max(n,s)+s'} S' \rightarrow (N \circ_{n+s'} S')$
- $(\lambda M) \circ_{m+1+s} S \rightarrow \lambda(M \circ_{m+s} (1 \cdot (S \circ_s \uparrow)))$
 $(M \circ_{m+s} S) \circ_{m+s+s'} S' \rightarrow M \circ_{m+s+s'} (S \circ_{s+s'} S')$
- $\text{id} \circ_s S \rightarrow S$ $S \circ_s \text{id} \rightarrow S$
- $1 \cdot \uparrow \rightarrow \text{id}$ $(1 \circ_s S) \cdot (\uparrow \circ_s S) \rightarrow S$

si $M \rightarrow N$ en (σ') ,
alors $\ell(M) \geq \ell(N)$

3/3: le lpo

(σ'')

❖

$$\begin{array}{ll}
 1 \circ_{\max(n,s)} (N \cdot S) \rightarrow N & \uparrow \circ_{\max(n,s)} (N \cdot S) \rightarrow S \\
 (N \cdot S) \circ_{\max(n,s)+s'} S' \rightarrow (N \circ_{n+s'} S') \cdot (S \circ_{s+s'} S') \\
 (\lambda M) \circ_{m+1+s} S \rightarrow \lambda(M \circ_{m+s} (S \circ_s \uparrow)) & \\
 (M \circ_m S) \circ_{m+s+s'} S' \rightarrow M \circ_{m+s+s'} (S \circ_{s+s'} S') \\
 \text{id} \circ_s S \rightarrow S & S \circ_s \text{id} \rightarrow S \\
 1 \cdot \uparrow \rightarrow \text{id} & (1 \circ_s S) \cdot (\uparrow \circ_s S) \rightarrow S \\
 M \circ_m S \rightarrow M \circ_n S \text{ si } m > n
 \end{array}$$

Termine!

- ❖ **Précédence:** $\dots > \circ_m > \dots > \circ_2 > \circ_1 > \circ_0$
 $> \cdot, \lambda, 1, \uparrow$
 $> \text{id}$

Terminaison de σ

- ❖ **Thm.** σ termine.
- ❖ (Ouf!)
- ❖ Preuve: (σ) termine si (σ') termine (version simplifiée)
 (σ') termine si (σ'') termine (semantic labeling)
et (σ'') termine par lpo. $\square$

Propriétés de σ

- ❖ **Thm.** σ termine.
- ❖ Rappel: σ est localement confluente
- ❖ Donc tout terme M a une σ -forme normale unique $\sigma(M)$
- ❖ Quelle est la forme des σ -formes normales?

Formes σ -normales (closes)

- ❖ $M, N, \dots ::= x \mid 1 \mid MN \mid \lambda M \mid M[S]$
 $S, S', \dots ::= \text{id} \mid N \cdot S \mid \uparrow \mid S \circ S'$

(β) $(\lambda M)N \rightarrow M[N \cdot \text{id}]$

(σ)

$\cdot 1[N \cdot S] \rightarrow N$

$\uparrow \circ (N \cdot S) \rightarrow S$

$\cdot (MN)[S] \rightarrow M[S](N[S])$

$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$

$\cdot (\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$

$\cdot M[S][S'] \rightarrow M[S \circ S']$

$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$

$\cdot M[\text{id}] \rightarrow M$

$\text{id} \circ S \rightarrow S$

$\cdot S \circ \text{id} \rightarrow S$

- ❖ Les piles σ -normales:

$$N_1 \cdot N_2 \cdot \dots \cdot N_k \cdot \uparrow^m$$

où $k, m \geq 0$, les N_i sont σ -normaux

et $\uparrow^0 = \text{id}$, $\uparrow^1 = \uparrow$, $\uparrow^{m+1} = \uparrow \circ \uparrow^m$ pour $m \geq 1$

- ❖ Les termes clos σ -normaux:

$$M, N, \dots ::= n \mid MN \mid \lambda M$$

où $n = 1[\uparrow^{n-1}]$ pour $n \geq 2$

- ❖ Ce sont juste les λ -termes en notation de de Bruijn!

Traductions de et vers le $\lambda\sigma$ -calcul

Comme avec de Bruijn

Traduction vers $\lambda\sigma$

- ❖ Paramétrées par une liste $\ell = [x_1; \dots; x_n]$ de variables [suffisamment longue]
(idée: on va remplacer la variable x_i par l'indice de de Bruijn i)

$$\underline{\lambda \rightarrow \lambda\sigma}$$

- ❖ $x^*(\ell) = i$ si $x = x_i$ (avec i minimal)
... et $x[\uparrow^n]$ si pas dans ℓ

$$(uv)^*(\ell) = u^*(\ell) (v^*(\ell))$$

$$(\lambda x . u)^*(\ell) = \lambda(u^*(x::\ell))$$

- ❖ **Prop.** si $u \rightarrow u'$
alors $u^*(\ell) \rightarrow^+ u'^*(\ell)$

Pénible à démontrer!
voir théorème 3 du poly
`machines.pdf`

En fait, $u^*(\ell)$:
— se réduit par une étape de β
en un terme...
— dont $u'^*(\ell)$ est la σ -forme normale

Comme avec de Bruijn
(en plus compliqué)

Traduction depuis $\lambda\sigma$

- ❖ Paramétrées par une liste $\ell = [u_1; \dots; u_n]$ de λ -termes [suffisamment longue]
(idée: on va remplacer i par u_i)

$$\underline{\lambda\sigma \rightarrow \lambda}$$

- ❖ $x^\circ(\ell) = x$
 $i^\circ(\ell) = u_i$
 $(MN)^\circ(\ell) = M^\circ(\ell) (N^\circ(\ell))$
 $(\lambda M)^\circ(\ell) = \lambda x . (M^\circ(x :: \ell))$
- ❖ $\text{id}^\circ(\ell) = \ell$
 $(N \cdot S)^\circ(\ell) = N^\circ(\ell) :: S^\circ(\ell)$
 $\uparrow^\circ([u_1; \dots; u_n]) = [u_2; \dots; u_n]$
 $(S \circ S')^\circ(\ell) = S^\circ(S'^\circ(\ell))$

$$(M[S])^\circ(\ell) = M^\circ(S^\circ(\ell))$$

Où $S^\circ(\ell)$ est
une **liste** de λ -termes

Comme avec de Bruijn
(en plus compliqué)

Traduction depuis $\lambda\sigma$

- ❖ Paramétrées par une liste $\ell = [u_1; \dots; u_n]$ de λ -termes [suffisamment longue]
(idée: on va remplacer i par u_i)

$$\underline{\lambda\sigma \rightarrow \lambda}$$

- ❖ $x^\circ(\ell) = x$
 $i^\circ(\ell) = u_i$
 $(MN)^\circ(\ell) = M^\circ(\ell) (N^\circ(\ell))$
 $(\lambda M)^\circ(\ell) = \lambda x . (M^\circ(x :: \ell))$
- ❖ $\text{id}^\circ(\ell) = \ell$
 $(N \cdot S)^\circ(\ell) = N^\circ(\ell) :: S^\circ(\ell)$
 $\uparrow^\circ([u_1; \dots; u_n]) = [u_2; \dots; u_n]$
 $(S \circ S')^\circ(\ell) = S^\circ(S'^\circ(\ell))$

$$(M[S])^\circ(\ell) = M^\circ(S^\circ(\ell))$$

- ❖ **Prop.** si $M \rightarrow M'$
alors $M^\circ(\ell) \rightarrow^* M'^\circ(\ell)$

Traductions de et vers $\lambda\sigma$

$$\underline{\lambda \rightarrow \lambda\sigma}$$

Prop. si $u \rightarrow u'$
alors $u^* (\ell) \rightarrow^+ u'^* (\ell)$

$$\underline{\lambda\sigma \rightarrow \lambda}$$

❖ **Prop.** si $M \rightarrow M'$
alors $M^\circ (\ell) \rightarrow^* M'^\circ (\ell)$

❖ De plus,

❖ **Prop.** Pour tout λ -terme clos u , $(u^* (\ell))^\circ (\ell) = u$.

❖ **Prop.** Pour tout $\lambda\sigma$ -terme clos M , $M \rightarrow^* (M^\circ (\ell))^* (\ell)$
... en n'utilisant que les règles de (σ)

❖ En fait, $(M^\circ (\ell))^* (\ell)$ est la forme σ -normale de M .

Confluence du $\lambda\sigma$ -calcul

Confluence du $\lambda\sigma$ -calcul

- ❖ Oui, le $\lambda\sigma$ -calcul (clos) est confluent...
- ❖ et c'est très simple!
- ❖ (On va tricher: on va se ramener au λ -calcul.)
- ❖ La tricherie s'appelle le **lemme d'interprétation** de Thérèse Hardin, au passage.

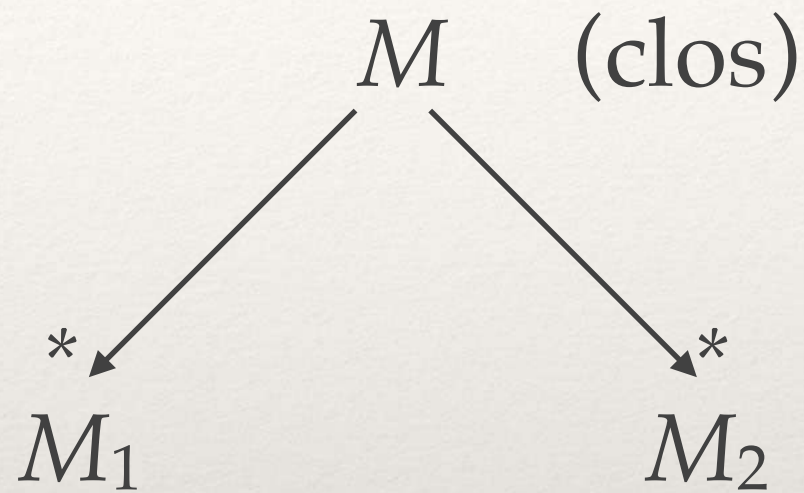


<https://diccan.com/Hebdo/h58/hhardin.JPG>

Confluence du $\lambda\sigma$ -calcul (clos)



<https://diccan.com/Hebdo/h58/hhardin.JPG>



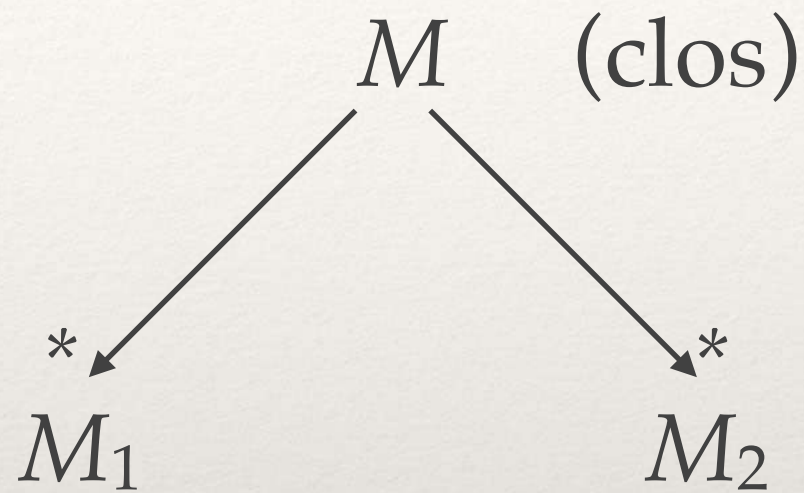
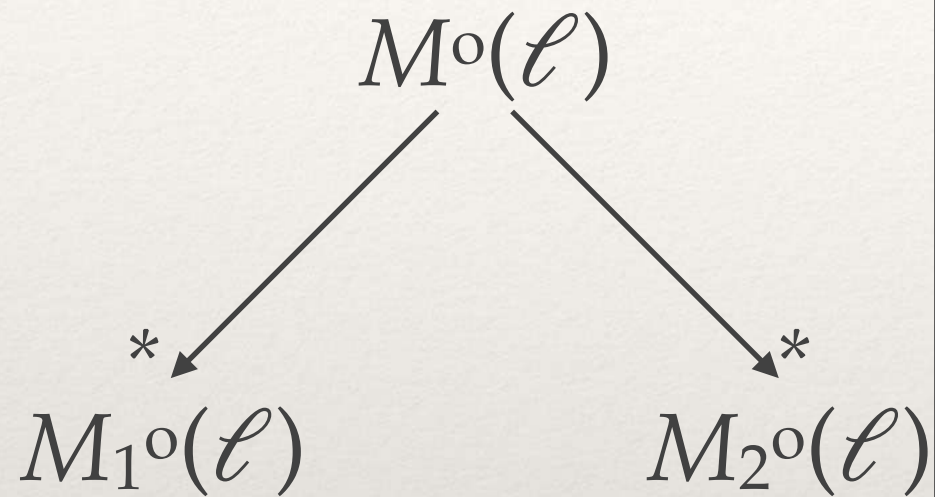
Côté λ -calcul

Côté $\lambda\sigma$ -calcul

Confluence du $\lambda\sigma$ -calcul (clos)



<https://diccan.com/Hebdo/h58/hhardin.JPG>



- ❖ **Prop.** si $M \rightarrow M'$
alors $M^o(\ell) \rightarrow^* M'^o(\ell)$
pour $\ell = [x_1; \dots; x_n]$,
 n assez grand

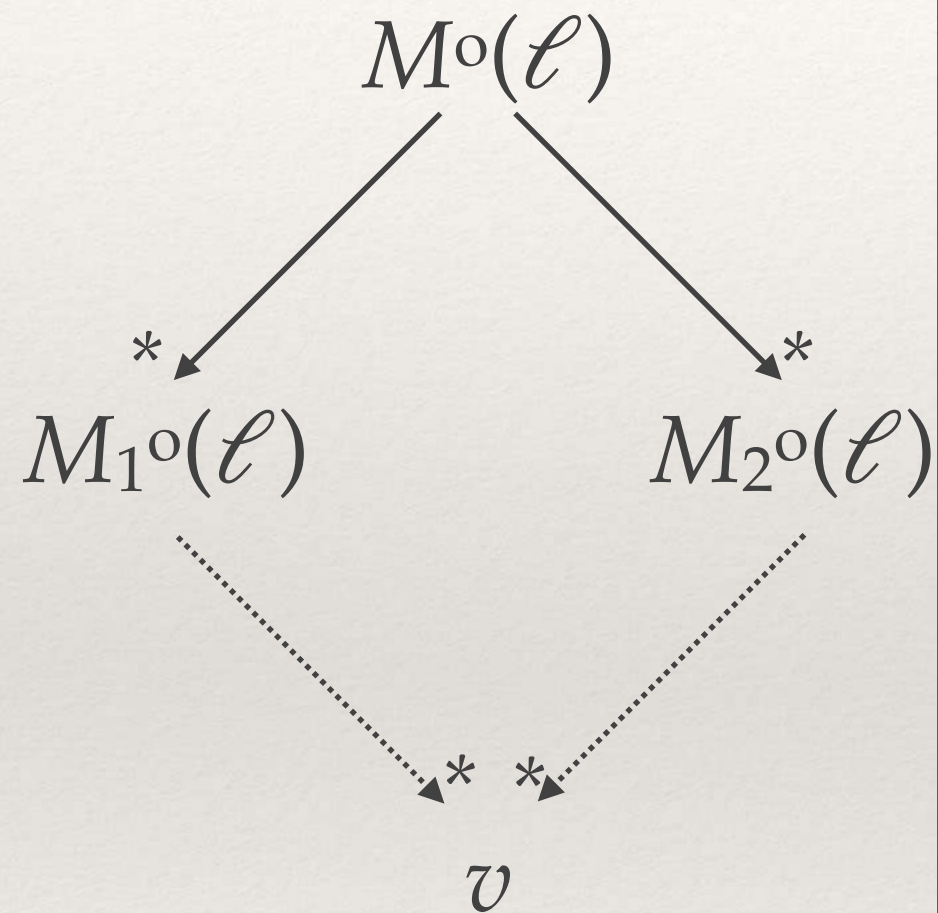
Côté λ -calcul

Côté $\lambda\sigma$ -calcul

Confluence du $\lambda\sigma$ -calcul (clos)

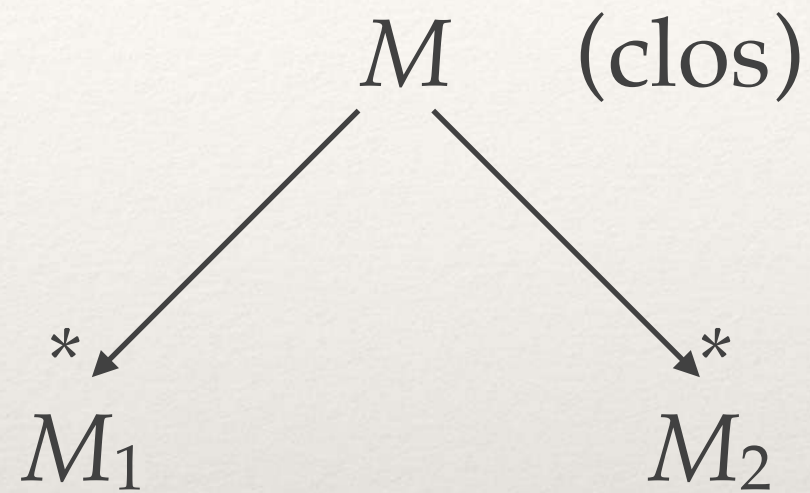


<https://diccan.com/Hebdo/h58/hhardin.JPG>



... car le λ -calcul
est confluent

Côté λ -calcul

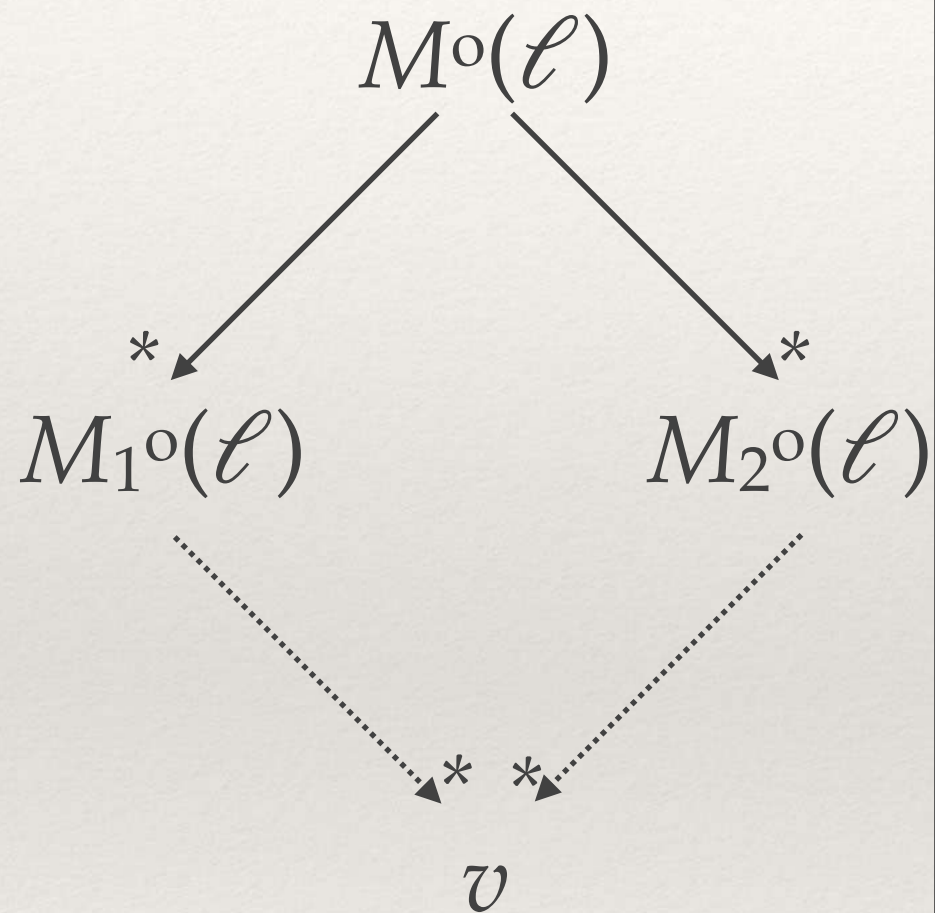


Côté $\lambda\sigma$ -calcul

Confluence du $\lambda\sigma$ -calcul (clos)

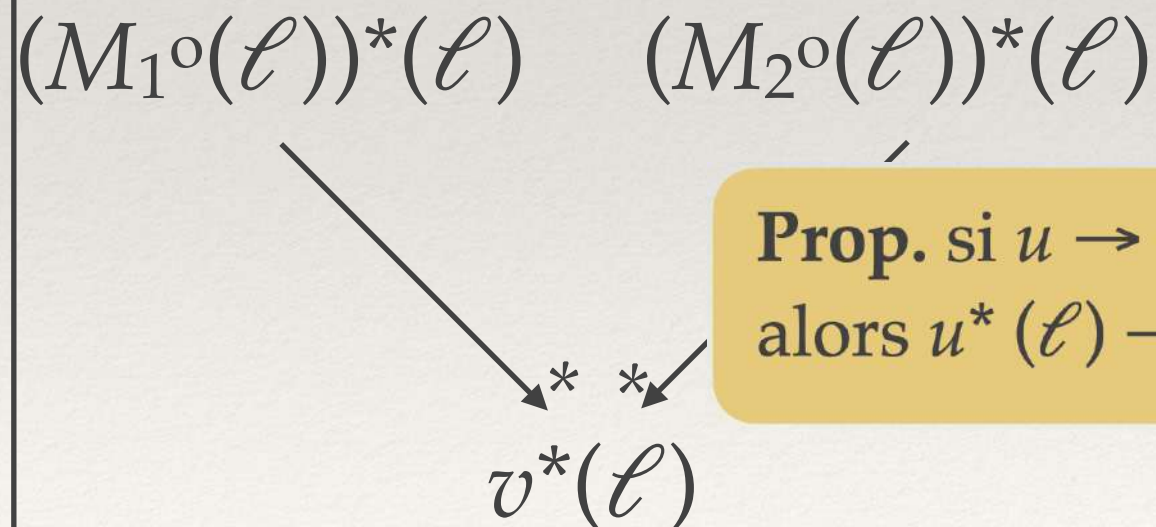
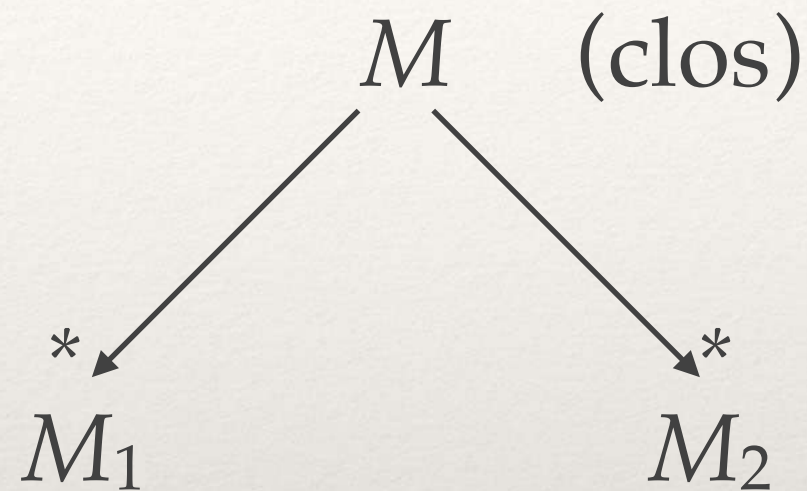


<https://diccan.com/Hebdo/h58/hhardin.JPG>



... car le λ -calcul
est confluent

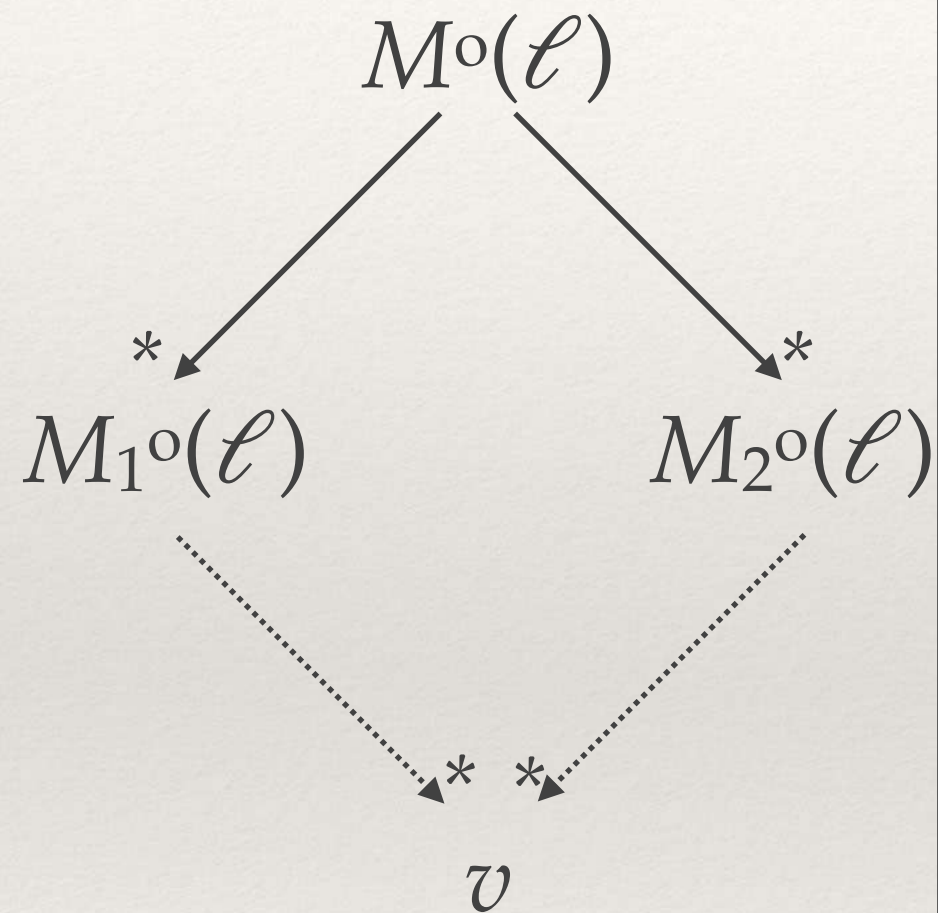
Côté λ -calcul



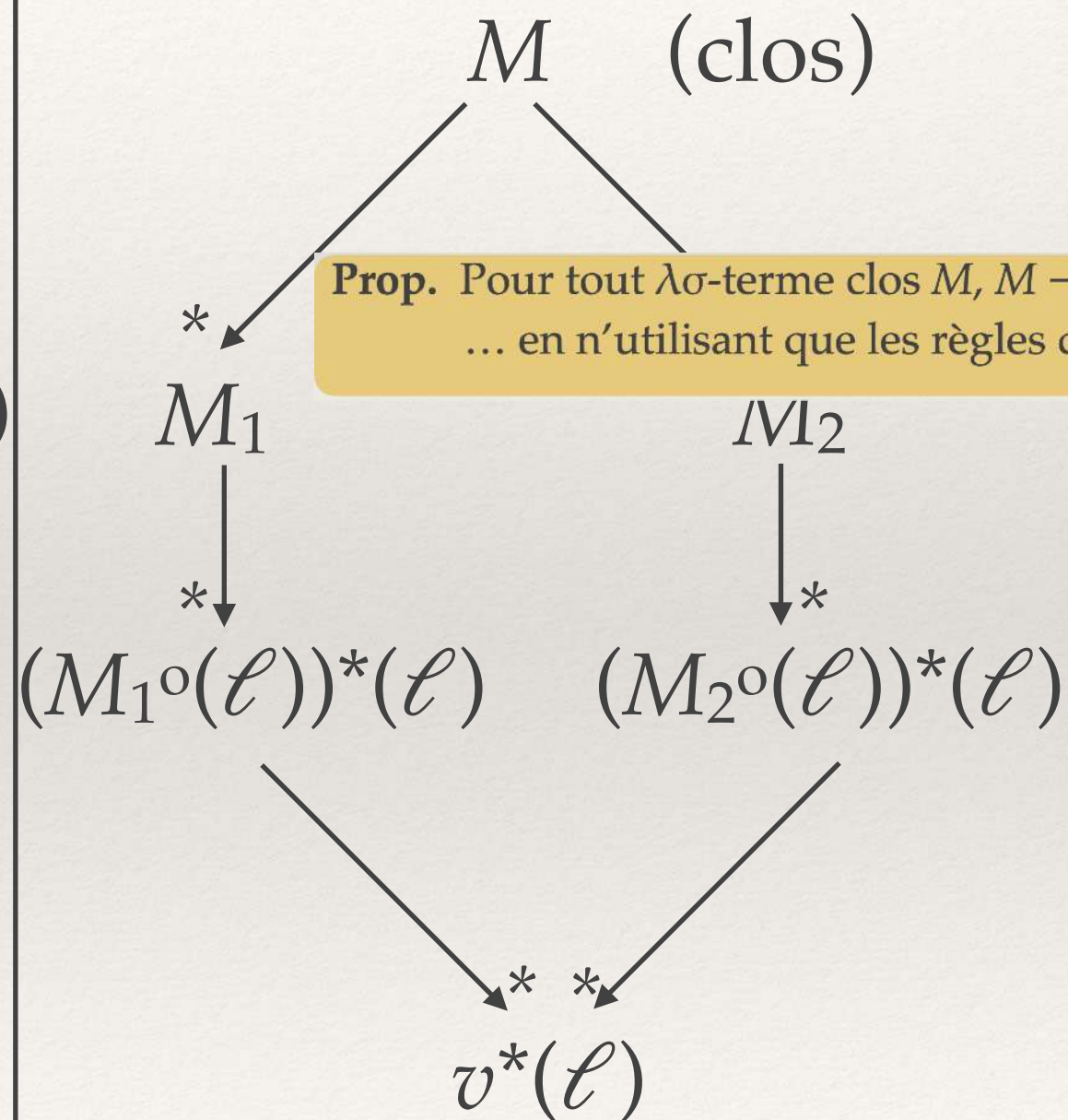
Prop. si $u \rightarrow u'$
alors $u^*(\ell) \rightarrow^+ u'^*(\ell)$

Côté $\lambda\sigma$ -calcul

Confluence du $\lambda\sigma$ -calcul (clos)



Côté λ -calcul



Côté $\lambda\sigma$ -calcul

Prop. Pour tout $\lambda\sigma$ -terme clos M , $M \rightarrow^* (M^o(\ell))^*(\ell)$
... en n'utilisant que les règles de (σ)

Confluence du $\lambda\sigma$ -calcul

- ❖ $M, N, \dots ::= x \mid 1 \mid MN \mid \lambda M \mid M[S]$
 $S, S', \dots ::= \text{id} \mid N \cdot S \mid \uparrow \mid S \circ S'$
- ❖ **Thm (A. Ríos).** Le $\lambda\sigma$ -calcul est confluent.
- ❖ ... y compris en présence de variables de termes (x).

Confluence du $\lambda\sigma$ -calcul?

❖ $M, N, \dots ::= x \mid 1 \mid MN \mid \lambda M \mid M[S]$
 $S, S', \dots ::= \text{id} \mid N \cdot S \mid \uparrow \mid S \circ S' \mid X$

❖ **Thm (A. Ríos).** Le $\lambda\sigma$ -calcul est confluent.

❖ ... y compris en présence de variables de termes (x).

❖ Mais attention:

❖ **Thm (Curien, Hardin, Lévy).** Le $\lambda\sigma$ -calcul **avec variables X de piles** n'est **pas confluent** (il est localement confluent).

La non-confluence du $\lambda\sigma$ -calcul

Proof: In the proof, $\rightarrow$ stands for a β , Id , IdR , $VarShift$ or $SCons$ step. Recall the term presented in [1] in support of the non confluence conjecture:

$B = YC$ where $C = YV$ where $V = \lambda\lambda X[1[x \circ (1 \cdot id)] \cdot (\uparrow \circ (x \circ ((21) \cdot id)))]$

where $Y = (\lambda\lambda 1(221))(\lambda\lambda 1(221))$ is Turing fixed point combinator. We shall actually need to consider a parameterized version of this term. For any substitution s , let:

$B_s = YC_s$ where $C_s = YV_s$ where $V_s = \lambda\lambda X[1[x \circ (1 \cdot s)] \cdot (\uparrow \circ (x \circ ((21) \cdot s)))]$

We shall omit subscripts s , when clear from the context. First observe, for any a , that:

$$(*) \quad Ca \xrightarrow{*} VCa \xrightarrow{*} X[1[x \circ (a \cdot s_1)] \cdot (\uparrow \circ (x \circ ((Ca) \cdot s_1)))]$$

where $s_1 = \sigma(s[a \cdot C \cdot id])$. Thus, for some s' :

$$\begin{aligned} B &\xrightarrow{*} CB \xrightarrow{*} X[1[x \circ (B \cdot s')] \cdot (\uparrow \circ (x \circ (CB \cdot s')))] \\ &\xrightarrow{*} X[1[x \circ (CB \cdot s')] \cdot (\uparrow \circ (x \circ (CB \cdot s')))] \rightarrow X[x \circ (CB \cdot s')] =_{\Delta} A \end{aligned}$$

and

$$B \xrightarrow{*} CB \xrightarrow{*} CA \quad (\text{since } B \xrightarrow{*} A).$$

We show that A and $C_s A$ cannot have a common reduct, whatever is s .



Types simples pour $\lambda\sigma$

Les contextes de typage
sont maintenant
des **listes** de formules

Types simples pour $\lambda\sigma$ (clos)

$$\frac{}{F_1, \dots, F_n \vdash 1 : F_1} (1)$$

$$\frac{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F}{\Gamma \vdash MN : G} (App) \quad \frac{F, \Gamma \vdash M : G}{\Gamma \vdash \lambda M : F \Rightarrow G} (\lambda)$$

$$\frac{\Delta \vdash M : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash M[S] : F} ([\cdot])$$

$$\frac{}{\Gamma \vdash id : \Gamma} (id)$$

$$\frac{}{F, \Gamma \vdash \uparrow : \Gamma} (\uparrow)$$

$$\frac{\Gamma \vdash N : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash N \cdot S : F, \Delta} (\cdot)$$

$$\frac{\Delta \vdash S' : \Lambda \quad \Gamma \vdash S : \Delta}{\Gamma \vdash S' \circ S : \Lambda} (\circ)$$

❖ **Thm (auto-contradiction)** Si $\Gamma \vdash M : F$ et $M \rightarrow^* N$ en $\lambda\sigma$

Le « type » d'une pile S
est une **liste** de formules

❖ **Thm (préservation du typage).**

Si $x_1:F_1, \dots, x_n:F_n \vdash u : F$ en λ -calcul (types simples)

alors $F_1, \dots, F_n \vdash u^*([x_1, \dots, x_n]) : F$ en $\lambda\sigma$

Types simples pour $\lambda\sigma$ (clos)

$$\frac{}{F_1, \dots, F_n \vdash 1 : F_1} (1)$$

$$\frac{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F}{\Gamma \vdash MN : G} (App) \quad \frac{F, \Gamma \vdash M : G}{\Gamma \vdash \lambda M : F \Rightarrow G} (\lambda)$$

$$\frac{\Delta \vdash M : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash M[S] : F} ([\cdot])$$

$$\frac{}{\Gamma \vdash id : \Gamma} (id)$$

$$\frac{}{F, \Gamma \vdash \uparrow : \Gamma} (\uparrow)$$

$$\frac{\Gamma \vdash N : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash N \cdot S : F, \Delta} (\cdot)$$

$$\frac{\Delta \vdash S' : \Lambda \quad \Gamma \vdash S : \Delta}{\Gamma \vdash S' \circ S : \Lambda} (\circ)$$

❖ ~~Thm (terminaison). Si $\Gamma \vdash M : F$
alors M est fortement normalisable (en $\lambda\sigma$)?~~

Typed λ -calculi with explicit substitutions may not terminate

Paul-André Mellies *

Ecole Normale Supérieure, 45 rue d'Ulm, 75005 Paris, France
INRIA Rocquencourt, Domaine de Voluceau, 78153 Le Chesnay Cedex, France
FWI, De Boelelaan 1081a, 1081 HV Amsterdam, Nederland
mellies@cs.vu.nl

Abstract. We present a simply typed λ -term whose computation in the $\lambda\sigma$ -calculus does not always terminate.



Paul-André Melliès

Le contre-ex

$$\begin{aligned}
 S_1 \circ S &\rightarrow^3 (\lambda 1)[S](1[S]) \cdot S \\
 &\rightarrow (\lambda(1[1 \cdot (S \circ \uparrow)]))(1[S]) \cdot S \\
 &\rightarrow (1[1 \cdot (S \circ \uparrow)])(1[S] \cdot \text{id}) \cdot S \quad (\beta) \\
 &\rightarrow 1[(1 \cdot (S \circ \uparrow)) \circ (1[S] \cdot \text{id})] \cdot S \quad (\text{assoc.}) \\
 &\rightarrow^+ 1[1[S] \cdot S'] \cdot S
 \end{aligned}$$

- ❖ Pour toute pile S , soit:

$$\text{rec}(S) = \uparrow \circ (1[S] \cdot \text{id}) \quad D_S(S') = 1[1[S] \cdot S'] \cdot S$$

et soit $S_1 = (\lambda 1)1 \cdot \text{id}$

- ❖ On vérifie que: $S_1 \circ S \rightarrow^+ D_S(S \circ \text{rec}(S))$

- ❖ Soit $C_S(S') = \uparrow \circ (1[S'] \cdot S)$

On vérifie que: $\text{rec}(S') \circ S \rightarrow^+ C_S(S' \circ S)$

- ❖ Soit $S_{n+1} = \text{rec}(S_n)$

On vérifie que: $S_n \circ S_{n+1} \rightarrow^+ C_{S_{n+1}}^{n-1}(D_{S_{n+1}}(S_{n+1} \circ S_{n+2}))$

- ❖ Donc $S_1 \circ S_1$ ne termine pas.

- ❖ Soit $u = \lambda z'.(\lambda x.(\lambda y.y)((\lambda z.z)x))((\lambda y.y)z') : \alpha \Rightarrow \alpha$

On a $u^*([\])\rightarrow^+ \lambda(1[S_1 \circ S_1])$: bien typé, mais ne termine pas!

Types simples pour $\lambda\sigma$ (clos)

$$\frac{}{F_1, \dots, F_n \vdash 1 : F_1} (1)$$

$$\frac{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F}{\Gamma \vdash MN : G} (App) \quad \frac{F, \Gamma \vdash M : G}{\Gamma \vdash \lambda M : F \Rightarrow G} (\lambda)$$

$$\frac{\Delta \vdash M : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash M[S] : F} ([\cdot])$$

$$\frac{}{\Gamma \vdash id : \Gamma} (id)$$

$$\frac{}{F, \Gamma \vdash \uparrow : \Gamma} (\uparrow)$$

$$\frac{\Gamma \vdash N : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash N \cdot S : F, \Delta} (\cdot)$$

$$\frac{\Delta \vdash S' : \Lambda \quad \Gamma \vdash S : \Delta}{\Gamma \vdash S' \circ S : \Lambda} (\circ)$$

❖ **Thm (normalisation).** Si $\Gamma \vdash M : F$
alors M est (faiblement) **normalisable** en $\lambda\sigma$

❖ Preuve (esquisse):
 $M^\circ(\ell)$ est un λ -terme typable et:

❖ **Prop.** si $M \rightarrow M'$
alors $M^\circ(\ell) \rightarrow^* M'^\circ(\ell)$

... et même $M^\circ(\ell) \rightarrow^+ M'^\circ(\ell)$
si M est σ -normal et $M \rightarrow M'$ par (β)
(note: $M^\circ(\ell) = M'^\circ(\ell)$ si σ -réduction)

Une machine de Krivine pour $\lambda\sigma$

Une machine de Krivine pour $\lambda\sigma$

- ❖ Une configuration $(M, S, args)$ où $args=[N_1;\dots;N_k]$ représente $M[S]N_1\dots N_k$

$$\begin{array}{lll} MN, S, args & \rightarrow & M, S, N[S] :: args \\ \lambda M, S, N :: args & \rightarrow & M, N \cdot S, args \\ M[S'], S, args & \rightarrow & M, S' \circ S, args \end{array}$$

$$\begin{array}{lll} 1, N \cdot S, args & \rightarrow & N, id, args \\ n+1, N \cdot S, args & \rightarrow & n, S, args \\ x[\uparrow^{n+1}], N \cdot S, args & \rightarrow & x[\uparrow^n], S, args \\ n, \uparrow, args & \rightarrow & n+1, id, args \\ x[\uparrow^n], \uparrow, args & \rightarrow & x[\uparrow^{n+1}], id, args \\ n, \uparrow \circ S, args & \rightarrow & n+1, S, args \\ x[\uparrow^n], \uparrow \circ S, args & \rightarrow & x[\uparrow^{n+1}], S, args \\ M, id \circ S, args & \rightarrow & M, S, args \\ M, (S'' \circ S') \circ S, args & \rightarrow & M, S'' \circ (S' \circ S), args \\ M, (N \cdot S') \circ S, args & \rightarrow & M, N[S] \cdot (S' \circ S), args \end{array}$$

$(\beta) (\lambda M)N \rightarrow M[N \cdot id]$	(σ)
$\cdot 1[N \cdot S] \rightarrow N$	$\uparrow \circ (N \cdot S) \rightarrow S$
$\cdot (MN)[S] \rightarrow M[S](N[S])$	$(N \cdot S) \circ S' \rightarrow N[S'] \cdot (S \circ S')$
$\cdot (\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$	
$\cdot M[S][S'] \rightarrow M[S \circ S']$	$(S \circ S') \circ S'' \rightarrow S \circ (S' \circ S'')$
$\cdot$	$id \circ S \rightarrow S$
$\cdot M[id] \rightarrow M$	$S \circ id \rightarrow S$
$\cdot$	

où M est de la forme n ou $x[\uparrow^n]$.

Une machine de Krivine pour $\lambda\sigma$

- ❖ Une configuration $(M, S, args)$ où $args=[N_1;\dots;N_k]$ représente $M[S]N_1\dots N_k$
- ❖ ... et l'on a les théorèmes de correction, et de progrès attendus:

Exercice 20 Montrer que si $M, S, [M_1;\dots;M_n] \rightarrow^* M', S', [M'_1;\dots;M'_{n'}]$ dans la machine ci-dessus, alors $M[S]M_1\dots M_n$ est σ -convertible avec un terme qui se réduit dans le $\lambda\sigma$ -calcul à un terme σ -convertible avec $M'[S']M'_1\dots M'_{n'}$.

Exercice 21 Montrer que dans le cas semi-clos, si la machine ci-dessus termine, alors c'est dans un état de la forme $M, id, args$ où M est de la forme n , x , ou $x[\uparrow^n]$, ou $\lambda M'$, et que dans ce dernier cas $args$ est la liste vide.

Exercice 22 Une forme normale de tête faible en $\lambda\sigma$ -calcul est un terme de la forme λM , ou bien $hM_1\dots M_n$, où la tête h est de la forme n ou x ou $x[\uparrow^n]$. Montrer que la machine ci-dessus calcule une forme normale de tête faible d'un $\lambda\sigma$ -terme semi-clos M s'il en a une, en démarrant de l'état $M, id, []$. (On utilisera la traduction $M \mapsto M^\circ(\ell)$ pour ℓ suffisamment grande et ses propriétés, le théorème de standardisation du λ -calcul, et l'exercice 20.)

(Oui, je fatigue)

Conclusions

Calculs à substitutions explicites

- ❖ Comme on l'a vu, le $\lambda\sigma$ -calcul présente un mélange de
 - bonnes propriétés
(confluence dans le cas clos, σ termine) et de
 - mauvaises propriétés
(pas de normalisation forte par ex.)
- ❖ Peut-on améliorer ça?
- ❖ Oui... des dizaines de calculs à substitutions explicites ont été proposés depuis; mais ce n'est plus un domaine actif en recherche aujourd'hui

Implémentations du λ -calcul

- ❖ Outre les calculs à substitutions explicites, on trouve:
- ❖ les systèmes de **combinateurs** (SKI, CBWK, SKInT, etc.)
- ❖ la **géométrie de l'interaction** (Girard)
et sa **machine** (Mackie)
- ❖ les calculs de **réduction optimale** (Lévy; Lamping)
- ❖ ... et puis toutes les machines à environnements utilisées en pratique (interprètes, compilateurs; ML, Lisp, Haskell, JavaScript, etc.)

Le cours est fini!

- ❖ Voilà, j'espère que tout ça vous a donné une bonne idée de ce qu'est le λ -calcul,
- ❖ avec ses propriétés fondamentales
- ❖ ses liens avec la théorie de la preuve
- ❖ et quelques problèmes liés à son implémentation.