Propriétés du λσ-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

$(\beta) \ (\lambda M) N \rightarrow M[N \cdot \mathrm{id}]$	(σ)
$ \begin{array}{ll} .1[N \cdot S] \to N & \uparrow \circ (N \cdot S) \to S \\ .(MN)[S] \to M[S](N[S]) & (N \cdot S) \circ S' \to I \\ (\lambda M)[S] \to \lambda (M[1 \cdot (S \circ \uparrow)]) \end{array} $	$\mathbb{V}[S'] \cdot (S \circ S')$
$(M(I)[S] \rightarrow M(M[I \cap (S \cup [)]))$ $. M[S][S'] \rightarrow M[S \cup S'] \qquad (S \cup S') \cup S'' \rightarrow$ $. \qquad id \cup S \rightarrow S$	$S \circ (S' \circ S'')$
$M[id] \to M \qquad S \circ id \to S$	

- * **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.

$(\beta) \ (\lambda M) N \rightarrow M[N \cdot \mathrm{id}]$		(σ)
$.1[N \cdot S] \rightarrow N$	$\uparrow \circ (N \cdot S) \to S$	
$. \ (MN)[S] \rightarrow M[S](N[S])$	$(N \cdot S) \circ S' \to N$	$V[S'] \cdot (S \circ S')$
. $(\lambda M)[S] \rightarrow \lambda (M[1 \cdot (S \circ \uparrow$)])	
$. M[S][S'] \to M[S \circ S']$	$(S \circ S') \circ S'' \to$	$S \circ (S' \circ S'')$
	$\mathrm{id} \circ S \twoheadrightarrow S$	
$M[id] \rightarrow M$	$S \circ \mathrm{id} \rightarrow S$	

- * Ceci nécessitera quelques efforts...
- Une fois ceci fait, le lemme de Newman nous dira que tout λσ-terme a une σ-forme normale unique.

Le sous-calcul σ des substitutions

- * σ est localement confluent.
- Nous allons montrer que σ termine.

* Ceci nécessitera quelques efforts	rts
-------------------------------------	-----

 Une fois ceci fait, le lemme de Newman nous dira que tout λσ-terme a une σ-forme normale unique.

100	$(\beta) (\lambda M) N \rightarrow M[N \cdot \mathrm{id}]$		(σ)
	$1[N \cdot S] \rightarrow N$ $(MN)[S] \rightarrow M[S](N[S])$ $(\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow M[S][S'] \rightarrow M[S \circ S'])$ $M[S][S'] \rightarrow M[S \circ S']$	$\uparrow \circ (N \cdot S) \rightarrow S$ $(N \cdot S) \circ S' \rightarrow N$ $(S \circ S') \circ S'' \rightarrow S$ $id \circ S \rightarrow S$ $S \circ id \rightarrow S$	$V[S'] \cdot (S \circ S')$ $S \circ (S' \circ S'')$

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 $\ell \to r$ entre expressions de la logique du premier ordre
- Comme σ

$(\beta) \ (\lambda M) N \rightarrow M[N \cdot \mathrm{id}]$		(σ)
$ \begin{array}{l} . \ 1[N \cdot S] \rightarrow N \\ . \ (MN)[S] \rightarrow M[S](N[S]) \\ . \ (\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow $	$\uparrow \circ (N \cdot S) \to S$ $(N \cdot S) \circ S' \to N[S'] \cdot$)])	$(S \circ S')$
$. M[S][S'] \to M[S \circ S']$	$(S \circ S') \circ S'' \to S \circ (S)$ id $\circ S \to S$	′ ° <i>S″</i>)
$M[id] \rightarrow M$	$S \circ \mathrm{id} \rightarrow S$	
•		

Terminaison des systèmes de réécriture

- * Systèmes de règles $\ell \to r$ entre expressions de la logique du premier ordre
- Comme σ
- * Comme $\rightarrow_{\mathbb{N}}$

$(\beta) \ (\lambda M) N \to M[N$	l∙id]			(σ)
$ 1[N \cdot S] \rightarrow N (MN)[S] \rightarrow M[S] (\lambda M)[S] \rightarrow \lambda(M[T] M[S][S'] \rightarrow M[S] . $	(N[S]) 1 · (S ○ ↑)]) ○ S']	$\uparrow \circ (N \cdot S) \rightarrow$ $(N \cdot S) \circ S' -$ $(S \circ S') \circ S'' -$ $id \circ S \rightarrow S$	S > N[S′ → S ○	$[] \cdot (S \circ S')$ $(S' \circ S'')$
. <i>M</i> [id] → <i>M</i>	$(0 \approx S)$ $(S \approx S)$ (+0) (+S) (*0) (*S)	$\begin{array}{c} 0 \approx \mathbf{S}(t) \\ \mathbf{S}(s) \approx \mathbf{S}(t) \\ s{+}0 \\ s{+}\mathbf{S}(t) \\ s{*}0 \\ s{*}\mathbf{S}(t) \end{array}$	$ \begin{array}{c} \rightarrow_{\mathbb{N}} \\ \rightarrow_{\mathbb{N}} \\ \rightarrow_{\mathbb{N}} \\ \rightarrow_{\mathbb{N}} \\ \rightarrow_{\mathbb{N}} \end{array} $	$ \begin{array}{c} \bot \\ s \approx t \\ s \\ \mathrm{S}(s{+}t) \\ \mathrm{O} \\ s{*}t{+}s \end{array} $

Terminaison des systèmes de réécriture

* Systèmes de règles $\ell \to r$ entre expressions de la logique du premier ordre

 $D_x($

 $D_x($

 $D_x($

D

 D_r

- Comme σ
- * Comme $\rightarrow_{\mathbb{N}}$
- Comme encore
 bien d'autres

	(β)	$(\lambda M)N \rightarrow M[N \cdot \mathrm{id}]$		(σ)
	. 1[1 . (Μ . (λ) . Μ	$N \cdot S] \rightarrow N$ $[N)[S] \rightarrow M[S](N[S])$ $[S][S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow))$ $[S][S'] \rightarrow M[S \circ S']$	$\uparrow \circ (N \cdot S) \to S$ $(N \cdot S) \circ S' \to N$ $])$ $(S \circ S') \circ S'' \to S$ $id \circ S \to S$	$V[S'] \cdot (S \circ S')$ $S \circ (S' \circ S'')$
$D_x(x)$	\rightarrow	1		$s \approx t$
$D_x(a)$	\rightarrow	0		8
M+N	\rightarrow	$D_x(M) + D_x(N)$		S(s+t)
$M \times N$)	\rightarrow	$D_x(M) \times N + M$	$\times D_x(N)$	0
M-N	\rightarrow	$D_x(M) - D_x(N)$		s*t+s
$D_x(-M)$	\rightarrow	$-D_x(M)$		-
$_{x}(M/N)$	\rightarrow	$\left(D_x(M) \times N - M\right)$	$M imes D_x(N) \Big) / M$	V^2
$\log(M))$	\rightarrow	$\grave{D}_x(M)/M$	/	

- Trouver un ordre strict bien fondé > sur les expressions tel que:
 pour toute règle ℓ → r, ℓ > r
 - passage au contexte: si u > v alors f(..., u, ...) > f(..., v, ...)

- ★ 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(..., u, ...) > f(..., v, ...)

- * 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(..., u, ...) > f(..., v, ...)
- * Si $u \rightarrow v$, alors u > v.

- Trouver un ordre strict bien fondé > sur les expressions tel que:
 pour toute règle ℓ → r, ℓ > r
 - passage au contexte:
 si u > v alors f(..., u, ...) > f(..., v, ...)

- * Si $u \to v$, alors u > v.
- * Donc pas de réduction infinie: si $u=u_0 \rightarrow u_1 \rightarrow ... \rightarrow u_n \rightarrow^{\infty} ...$ alors $u=u_0 > u_1 > ... > u_n > ...$

- Trouver un ordre strict bien fondé > sur les expressions tel que:
 pour toute règle ℓ → r, ℓ > r
 - passage au contexte: si u > v alors f(..., u, ...) > f(..., v, ...)

- * Si $u \to v$, alors u > v.
- * Donc pas de réduction infinie: si $u=u_0 \rightarrow u_1 \rightarrow ... \rightarrow u_n \rightarrow^{\infty} ...$ alors $u=u_0 > u_1 > ... > u_n > ...$
- * Pour $\rightarrow_{\mathbb{N}}$, on avait défini e > e' ssi [e] > [e']

$(0 \approx S)$	$0 pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	1
$(S \approx S)$	$\mathtt{S}(s) pprox \mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
(+0)	s+0	$\rightarrow_{\mathbb{N}}$	8
(+S)	$s+\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	S(s+t)
(*0)	<i>s</i> *0	$\rightarrow_{\mathbb{N}}$	0
(*S)	$s*\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s{*}t{+}s$

Ordre lexicographique

- * Soit >, >' deux ordres stricts. On pose (u, u') (> × >')_{lex} (v, v') ssi -u > v ou -u = v et u' > v'
- * **Prop.** Si > et >' bien fondés, alors $(> \times >')_{lex}$ 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. □

Extension lexicographique

- * Soit > ordre strict. On pose $(u_1, ..., u_m) >_{lex} (v_1, ..., v_n)$ 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 $i, 1 \le i \le n$
- Prop. Si > bien fondé, alors >_{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*. □

Le lpo

- On se fixe un ordre strict > sur les symboles de fonction (la précédence)
- * On définit >^{lpo} sur les expressions closes par les règles:

 $[\geq^{lpo} abrège (>^{lpo} ou =)]$

$$(f_{i}(s_{1}, ..., s_{m})) \gg (g_{i}(t_{1}, ..., t_{n}))$$

$$\exists i) \quad s_{i} \geq^{lpo} t \qquad (\forall j) \quad s \geq^{lpo} t_{j}$$

$$s \geq^{lpo} t \qquad (\forall s \geq^{lpo} t) \qquad [\gg abrège (> \times (>^{lpo})_{lex})_{lex}]$$

où
$$s=f(s_1, ..., s_m)$$

 $t=g(t_1, ..., t_n)$

Le lpo

- On se fixe un ordre strict > sur les symboles de fonction (la précédence)
- * On définit >^{lpo} sur les expressions closes par les règles:

$$(f_{i}(s_{1}, ..., s_{m})) \gg (g_{i}(t_{1}, ..., t_{n}))$$

$$(\exists i) \quad s_{i} \geq^{lpo} t \qquad (\forall j) \quad s >^{lpo} t_{j}$$

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

$$\geq^{lpo} i \quad brège (>^{lpo} ou =)]$$

$$Où \quad s = f(s_{1}, ..., s_{m})$$

$$t = g(t_{1}, ..., t_{n})$$

$$t = g(t_{1}, ..., t_{n})$$

Le lpo

- On se fixe un ordre strict > sur les symboles de fonction (la précédence)
- * On définit >^{lpo} sur les expressions closes par les règles:

$$(f_{i}(s_{1}, ..., s_{m})) \gg (g_{i}(t_{1}, ..., t_{n}))$$

$$(\exists i) s_{i} \geq lpo t \qquad (\forall j) s \geq lpo t_{j}$$

$$(\forall j) s \geq lpo t \qquad [\gg abrège (> \times (> lpo)_{lex})_{lex}]$$

$$s \geq lpo t \qquad [\gg abrège (> \times (> lpo)_{lex})_{lex}]$$

$$C'est-à-dire: f>g ou bien \qquad =f(s_{1}, ..., s_{m})$$

$$(f=g et: m>n ou \qquad =f(s_{1}, ..., s_{m})$$

$$(f=g et: m>n ou \qquad =n et: s_{1}>lpot_{1} ou \qquad =g(t_{1}, ..., t_{n})$$

$$m=n et: s_{1}>lpot_{2} ou \qquad etc.)$$

Le	e lpo			
 On se fixe un ordre strict > (la précédence) 	\bigvee Ça, c'est pour éviter que les t_j ne grossissent trop. Par ex., si $f > g$: $s = f(s_1) \gg g(s)$ ion ion $\max s \rightarrow g(s) \rightarrow g(g(s)) \rightarrow \dots$ réduction infinie			
On définit > ^{lpo} sur les expressions closes par les règles:				
$(f_{1}(s_{1},, s_{m})) \gg (g_{1}(t_{1},, t_{n}))$				
$(\exists i) \ S_i \geq^{\text{lpo}} t$	$(\forall j) \ s >^{\text{lpo}} t_j$			
s >lpo t	s > lpo t [» abrège (> × (>lpo) _{lex}) _{lex}]			
$[\geq^{lpo} i brège (>^{lpo} ou =)]$ f>g ou =)	-à-dire: u bien et: $m > n$ ou $= \sigma(t_1, \dots, s_m)$			
Permet notamment de conclure $f(s_1,, s_m) >^{lpo} s_i$	$m=n \text{ et: } s_1 >^{\text{lpo}} t_1 \text{ ou}$ $s_1=t_1 \text{ et: } s_2 >^{\text{lpo}} t_2 \text{ ou}$ etc.)			

>lpo est un ordre strict
 (et même total si > est total)

	$(f_{1}(s_{1},, s_{m})) \gg (g_{1}(t_{1},, t_{n}))$
$(\exists i) \ s_i \geq^{\text{lpo}} t$	$(\forall j) \ s >^{\mathrm{lpo}} t_j$
$s >^{\text{lpo}} t$	$s >^{\text{lpo}} t$

>^{lpo} est un ordre strict
 (et même total si > est total)

	$(f_{1}(s_{1},, s_{m})) \gg (g_{1}(t_{1},, t_{n}))$
$(\exists i) \ S_i \geq^{\text{lpo}} t$	$(\forall j) \ s >^{\mathrm{lpo}} t_j$
$s >^{\text{lpo}} t$	$s >^{ m lpo} t$

* ... qui passe au contexte

>^{lpo} est un ordre strict
 (et même total si > est total)



- * ... qui passe au contexte
- * … qui a la propriété de sous-terme:
 si *t* est un sous-terme strict de *s* (en notation *s* ⊳ *t*),
 alors *s* >^{lpo} *t*

>^{lpo} est un ordre strict
 (et même total si > est total)



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

- Comment montrer que →_N termine sans se fatiguer
- * Précédence: $\approx > \bot \quad * > + > S$

$(0 \approx S)$	$0 pprox \mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	
$(\mathtt{S}\approx\mathtt{S})$	$\mathbf{S}(s) pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	s pprox t
(+0)	s+0	$\rightarrow_{\mathbb{N}}$	s
(+S)	$s+\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$\mathtt{S}(s{+}t)$
(*0)	<i>s</i> *0	$\rightarrow_{\mathbb{N}}$	0
(*S)	$s*\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s{*}t{+}s$

	$(f_{1}(s_{1},, s_{m})) \gg (g_{1}(t_{1},, t_{n}))$
$(\exists i) \ s_i \geq^{\text{lpo}} t$	$(\forall j) \ s >^{\mathrm{lpo}} t_j$
s > lpo t	$s >^{\text{lpo}} t$

?

s+S(t) > lpo S(s+t)

- Comment montrer que →_N termine sans se fatiguer
- * Précédence: $\approx > \perp * > + > S$

$(0 \approx S)$	$\mathtt{O}pprox \mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	\perp
$(\mathtt{S}\approx\mathtt{S})$	$\mathbf{S}(s) pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
(+0)	s+0	$\rightarrow_{\mathbb{N}}$	s
(+S)	$s+\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$\mathtt{S}(s{+}t)$
(*0)	<i>s</i> *0	$\rightarrow_{\mathbb{N}}$	0
(*S)	$s*\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s{*}t{+}s$

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

?

$$\frac{s+S(t)>^{\text{lpo}}s+t}{s+S(t)>^{\text{lpo}}S(s+t)}$$
Règle de droite (car + > S)

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

$(0 \approx S)$	$\mathtt{O} pprox \mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	\perp
$(\mathtt{S}\approx\mathtt{S})$	$\mathbf{S}(s) pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
(+0)	s+0	$\rightarrow_{\mathbb{N}}$	s
(+S)	$s+\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	S(s+t)
(*0)	<i>s</i> *0	$\rightarrow_{\mathbb{N}}$	0
(*S)	$s*\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s{*}t{+}s$

	$(f_{r}(s_{1},, s_{m})) \gg (g_{r}(t_{1},, t_{n}))$
$Ii) S_i \geq^{lpo} t$	$(\forall j) \ s >^{\mathrm{lpo}} t_j$
s >lpo t	$s >^{\text{lpo}} t$

$$S(t) >^{lpo} t$$

$$s+S(t) >^{lpo} s \quad s+S(t) >^{lpo} t$$

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

- Comment montrer que →_N termine sans se fatiguer
- * Précédence: $\approx > \bot \quad * > + > S$

$(0 \approx S)$	$0 pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	\perp
$(\mathtt{S}\approx\mathtt{S})$	$\mathbf{S}(s) pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
(+0)	s+0	$\rightarrow_{\mathbb{N}}$	s
(+S)	$s+\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$\mathtt{S}(s{+}t)$
(*0)	<i>s</i> *0	$\rightarrow_{\mathbb{N}}$	0
(*S)	$s*\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s{*}t{+}s$

	$(f_{1}(s_{1},, s_{m})) \gg (g_{1}(t_{1},, t_{n}))$
$(\exists i) \ s_i \geq^{\text{lpo}} t$	$(\forall j) \ s >^{\text{lpo}} t_j$
$s >^{\text{lpo}} t$	$s >^{ ext{lpo}} t$

$$\begin{split} \mathbf{S}(t) > & \mathrm{lpo} \ t \\ s + \mathbf{S}(t) > & \mathrm{lpo} \ s \quad s + \mathbf{S}(t) > & \mathrm{lpo} \ t \\ \hline s + \mathbf{S}(t) > & \mathrm{lpo} \ s + t \\ s + \mathbf{S}(t) > & \mathrm{lpo} \ \mathbf{S}(s + t) \end{split}$$

Sous-terme! (ou bien règle de gauche)

Règle de droite (ici +=+, s=s et on vérifie S(t) > lpo t) Règle de droite (car + > S)

- Comment montrer que →_N termine sans se fatiguer
- * Précédence: $\approx > \bot \quad * > + > S$

$(0 \approx S)$	$\mathtt{O} pprox \mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	\perp
$(\mathtt{S}\approx\mathtt{S})$	$\mathbf{S}(s) pprox \mathbf{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s \approx t$
(+0)	s+0	$\rightarrow_{\mathbb{N}}$	\boldsymbol{s}
(+S)	$s+\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	S(s+t)
(*0)	<i>s</i> *0	$\rightarrow_{\mathbb{N}}$	0
(*S)	$s*\mathtt{S}(t)$	$\rightarrow_{\mathbb{N}}$	$s{*}t{+}s$

	$(f_{1}(s_{1},, s_{m})) \gg (g_{1}(t_{1},, t_{n}))$
$(\exists i) \ s_i \geq^{\text{lpo}} t$	$(\forall j) \ s >^{lpo} t_j$
$s >^{\text{lpo}} t$	$s >^{ m lpo} t$

$$\frac{S(t) >^{lpo} t}{s+S(t) >^{lpo} s \quad s+S(t) >^{lpo} t}$$
Sous-terme! (ou bien règle de gauche)

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

Je vous laisse vérifier les autres règles (exercice!)

- * Plus compliqué?
- Pas du tout,
 c'est totalement
 mécanique
- Précédence:
 D_x > 1, 0, +, ×, -, /, ^

$D_x(x)$	\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	$\dot{D}_x(M)/M$

	$(f_r(s_1,, s_m)) \gg (g_r(t_1,, t_n))$
($\exists i$) $s_i \geq^{\operatorname{lpo}} t$	$(\forall j) \ s >^{\mathrm{lpo}} t_j$
s > lpo t	$s >^{\text{lpo}} t$

Soit: Acc la partie bien fondée de >lpo

Supposons > bien fondé

- $= \{s \mid pas de chaîne infinie décroissante$
 - pour >lpo partant de s}

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

Soit: Acc la partie bien fondée de >lpo

- $= \{s \mid pas de chaîne infinie décroissante$
 - pour >lpo partant de s}
- $\underline{Acc} = \{f(s_1, \dots, s_m) \mid \text{ tout } s_i \text{ est dans Acc}\}$

Supposons <u>Acc</u> – Acc non vide

Supposons > bien fondé




























Soit: Acc la partie bien fondée de >lpo

- Supposons > bien fondé
- = {s | pas de chaîne infinie décroissante pour >^{lpo} partant de s}
- $\underline{Acc} = \{f(s_1, \dots, s_m) \mid \text{ tout } s_i \text{ est dans Acc}\}$
- Donc <u>Acc</u> Acc est vide.

Soit: Acc la partie bien fondée de >lpo

- Supposons > bien fondé
- = {s | pas de chaîne infinie décroissante pour >^{lpo} partant de *s*}
- $\underline{Acc} = \{f(s_1, \dots, s_m) \mid \text{ tout } s_i \text{ est dans Acc}\}$
- Donc <u>Acc</u> Acc est vide.

Autrement dit: pour tout *s*=*f*(*s*₁,...,*s*_m),
si tous les *s*_i sont dans Acc (i.e., si *s* ∈ <u>Acc</u>)
alors *s* est dans Acc.

Soit: Acc la partie bien fondée de >lpo = {s | pas de chaîne infinie décroissante pour >lpo partant de s} Acc = {f($s_1, ..., s_m$) | tout s_i est dans Acc} * Donc Acc – Acc est vide. * Autrement dit: pour tout $s=f(s_1, ..., s_m)$, si tous les s_i sont dans Acc (i.e., si $s \in Acc$) alors s est dans Acc. * Donc, par récurrence sur (la taille de) s:

 Supposons > bien fondé Soit: Acc la partie bien fondée de >lpo $= \{s \mid pas de chaîne infinie décroissante$ pour >lpo partant de s} $\underline{Acc} = \{f(s_1, \dots, s_m) \mid \text{ tout } s_i \text{ est dans Acc}\}$ * Donc <u>Acc</u> – Acc est vide. * Autrement dit: pour tout $s=f(s_1,...,s_m)$, si tous les s_i sont dans Acc (i.e., si $s \in Acc$) alors s est dans Acc. * Donc, par récurrence sur (la taille de) s: **Thm.** Toute expression *s* est dans Acc.

Soit: Acc la partie bien fondée de $>^{lpo}$ = {s pas de chaîne infinie décroissant pour $>^{lpo}$ partant de s}	 Supposons > bien fondé
$\underline{Acc} = \{f(s_1, \dots, s_m) \mid \text{ tout } s_i \text{ est dans Acc}\}$	
* Donc <u>Acc</u> – Acc est vide.	
* Autrement dit: pour tout $s=f(s_1,$	$(,S_m),$
si tous les s_i sont d	lans Acc (i.e., si $s \in \underline{Acc}$)
alors s est dans Ac	CC.
* Donc, par récurrence sur (la taille	de) s:
* Thm. Toute expression <i>s</i> est dans	Acc.
 I.e., >^{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 avec $(\lambda M)[S] \rightarrow \lambda(M[1 \cdot (S \circ \uparrow)])$



Et pour σ ?

- * Ça ne fonctionne pas
- * Quelle que soit la précédence choisie. vous aurez des problèm $(\lambda M)[S] \rightarrow$

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





- * En fait, la terminaison de σ est un problème **difficile**.
- Nous verrons une preuve due à Hans Zantema.

1/3: simplification

On traduit les λσ-termes
en remplaçant
[] par °, @ par ·

$(\beta) (\lambda M) N \rightarrow M[N \cdot id]$		(σ)
$.1[N \cdot S] \rightarrow N$	$\uparrow \circ (N \cdot S) \to S$	
$. \ (MN)[S] \to M[S](N[S])$	$(N \cdot S) \circ S' \to N$	$V[S'] \cdot (S \circ S')$
$. \ (\lambda M)[S] \to \lambda(M[1 \cdot (S \circ \uparrow)$])	
$. M[S][S'] \to M[S \circ S']$	$(S \circ S') \circ S'' \to S''$	$S \circ (S' \circ S'')$
	$\mathrm{id} \circ S \twoheadrightarrow S$	
$M[id] \rightarrow M$	$S \circ \mathrm{id} \rightarrow S$	

1/3: simplification

On traduit les λσ-termes en remplaçant
[] par 0, @ par .

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

(**σ**')

1/3: simplification

On traduit les λσ-termes en remplaçant
[] par 0, @ par .

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

 (σ')

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

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

* Le « **semantic labeling** » de Hans Zantema:



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

- * Le « **semantic labeling** » de Hans Zantema:
- on va étiqueter chaque opérateur o par un entier:



- qui est la valeur sémantique de quelque chose
- qui reste invariant (ou décroît)
 le long des réductions

- * Le « **semantic labeling** » de Hans Zantema:
- on va étiqueter chaque opérateur o 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 $M \circ S$



- * Le « **semantic labeling** » de Hans Zantema:
- on va étiqueter chaque opérateur o 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 $M \circ S$





- * Le « **semantic labeling** » de Hans Zantema:
- on va étiqueter chaque opérateur o 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 $M \circ S$





si $M \to N$ en (σ'), alors $\ell(M) \ge \ell(N)$

* On étiquette \circ dans $M \circ S$						
	pai	:t	$\mathcal{C}(M) + \mathcal{C}(S)$	5)		
	$\ell(x)$	^	0	$\ell(1)$	^	0
<i>ℓ</i> (.	$\ell(\uparrow)$ $M \cdot N)$	≡ ≙	$0 \\ \max(\ell(M), \ell(N))$	$\ell(\lambda M)$	=	$\ell(M) + 1$
	$\ell(id)$	Ê	0	$\ell(M \circ N)$	Ê	$\ell(M) + \ell(N)$

		(<u>ơ</u> ′)
$1 \circ (N \cdot S) \rightarrow N$	$\uparrow \circ (N \cdot S) \to S$	
$(N \cdot S)$	$\circ S' \to (N \circ S') \cdot (S \circ S')$	
$(\lambda M)\circ S\to\lambda(M\circ(1$	$\cdot (S \circ \uparrow)))$	
$(M \circ S)$	$\circ S' \to M \circ (S \circ S')$	
$\mathrm{id} \circ S \to S$	$S \circ \mathrm{id} \rightarrow S$	
$1 \cdot \uparrow \rightarrow id$	$(1 \circ S) \cdot (\uparrow \circ S) \rightarrow$	• S

* On étiquette \circ dans $M \circ S$					
par $\ell(M) + \ell(S)$	5)				
$\ell(x) \stackrel{\circ}{=} 0$ $\ell(\uparrow) \stackrel{\circ}{=} 0$	$\ell(1) \ \ell(\lambda M)$		$egin{array}{c} 0 \ \ell(M)+1 \end{array}$		(
$\begin{array}{rcl} \ell(M \cdot N) &=& \max(\ell(M), \ell(N)) \\ \ell(id) & \stackrel{\circ}{=} & 0 \end{array}$	$\ell(M \circ N)$	ê	$\ell(M) + \ell(N)$		1



La règle problématique devient:
 (λM) ◦_{m+1+s} S → λ(M ◦_{m+s} (1 · (S ◦_s ↑)))
 où m=ℓ(M), s=ℓ(S)

* On étiquette \circ dans $M \circ S$						
par $\ell(M) + \ell(S)$	5)					
$ \begin{array}{rcl} \ell(x) & \hat{=} & 0 \\ \ell(\uparrow) & \hat{=} & 0 \\ \ell(M,N) & \hat{=} & \max(\ell(M),\ell(N)) \end{array} $	$\ell(1) \ \ell(\lambda M)$	^ ^	$egin{array}{c} 0 \ \ell(M)+1 \end{array}$			
$\ell(M \cdot N) = \max(\ell(M), \ell(N))$ $\ell(id) = 0$	$\ell(M \circ N)$	Ê	$\ell(M) + \ell(N)$			



La règle problématique devient:
 (λM) ◦_{m+1+s} S → λ(M ◦_{m+s} (1 ⋅ (S ◦_s ↑)))
 où m=ℓ(M), s=ℓ(S)
 On a maintenant une infinité

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 !

* On	• On étiquette \circ dans $M \circ S$							
par	$\mathcal{C}(M)$	$)+\ell(S)$	5)					
$\ell(x) \ \ell(\uparrow) \ \ell(M \cdot N)$	$ \hat{=} 0 \hat{=} 0 \hat{=} \max(\ell) $	$(M) \ \ell(N))$	$\ell(1)\\\ell(\lambda M)$		$egin{array}{c} 0 \ \ell(M)+1 \end{array}$			
$\ell(id)$	$\hat{=}$ 0	$(\mathbf{N}, \mathbf{C}(\mathbf{N}))$	$\ell(M \circ N)$	Â	$\ell(M) + \ell(N)$			

	(<u>o</u> ')	
$1 \circ (N \cdot S) \to N$	$\uparrow \circ (N \cdot S) \to S$	
$(N \cdot S)$	$S' \rightarrow (N \circ S') \cdot (S \circ S')$	
$(\lambda M) \circ S \rightarrow \lambda (M \circ (1$	$\cdot (S \circ \uparrow)))$	
$(M \circ S)$	$\circ S' \to M \circ (S \circ S')$	
$\mathrm{id} \circ S \to S$	$S \circ \mathrm{id} \rightarrow S$	
$1 \cdot \uparrow \rightarrow id$	$(1 \circ S) \cdot (\uparrow \circ S) \rightarrow S$	

*	On étiquette	\circ dans $M \circ S$				(<u>o</u> ′)
	par $\ell(M) + \ell(S)$	5)		$1 \circ (N \cdot S) \rightarrow N$ $(N \cdot S) \circ S' -$	$\uparrow \circ (N \cdot S) \rightarrow S$ $\Rightarrow (N \circ S') \cdot (S \circ S')$	
l($egin{array}{cccc} \ell(x) & \hat{=} & 0 \ \ell(\uparrow) & \hat{=} & 0 \ (M \cdot N) & \hat{=} & \max(\ell(M), \ell(N)) \ \ell(id) & \hat{=} & 0 \end{array}$	$\begin{array}{rcl} \ell(1) & \stackrel{\circ}{=} & 0 \\ \ell(\lambda M) & \stackrel{\circ}{=} & \ell(M) + 1 \end{array}$ $\ell(M \circ N) & \stackrel{\circ}{=} & \ell(M) + \ell(N) \end{array}$		$(\lambda M) \circ S \to \lambda (M \circ (1 \cdot (S \circ (M \circ S) \circ S')))$ $(M \circ S) \circ S'$ $(M \circ S) \circ S'$ $1 \cdot \uparrow \to id$	$ \stackrel{()}{\to} M \circ (S \circ S') $ $ S \circ \text{id} \rightarrow S $ $ (1 \circ S) \cdot (\uparrow \circ S) \rightarrow S $	
					(o'')	
*	$1 \circ_{\max(n,s)} (N \cdot$	$S) \rightarrow N$		$\uparrow \circ_{\max(n,s)}$	$(N \cdot S) \to S$	
		$(N \cdot S) \circ_{\max(n,s)}$)+s	$S' S' \rightarrow (N \circ_{n+S'})$	$(S') \cdot (S \circ_{S+s})$	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+	$_{+s'}S' \longrightarrow M \circ_{m+s}$	$_{S+S'}(S \circ_{S+S'} S)$	5')
	id o C v C		C	So id S		

- * Fait. Si $M \rightarrow N$ en (σ') alors la version étiquetée [M] de Mse réécrit en celle, [N], de N en (σ'')
- Donc si (σ") termine,
 alors (σ'), donc aussi (σ) termine.

$$(\sigma')$$

$$1 \circ (N \cdot S) \rightarrow N \qquad \uparrow \circ (N \cdot S) \rightarrow S \\ (N \cdot S) \circ S' \rightarrow (N \circ S') \cdot (S \circ S') \\ (\lambda M) \circ S \rightarrow \lambda (M \circ (1 \cdot (S \circ \uparrow)))) \\ (M \circ S) \circ S' \rightarrow M \circ (S \circ S') \\ \text{id} \circ S \rightarrow S \qquad S \circ \text{id} \rightarrow S \\ 1 \cdot \uparrow \rightarrow \text{id} \qquad (1 \circ S) \cdot (\uparrow \circ S) \rightarrow S \end{cases}$$

$$(\sigma'')$$

$$\uparrow 0_{\max(n,s)} (N \cdot S) \to N \qquad \uparrow 0_{\max(n,s)} (N \cdot S) \to S \\ (N \cdot S) 0_{\max(n,s)+s'} S' \to (N \circ_{n+'s'} S') \cdot (S \circ_{s+s'} S') \\ (\lambda M) 0_{m+1+s} S \to \lambda (M \circ_{m+s} (1 \cdot (S \circ_s \uparrow))) \\ (M \circ_{m+s} S) 0_{m+s+s'} S' \to M \circ_{m+s+s'} (S \circ_{s+s'} S') \\ \text{id } 0_s S \to S \qquad S \circ_s \text{id} \to S \\ 1 \cdot \uparrow \to \text{id} \qquad (1 \circ_s S) \cdot (\uparrow \circ_s S) \to S$$

- * Fait. Si M → N en (σ')
 alors la version étiquetée [[M]] de l
 se réécrit en celle, [[N]], de N en (σ')
- Donc si (σ") termine,
 alors (σ'), donc aussi (σ) termine.

Faux!

$$(1 \circ (N \cdot S)) \circ S' \rightarrow N \circ S' \text{ mais}$$

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

$$\rightarrow \llbracket N \rrbracket \circ_{n+s'} \llbracket S' \rrbracket ???$$

 (σ'')

* 1 ◦_{max(n,s)} (N ⋅ S) → N (N ⋅ S) ◦_{max(n,s)+s'} S' → (N ◦_{n+'s'} S') ⋅ (S ◦_{s+s'} S') $(AM) ◦_{m+1+s} S → \lambda(M ◦_{m+s} (1 ⋅ (S ◦_s ↑))))$ $(M ◦_{m+s} S) ◦_{m+s+s'} S' → M ◦_{m+s+s'} (S ◦_{s+s'} S')$ $id ◦_s S → S$ $S ◦_s id → S$ 1 ⋅ ↑ → id $(1 ◦_s S) ⋅ (↑ ◦_s S) → S$

* Fait. Si $M \rightarrow N$ en (σ') Faux! alors la version étiquetée [M] de l $(1 \circ (N \cdot S)) \circ S' \rightarrow N \circ S'$ mais se réécrit en celle, [N], de N en (σ' $(1 \circ_{\max(n,s)} (\llbracket N \rrbracket \cdot \llbracket S \rrbracket)) \circ_{\max(n,s)+s'} \llbracket S' \rrbracket$ $\rightarrow [N] \circ_{n+s'} [S'] ???$ * Donc si (σ'') termine, $(1 \circ S) \cdot (\uparrow \circ S) \rightarrow S$ alors (σ'), donc aussi (σ) termine. ajouter: $M \circ_m S \to M \circ_n S$ si m > nsi $M \rightarrow N$ en (σ'), * $1 \circ_{\max(n,s)} (N \cdot S) \rightarrow N$ $\uparrow \circ_{max}($ alors $\ell(M) \ge \ell(N)$ $(N \cdot S) \circ_{\max(n,s)+s'} S' \rightarrow (N \in$ $(\lambda M) \circ_{m+1+s} S \longrightarrow \lambda(M \circ_{m+s} (1 \cdot (S \circ_s \uparrow)))$ $(M \circ_{m+s} S) \circ_{m+s+s'} S' \to M \circ_{m+s+s'} (S \circ_{s+s'} S')$ id $\circ_s S \to S$ $S \circ_{s} id \rightarrow S$ $(1 \circ_s S) \cdot (\uparrow \circ_s S) \rightarrow S$ $1 \cdot \uparrow \rightarrow \text{id}$



3/3: le lpo

* 1 ◦_{max(n,s)} (N ⋅ S) → N ↑ ◦_{max(n,s)} (N ⋅ S) → S
(N ⋅ S) ◦_{max(n,s)+s'} S' → (N ◦_{n+s'} S') ⋅ (S ◦_{s+s'} S')
(AM) ◦_{m+1+s} S → λ(M ◦_{m+s} (1 ⋅ (S ◦_s ↑)))
(M ◦_{m+s} S) ◦_{m+s+s'} S' → M ◦_{m+s+s'} (S ◦_{s+s'} S')
id ◦_s S → S S ◦_s id → S
1 ⋅ ↑ → id (1 ◦_s S) ⋅ (↑ ◦_s S) → S
M ◦_m S → M ◦_n S si m>n

3/3: le lpo

* 1 ◦_{max(n,s)} (N ⋅ S) → N $\uparrow ◦_{max(n,s)} (N ⋅ S) → S$ $(N ⋅ S) ◦_{max(n,s)+s'} S' → (N ◦_{n+s'} S') ⋅ (S ◦_{s+s'} S')$ $(\lambda M) ◦_{m+1+s} S → \lambda(M ◦_{m+s} (1 ⋅ (S ◦_s ↑))))$ $(M ◦_{m+s} S) ◦_{m+s+s'} S' → M ◦_{m+s+s'} (S ◦_{s+s'} S')$ $id ◦_s S → S$ 1 ⋅ ↑ → id $(1 ◦_s S) ⋅ (↑ ◦_s S) → S$ $M ◦_m S → M ◦_n S \text{ si } m > n$

* **Précédence:** ... > \circ_m > ... > \circ_2 > \circ_1 > \circ_0 > \cdot , λ , 1, \uparrow

>id

3/3: le lpo

* $1 \circ_{\max(n,s)} (N \cdot S) \rightarrow N$ $(N \cdot S) \circ_{\max(n,s)+s'} S' \rightarrow (N \circ_{n+s'} S') \cdot (S \circ_{s+s'} S')$ $(\lambda M) \circ_{m+1+s} S \rightarrow \lambda(\lambda' \text{ termine! } S \circ_s \uparrow)))$ $(M \circ_m \text{ Termine! } S \circ_s \uparrow)))$ $(M \circ_m M \circ_{m+s+s'} S' \rightarrow M \circ_{m+s+s'} (S \circ_{s+s'} S')$ $id \circ_s S \rightarrow S$ $1 \cdot \uparrow \rightarrow id$ $M \circ_m S \rightarrow M \circ_n S \text{ si } m > n$

* **Précédence:** ... > \circ_m > ... > \circ_2 > \circ_1 > \circ_0

 $> \cdot, \lambda, 1, \uparrow$ > 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. \Box

Propriétés de o

* **Thm.** σ termine.

Propriétés de o

- * **Thm.** σ termine.
- * Rappel: σ est localement confluente

Propriétés de o

- * **Thm.** σ termine.
- * Rappel: σ est localement confluente
- * Donc tout terme *M* a une σ -forme normale unique $\sigma(M)$

Propriétés de o

- * **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)

- * $M, N, \dots ::= x 1 \mid MN \mid \lambda M \mid M[S]$ $S, S', \dots ::= id \mid N \cdot S \mid \uparrow \mid S \circ S'$
- * Les piles σ -normales: $N_1 \cdot N_2 \cdot \ldots \cdot N_k \cdot \uparrow^m$

$(\beta) (\lambda M) N \rightarrow M[N \cdot id]$		(σ)
$.1[N \cdot S] \rightarrow N$	$\uparrow \circ (N \cdot S) \to S$	
$. (MN)[S] \rightarrow M[S](N[S])$	$(N \cdot S) \circ S' \to N[S'] \cdot (S \circ S')$	
$. \ (\lambda M)[S] \to \lambda(M[1 \cdot (S \circ \uparrow)])$		
$. M[S][S'] \to M[S \circ S']$	$(S \circ S') \circ S'' \to S''$	$S \circ (S' \circ S'')$
•	$\mathrm{id}\circ S\to S$	
$M[id] \rightarrow M$	$S \circ \mathrm{id} \rightarrow S$	

où $k, m \ge 0$, les N_i sont σ -normaux et $\uparrow^0 = id, \uparrow^1 = \uparrow, \uparrow^{m+1} = \uparrow \circ \uparrow^m$ pour $m \ge 1$

Formes σ-normales (closes)

- * $M, N, \dots ::= x 1 \mid MN \mid \lambda M \mid M[S]$ $S, S', \dots ::= id \mid N \cdot S \mid \uparrow \mid S \circ S'$
- * Les piles σ -normales: $N_1 \cdot N_2 \cdot \ldots \cdot N_k \cdot \uparrow^m$

$(\beta) (\lambda M) N \rightarrow M[N \cdot id]$		(σ)	
$. 1[N \cdot S] \rightarrow N$ $. (MN)[S] \rightarrow M[S](N[S])$	$\uparrow \circ (N \cdot S) \to S$ $(N \cdot S) \circ S' \to N$	$V[S'] \cdot (S \circ S')$	
$(\lambda M)[S] \rightarrow \lambda (M[1 \cdot (S \circ M[S][S'])) \rightarrow M[S \circ S']$	$\begin{split} &(\lambda M)[S] \to \lambda(M[1 \cdot (S \circ \uparrow)]) \\ &M[S][S'] \to M[S \circ S'] \qquad (S \circ S') \circ S'' \to S \circ (S' \circ S'') \end{split}$		
$M[id] \rightarrow M$	$id \circ S \to S$ $S \circ id \to S$		

où $k, m \ge 0$, les N_i sont σ -normaux et $\uparrow^0 = id, \uparrow^1 = \uparrow, \uparrow^{m+1} = \uparrow \circ \uparrow^m$ pour $m \ge 1$

* Les termes clos σ-normaux:

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

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

Formes σ-normales (closes)

- * $M, N, \dots ::= x 1 \mid MN \mid \lambda M \mid M[S]$ $S, S', \dots ::= id \mid N \cdot S \mid \uparrow \mid S \circ S'$
- * Les piles σ -normales: $N_1 \cdot N_2 \cdot \ldots \cdot N_k \cdot \uparrow^m$

$(\beta) \ (\lambda M) N \rightarrow M[N \cdot \mathrm{id}]$	(σ)	
$. 1[N \cdot S] \rightarrow N$ $. (MN)[S] \rightarrow M[S](N[S])$ $. (\lambda M)[S] \rightarrow \lambda (M[1 \cdot (S \circ \uparrow)])$	$\uparrow \circ (N \cdot S) \to S$ $(N \cdot S) \circ S' \to N$	$V[S'] \cdot (S \circ S')$
$M[S][S'] \rightarrow M[S \circ S']$. M[id] $\rightarrow M$	$(S \circ S') \circ S'' \rightarrow S$ id $\circ S \rightarrow S$ $S \circ id \rightarrow S$	$S \circ (S' \circ S'')$

où $k, m \ge 0$, les N_i sont σ -normaux et $\uparrow^0 = id, \uparrow^1 = \uparrow, \uparrow^{m+1} = \uparrow \circ \uparrow^m$ pour $m \ge 1$

* Les termes clos σ -normaux: $M, N, \dots ::= n \mid MN \mid \lambda M$ où $n=1[\uparrow^{n-1}]$ pour $n \ge 2$

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

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



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

$\underline{\lambda \to \lambda \sigma}$

* $x^*(\ell) = i \text{ si } x = x_i \text{ (avec } i \text{ minimal)}$... et $x[\uparrow^n]$ si pas dans ℓ $(uv)^*(\ell) = u^*(\ell) (v^*(\ell))$ $(\lambda x \cdot u)^*(\ell) = \lambda(u^*(x::\ell))$

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

Comme avec ue σ **Traduction vers λσ**

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

$\underline{\lambda \to \lambda \sigma}$

* $x^*(\ell) = i \text{ si } x = x_i \text{ (avec } i \text{ minimal)}$... et $x[\uparrow^n]$ si pas dans ℓ $(uv)^*(\ell) = u^*(\ell) (v^*(\ell))$ $(\lambda x \cdot u)^*(\ell) = \lambda(u^*(x::\ell))$ Pénible à démontrer! voir théorème 3 du poly machines.pdf

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

Comme avec uc **Traduction vers λσ**

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

$\underline{\lambda \to \lambda \sigma}$

* $x^*(\ell) = i \text{ si } x = x_i \text{ (avec } i \text{ minimal)}$... et $x[\uparrow^n]$ si pas dans ℓ $(uv)^*(\ell) = u^*(\ell) (v^*(\ell))$ $(\lambda x \cdot u)^*(\ell) = \lambda(u^*(x::\ell))$

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



$\underline{\lambda\sigma \to \lambda}$

* $x^{\circ}(\ell) = x$ $i^{\circ}(\ell) = u_i$ $(MN)^{\circ}(\ell) = M^{\circ}(\ell) (N^{\circ}(\ell))$ $(\lambda M)^{\circ}(\ell) = \lambda x . (M^{\circ}(x::\ell))$

 $(M[S])^{\circ}(\mathcal{E}) = M^{\circ}(S^{\circ}(\mathcal{E}))$



$\underline{\lambda\sigma \to \lambda}$

* $x^{\circ}(\ell) = x$ $i^{\circ}(\ell) = u_i$ $(MN)^{\circ}(\ell) = M^{\circ}(\ell) (N^{\circ}(\ell))$ $(\lambda M)^{\circ}(\ell) = \lambda x . (M^{\circ}(x::\ell))$

 $(M[S])^{\circ}(\mathscr{C}) = M^{\circ}(S^{\circ}(\mathscr{C}))$

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



 $\lambda \sigma \rightarrow \lambda$

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

 $(M[S])^{\circ}(\mathscr{C}) = M^{\circ}(S^{\circ}(\mathscr{C}))$

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



 $\underline{\lambda\sigma \to \lambda}$

* $x^{\circ}(\ell) = x$ $i^{\circ}(\ell) = u_i$ $(MN)^{\circ}(\ell) = M^{\circ}(\ell) (N^{\circ}(\ell))$ $(\lambda M)^{\circ}(\ell) = \lambda x . (M^{\circ}(x::\ell))$

 $(M[S])^{\circ}(\mathcal{C}) = M^{\circ}(S^{\circ}(\mathcal{C}))$

* $\operatorname{id}^{\circ}(\ell) = \ell$ $(N \cdot S)^{\circ}(\ell) = N^{\circ}(\ell)::S^{\circ}(\ell)$ $\uparrow^{\circ}([u_1; \ldots; u_n]) = [u_2; \ldots; u_n]$ $(S \circ S')^{\circ}(\ell) = S^{\circ}(S'^{\circ}(\ell))$

Traductions de et vers $\lambda \sigma$

$\underline{\lambda \to \lambda \sigma}$

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

* De plus,

 $\underline{\lambda\sigma \to \lambda}$

Traductions de et vers $\lambda \sigma$

$\lambda \rightarrow \lambda \sigma$

Prop. si $u \rightarrow u'$ alors $u^*(\ell) \rightarrow^+ u'^*(\ell)$ $\underline{\lambda\sigma \to \lambda}$

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

Traductions de et vers $\lambda \sigma$

$\lambda \rightarrow \lambda \sigma$

Prop. si $u \rightarrow u'$ alors $u^*(\ell) \rightarrow^+ u'^*(\ell)$ $\underline{\lambda\sigma \to \lambda}$

- De plus,
- * **Prop.** Pour tout λ -terme clos u, $(u^*(\ell))^{\circ}(\ell) = u$.
- *** Prop.** Pour tout λσ-terme clos M, M →* (M^o (ℓ))* (ℓ)
 ... en n'utilisant que les règles de (σ)

Traductions de et vers $\lambda \sigma$

$\lambda \rightarrow \lambda \sigma$

Prop. si $u \rightarrow u'$ alors $u^*(\ell) \rightarrow^+ u'^*(\ell)$ $\underline{\lambda\sigma \to \lambda}$

* **Prop.** si $M \to M'$ alors $M^{\circ}(\ell) \to M'^{\circ}(\ell)$

De plus,

 \sim

- * **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.

- * 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.JF

(clos) M * * M_1 M_2



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

<u>Côté λ-calcul</u>



*

 M_1

 $M^{o}(\ell)$ $M_2^{\rm o}(\ell)$ $M_1^{o}(\ell)$

* **Prop.** si $M \to M'$ alors $M^{\circ}(\ell) \to M'^{\circ}(\ell)$ pour $\ell = [x_1; ...; x_n],$ *n* assez grand

M



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



<u>Côté $\lambda\sigma$ -calcul</u>

(clos)

*

 M_2

 $M^{o}(\ell)$ * $M_2^{o}(\ell)$ $M_1^{o}(\ell)$ \mathcal{U} ... car le λ -calcul est confluent

<u>Côté λ-calcul</u>

* M1

M



https://diccan.com/Hebdo/h58/hhardin.JPC

<u>Côté λσ-calcul</u>

(clos)

*

 M_2





 $M^{o}(\ell)$ (clos) M **Prop.** Pour tout $\lambda \sigma$ -terme clos $M, M \rightarrow (M^{\circ}(\ell)) (\ell)$ * ... en n'utilisant que les règles de (σ) $M_1^{o}(\ell)$ $M_2^{o}(\ell)$ M_1 $IV1_2$ * 1* $(M_1^{o}(\ell))^*(\ell) \quad (M_2^{o}(\ell))^*(\ell)$ \mathcal{U} ... car le λ -calcul * est confluent $v^*(\ell)$ <u>Côté λ-calcul</u> <u>Côté λσ-calcul</u>



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

Confluence du $\lambda\sigma$ -calcul?

- * $M, N, \ldots ::= x \mid 1 \mid MN \mid \lambda M \mid M[S]$ $S, S', \ldots ::= id \mid N \cdot S \mid \uparrow \mid S \circ S' \mid X$
- * **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 λσ-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:

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

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

$$B \xrightarrow{\star} CB \xrightarrow{\star} X[\mathbf{1}[\mathbf{x} \circ (B \cdot s')] \cdot (\uparrow \circ (\mathbf{x} \circ (CB \cdot s')))]$$

$$\xrightarrow{\star} X[\mathbf{1}[\mathbf{x} \circ (CB \cdot s')] \cdot (\uparrow \circ (\mathbf{x} \circ (CB \cdot s')))] \to X[\mathbf{x} \circ (CB \cdot s')] =_{\Delta} A$$

and
$$B \xrightarrow{\star} CB \xrightarrow{\star} CA \text{ (since } B \xrightarrow{\star} A\text{).}$$

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









Types simples pour $\lambda \sigma$

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

$$\frac{\overline{F_1, \dots, F_n \vdash 1 : F_1}}{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F} (App) \frac{F, \Gamma \vdash M : G}{\Gamma \vdash \lambda M : F \Rightarrow G} (\lambda)$$












Les contextes de typage sont maintenant des **listes** de formules S simples pour $\lambda\sigma$ (clos)

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

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

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

$$\frac{\overline{\Gamma \vdash id:\Gamma}^{(id)}}{\overline{\Gamma \vdash N:F} \Gamma \vdash S:\Delta} (1)$$

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

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

(. 1)

 (Λ)

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



* Thm (autoréduction). Si $\Gamma \vdash M : F$ et $M \rightarrow N$ en $\lambda \sigma$ alors $\Gamma \vdash N : F$

$$\frac{\overline{F_{1}, \dots, F_{n} \vdash 1: F_{1}}(1)}{\Gamma \vdash M: F \Rightarrow G \quad \Gamma \vdash N: F \quad (App)} \frac{F, \Gamma \vdash M: G}{\Gamma \vdash \lambda M: F \Rightarrow G}(\lambda)$$

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

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

- * Thm (autoréduction). Si $\Gamma \vdash M : F$ et $M \rightarrow N$ en $\lambda \sigma$ alors $\Gamma \vdash N : F$
- * Thm (préservation du typage). Si $x_1:F_1, ..., x_n:F_n \vdash u : F$ en λ -calcul (types simples) alors $F_1, ..., F_n \vdash u^*([x_1, ..., x_n]) : F$ en $\lambda \sigma$

$$\frac{\overline{F_{1}, \dots, F_{n} \vdash 1 : F_{1}}(1)}{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F} (App) \frac{F, \Gamma \vdash M : G}{\Gamma \vdash \lambda M : F \Rightarrow G}(\lambda)$$

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

$$\frac{\overline{\Gamma} \vdash id:\Gamma}{\Gamma \vdash N:F} \stackrel{(id)}{\Gamma \vdash N:F} \frac{\overline{\Gamma} \vdash F}{\Gamma \vdash S:\Delta} \stackrel{(\uparrow)}{\Gamma \vdash N \cdot S:F,\Delta} \\
\frac{\Delta \vdash S':\Lambda \quad \Gamma \vdash S:\Delta}{\Gamma \vdash S:\Delta} \stackrel{(\circ)}{\Gamma \vdash S' \circ S:\Lambda} (\circ)$$

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

$$\frac{\overline{F_1, \dots, F_n \vdash 1 : F_1}}{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F} (1)}$$

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

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

$$\frac{\overline{\Gamma \vdash id:\Gamma}(id)}{\Gamma \vdash N:F \quad \Gamma \vdash S:\Delta} \stackrel{(\uparrow)}{\overline{\Gamma \vdash N:F}(\cdot)} \\
\frac{\overline{\Gamma \vdash N:F \quad \Gamma \vdash S:\Delta}(\cdot)}{\Gamma \vdash N\cdot S:F,\Delta} \stackrel{(\circ)}{\overline{\Gamma \vdash S':\Lambda}(\circ)}$$

Thm (terminaison). Si $\Gamma \vdash M : F$

alors M est fortement normalisable (en Ao)

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

* Pour toute pile *S*, soit:

```
\operatorname{rec}(S) = \uparrow \circ (1[S] \cdot \operatorname{id}) D_S(S') = 1[1[S] \cdot S'] \cdot S
et soit S_1 = (\lambda 1)1 \cdot \operatorname{id}
```

* On vérifie que: $S_1 \circ S \to D_S(S \circ rec(S))$



* Pour toute pile *S*, soit:

- * On vérifie que: $S_1 \circ S \rightarrow^+ D_S(S \circ \operatorname{rec}(S))$
- Soit C_S(S') = ↑ (1[S'] · S)
 On vérifie que: rec(S') S →⁺ C_S(S' S)

* Pour toute pile *S*, soit:

- * On vérifie que: $S_1 \circ S \rightarrow^+ D_S(S \circ rec(S))$
- * Soit $C_S(S') = \uparrow \circ (1[S'] \cdot S)$ On vérifie que: $\operatorname{rec}(S') \circ S \to^+ C_S(S' \circ S)$
- * Soit $S_{n+1} = \operatorname{rec}(S_n)$ On vérifie que: $S_n \circ S_{n+1} \rightarrow^+ C_{Sn+1} n^{-1} (D_{Sn+1}(S_{n+1} \circ S_{n+2}))$

* Pour toute pile *S*, soit:

- * On vérifie que: $S_1 \circ S \rightarrow^+ D_S(S \circ rec(S))$
- * Soit $C_S(S') = \uparrow \circ (1[S'] \cdot S)$ On vérifie que: $\operatorname{rec}(S') \circ S \to^+ C_S(S' \circ S)$
- * Soit $S_{n+1} = \operatorname{rec}(S_n)$ On vérifie que: $S_n \circ S_{n+1} \rightarrow^+ C_{Sn+1} n-1 (D_{Sn+1}(S_{n+1} \circ S_{n+2}))$
- * Donc $S_1 \circ S_1$ ne termine pas.

* Pour toute pile *S*, soit:

- * On vérifie que: $S_1 \circ S \rightarrow^+ D_S(S \circ \operatorname{rec}(S))$
- * Soit $C_S(S') = \uparrow \circ (1[S'] \cdot S)$ On vérifie que: $\operatorname{rec}(S') \circ S \to^+ C_S(S' \circ S)$
- * Soit $S_{n+1} = \operatorname{rec}(S_n)$ On vérifie que: $S_n \circ S_{n+1} \rightarrow^+ C_{Sn+1} n-1 (D_{Sn+1}(S_{n+1} \circ S_{n+2}))$
- * 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!

$$\frac{\overline{F_1, \dots, F_n \vdash 1 : F_1}}{\Gamma \vdash M : F \Rightarrow G \quad \Gamma \vdash N : F} (App) \frac{F, \Gamma \vdash M : G}{\Gamma \vdash \lambda M : F \Rightarrow G} (\lambda)$$
$$\frac{\Delta \vdash M : F \quad \Gamma \vdash S : \Delta}{\Gamma \vdash M[S] : F} (I)$$

$$\frac{\overline{\Gamma \vdash id:\Gamma}^{(id)}}{\overline{\Gamma \vdash N:F} \Gamma \vdash S:\Delta} \stackrel{(\uparrow)}{\overline{\Gamma \vdash N:F} \Gamma \vdash S:\Delta}{(\cdot)}$$

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

- * 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 configuration (M, S, args) où args=[N₁;...;N_k]
 représente M[S]N₁...N_k

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

Une configuration (M, S, args) où args=[N₁;...;N_k]
 représente M[S]N₁...N_k

	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	
1	NI C			77	: 1	
1	$, \qquad N \cdot S,$	args	\rightarrow	IV,	ia,	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	$, \qquad \uparrow,$	args	\rightarrow	n+1,	id,	args
$x[\uparrow^n]$	$, \qquad \uparrow,$	args	\rightarrow	$x[\uparrow^{n+1}],$	id,	args
n	$, \qquad \uparrow \circ S,$	args	\rightarrow	n+1,	S,	args
$x[\uparrow^n]$	$, \qquad \uparrow \circ S,$	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

$.1[N \cdot S] \rightarrow N$	$\uparrow \circ (N \cdot S) \to S$	
$. \ (MN)[S] \to M[S](N[S])$	$(N \cdot S) \circ S' \to N$	$V[S'] \cdot (S \circ S')$
. $(\lambda M)[S] \rightarrow \lambda (M[1 \cdot (S \circ \uparrow$)])	
$. M[S][S'] \to M[S \circ S']$	$(S \circ S') \circ S'' \to S''$	$S \circ (S' \circ S'')$
	$\mathrm{id} \circ S \to S$	
$M[id] \rightarrow M$	$S \circ \mathrm{id} \rightarrow S$	

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

- Une configuration (*M*, *S*, args) où args=[N₁;...;N_k] représente *M*[*S*]N₁...N_k
- * ... et l'on a les théorèmes de correction, et de progrès attendus:

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

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.