Examen, λ-calcul, 2019

Note. Comme promis, l'examen commence par une suite à une partie du DM.

Important. Soyez clair.

1 Un cas particulier terminant du λ-calcul algébrique

J'avais initialement imaginé vous demander de montrer un théorème de terminaison du λ-calcul algébrique très général. A la place, on va étudier un cas particulier, dont l'indication était déjà donnée par les exemples du DM.

On se donne l'algèbre de types :

$$\begin{aligned}\sigma, \tau, \cdots ::= &\ \mathtt{nat} \\ &\mid \mathtt{list}(\sigma) \\ &\mid \sigma \to \tau\end{aligned}$$

Les termes, ainsi que leurs règles de typage, sont donnés par :

$$\frac{}{\vdash x_\sigma : \sigma} \qquad \frac{\vdash u : \sigma \to \tau \quad \vdash v : \sigma}{\vdash uv : \tau} \qquad \frac{\vdash u : \tau}{\vdash \lambda x_\sigma . u : \sigma \to \tau}$$

$$\frac{}{\vdash 0 : \mathtt{nat}} \qquad \frac{\vdash u : \mathtt{nat}}{\vdash \mathtt{s}(u) : \mathtt{nat}}$$

$$\frac{\vdash u : \sigma \quad \vdash v : \mathtt{nat} \to \sigma \to \sigma \quad \vdash w : \mathtt{nat}}{\vdash R_\sigma(u, v, w) : \sigma}$$

$$\frac{}{\vdash \mathtt{nil}_\sigma : \mathtt{list}(\sigma)} \qquad \frac{\vdash u : \sigma \quad \vdash v : \mathtt{list}(\sigma)}{\vdash \mathtt{cons}_\sigma(u, v) : \mathtt{list}(\sigma)}$$

$$\frac{\vdash u : \tau \quad \vdash v : \sigma \to \mathtt{list}(\sigma) \to \tau \to \tau \quad \vdash w : \mathtt{list}(\sigma)}{\vdash M_{\sigma,\tau}(u, v, w) : \tau}$$

On rappelle que tout terme typable a un unique type. Ceci a été démontré dans le DM. On dira simplement *terme* plutôt que terme typable, sauf ambiguïté. Il y a une infinité dénombrable de variables de chaque type.

Les règles de réduction sont :

$$\begin{aligned}
(\beta) \quad & (\lambda x_\sigma . u)v \to u[x_\sigma := v] \\
(R0) \quad & R_\sigma(u, v, 0) \to u \\
(R\mathtt{s}) \quad & R_\sigma(u, v, \mathtt{s}(w)) \to vw(R_\sigma(u, v, w)) \\
(M\mathtt{nil}) \quad & M_{\sigma,\tau}(u, v, \mathtt{nil}_\sigma) \to u \\
(M\mathtt{cons}) \quad & M_{\sigma,\tau}(u, v, \mathtt{cons}_\sigma(s, t)) \to vst(M_{\sigma,\tau}(u, v, t))
\end{aligned}$$

Ce calcul vérifie la propriété d'auto-réduction. Le but de l'exercice est de démontrer que ce calcul termine.

Comme dans le λ-calcul (mais le langage a changé), on définit une notion de candidat. Cependant, nos candidats vont être typés. On définit donc un *candidat de type* τ comme étant un ensemble de termes S *de type* τ satisfaisant :

CR1 $S \subseteq SN_\tau$, où SN_τ est l'ensemble des termes fortement normalisables de type τ ;

CR2 pour tout $u \in S$, tout réduit en une étape u' de u est dans S ;

CR3 tout terme u de type τ qui est neutre et dont tous les réduits en une étape u' sont dans S, est lui-même dans S. Un terme est *neutre* si et seulement s'il n'est pas une λ-abstraction.

Pour tout ensemble S de termes d'un même type σ, pour tout ensemble S' de termes d'un même type τ, on définit $S \Rightarrow S'$ comme étant l'ensemble des termes u de type $\sigma \to \tau$ tels que pour tout $v \in S$, uv est dans S'. On notera que uv est toujours de type τ dans ce cas.

1. Montrer que pour tous candidats S de type σ et S' de type τ, $S \Rightarrow S'$ est un candidat de type $\sigma \to \tau$. Noter qu'on ne peut pas simplement utiliser des résultats du cours : le langage a changé, et la définition de $S \Rightarrow S'$ a changé aussi.

On définit maintenant :

$$\begin{aligned}
RED_{\mathtt{nat}} &= SN_{\mathtt{nat}} \\
RED_{\mathtt{list}(\sigma)} &= SN_{\mathtt{list}(\sigma)} \\
RED_{\sigma\to\tau} &= RED_\sigma \Rightarrow RED_\tau .
\end{aligned}$$

Alors RED_τ est un candidat de type τ, pour tout type τ.

2. Montrer que pour tout type σ, pour tout $u \in RED_\sigma$, pour tout $v \in RED_{\mathtt{nat}\to\sigma\to\sigma}$, pour tout $w \in RED_{\mathtt{nat}}$, $R_\sigma(u, v, w)$ est dans RED_σ.

3. Montrer que pour tous types σ et τ, pour tout $u \in RED_\tau$, pour tout $v \in RED_{\sigma \to \mathtt{list}(\sigma) \to \tau \to \tau}$, pour tout $w \in RED_{\mathtt{list}(\sigma)}$, $M_{\sigma,\tau}(u, v, w)$ est dans RED_τ. Il est recommandé de se contenter de dire ce qui change par rapport à la question précédente, sans tout redécrire en détail.

On admet le lemme suivant, similaire à un lemme vu en cours : pour tous types σ et τ, pour tout terme u de type τ, si pour tout $v \in RED_\sigma$ on a $u[x_\sigma := v] \in RED_\tau$, alors $\lambda x_\sigma.u \in RED_{\sigma \to \tau}$.

On dira qu'une subsitution $\theta = [x_{1\tau_1} := v_1, \cdots, x_{n\tau_n} := v_n]$ (où pour tout i, le type de v_i vaut τ_i) est *réductible* si et seulement $v_i \in RED_{\tau_i}$ pour tout i, $1 \leq i \leq n$.

4. Montrer que pour tout terme u de type τ, pour toute substitution réductible θ, $u\theta$ est dans RED_τ, pour tout type τ. En déduire que tout terme (typable) termine.

On souhaite maintenant ajouter une nouvelle fonction $\mathtt{map}_{\sigma,\tau}$ au langage, avec les règles de typage et de réduction suivantes.

$$\frac{\vdash u : \sigma \to \tau \qquad \vdash v : \mathtt{list}(\sigma)}{\vdash \mathtt{map}_{\sigma,\tau}(u, v) : \mathtt{list}(\tau)}$$

$$\begin{array}{rl}
(\mathtt{mapnil}) & \mathtt{map}_{\sigma,\tau}(u, \mathtt{nil}_\sigma) \to \mathtt{nil}_\tau \\
(\mathtt{mapcons}) & \mathtt{map}_{\sigma,\tau}(u, \mathtt{cons}_\sigma(s, t)) \to \mathtt{cons}_\tau(us, \mathtt{map}_{\sigma,\tau}(u, t)).
\end{array}$$

Le calcul obtenu en ajoutant ces nouvelles fonctionnalités sera simplement appelé le *nouveau calcul*. Le calcul sans ces fonctionnalités sera appelé l'*ancien calcul*.

5. Les termes (typables) du nouveau calcul terminent-ils toujours ? Justifier.

2 Le $\lambda\upsilon$-calcul

On étudie un autre λ-calcul à substitutions explicites, le *$\lambda\upsilon$-calcul*. Les *expressions* du $\lambda\upsilon$-calcul sont définies par la grammaire :

			Termes
$M, N, P, \ldots$	$::=$	x	Variables
	$\mid$	$\underline{n}$	$(n \in \mathbb{N}, n \geq 1)$ Indices de de Bruijn 1, 2, 3, ...
	$\mid$	MM	Applications
	$\mid$	λM	Abstractions
	$\mid$	$M[S]$	Application de substitution
			Piles (substitutions explicites)
S	$::=$	$\uparrow$	Shift
	$\mid$	$\Uparrow S$	Lift
	$\mid$	$M/$	Slash

$M/$ (« M slash ») représente la même chose que $M \cdot \mathrm{id}$ en $\lambda\sigma$-calcul, et $\Uparrow S$ (« lift S ») représente la même chose que $1 \cdot (M \circ \uparrow)$ en $\lambda\sigma$-calcul.

(β)	$(\lambda M)N \to M[N/]$		
$(Var/)$	$\underline{1}[M/] \to M$	$(Var \Uparrow)$	$\underline{1}[\Uparrow S] \to \underline{1}$
$(\uparrow /)$	$\underline{n+1}[N/] \to \underline{n}$	$(\uparrow\Uparrow)$	$\underline{n+1}[\Uparrow S] \to \underline{n}[S][\uparrow]$
(App)	$(MN)[S] \to M[S](N[S])$	(Lam)	$(\lambda M)[S] \to \lambda(M[\Uparrow S])$
$(Var \uparrow)$	$\underline{n}[\uparrow] \to \underline{n+1}$		

FIGURE 1 – Les règles de réduction de $\lambda\upsilon$ $(n \geq 1)$

Les règles de réduction du $\lambda\upsilon$-calcul sont données en figure 1. Le système formé de toutes les règles sauf (β) s'appelle υ.

La traduction d'un λ-terme u en un $\lambda\upsilon$-terme $u^\bullet(\ell)$ (dans le contexte de la liste de variables ℓ, comme pour le $\lambda\sigma$-calcul étudié dans le cours) est définie comme suit. (Contrairement au $\lambda\sigma$-calcul, le terme $\underline{n}$ n'est *pas* une abréviation.)

$$
\begin{array}{rcll}
x^\bullet(\ell) & \hat{=} & \underline{n} & \text{si } n \text{ est le premier entier tel que } x = \ell(n) \\
 & & x\underbrace{[\uparrow][\uparrow]\ldots[\uparrow]}_{k \text{ fois}} & \text{si } x \text{ n'est pas dans } \ell,\ k \text{ est la longueur de } \ell \\
(uv)^\bullet(\ell) & \hat{=} & u^\bullet(\ell)v^\bullet(\ell) & \\
(\lambda x \cdot u)^\bullet(\ell) & \hat{=} & \lambda(u^\bullet(x :: \ell)) &
\end{array}
$$

(La liste ℓ de longueur k est par convention la suite $\ell(1)$, $\ell(2)$, …, $\ell(k)$, où $\ell(i)$ dénote donc le $i$ième élément de la liste.)

6. Montrer, en considérant le $\lambda\upsilon$-terme $(\lambda x)N[\uparrow]$, où x est une variable et N est un terme normal que l'on choisira, que le $\lambda\upsilon$-calcul n'est *pas* confluent.

Le reste de l'exercice consiste à montrer que le $\lambda\upsilon$-calcul restreint aux termes clos est confluent, lui.

7. Démontrer que toute pile du $\lambda\upsilon$-calcul est de la forme $\Uparrow^p (M/)$ pour un certain terme M, ou bien $\Uparrow^p (\uparrow)$, $p \in \mathbb{N}$.
8. On définit deux mesures des $\lambda\upsilon$-termes comme suit :

$$\begin{array}{rrr}
\mu_1(\underline{n}) = 2^n & \mu_1(MN) = \mu_1(M) + \mu_1(N) & \mu_1(x) = 2 \\
\mu_1(\lambda M) = \mu_1(M) + 1 & \mu_1(M[S]) = \mu_1(M)\mu_1(S) & \\
\mu_1(\uparrow) = 2 & \mu_1(\Uparrow S) = \mu_1(S) & \mu_1(M/) = \mu_1(M)
\end{array}$$

$$\begin{array}{rrr}
\mu_2(\underline{n}) = 2^n & \mu_2(MN) = \mu_2(M) + \mu_2(N) + 1 & \mu_2(x) = 2 \\
\mu_2(\lambda M) = \mu_2(M) + 1 & \mu_2(M[S]) = \mu_2(M)(\mu_2(S) + 1) & \\
\mu_2(\uparrow) = 2 & \mu_2(\Uparrow S) = 2\mu_2(S) + 1 & \mu_2(M/) = \mu_2(M)
\end{array}$$

En utilisant μ_1 et μ_2, démontrer que υ termine (normalise fortement).

9. On admet que υ est localement confluent. Pourquoi υ est-il confluent sur les termes clos ? Citer le nom du théorème que vous utilisez, le cas échéant.

On admettra que si $u \to v$ en λ-calcul, alors pour toute liste ℓ de variables contenant les variables libres de u, $u^\bullet(\ell) \to^+ v^\bullet(\ell)$.

On définit une traduction inverse par :

$$\begin{array}{rl}
(\underline{n})^\circ(u_1 :: \ldots :: u_n :: P) = u_n & \uparrow^\circ (u :: P) = P \\
(MN)^\circ(P) = M^\circ(P)N^\circ(P) & (\Uparrow S)^\circ(u :: P) = u :: S^\circ(P) \\
(\lambda M)^\circ(P) = \lambda x \cdot M^\circ(x :: P) & (M/)^\circ(P) = M^\circ(P) :: P \\
(M[S])^\circ(P) = M^\circ(S^\circ(P)), &
\end{array}$$

où P est une liste suffisamment longue de λ-termes, et x est une variable fraîche dans le cas de λM.

On admettra que si P est de longueur au moins $d(M)$, alors $M^\circ(P)$ est défini.

10. Par *suffisamment longue*, on entend précisément que $M^\circ(P)$ est défini pour toute liste P de longueur au moins $d(M)$, où $d(M)$ est défini par récurrence sur M par :

$$\begin{array}{rl}
d(\underline{n}) = ? & \\
d(MN) = ? & d(\lambda M) = ? \\
d(M[\Uparrow^p (N/)]) = ? & d(M[\Uparrow^p \uparrow]) = ?
\end{array}$$

Remplacer les points d'interrogation. On demande les meilleures valeurs possibles.

On admettra que si M se réduit en N en $\lambda\upsilon$-calcul, alors $d(M) \geq d(N)$. Pour toute liste P de longueur au moins $d(M)$, $M^\circ(P)$, et si $M \to N$, alors (comme $d(N) \leq d(M)$), $N^\circ(P)$ est aussi défini. On admettra qu'on a alors aussi $M^\circ(P) \to^* N^\circ(P)$ et que de plus, si la règle utilisée n'est pas (β), alors $M^\circ(P) = N^\circ(P)$.

Un $\lambda\upsilon$-terme est *clos* si et seulement s'il ne contient pas de variable libre.

11. Montrer que les $\lambda\upsilon$-termes clos υ-normaux ne contiennent pas d'occurrence de l'opérateur $\_[\_]$.
12. Montrer que pour tout $\lambda\upsilon$-terme M clos υ-normal, pour toute liste ℓ de variables distinctes deux à deux de longueur au moins $d(M)$, $(M^\circ(\ell))^\bullet(\ell) = M$.
13. Montrer que pour tout $\lambda\upsilon$-terme M clos, pour toute liste ℓ de variables distinctes deux à deux de longueur au moins $d(M)$, M se réécrit par υ en $(M^\circ(\ell))^\bullet(\ell)$.
14. Montrer que le $\lambda\upsilon$-calcul est confluent *sur les termes clos*. Autrement dit, si $M \to^* N_1$ et $M \to^* N_2$ en $\lambda\upsilon$-calcul, où M est clos, alors il existe un $\lambda\upsilon$-terme (clos) P tel que $N_1 \to^* P$ et $N_2 \to^* P$.