# Le $\lambda\upsilon$-calcul (suite du sujet de 2001).

## *Correction.*

Documents autorisés (en particulier le poly).

Les questions sont annotées par un niveau de difficulté, variant de $(0)$ (facile) à $(3)$ (difficile). On notera bien que les questions les plus difficiles sont aussi celles qui rapportent le plus de points. Un candidat qui ne répondrait qu'au questions évaluées $(0)$ ou $(1)$, même à la perfection, n'aura ainsi (probablement) aucune chance d'atteindre la moyenne.

Le but de ce problème est d'étudier le $\lambda\upsilon$-calcul, un $\lambda$-calcul à substitutions explicites proposé par J. Rouyer-Degli et P. Lescanne en 1994. Note : la version exposée ici a une infinité de constantes $\underline{1}$, $\underline{2}$, $\underline{3}$, etc., pour dénoter les variables liées, alors que l'examen de 2001 utilisait une seule constante 1, et codait 2 comme $1[\uparrow]$, 3 comme $1[\uparrow][\uparrow]$, etc.

Comme dans le $\lambda\sigma$-calcul, les *expressions* du $\lambda\upsilon$-calcul sont ceux définis par la grammaire :

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

Les changements par rapport au $\lambda\sigma$-calcul sont : 1. on a un opérateur explicite $\Uparrow$, et $\Uparrow S$ correspond à la pile $1 \cdot (S \circ \uparrow)$ du $\lambda\sigma$-calcul ; 2. on a un nouvel opérateur de pile $/$, et $M/$ correspond à la pile $M \cdot id$ du $\lambda\sigma$-calcul ; 3. on n'a pas les constructions $M \cdot S$ et $S \circ S'$ du $\lambda\sigma$-calcul ; 4. on a une infinité d'indices de de Bruijn $\underline{1}$, $\underline{2}$, $\underline{3}$, etc. De plus, on n'a pas de variable de pile.

La traduction d'un $\lambda$-terme $u$ en un $\lambda\upsilon$-terme $u^\bullet(\ell)$ (dans le contexte de la liste de variables $\ell$, 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.)

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

(On rappelle que la liste $\ell$ de longueur $k$ est par convention la suite $\ell(1), \ell(2), \ldots, \ell(k)$, où $\ell(i)$ dénote donc le $i$ième élément de la liste.)

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

FIG. 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 $(\beta)$ s'appelle $\upsilon$.

### I. Réduction.

1. (2) Montrer que, pour tout $\lambda$-terme $v$, pour toute liste $\ell$ de variables, pour tout entier $k$, pour toutes variables $y_1, \ldots, y_p$, et pour toute variable $z$ non libre dans $v$, si toutes les variables libres de $v$ sont dans $\ell$ ou parmi $y_1, \ldots, y_p$, alors

$$v^\bullet(y_1 :: \ldots :: y_p :: \ell)[\Uparrow^p\uparrow] \to_\upsilon^* v^\bullet(y_1 :: \ldots :: y_p :: z :: \ell)$$

où $\Uparrow^p S$ dénote $\underbrace{\Uparrow \ldots \Uparrow}_{p \text{ fois}} S$.

Important : dans cette question et toutes les autres, on donnera systématiquement les noms des règles employées à chaque étape de chaque réduction.

*(Note : il fallait évidemment lire "pour tout entier $p$" et non "pour tout entier $k$".)*
*Par récurrence structurelle sur $v$. Si $v = y_i$ ($i$ étant minimal tel que $v = y_i$, $1 \leq i \leq p$), alors le côté gauche vaut $\underline{i}[\Uparrow^p\uparrow]$, qui se réduit en $\underline{i-1}[\Uparrow^{p-1}\uparrow][\uparrow]$, puis en ..., puis en $\underline{1}[\Uparrow^{p-i+1}\uparrow]\underbrace{[\uparrow]\ldots[\uparrow]}_{i-1}$ par $(\uparrow\Uparrow)$, puis en $\underline{1}\underbrace{[\uparrow]\ldots[\uparrow]}_{i-1}$ par $(Var \Uparrow)$, puis en $\underline{i}$ par $(Var \uparrow)$. Et ce dernier est exactement le côté droit.*
*Si $v$ est une variable $x$ différente des $y_i$, alors par hypothèse $x$ apparait dans $\ell$, disons $x = \ell(n)$ avec $n$ minimal, alors le côté gauche est $\underline{p+n}[\Uparrow^p\uparrow]$, qui se réduit en $\underline{n}[\uparrow]\underbrace{[\uparrow]\ldots[\uparrow]}_{p}$ par $(\uparrow\Uparrow)$, puis en $\underline{n+p+1}$ par $(Var \uparrow)$. Mais c'est encore exactement le côté droit, car $z$ n'est pas $x$, puisque $z$ n'est pas libre dans $v$ par hypothèse.*
*Le cas où $v$ est une application est un appel direct à l'hypothèse de récurrence. Si $v$ est une $\lambda$-abstraction $\lambda x \cdot w$, on a*

$$\begin{array}{rcll} v^\bullet(y_1 :: \ldots :: y_p :: \ell)[\Uparrow^p\uparrow] & = & \lambda(w^\bullet(x :: y_1 :: \ldots :: y_p :: \ell))[\Uparrow^p\uparrow] & \\ & \to_\upsilon & \lambda(w^\bullet(x :: y_1 :: \ldots :: y_p :: \ell)[\Uparrow^{p+1}\uparrow]) & \textit{(par } (Lam)\textit{)} \\ & \to_\upsilon^* & \lambda(w^\bullet(x :: y_1 :: \ldots :: y_p :: z' :: \ell)) & \\ & = & v^\bullet(y_1 :: \ldots :: y_p :: z' :: \ell) = v^\bullet(y_1 :: \ldots :: y_p :: z :: \ell) & \end{array}$$

*par hypothèse de récurrence, à condition que $z'$ ne soit pas libre dans $w$. On déduit la dernière égalité en remarquant que $v^\bullet(y_1 :: \ldots :: y_p :: z' :: \ell)$ est indépendant de la variable $z'$ du moment que $z'$ n'est pas libre dans $v$ ; on peut donc prendre $z' = z$, où $z$ est n'importe quelle variable non libre dans $v$.*

2. (0) En déduire que, pour tout $\lambda$-terme $v$, pour toute liste $\ell$ de variables contenant toutes les variables libres de $v$, pour tout entier $k$, pour toutes variables $x_1, \ldots, x_k$ non libres dans $v$ :

$$v^\bullet(\ell)\underbrace{[\uparrow]\ldots[\uparrow]}_{k \text{ fois}} \to_v^* v^\bullet(x_1 :: \ldots :: x_k :: \ell)$$

*Le cas $p = 0$ de la question précédente donne $v^\bullet(\ell)[\uparrow] \to_v^* v^\bullet(z :: \ell)$. On itère ensuite $k$ fois cette égalité avec $z = x_k$, ..., $z = x_1$.*

3. (2) Montrer que, pour tous $\lambda$-termes $u$ et $v$, pour toute liste de variables $\ell$ contenant toutes les variables libres de $u$ et de $v$, pour toute variable $x$, pour tout entier $k$, pour toutes variables $x_1, \ldots, x_k$ distinctes de $x$ et non libres dans $v$ :

$$u^\bullet(x_1 :: \ldots :: x_k :: x :: \ell)[\Uparrow^k v^\bullet(\ell)/] \to_v^* (u[x := v])^\bullet(x_1 :: \ldots :: x_k :: \ell)$$

où $\Uparrow^k S$ dénote $\underbrace{\Uparrow \ldots \Uparrow}_{k \text{ fois}} S$.

*Note : l'énoncé, bien que juste, n'était pas démontrable directement par récurrence. En fait, la récurrence ne passe pas lorsque $u$ est de la forme $\lambda x_0 \cdot u_0$. Il fallait donc généraliser le résultat à démontrer en :*

> *pour tous $\lambda$-termes $u$ et $v$, pour toute liste de variables $\ell$ contenant toutes les variables libres de $v$, pour toute variable $x$, pour tout entier $k$, pour toutes variables $x_1$, $\ldots$, $x_k$ distinctes de $x$ et non libres dans $v$, si toutes les variables libres de $u$ sont parmi $x_1, \ldots, x_k$, $x$ ou dans $\ell$ :*
>
> $$u^\bullet(x_1 :: \ldots :: x_k :: x :: \ell)[\Uparrow^k v^\bullet(\ell)/] \to_v^* (u[x := v])^\bullet(x_1 :: \ldots :: x_k :: \ell)$$

*Ceci se démontre par récurrence structurelle sur $u$. Si $u = x_i$ (avec $i$ minimal, $1 \leq i \leq k$), le côté gauche est $\underline{i}[\Uparrow^k v^\bullet(\ell)/]$, qui se réduit en $\underline{1}[\Uparrow^{k-i+1} v^\bullet(\ell)/]\underbrace{[\uparrow]\ldots[\uparrow]}_{i-1}$ par $(\uparrow\Uparrow)$, puis en $\underline{1}\underbrace{[\uparrow]\ldots[\uparrow]}_{i-1}$ par $(Var \Uparrow)$, puis en $\underline{i}$ par $(Var \uparrow)$. Or le côté droit est $x_i^\bullet(x_1 :: \ldots :: x_k :: \ell)$, puisque $x_i \neq x$, c'est-à-dire $\underline{i}$ de nouveau.*
*Si $u = x$, le côté gauche est $\underline{k+1}[\Uparrow^k v^\bullet(\ell)/]$, qui se réduit par $k$ applications de $(\uparrow\Uparrow)$ en $\underline{1}[v^\bullet(\ell)/]\underbrace{[\uparrow]\ldots[\uparrow]}_{k}$, puis en $v^\bullet(\ell)\underbrace{[\uparrow]\ldots[\uparrow]}_{k}$ par $(Var/)$, puis en $v^\bullet(x_1 :: \ldots :: x_k :: \ell)$ par la question précédente. Mais c'est le côté droit, puisque $x_i \neq x$ pour tout $i$.*
*Si $u$ est une variable $z$ différente de $x$, $x_1, \ldots, x_k$, donc dans $\ell$, disons $z = \ell(i)$ avec $i$ minimal, alors le côté gauche est $\underline{k+1+i}[\Uparrow^k v^\bullet(\ell)/]$ qui se réduit par $(\uparrow\Uparrow)$ en $\underline{1+i}[v^\bullet(\ell)/]\underbrace{[\uparrow]\ldots[\uparrow]}_{k}$, puis en $\underline{i}\underbrace{[\uparrow]\ldots[\uparrow]}_{k}$ par $(\uparrow /)$, puis en $\underline{i+k}$ par $(Var \uparrow)$. Encore une fois, ceci est le côté droit.*
*Les cas où $u$ est une application ou une $\lambda$-abstraction sont des appels directs à l'hypothèse de récurrence. Ce dernier cas, $u = \lambda x_0 \cdot u_0$, se traite en remarquant que*

$$\begin{array}{ll} & u^\bullet(x_1 :: \ldots :: x_k :: x :: \ell)[\Uparrow^k v^\bullet(\ell)/] \\ = & \lambda(u_0^\bullet(x_0 :: x_1 :: \ldots :: x_k :: x :: \ell))[\Uparrow^k v^\bullet(\ell)/] \\ \to_v & \lambda(u_0^\bullet(x_0 :: x_1 :: \ldots :: x_k :: x :: \ell)[\Uparrow^{k+1} v^\bullet(\ell)/] \end{array}$$

*par $(Lam)$, et l'on peut maintenant appliquer l'hypothèse de récurrence, alors même que $x_0$ peut être libre dans $u_0$ ; c'est là où l'énoncé avait besoin d'être généralisé.*

4. (1) En déduire que si $u \to v$ en $\lambda$-calcul, alors pour toute liste $\ell$ de variables contenant les variables libres de $u$, $u^\bullet(\ell) \to^+ v^\bullet(\ell)$.

*Par récurrence sur la profondeur $d$ du rédex contracté dans $u$. Si $d = 0$, alors $u = (\lambda x \cdot u_1)v_1$, $v = u_1[x := v_1]$, et $u^\bullet(\ell) = (\lambda u_1^\bullet(x :: \ell))v_1^\bullet(\ell) \to u_1^\bullet(x :: \ell)[v_1^\bullet(\ell)/]$ (par $(\beta)$), qui se réduit en $v^\bullet(\ell)$ par le cas $k = 0$ de la question précédente.*
*Si $d \geq 1$, c'est un appel direct à l'hypothèse de récurrence.*

## II. Confluence et terminaison de $\upsilon$.

1. (1) On définit l'interprétation suivante $\mu_1$ des $\lambda\upsilon$-termes (et piles) vers les entiers naturels supérieurs ou égaux à 2 :

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

Montrer que si $M$ se réécrit en une étape vers $N$ par $\upsilon$, alors $\mu_1(M) \geq \mu_1(N)$. Montrer de plus que l'inégalité est stricte, c'est-à-dire que $\mu_1(M) > \mu_1(N)$, dès que la règle utilisée n'est ni $(App)$ ni $(\uparrow\Uparrow)$.

*Note : encore une fois l'énoncé contenait une erreur mineure. Il fallait lire "dès que la règle utilisée n'est ni $(App)$ ni $(\uparrow\Uparrow)$ ni $(Var\uparrow)$".*
*Montrons déjà que $\mu_1(l) \geq \mu_1(r)$, ou $\mu_1(l) > \mu_1(r)$, pour toute règle $l \to r$ :*

| | $l \to r$ | $\mu_1(l)$ | $\mu_1(r)$ | $>$ / $\geq$ | *Commentaire* |
|---|---|---|---|---|---|
| $(Var/)$ | $\underline{1}[M/] \to M$ | $2\mu_1(M)$ | $\mu_1(M)$ | $>$ | *car* $\mu_1(M) \geq 1$ |
| $(Var\Uparrow)$ | $\underline{1}[\Uparrow S] \to \underline{1}$ | $2\mu_1(M)$ | $2$ | $>$ | *car* $\mu_1(S) \geq 2$ |
| $(\uparrow/)$ | $\underline{n+1}[N/] \to \underline{n}$ | $2^{n+1}\mu_1(N)$ | $2^n$ | $>$ | *car* $\mu_1(N) \geq 1$ |
| $(\uparrow\Uparrow)$ | $\underline{n+1}[\Uparrow S] \to \underline{n}[S][\uparrow]$ | $2^{n+1}\mu_1(S)$ | $2^{n+1}\mu_1(S)$ | $=$ | |
| $(App)$ | $(MN)[S] \to M[S](N[S])$ | $(\mu_1(M) + \mu_1(N))\mu_1(S)$ | $\mu_1(M)\mu_1(S)+$ $\mu_1(N)\mu_1(S)$ | $=$ | |
| $(Lam)$ | $(\lambda M)[S] \to \lambda(M[\Uparrow S])$ | $(\mu_1(M) + 1)\mu_1(S)$ | $\mu_1(M)\mu_1(S) + 1$ | $>$ | *car* $\mu_1(S) \geq 2$ |
| $(Var\uparrow)$ | $\underline{n}[\uparrow] \to \underline{n+1}$ | $2.2^n$ | $2^{n+1}$ | $=$ | |

*On montre ensuite par récurrence sur la profondeur du $\upsilon$-rédex contracté dans $M \to_\upsilon N$ que $\mu_1(M) > \mu_1(N)$ ; nous venons de traiter le cas où cette profondeur est 0, et les cas de récurrence sont immédiats.*

2. (2) De façon similaire, trouver une interprétation $\mu_2$ des $\lambda\upsilon$-termes (et piles) vers les entiers naturels supérieurs ou égaux à 1 telle que, pour tous termes ou piles $M$ et $N$, si $M$ se réécrit en une étape vers $N$ par l'une des règles $(App)$ ou $(\uparrow\Uparrow)$, alors $\mu_2(M) > \mu_2(N)$.

*Note : encore une fois, il y avait une petite erreur et il fallait lire "par l'une des règles $(App)$ ou $(\uparrow\Uparrow)$ ou $(Var\uparrow)$".*
*Prenons par exemple :*

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

| | $l \to r$ | $\mu_2(l)$ | $\mu_2(r)$ | $>$ / $\geq$ |
|---|---|---|---|---|
| $(\uparrow\Uparrow)$ | $\underline{n+1}[\Uparrow S] \to \underline{n}[S][\uparrow]$ | $2^{n+1}(2\mu_2(S)+2)$ | $3.2^n(\mu_2(S)+1)$ | $>$ |
| $(App)$ | $(MN)[S] \to M[S](N[S])$ | $(\mu_2(M)+\mu_2(N)+1)\times$ $(\mu_2(S)+1)$ | $\mu_2(M)(\mu_2(S)+1)+$ $\mu_2(N)(\mu_2(S)+1)+1$ | $>$ |
| $(Var\uparrow)$ | $\underline{n}[\uparrow] \to \underline{n+1}$ | $3.2^n$ | $2^{n+1}$ | $>$ |

*Encore une fois, il est facile de voir que cette mesure passe au contexte, c'est-à-dire que si $M \to N$ par l'une des trois règles ci-dessus, alors $\mu_2(M) > \mu_2(N)$.*

3. (1) En déduire que $\upsilon$ termine.

   *Posons $\mu(M) = (\mu_1(M), \mu_2(M))$, ordonné par l'ordre lexicographique sur $\mathbb{N}^2$. Cet ordre n'a aucune chaîne infinie décroissante, et les deux questions précédentes montrent que si $M \to_\upsilon N$ alors $\mu(M)$ est strictement plus grand que $\mu(N)$ dans cet ordre. Donc $\upsilon$ termine.*

4. (0) On admettra que $\upsilon$ est localement confluent. (Comme dit en cours, ceci est vérifiable par machine, bien que le cas présent ayant un nombre infini de règles de réduction pose quelques problèmes pratiques.) Pourquoi $\upsilon$ est-il confluent sur les termes clos ?

   *Par le lemme de Newman, $\upsilon$ étant localement confluent et terminant sur les termes clos. La confluence de $\upsilon$ est en fait évidente, si l'on remarque que $\upsilon$ est un système de réécriture* orthogonal*, et que tout système orthogonal est automatiquement confluent, par un argument fondé sur une notion de réductions parallèles inspirées du $\lambda$-calcul (Gérard Huet,* Confluent Reductions : Abstract Properties and Applications to Term Rewriting Systems*, Journal of the Association for Computing Machinery 27(4), Oct. 1980, pp.797–821). On ne demandait évidemment pas aux étudiants de trouver tous seuls cette référence. S'il y en a qui y sont arrivés, c'est peut-être qu'ils ont bien écouté le dernier cours ; ou peut-être qu'ils ont eu la présence d'esprit d'aller chercher l'un des papiers disponibles sur le $\lambda\upsilon$-calcul : ce qui est la marque d'un bon chercheur !*

## III. Confluence du $\lambda\upsilon$-calcul.

1. (0) 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.

   Dans la suite, on va montrer que le problème est dû aux variables ($x$ en l'occurrence ici). On ne considérera dans la suite de la partie III et même en partie IV que des $\lambda\upsilon$-termes *clos*.

   *$(\lambda x)N[\uparrow]$ se réduit par $(\beta)$ en $x[N/][\uparrow]$, qui est normal ; et d'autre part en $((\lambda x)[\uparrow])(N[\uparrow])$ par $(App)$. Ce dernier terme se réduit par $(Lam)$ en $(\lambda x[\Uparrow\uparrow])(N[\uparrow])$, qui se réduit par $(\beta)$ en $x[\Uparrow\uparrow][N[\uparrow]/]$. Mais ce dernier terme est normal et différent de $x[N/][\uparrow]$, par exemple si $N$ est une variable $y$. On a donc obtenu un diagramme de confluence locale qui ne se joint pas. En particulier, le calcul n'est pas confluent.*

2. (0) Démontrer que toute pile du $\lambda\upsilon$-calcul est de la forme $\Uparrow^p (M/)$ pour un certain terme $M$, ou bien $\Uparrow^p (\uparrow)$.

   *C'est pratiquement une évidence. Formellement, on peut montrer que toute pile $S$ est d'une de ces formes par récurrence structurelle sur la pile $S$. C'est le cas si $S$ est de la forme $\uparrow$ ou $M/$, auquel cas $p = 0$, et c'est par hypothèse de récurrence si $S$ est de la forme $\Uparrow S'$.*

3. (3) Définir une traduction des $\lambda\upsilon$-termes clos $M$ en $\lambda$-termes $M^\circ(P)$, paramétrée par une liste $P$ de $\lambda$-termes, comme dans le cours (partie sur le $\lambda\sigma$-calcul) ; comme dans le cours, $M^\circ(P)$ ne sera défini que pour $P$ suffisamment longue. Cette traduction devra être telle que :

(a) $\underline{n}^\circ(P)$ est défini dès que la longueur de $P$ est d'au moins $n$, et vaut le $n$ième élément de la liste $P$ ;

(b) pour tous $\lambda\upsilon$-termes $M$ et $N$, si $M \to N$ en $\lambda\upsilon$-calcul et $M^\circ(P)$ est défini, alors $N^\circ(P)$ aussi et $M^\circ(P) \to^* N^\circ(P)$ en $\lambda$-calcul ;

(c) si $M^\circ(P)$ est défini et $P = [u_1; \ldots; u_m]$, alors $(M^\circ(P))^\bullet(\ell)$ et

$$M \underbrace{[\Uparrow^m\uparrow]...[\Uparrow^m\uparrow]}_{n \text{ fois}}[\Uparrow^{m-1}\ (u_m^\bullet(\ell)/)]...[\Uparrow^2\ u_3^\bullet(\ell)/][\Uparrow u_2^\bullet(\ell)/][u_1^\bullet(\ell)/]$$

sont $\sigma$-convertibles, où $n$ est la longueur de $\ell$.

Justifier.

*Il s'agissait de loin de la question la plus difficile du devoir. (Note : il fallait bien sûr lire "$\upsilon$-convertibles" et non "$\sigma$-convertibles".)*

*Définissons $M^\circ(P)$ en s'inspirant de la traduction similaire pour le $\lambda\sigma$-calcul du cours, et en utilisant les intuitions sur la correspondance entre $\lambda\upsilon$ et $\lambda\sigma$ du début de l'énoncé :*

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

*La notation $M^\circ(P)$ est donc définie pour $P$ suffisamment longue, et la propriété (a) est alors évidente.*

*Pour démontrer la propriété (b), on peut commencer par faire la remarque utile suivante, qui nous sera aussi utile pour simplifier la démonstration de (c) :*

**Remarque 1** *si $M \to_\upsilon N$ et $M^\circ(P)$ est défini, alors $N^\circ(P)$ aussi et $M^\circ(P) = N^\circ(P)$.*

*La démonstration est une récurrence sur la profondeur du rédex contracté. Les cas de base, où $M$ est lui-même le rédex contracté, sont des calculs relativement fastidieux et que je ne reproduirai pas ici.*

*On peut démontrer, sur le même principe, que si $M \to N$ par $(\beta)$, alors $M^\circ(P) \to^* N^\circ(P)$ en $\lambda$-calcul (si le côté de gauche est bien défini). Ceci nécessite cependant de faire deux remarques auxiliaires.*

**Remarque 2** *pour tout $\lambda\upsilon$-terme clos $M$, pour tout $\lambda$-terme $v$, $M^\circ([u_1; \ldots; u_n])[x := v] = M^\circ([u_1[x := v]; \ldots; u_n[x := v]])$.*

**Remarque 3** *si $u_1 \to^* u_1'$ et $\ldots$ et $u_n \to^* u_n'$ en $\lambda$-calcul, alors $M^\circ([u_1; \ldots; u_n]) \to^* M^\circ([u_1'; \ldots; u_n'])$.*

*Ces deux remarques se prouvent par récurrence structurelle sur $M$, et ne présentent aucune difficulté. Le seul cas subtil est celui où $M$ est de la forme $M_1[\Uparrow^p (N/)]$, où l'on doit appliquer l'hypothèse de récurrence d'abord sur $M_1$, ensuite sur $N$.*

*Pour démontrer que si $M \to N$ par $(\beta)$, alors $M^\circ(P) \to^* N^\circ(P)$ en $\lambda$-calcul par récurrence sur la profondeur du rédex contracté, le cas de base consiste à démontrer que $((\lambda M)N)^\circ(P)$, si défini, se réduit en $(M[x := N])^\circ(P)$. Or*

$$\begin{array}{rcl}
((\lambda M)N)^\circ(P) & = & (\lambda x \cdot M^\circ(x :: P))N^\circ(P) \\
& \to & M^\circ(x :: P)[x := N^\circ(P)] \to M^\circ(N^\circ(P) :: P)
\end{array}$$

*par la remarque 2 et parce que $x$ est fraîche, donc pas libre dans $P$. Or $M^\circ(N^\circ(P) :: P) = (M[N/])^\circ(P)$.*
*Le cas de récurrence est essentiellement trivial. Le seul cas un peu subtil est celui où $M$ est de la forme $M_1[S]$, et le rédex contracté est dans $S$ ; donc $N = M_1[S']$ avec $S \to S'$. Par la question précédente, $S$ est de la forme $\Uparrow^p (N_1/)$, $S' = \Uparrow^p (N'_1/)$ et $N_1$ se réduit en $N'_1$. Comme $M^\circ(P)$ est défini, $P$ est de la forme $u_1 :: \ldots :: u_p :: P'$, et*

$$\begin{array}{rcl} M^\circ(P) & = & M_1^\circ(u_1 :: \ldots :: u_p :: N_1^\circ(P') :: P') \\ N^\circ(P) & = & M_1^\circ(u_1 :: \ldots :: u_p :: {N'_1}^\circ(P') :: P') \end{array}$$

*Or par hypothèse de récurrence $N_1^\circ(P') \to^* {N'_1}^\circ(P')$, donc $M^\circ(P) \to^* N^\circ(P)$ par la remarque 3.*
*Passons maintenant à la propriété (c), qui était la plus difficile à démontrer. Un élève a remarqué qu'on pouvait en fait simplifier la question, et montrer que $(M^\circ(P))^\bullet(\ell)$ est $\upsilon$-convertible avec*

$$M[\Uparrow^{m-1} (u_m^\bullet(\ell)/)]...[\Uparrow^2 u_3^\bullet(\ell)/][\Uparrow u_2^\bullet(\ell)/][u_1^\bullet(\ell)/]$$

*Le cas où $M$ commence par $\lambda$ nous force à démontrer un résultat plus général, et à montrer que pour tout entier $k$, pour toutes variables fraîches $x_1, \ldots, x_k$, $(M^\circ(x_1 :: \ldots :: x_k :: P))^\bullet(x_1 :: \ldots :: x_k :: \ell)$ est $\upsilon$-convertible avec*

$$M[\Uparrow^{m+k-1} (u_m^\bullet(\ell)/)]...[\Uparrow^{2+k} u_3^\bullet(\ell)/][\Uparrow^{1+k} u_2^\bullet(\ell)/][\Uparrow^k u_1^\bullet(\ell)/]$$

*Démontrer ceci par une récurrence directe est sans doute extrêmement pénible. On peut simplifier en remarquant qu'on peut supposer $M$ $\upsilon$-normal sans perte de généralité. Soit en effet $M'$ la $\upsilon$-forme normale de $M$ (qui existe et est même unique par les résultats de la partie II). Alors comme $M \to_\upsilon^* M'$, par la remarque 1 $M^\circ(x_1 :: \ldots :: x_k :: P) = M'^\circ(x_1 :: \ldots :: x_k :: P)$, donc $(M^\circ(x_1 :: \ldots :: x_k :: P))^\bullet(x_1 :: \ldots :: x_k :: \ell) = (M'^\circ(x_1 :: \ldots :: x_k :: P))^\bullet(x_1 :: \ldots :: x_k :: \ell)$. Si ce dernier est $\upsilon$-convertible avec $M'[\Uparrow^{m+k-1} (u_m^\bullet(\ell)/)]...[\Uparrow^{2+k} u_3^\bullet(\ell)/][\Uparrow^{1+k} u_2^\bullet(\ell)/][\Uparrow^k u_1^\bullet(\ell)/]$, il l'est donc avec $M[\Uparrow^{m+k-1} (u_m^\bullet(\ell)/)]...[\Uparrow^{2+k} u_3^\bullet(\ell)/][\Uparrow^{1+k} u_2^\bullet(\ell)/][\Uparrow^k u_1^\bullet(\ell)/]$.*
*Sachant qu'on peut se restreindre à des termes $\upsilon$-normaux clos, on remarque maintenant :*

**Remarque 4** *Un terme $\upsilon$-normal clos ne contient aucun sous-terme de la forme $N[S]$.*

*En effet, sinon il contiendrait un plus petit sous-terme de la forme $N[S]$, et ce dernier serait normal. Ceci implique que $N$ n'est ni une application ni une abstraction (sinon $(App)$ our $(Lam)$ s'appliquerait), ni de la forme $N_1[S_1]$ (par minimalité), donc est de la forme $\underline{n}$. Si $n = 1$, alors $(Var/)$ ou $(Var \Uparrow)$ ou $(Var \uparrow)$ s'appliquerait. Si $n \geq 2$, alors $(\uparrow /)$ ou $(\uparrow\Uparrow)$ ou $(Var \uparrow)$ s'appliquerait. Dans tous les cas on a donc une contradiction.*
*Pour montrer que $(M^\circ(x_1 :: \ldots :: x_k :: P))^\bullet(x_1 :: \ldots :: x_k :: \ell)$ est $\upsilon$-convertible avec*

$$M[\Uparrow^{m+k-1} (u_m^\bullet(\ell)/)]...[\Uparrow^{2+k} u_3^\bullet(\ell)/][\Uparrow^{1+k} u_2^\bullet(\ell)/][\Uparrow^k u_1^\bullet(\ell)/]$$

*, sachant que $M$ est $\upsilon$-normal et clos, on n'a donc qu'à examiner les cas où $M$ est une application, une abstraction, ou de la forme $\underline{i}$. Les deux premiers cas sont par hypothèse de récurrence, directement. Reste le cas $M = \underline{i}$.*
*On calcule alors :*

**Remarque 5** *$\underline{i}[\Uparrow^j (N/)] \to_\upsilon^* \underline{i}$ si $i \leq j$, $N[\uparrow]^{i-1}$ si $i = j + 1$ (où $[\uparrow]^{i-1}$ dénote l'application de $[\uparrow]$ $i - 1$ fois), $\underline{i - 1}$ si $i > j + 1$.*

*En utilisant la remarque 5, on a deux cas. Si $i \leq k$, alors*

$$\begin{array}{rl} & \underline{i}[\Uparrow^{m+k-1} (u^\bullet_m(\ell)/)]...[\Uparrow^{2+k} u^\bullet_3(\ell)/][\Uparrow^{1+k} u^\bullet_2(\ell)/][\Uparrow^k u^\bullet_1(\ell)/] \\ \to^*_\upsilon & \underline{i} = (\underline{i}^\circ(x_1 :: \ldots :: x_k :: P))^\bullet(x_1 :: \ldots :: x_k :: \ell) \end{array}$$

*Si $i > k$, en remarquant que comme $\underline{i}^\circ(x_1 :: \ldots :: x_k :: P)$ est bien défini et $P$ est de longueur $m$, alors $i \leq k+m$, d'où :*

$$\begin{array}{rl} & \underline{i}[\Uparrow^{m+k-1} (u^\bullet_m(\ell)/)]...[\Uparrow^{2+k} u^\bullet_3(\ell)/][\Uparrow^{1+k} u^\bullet_2(\ell)/][\Uparrow^k u^\bullet_1(\ell)/] \\ \to^*_\upsilon & \underline{i}[\Uparrow^{m+k-2} (u^\bullet_{m-1}(\ell)/)]...[\Uparrow^{2+k} u^\bullet_3(\ell)/][\Uparrow^{1+k} u^\bullet_2(\ell)/][\Uparrow^k u^\bullet_1(\ell)/] \\ \to^*_\upsilon & \ldots \\ \to^*_\upsilon & \underline{i}[\Uparrow^{i-1} (u^\bullet_{i-k}(\ell)/)]...[\Uparrow^{2+k} u^\bullet_3(\ell)/][\Uparrow^{1+k} u^\bullet_2(\ell)/][\Uparrow^k u^\bullet_1(\ell)/] \\ \to^*_\upsilon & u^\bullet_{i-k}(\ell)[\uparrow]^{i-1}[\Uparrow^{i-2} (u^\bullet_{i-k-1}(\ell)/)]...[\Uparrow^{2+k} u^\bullet_3(\ell)/][\Uparrow^{1+k} u^\bullet_2(\ell)/][\Uparrow^k u^\bullet_1(\ell)/] \\ \to^*_\upsilon & u^\bullet_{i-k}(x_1 :: \ldots :: x_{i-1} :: \ell)[\Uparrow^{i-2} (u^\bullet_{i-k-1}(\ell)/)]...[\Uparrow^{2+k} u^\bullet_3(\ell)/][\Uparrow^{1+k} u^\bullet_2(\ell)/][\Uparrow^k u^\bullet_1(\ell)/] \\ & \quad \textit{par la question I.2} \\ \to^*_\upsilon & (u_{i-k}[x_{i-1} := u_{i-k-1}] \ldots [x_{k+1} := u_1])^\bullet(x_1 :: \ldots :: x_k :: \ell) \\ & \quad \textit{en appliquant } i-k-1 \textit{ fois la question I.3} \\ = & u^\bullet_{i-k}(x_1 :: \ldots :: x_k :: \ell) \\ & \quad \textit{car } x_1, \ldots, x_k \textit{ sont fraîches} \\ = & (\underline{i}^\circ(x_1 :: \ldots :: x_k :: P))^\bullet(x_1 :: \ldots :: x_k :: \ell) \end{array}$$

4. (2) Déduire du dernier point ci-dessus que si $P = \ell = [x_1; \ldots; x_n]$, et si $M^\circ(P)$ est défini, alors $M \to^*_\upsilon (M^\circ(P))^\bullet(\ell)$.

   *Il suffit de démontrer que $M$ et $(M^\circ(P))^\bullet(\ell)$ sont $\upsilon$-convertibles. Comme $\_^\bullet(\ell)$ ne fabrique que des termes $\upsilon$-normaux, il s'ensuit par confluence de $\upsilon$ que $M \to^*_\upsilon (M^\circ(P))^\bullet(\ell)$.*
   *L'idée de la question était de remarquer que par (c) (modifié), $(M^\circ(P))^\bullet(\ell)$ est $\upsilon$-convertible avec*

   $$M[\Uparrow^{m-1} (\underline{m}/)]...[\Uparrow^2 (\underline{3}/)][\Uparrow (\underline{2}/)][\underline{1}/]$$

   *Sans perte de généralité, on peut supposer que $M$ est $\upsilon$-normal. Comme dans la question précédente, on peut montrer que $M$ est $\upsilon$-convertible avec*

   $$M[\Uparrow^{m-1} (\underline{m}/)]...[\Uparrow^2 (\underline{3}/)][\Uparrow (\underline{2}/)][\underline{1}/]$$

   *Ceci se fait par récurrence structurelle sur $M$, le cas où $M = \underline{i}$ se traitant via la remarque 5.*
   *Il y avait en fait plus simple. Comme on peut supposer sans perte de généralité que $M$ est $\upsilon$-normal (et clos), on peut tout simplement montrer directement par récurrence sur $M$ que $(M^\circ(P))^\bullet(\ell)$ est exactement $M$. En utilisant la remarque 4, on n'a à considérer que les cas où $M$ est une application, ou une abstraction (autant de cas triviaux), ou un indice $\underline{i}$, auquel cas :*

   $$(\underline{i}^\circ([x_1; \ldots; x_m]))^\bullet([x_1; \ldots; x_m]) \quad = \quad x^\bullet_i([x_1; \ldots; x_m]) = \underline{i}$$

   *Il s'ensuite au passage qu'il n'est pas réellement nécessairement d'établir la propriété (c) pour pouvoir effectuer le reste du sujet.*

5. (1) Utiliser la technique d'interprétation de Hardin et en déduire que $\lambda\upsilon$ est confluent sur les $\lambda\upsilon$-termes clos. Pour être précis, montrer que si $M$, $N_1$, $N_2$ sont des $\lambda\upsilon$-termes clos et que $M \to^* N_1$ et $M \to^* N_2$, alors il existe un $\lambda\upsilon$-terme clos $N$ tel que $N_1 \to^* N$ et $N_2 \to^* N$.

*Par hypothèse, on a :*

$$\begin{tikzcd}
 & M \arrow[dl, "*"'] \arrow[dr, "*"] & \\
N_1 & & N_2
\end{tikzcd}$$

*Donc pour $P = \ell = [x_1; \ldots; x_m]$ avec $m$ suffisamment grand :*

$$\begin{tikzcd}
 & M^\circ(P) \arrow[dl, "*"'] \arrow[dr, "*"] & \\
N_1^\circ(P) & & N_2^\circ(P)
\end{tikzcd}$$

*en $\lambda$-calcul. Comme le $\lambda$-calcul est confluent, il existe un $\lambda$-terme $u$ tel que :*

$$\begin{tikzcd}
 & M^\circ(P) \arrow[dl, "*"'] \arrow[dr, "*"] & \\
N_1^\circ(P) \arrow[dr, "*"'] & & N_2^\circ(P) \arrow[dl, "*"] \\
 & u &
\end{tikzcd}$$

*Donc :*

$$\begin{tikzcd}
 & (M^\circ(P))^\bullet(\ell) \arrow[dl, "*"'] \arrow[dr, "*"] & \\
(N_1^\circ(P))^\bullet(\ell) \arrow[dr, "*"'] & & (N_2^\circ(P))^\bullet(\ell) \arrow[dl, "*"] \\
 & u^\bullet(\ell) &
\end{tikzcd}$$

*Comme $N_i \to^* (N_i^\circ(P))^\bullet(\ell)$, on peut compléter le premier et le dernier diagramme comme suit :*

$$\begin{tikzcd}
 & M \arrow[dl, "*"'] \arrow[dr, "*"] & \\
N_1 \arrow[d, "*"'] & (M^\circ(P))^\bullet(\ell) \arrow[dl, "*"'] \arrow[dr, "*"] & N_2 \arrow[d, "*"] \\
(N_1^\circ(P))^\bullet(\ell) \arrow[dr, "*"'] & & (N_2^\circ(P))^\bullet(\ell) \arrow[dl, "*"] \\
 & u^\bullet(\ell) &
\end{tikzcd}$$

## IV. Machines à réduction pour $\lambda\upsilon$.

On cherche à définir une machine à réduction, dans le style de la machine de Krivine pour $\lambda\sigma$ du cours. Une *configuration* de la machine est un triplet $M, R, args$, où $M$ est un $\lambda\upsilon$-terme clos, $R$ est une liste $S_1; S_2; \ldots; S_n$ de $\lambda\upsilon$-piles closes $S_i$, $1 \leq i \leq n$, et $args$ est une liste d'arguments

$N_1; N_2; \ldots; N_k$, tous clos. La sémantique intuitive de $M, R, args$ est alors le $\lambda\upsilon$-terme $S(M, R, args) = M[S_1][S_2]\ldots[S_n]N_1N_2\ldots N_k$.

Si $R = S_1; S_2; \ldots; S_n$, on notera $M[[R]]$ le terme $M[S_1][S_2]\ldots[S_n]$. On notera $\Uparrow R$ la liste $\Uparrow S_1; \Uparrow S_2; \ldots; \Uparrow S_n$. On notera aussi $R; S$ la liste $S_1; S_2; \ldots; S_n; S$ et $S; R$ la liste $S; S_1; S_2; \ldots; S_n$.

1. (3) Produire la machine, sous forme d'un nombre fini de règles $M, R, args \to M', R', args'$. Cette machine devra vérifier que :
   (a) si $M, R, args \to M', R', args'$ alors $S(M, R, args)$ se réduit en zéro, une ou plusieurs étapes du $\lambda\upsilon$-calcul en $S(M', R', args')$ ;
   (b) lorsque la machine termine, c'est sur une configuration de la forme $\lambda M, R, []$ ou bien $\underline{n}, [], args$.

   Justifier. Pourquoi est-il légitime de dire qu'il s'agit d'une machine de réduction de tête faible ?

   *La machine est définie par les règles suivantes :*

$$
\begin{array}{rccccccc}
MN, & R, & args & \to & M, & R, & N[[R]] :: args \\
\lambda M, & R, & N :: args & \to & M, & \Uparrow R; N/, & args \\
M[S], & R, & args & \to & M, & S; R, & args \\
\underline{1}, & M/; R, & args & \to & M, & R, & args \\
\underline{1}, & \Uparrow S; R, & args & \to & \underline{1}, & R, & args \\
\underline{n+1}, & N/; R, & args & \to & \underline{n}, & R, & args \\
\underline{n+1}, & \Uparrow S; R, & args & \to & \underline{n}, & S; \uparrow; R, & args \\
\underline{n}, & \uparrow; R, & args & \to & \underline{n+1}, & R, & args
\end{array}
$$

   *La propriété (a) est vérifiée : la première règle est par $(App)$, la deuxième par $(Lam)$, la troisième ne change pas la sémantique, les règles suivantes sont par $(Var/)$, $(Var \Uparrow)$, $(\uparrow /)$, $(\uparrow\Uparrow)$ et $(Var \uparrow)$ respectivement.*

   *La propriété (b) est prouvée comme suit. Supposons que la machine termine sur $M, R, args$. $M$ ne peut pas être une application à cause de la première règle, ni une variable car $M$ est clos, ni une application de substitution à cause de la troisième règle. Donc $M$ est soit une abstraction soit un indice de de Bruijn.*

   – *Si la machine termine sur $\lambda M, R, args$, alors $args = []$ sinon la deuxième règle s'applique.*
   – *Si la machine termine sur $\underline{n}, R, args$, alors $R = []$ sinon l'une des cinq dernières règles s'applique.*