DM, λ-calcul, 2020

à rendre le 06 mai 2020 au plus tard

Correction.

Notes préliminaires. Je demande en général à ce que les réponses aux DM soient claires. Mon but est égoïste : j'essaie de minimiser mon temps et mes efforts de correction. Dans les circonstances actuelles, je ne pourrai même pas imprimer vos copies, or j'éprouve encore plus de difficultés à lire des copies électroniquement : pas moyen de classer les copies, chacune ouverte à une certain page, en piles savamment agencées ; pas moyen de feuilleter rapidement un ensemble de copies, en gardant des doigts à certains endroits où je voudrais retourner rapidement, etc. En plus, je corrige le plus souvent question par question, et non copie par copie... Vous pouvez donc imaginer que je préférerais que ce que vous écrirez sera non seulement juste, mais *clairement* et *manifestement* juste. La concision est une qualité allant dans le même sens, ainsi que le soin porté aux références aux résultats du cours ou aux questions précédentes. Je compte sur vous !

Si vous le souhaitez, je mettrai le source LaTeX de ce sujet en ligne, et vous pourrez écrire votre solution en donnant votre nom comme argument de la commande `\author` en tête de document, puis écrire vos réponses entre les différentes `\begin{solution}` et `\end{solution}`. Envoyez-moi ensuite le pdf, pas le source.

Au fait, l'examen contiendra peut-être une suite à ce DM.

1 Le système CBWK

On considère une variante du système de logique combinatoire vu en TD, qui était basé sur les combinateurs **S**, **K**, et **I**. Celui-ci est basé sur les combinateurs

$\mathbf{C}$, $\mathbf{B}$, $\mathbf{W}$, et $\mathbf{K}$, avec comme règles de réduction :

$$\begin{aligned} \mathbf{C}MNP &\to (MP)N \\ \mathbf{B}MNP &\to M(NP) \\ \mathbf{W}MN &\to MNN \\ \mathbf{K}MN &\to M \end{aligned}$$

Explicitement, la syntaxe des termes du système CBWK (ou *CBWK-termes*) est :

$$\begin{aligned} M, N, P, \ldots = x & \qquad\qquad \text{variables} \\ \mid \mathbf{C} & \\ \mid \mathbf{B} & \\ \mid \mathbf{W} & \\ \mid \mathbf{K} & \\ \mid MN & \qquad\qquad \text{applications.} \end{aligned}$$

Comme d'habitude, l'application associe à gauche : dans la règle de $\mathbf{C}$, j'aurais pu écrire MPN plutôt que $(MP)N$.

Question 1 Trouvez un CBWK-terme $\mathbf{I}$ tel que, pour tout N, $\mathbf{I}N \to^* N$.

2 lignes pour moi.

$\mathbf{I} = \mathbf{CK}*$*, où* $*$ *est n'importe quel CBWK-terme, par exemple* $\mathbf{K}$ *:* $\mathbf{CK} * N \to \mathbf{K}N* \to N$*.*

Question 2 On cherche à définir une forme d'abstraction $[x]M$ en CBWK , autrement dit on cherche à définir, pour chaque variable x et chaque CBWK-terme M, un CBWK-terme $[x]M$ tel que, pour tout CBWK-terme N, $([x]M)N \to^+ M[x := N]$. ($\to^+$ signifie : se réduit en au moins une étape.) Donnez une telle définition. Je vous donne deux des clauses :

Je vous ai donné le cas difficile, les autres cas sont beaucoup plus faciles.

$$\begin{aligned} [x]M &= \mathbf{K}M && \text{si } x \text{ pas libre dans } M \\ [x](M_1M_2) &= \mathbf{W}(\mathbf{B}(\mathbf{C}([x]M_1))([x]M_2)) && \text{si } x \text{ libre dans } M_1 \text{ et dans } M_2. \end{aligned}$$

Attention : il se peut que vous ayez à retravailler votre définition, pour que les réponses aux questions suivantes soient correctes.

$$\begin{aligned} [x]M &= \mathbf{K}M && \textit{si } x \textit{ pas libre dans } M \\ [x]x &= \mathbf{I} && \\ [x](M_1M_2) &= \mathbf{C}([x]M_1)M_2 && \textit{si } x \textit{ libre dans } M_1\textit{, pas dans } M_2 \\ [x](M_1M_2) &= \mathbf{B}M_1([x]M_2) && \textit{si } x \textit{ libre dans } M_2\textit{, pas dans } M_1 \\ [x](M_1M_2) &= \mathbf{W}(\mathbf{B}(\mathbf{C}([x]M_1))([x]M_2)) && \textit{sinon.} \end{aligned}$$

Dans ce dernier cas, on peut aussi utiliser le terme $\mathbf{W}(\mathbf{C}(\mathbf{B}(\mathbf{KB})([x]M_1))([x]M_2))$, *c'est-à-dire* $\mathbf{W}([x][y](M_1M_2'))$, *où* M_2' *est le terme* $M_2[x := y]$. *Le terme proposé est* $\mathbf{W}([y][x](M_1M_2'))$.
On effectue la vérification de $([x]M)N \to^+ M[x := N]$ *par récurrence sur* M *(ce n'est pas demandé). Si* x *est libre dans* M_1 *et* M_2*, on a :*

$$\begin{aligned}([x](M_1M_2))N &= \mathbf{W}(\mathbf{B}(\mathbf{C}([x]M_1))([x]M_2))N\\ &\to \mathbf{B}(\mathbf{C}([x]M_1))([x]M_2)NN\\ &\to \mathbf{C}([x]M_1)(([x]M_2)N)N\\ &\to (([x]M_1)N)(([x]M_2)N)\\ &\to^+ (M_1[x := N])(M_2[x := N]) = (M_1M_2)[x := N],\end{aligned}$$

par hypothèse de récurrence. Si x *est libre dans* M_1 *et pas dans* M_2*,*

$$\begin{aligned}([x](M_1M_2))N &= \mathbf{C}([x]M_1)M_2N\\ &\to ([x]M_1)NM_2 \quad \to^+ M_1[x := N]M_2 = (M_1M_2)[x := N],\end{aligned}$$

parce que x *n'est pas libre dans* M_2*. Les autres cas sont faciles.*

Question 3 Montrez que si $z \neq x$ et z n'est pas libre dans N, alors $([z]M)[x := N] = [z](M[x := N])$.

Par récurrence sur la taille de M.
Si $M = z$, $([z]M)[x := N] = \mathbf{I}[x := N] = \mathbf{I}$, *et* $[z](M[x := N]) = [z]z$ *(parce que* $z \neq x$*)* $= \mathbf{I}$.
Si z *n'est pas libre dans* M*, alors* $([z]M)[x := N] = (\mathbf{K}M)[x := N] = \mathbf{K}(M[x := N])$, *et* $[z](M[x := N]) = \mathbf{K}(M[x := N])$, *car* z *n'étant pas libre non plus dans* N*, elle n'est pas libre dans* $M[x := N]$.
On regarde finalement les cas où $M = M_1M_2$*, et* z *est libre dans* M_1*,* M_2*, ou les deux.*
Si z *est libre dans* M_1 *et pas dans* M_2*,* $([z]M)[x := N] = (\mathbf{C}([z]M_1)M_2)[x := N] = \mathbf{C}([z](M_1[x := N]))(M_2[x := N])$*, en utilisant l'hypothèse de récurrence ; et c'est aussi ce que vaut* $[z](M[x := N])$*, puisque* z *restera libre dans* $M_1[x := N]$ *et non libre dans* $M_2[x := N]$.
L'argument est similaire si z *est libre dans* M_2 *et pas dans* M_1.
Si z *est libre dans* M_1 *et dans* M_2*, l'argument est encore une fois similaire, mais avec des* $\mathbf{W}(\mathbf{B}(\mathbf{C}$ *en tête de termes.*

Question 4 On définit une traduction des λ-termes vers les CBWK-termes par : $x^* = x$, $(uv)^* = u^*v^*$, $(\lambda x \cdot u)^* = [x]u^*$. Montrer que si $u \to v$ par β-réduction faible (c'est-à-dire pas sous une λ), alors $u^* \to^+ v^*$. Vous aurez besoin de lemmes auxiliaires : énoncez-les moi, et prouvez-les, dans l'ordre (aucune référence en avant autorisée).

Les élèves les plus sérieux auront géré le α-renommage, ce que j'ignorerai ici. Si l'on souhaite le traiter, on doit montrer :

— ***Lemme 0*** *: si y n'est pas libre dans M, alors $[y](M[x := y]) = [x]M$ (récurrence immédiate sur M).*

— *si $s =_\alpha t$ alors $s^* = t^*$. Il suffit de démontrer que si $s \; \alpha \; t$, alors $s^* = t^*$, et c'est une récurrence facile sur la position où est effectuée le α-renommage, en utilisant le lemme 0 dans le cas de base.*

Lemme A. *$s^*[x := t^*] = (s[x := t])^*$.*

*Par récurrence sur s. Si $s = x$, $s^*[x := t^*] = x[x := t^*] = t^* = (s[x := t])^*$. Si s est une autre variable y, $s^*[x := t^*] = y[x := t^*] = y = y^* = (y[x := t])^* = (s[x := t])^*$. Si s est une application, c'est un appel direct à l'hypothèse de récurrence. Si $s = \lambda z.s'$, $s^*[x := t^*] = ([z]s'^*)[x := t^*]$ par la **Question 3**.*

Lemme B. *$((\lambda x.s)t)^* \to^+ (s[x := t])^*$. On a $((\lambda x.s)t)^* = ([x]s^*)t^* \to^+ s^*[x := t^*]$ (par la **Question 2**) $= (s[x := t])^*$ (par le Lemme A).*

Lemme C. *Si $u \to v$ par réduction faible, alors $u^* \to^+ v^*$.*

Par récurrence sur la profondeur du β-rédex contracté. Si u est le β-rédex lui-même, c'est le lemme B. Sinon, $u = u_1u_2$ et la réduction a lieu dans u_1 ou dans u_2. Dans le premier cas, on $u_1 \to v_1$ par réduction faible, $v = v_1u_2$, et donc $u^* = u_1^*u_2^* \to^+ v_1^*u_2^* = v^*$, en utilisant l'hypothèse de récurrence. Le cas où la réduction a lieu dans u_2 est symétrique.

Question 5 Donner un exemple montrant que l'on ne peut pas étendre le résultat de la question précédente lorsque la réduction de u vers v a lieu sous un λ.

On choisit $u = \lambda x \cdot (\lambda y \cdot x)x$, par exemple, avec $v = \lambda x \cdot x$. Alors $u^* = \mathbf{W}(\mathbf{B}(\mathbf{C}(\mathbf{K}x)\mathbf{I}))$, qui est en forme normale, et $v^* = \mathbf{I}$.

2 Entiers de Parigot

On a vu en cours les entiers de Church. Il y a de nombreuses autres façons de coder les entiers. Nous allons étudier les *entiers de Parigot* :

$$\begin{array}{rcl} \overline{0} & = & \lambda x, f.x \\ \overline{n+1} & = & \lambda x, f.f\overline{n}(\overline{n}xf) \end{array}$$

Question 6 Vérifier que $\overline{succ} = \lambda n, x, f.fn(nxf)$ est une fonction successeur, c'est-à-dire que $\overline{succ}\ \overline{n} \to^* \overline{n+1}$ pour tout $n \in \mathbb{N}$. En combien d'étapes de β-réduction cela se fait-il ?

N'oubliez pas de mentionner le nombre d'étapes !

Une étape :

$$\overline{succ}\,\overline{n} \quad \to \quad \lambda x, f.f\overline{n}(\overline{n}xf) = \overline{n+1}$$

Question 7 On définit un récurseur $\overline{R}$ par $\overline{R}(u, v, n) = nuv$. Vérifier qu'il se comporte bien comme un récurseur, c'est-à-dire :

N'oubliez pas de mentionner le nombre d'étapes !

$$\overline{R}(u, v, \overline{0}) \quad \to^* \quad u \qquad (1)$$
$$\overline{R}(u, v, \overline{n+1}) \quad \to^* \quad v\overline{n}(\overline{R}(u, v, \overline{n})) \qquad (2)$$

Combien de β-réductions utilisez-vous dans chaque cas ?

2 étapes pour la première :

$$\begin{aligned} \overline{R}(u, v, \overline{0}) &= \overline{0}uv \\ &\to^2 u \end{aligned}$$

et 2 étapes pour la seconde :

$$\begin{aligned} \overline{R}(u, v, \overline{n+1}) &= \overline{n+1}\ uv \\ &\to^2 v\overline{n}(\overline{n}uv) = v\overline{n}(\overline{R}(u, v, \overline{n})) \end{aligned}$$

Question 8 Donnez un terme $\overline{pred}$ calculant le prédécesseur. On devra avoir :

$$\overline{pred}\,\overline{n+1} \quad \to^* \quad \overline{n} \qquad (3)$$
$$\overline{pred}\,\overline{0} \quad \to^* \quad \overline{0} \qquad (4)$$

La réduction (3) devra compter au plus 5 étapes, la réduction (4) au plus 3.

On peut prendre $\overline{pred} = \lambda n.\overline{R}(\overline{0}, \lambda x, y.x, n)$, qui est correct par la question précédente : $\overline{pred}\ \overline{n+1} \to \overline{R}(\overline{0}, \lambda x, y.x, \overline{n+1}) \to^+ (\lambda x, y.x)\overline{n}(\overline{R}(\overline{0}, \lambda x, y.x, \overline{n})) \to^2 \overline{n}$ (5 étapes de réduction) et $\overline{pred}\ \overline{0} \to \overline{R}(\overline{0}, \lambda x, y.x, \overline{0}) \to^+ \overline{0}$ (3 étapes).
On peut aussi définir directement $\overline{pred} = \lambda n.n\overline{0}(\lambda x, y.x)$.

Question 9 On note $[n]$ le codage de Church de l'entier n, et on rappelle que le prédécesseur est donné par $P = \lambda n.\pi_2(n(\lambda s \cdot \langle S(\pi_1 s), \pi_1 s\rangle)\langle[0], [0]\rangle)$ sur les entiers de

Church, où $\langle \_, \_ \rangle$ dénote la paire usuelle en λ-calcul et π_1, π_2 les deux projections correspondantes, et S est le successeur sur les entiers de Church. Combien de réductions sont-elles nécessaires pour réduire $P[n+1]$ en $[n]$? On donnera une estimation asymptotiquement sous forme d'un grand O d'une fonction de n. En quoi les entiers de Parigot sont-ils un progrès par rapport aux entiers de Church ?

Grand O de n : on a besoin d'un nombre linéaire de réductions.

Le prédécesseur sur les entiers de Parigot n'a besoin que d'un nombre constant de réductions.

Question 10 Proposer des termes du λ-calcul implémentant l'addition, la multiplication, et la fonction $n \mapsto 2^n$ sur les entiers de Parigot.

Pas besoin de preuve, mais une petite explication sera bienvenue, surtout si votre codage est bizarre.

Ce sont des fonctions primitives récursives. Il suffit de les coder à l'aide du récurseur :

$$\begin{array}{rcl} \overline{+} & = & \lambda m, n.\overline{R}(m, \lambda k.\overline{succ}, n) & = & \lambda m, n.nm(\lambda k.\overline{succ}) \\ \overline{\times} & = & \lambda m, n.\overline{R}(\overline{0}, \lambda k, x.\overline{+}xm, n) & = & \lambda m, n.n\overline{0}(\lambda x.\overline{+}xm) \\ \overline{exp} & = & \lambda n.\overline{R}(\overline{1}, \lambda k, x.\overline{+}xx, n) & = & \lambda n.n\overline{1}(\lambda x.\overline{+}xx) \end{array}$$

par exemple.

3 Réductions de tête et types intersection simples

On considère les *types intersection simples*, définis comme suit. Il s'agit d'une variante simplifiée des types du système $\mathcal{D}\Omega$ vu en TD.

$$\sigma, \tau, \ldots \quad ::= \quad b \quad | \quad [\sigma_1, \cdots, \sigma_n] \to \tau$$

Ici, b est un type de base, et $n \in \mathbb{N}$. La notation $[\sigma_1, \cdots, \sigma_n]$ dénote un multi-ensemble, possiblement vide, de types. (Un multi-ensemble est une liste finie modulo permutation.) Intuitivement, un terme est de « type » $[\sigma_1, \cdots, \sigma_n]$ si et seulement il est de type σ_i pour tout i, $1 \leq i \leq n$, en même temps. Lorsque $n = 0$, on voit que tout terme est de « type » $[]$; c'est l'équivalent du symbole Ω du TD.

Nous écrirons μ pour un multi-ensemble de types, lorsque les types qu'il contient ne sont pas particulièrement importants.

Les jugements de typage sont de la forme $\Gamma \vdash u : \tau$, où Γ est un ensemble d'hypothèses de typage $x : \mu$, les variables x étant distinctes deux à deux. Les règles de typage sont :

$$\frac{}{\Gamma, x : [\sigma_1, \cdots, \sigma_n] \vdash x : \sigma_i} (Var) \quad (\text{où } 1 \leq i \leq n)$$

$$\frac{\Gamma \vdash u : [\sigma_1, \cdots, \sigma_n] \to \tau \quad \Gamma \vdash v : \sigma_1 \quad \dots \quad \Gamma \vdash v : \sigma_n}{\Gamma \vdash uv : \tau} (App) \quad \frac{\Gamma, x : \mu \vdash u : \tau}{\Gamma \vdash \lambda x \cdot u : \mu \to \tau} (Abs)$$

Comme en cours, chaque terme est lu à α-renommage près, ce qui permet notamment de changer le nom de x à volonté dans la règle (Abs).

On définit RED_τ pour chaque type τ, et RED_μ pour chaque multi-ensemble μ de types, comme suit :

— RED_b vaut HN, l'ensemble des λ-termes qui ont une forme normale de tête (ceci signifie : dont la réduction de tête termine).

— Lorsque $u = [\sigma_1, \cdots, \sigma_n]$, $RED_\mu = \bigcap_{i=1}^n RED_{\sigma_i}$; en particulier, $RED_{[]}$ est l'ensemble Λ de *tous* les λ-termes.

— $RED_{\mu \to \tau}$ vaut $RED_\mu \Rightarrow RED_\tau$, c'est-à-dire l'ensemble des termes u tels que pour tout $v \in RED_\mu$, uv est dans RED_τ.

On notera $\to_h$ la réduction de tête en une étape, et $\to_w$ la réduction de tête *faible*, c'est-à-dire n'opérant pas sous une λ-abstraction. On pourra admettre, sans le démontrer, que $w \to_h w'$ implique $w[x := v] \to_h w'[x := v]$, pour tout terme v, et de même pour $\to_w$ a la place de $\to_h$.

Question 11 On sait que si $u \to_w u'$, alors pour tout v, $uv \to_w u'v$. Nous l'avons utilisé dans la démonstration du théorème de standardisation, par exemple. Montrer que ceci n'est plus vrai si on remplace $\to_w$ par $\to_h$. Cette question a pour but de vous éviter de tomber dans un piège à la question suivante.

> *Par exemple, $u = \lambda x.(\lambda y.y)x \to_h u' = \lambda x.x$, mais $uv \to_h (\lambda y.y)v \neq u'v$. (Oh oui, bien sûr, $u'v \to_h v$ et $(\lambda y.y)v \to_h v$ aussi, mais ce n'est pas ce qu'on demande.)*

Question 12 Montrer que, pour tout type τ :

Même si ça ressemble à une question de cours, c'est *la* question de réflexion du DM !

H1 $RED_\tau \subseteq HN$;

H2 pour tout $u \in RED_\tau$, si $u \to_h u'$ alors $u' \in RED_\tau$;

H3 tout terme neutre u dont tous les réduits de tête en une étape u' sont dans RED_τ, est aussi dans RED_τ.

On notera que H3 s'exprime de façon équivalente comme suit : toute forme normale de tête neutre est dans RED_τ, et tout terme neutre non normal de tête dont l'unique réduit de tête est dans RED_τ, est lui aussi dans RED_τ.

Types de base. Evident.

Types $\mu \to \tau$.

- ***H1**. Soit $u \in RED_{\mu\to\tau}$. On choisit un terme v dans RED_μ. Alors $uv \in RED_\tau \subseteq HN$, et on va en conclure que $u \in HN$. Deux difficultés :*
  - *On doit d'abord démontrer que RED_μ est non vide. Par **H3** sur μ, RED_μ contient toutes les formes normales de tête. On peut par exemple prendre une variable x pour v.*
  - *Sachant que $ux \in HN$, on souhaite en déduire que $u \in HN$. Comme on l'a vu lors de la démonstration du théorème de standardisation en cours, il faut prendre des précautions. Imaginons une réduction de tête infinie partant de u, $u = u_0 \to_h u_1 \to_h \cdots$. Si c'est une réduction de tête* faible*, alors $ux = u_0x \to_h u_1x \to_h \cdots$ est aussi une réduction de tête (faible), contradiction. Sinon, soit n le premier indice où u_n est une λ-abstraction. Tous les termes suivants sont aussi des λ-abstractions, et il sera commode (à α-renommage près) de considérer que cette abstraction porte sur la variable x elle-même. La réduction de tête partant de u est donc $u = u_0 \to_h u_1 \to_h \cdots \to_h u_n = \lambda x \cdot v_n \to_h \lambda x \cdot v_{n+1} \to_h \cdots$, avec $v_n \to_h v_{n+1} \to_h \cdots$. On a alors une réduction de tête $ux = u_0x \to_h u_1x \to_h \cdots \to_h u_nx = (\lambda x \cdot v_n)x \to_h v_n \to_h v_{n+1} \to_h \cdots$, qui est infinie : contradiction, de nouveau.*
- ***H2**. Soit $u \in RED_{\mu\to\tau}$, et $u \to_h u'$. Pour démontrer que $u' \in RED_{\mu\to\tau}$, on doit démontrer que pour tout $v \in RED_\mu$, $u'v$ est dans RED_τ. Ceci sera par **H2** sur τ, si on arrive à démontrer que $uv \to_h u'v$.*

  C'est le cas si $u \to_h u'$ est une réduction de tête faible.

  *Sinon, $u = \lambda x \cdot w$, $u' = \lambda x \cdot w'$, et $w \to_h w'$. Alors $uv \to_h w[x := v] \to_h w'[x := v]$ (par le résultat admis), donc par **H2** utilisée deux fois au type τ, $w'[x := v]$ est dans RED_τ. Or $u'v \to_h w'[x := v]$, et $u'v$ est neutre ! Donc par **H3** au type τ, $u'v$ est dans RED_τ.*
- ***H3**. Soit u un terme neutre.*

  *Si u est en forme normale de tête, on doit montrer que u est dans $RED_{\mu\to\tau}$. Mais u est de la forme $xu_1 \cdots u_n$ dans ce cas. Pour tout $v \in RED_\mu$, uv est de la forme $xu_1 \cdots u_nv$, qui est encore neutre et en forme normale de tête, donc dans RED_τ par **H3** au type τ. Donc u est bien dans $RED_{\mu\to\tau}$.*

Sinon, soit u' l'unique réduit en une étape de réduction de tête à partir de u, et supposons que $u' \in RED_{\mu\to\tau}$. On fixe $v \in RED_\mu$, et on souhaite démontrer que $uv \in RED_\tau$. La clé, c'est que u est neutre, donc la réduction $u \to_h u'$ est une réduction de tête faible. *Alors $uv \to_w u'v$, $u'v$ est dans RED_τ par définition. Comme uv est neutre, il est aussi dans RED_τ par* ***H3*** *au type τ, et l'on conclut.*

Question 13 Montrer que pour tout λ-terme u tel que $u[x := v] \in RED_\tau$ pour tout $v \in RED_\mu$, on a $\lambda x \cdot u \in RED_{\mu\to\tau}$.

Plus facile que vous ne le croyez.

Comme dans le cours, mais plus simple (aucune récurrence auxiliaire !). Soit $v \in RED_\mu$ arbitraire. Le terme $(\lambda x \cdot u)v$ est neutre, et son unique réduit de tête en une étape est $u[x := v]$, qui est dans RED_τ par hypothèse.

Question 14 Conclure que tout terme typable dans le système des types intersection simples a une forme normale de tête (i.e., sa réduction de tête termine).

On montre que si $\Gamma \vdash u : \tau$ est dérivable, alors $u\theta \in RED_\tau$ pour toute substitution $\theta \in \underline{RED}_\Gamma$, comme dans le cours, où $\underline{RED}_\Gamma$ est l'ensemble des substitutions qui à chaque variable x apparaissant comme $x : \mu$ dans Γ associe un terme de RED_μ.

C'est une récurrence sur la dérivation. On utilise la définition de $\underline{RED}_\Gamma$ dans le cas de (Ax), la définition de $RED_{\mu\to\tau}$ dans le cas de (App), et la ***Question 13*** *dans le cas de (Abs).*

Nous allons démontrer la réciproque. Pour ceci, nous admettrons quelques résultats faciles, mais ennuyeux :

— **Affaiblissement 1** : si l'on peut dériver $\Gamma \vdash s : \tau$, on peut aussi dériver $\Gamma, \Delta \vdash s : \tau$;

— **Affaiblissement 2** : si l'on peut dériver $\Gamma, x : \mu \vdash s : \tau$ alors on peut aussi dériver $\Gamma, x : \mu \cup \mu' \vdash s : \tau$, où $\cup$ est l'union multi-ensemble (la concaténation de listes, à permutation près) ;

— **Renforcement** : si l'on peut dériver $\Gamma, x : \mu \vdash s : \tau$ et que x n'est pas libre dans s, alors on peut dériver $\Gamma \vdash s : \tau$.

Question 15 Montrer que tout terme $u = \lambda x_1, \ldots, x_m \cdot x v_1 \ldots v_n$ en forme normale de tête est typable, autrement dit qu'il existe un jugement de la forme $\Gamma \vdash u : \tau$ dérivable dans la discipline des types intersection simples. On donnera explicitement Γ et τ selon les différents cas considérés, et une explication sommaire des « types » à donner aux variables x_i, mais pas la peine de me dessiner une dérivation de typage bien formelle.

Donnez-moi la solution la plus simple possible !

On donne à x le type $\sigma = \underbrace{[] \to \cdots \to []}_{n\ fois} \to b$, où b est un type de base (par exemple ; on peut en fait prendre ce qu'on veut). Le « type » de x_i sera $\mu_i = []$, sauf si $x_i = x$, auquel cas on pose $\mu_i = [\sigma]$.

On pose $\tau = \mu_1 \to \cdots \to \mu_m \to b$.

Si x est l'une des x_i, alors on peut prendre Γ vide, sinon on prend $\Gamma = x : [\sigma]$ (non, pas $x : \sigma$, qui n'a pas de sens).

Question 16 Montrer que si on peut dériver $\Gamma \vdash s[x := t] : \tau$ (avec x absente de Γ), alors il existe des types $\sigma_1, \ldots, \sigma_n$ tels que l'on peut dériver $\Gamma, x : [\sigma_1, \cdots, \sigma_n] \vdash s : \tau$, et aussi $\Gamma \vdash t : \sigma_i$ pour chaque i, $1 \leq i \leq n$.

En plus de la preuve, si vous pouviez me donner une explication intuitive, ça serait super.

Intuitivement, les σ_i sont les types des différentes occurrences de x dans s. Ceci se démontre par une récurrence sur la dérivation donnée.

Si la dernière règle est (Var), et que s est la variable x, on peut dériver $\Gamma, x : [\sigma_1] \vdash s : \tau$, avec $\sigma_1 = \tau$, et $\Gamma \vdash t : \sigma_1$ puisque $s[x := t] = t$ et $\sigma_1 = \tau$. Si c'est une autre variable, on peut dériver $\Gamma, x : [] \vdash s : \tau$.

Si la dernière règle est (App), et que s est une application uv, alors on a pu dériver $\Gamma \vdash u[x := t] : \mu \to \tau$ et $\Gamma \vdash v[x := t] : \varphi_i$ pour chaque i, $1 \leq i \leq m$, où $\mu = [\varphi_1, \cdots, \varphi_m]$.

*Par hypothèse de récurrence, on obtient une dérivation de $\Gamma, x : \mu_0 \vdash u : \mu \to \tau$, et m dérivations de $\Gamma, x : \mu_i \vdash v : \varphi_i$, plus les dérivations appropriées de typage $\Gamma \vdash t : \sigma$, pour chaque σ de l'un quelconque des multi-ensembles μ_i, $0 \leq i \leq m$. On pose $\mu' = \mu_0 \cup \mu_1 \cup \cdots \cup \mu_m$. Par **Affaiblissement 2**, on obtient des dérivations de $\Gamma, x : \mu' \vdash u : \mu \to \tau$ et de $\Gamma, x : \mu' \vdash v : \varphi_i$ pour chaque i. On utilise (App), et on obtient une dérivation de $\Gamma, x : \mu' \vdash uv : \tau$.*

Si la dernière règle est (Abs), alors s est de la forme $\lambda z.u$, où l'on prend soit d'α-renommer pour s'assurer que $z \neq x$ et que z ne soit pas libre dans t. De plus, on a $\tau = \mu \to \tau'$, et une dérivation de $\Gamma, z : \mu \vdash u[x := t] : \tau'$. Par hypothèse de récurrence, on obtient une dérivation de $\Gamma, z : \mu, x : \mu_0 \vdash u : \tau'$, avec $\mu_0 = [\sigma_1, \cdots, \sigma_m]$, et m dérivations de $\Gamma, z : \mu \vdash t : \sigma_i$, $1 \leq i \leq m$. Par Renforcement, comme z n'est pas libre dans t, on obtient m dérivations de $\Gamma \vdash t : \sigma_i$, $1 \leq i \leq m$. Et par (Abs), on obtient une dérivation de $\Gamma, x : \mu_0 \vdash \lambda z.u : \mu \to \tau'$, c'est-à-dire de $\Gamma, x : \mu_0 \vdash s : \tau$.

Question 17 Montrer que si on peut dériver $\Gamma \vdash s[x := t] : \tau$, alors on peut aussi dériver $\Gamma \vdash (\lambda x.s)t : \tau$.

Si x apparaît dans Γ, on peut choisir une variable fraîche y, donc absente de Γ (et différente de x, ni libre dans s), et l'hypothèse se réécrit : on peut dériver $\Gamma \vdash s[x := y][y := t] : \tau$. À α-renommage près ($\lambda x.s$ et $\lambda y.s[x := y]$ sont α-équivalents), on peut donc supposer que x est absente de Γ.

*Par la **Question 16**, il existe des types $\sigma_1, \ldots, \sigma_n$ tels que l'on peut dériver $\Gamma, x : [\sigma_1, \cdots, \sigma_n] \vdash s : \tau$, et que l'on ait aussi des dérivations de $\Gamma \vdash t : \sigma_i$ pour chaque i, $1 \leq i \leq n$. On obtient la dérivation souhaitée en appliquant (Abs) sur la première, puis (App) en combinaison avec les autres.*

Question 18 Montrer que si $u \to_h u'$ et si l'on peut dériver $\Gamma \vdash u' : \tau$, alors on peut aussi dériver $\Gamma \vdash u : \tau$. On pourra utiliser l'abréviation $\Gamma \vdash v : \mu'$ pour dire que $\Gamma \vdash v : \varphi$ est dérivable pour chaque type $\varphi \in \mu'$.

*On écrit $u = \lambda x_1, \ldots, x_m.(\lambda x.s)tt_2 \cdot t_n$, et $u' = \lambda x_1, \ldots, x_m.s[x := t]t_2 \cdot t_n$. Par analyse de la forme de la dérivation de $\Gamma \vdash u' : \tau$, on a nécessairement une sous-dérivation de $\Gamma, x_1 : \mu_1, \ldots, x_m : \mu_m \vdash s[x := t] : \mu'_2 \to \cdots \mu'_n \to \tau$, et des sous-dérivations de $\Gamma, x_1 : \mu_1, \ldots, x_m : \mu_m \vdash t_i : \mu'_i$, $2 \leq i \leq n$. Par la **Question 17**, la première sous-dérivation donne lieu à une autre dérivation de $\Gamma, x_1 : \mu_1, \ldots, x_m : \mu_m \vdash (\lambda x.s)t : \mu'_2 \to \cdots \mu'_n \to \tau$, et en appliquant (App) $n-1$ fois puis (Abs) m fois, on obtient une dérivation de $\Gamma \vdash u : \tau$.*

Question 19 Conclure que les λ-termes qui ont une forme normale de tête sont exactement ceux qui sont typables dans la discipline des types intersection simples.

*Une direction est la **Question 14**. Réciproquement, si $u = u_0 \to_h \cdots \to_n u_n$ avec u_n en forme normale de tête, alors u_n est typable par la **Question 15**, et donc $u_{n-1}, \ldots, u_0$, de proche en proche, par la **Question 18**.*