Le λ-calcul typé avec un seul lieur

Fairouz Kamareddine

Université de Heriot-Watt,
Edimbourgh, Ecosse

13 October 2005

Les types simples

- Soient $f : \mathbb{N} \to \mathbb{N}$ et $g_f : \mathbb{N} \to \mathbb{N}$ tel que $g_f(x) = f(f(x))$.
  Soit $F_{\mathbb{N}} : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$ tel que
  $F_{\mathbb{N}}(f)(x) = g_f(x) = f(f(x))$.
- Dans le lambda calcul typé de Church on écrit la fonction $F_{\mathbb{N}}$ comme suit:

$$\lambda_{f:\mathbb{N}\to\mathbb{N}}.\lambda_{x:\mathbb{N}}.f(f(x))$$

- Si on veut la même fonction sur les Booléens $\mathcal{B}$, on écrit:

$$\begin{array}{rl} \text{la fonction } F_{\mathcal{B}} \text{ est} & \lambda_{f:\mathcal{B}\to\mathcal{B}}.\lambda_{x:\mathcal{B}}.f(f(x)) \\ \text{le type de la fonction } F_{\mathcal{B}} \text{ est} & (\mathcal{B} \to \mathcal{B}) \to (\mathcal{B} \to \mathcal{B}) \end{array}$$

Polymorphisme: le λ-calcul typé après Church

- A la place de répéter le travail, on prend $\alpha : *$ (α est un type quelconque) et on définit une fonction polymorphique F comme suit:

$$\lambda_{\alpha:*}.\lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x))$$

On donne à F le type:

$$\Pi_{\alpha:*}.(\alpha \to \alpha) \to (\alpha \to \alpha)$$

- Comme ça,
$F(\alpha) = \lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)) : (\alpha \to \alpha) \to (\alpha \to \alpha)$
- On peut instancier α selon notre besoin:
  - $F(\mathbb{N}) = \lambda_{f:\mathbb{N}\to\mathbb{N}}.\lambda_{x:\mathbb{N}}.f(f(x)) : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$
  - $F(\mathcal{B}) = \lambda_{f:\mathcal{B}\to\mathcal{B}}.\lambda_{x:\mathcal{B}}.f(f(x)) : (\mathcal{B} \to \mathcal{B}) \to (\mathcal{B} \to \mathcal{B})$
  - $F(\mathcal{B} \to \mathcal{B}) = \lambda_{f:(\mathcal{B}\to\mathcal{B})\to(\mathcal{B}\to\mathcal{B})}.\lambda_{x:(\mathcal{B}\to\mathcal{B})}.f(f(x)) :$
  $((\mathcal{B} \to \mathcal{B}) \to (\mathcal{B} \to \mathcal{B})) \to ((\mathcal{B} \to \mathcal{B}) \to (\mathcal{B} \to \mathcal{B}))$

- Comme ça, les types sont comme les fonctions:
  - On peut les construire par abstraction.
  - On peut les instancier
- Mais, pendant le passage des types simples aux types polymorphiques, on a oublié d'atapter la règle

$$(\beta) \qquad (\lambda_{x:A}.b)C \to b[x := C]$$

  à une règle semblable pour le Π:

$$(\Pi) \qquad (\Pi_{x:A}.B)C \to B[x := C]$$

- D'habitude, si $b : B$, on prend $(\lambda_{x:A}.b)C : B[x := C]$ à la place de prendre $(\lambda_{x:A}.b)C : (\Pi_{x:A}.B)C$
- Dans la literature, il y a plusieures études (comme le Automath de de Bruijn) qui montrent que la règle Π est utile.

- Il parait que le development des théories des types nous amène de plus en plus à adopter des rôle identiques pour les lieurs λ et Π. Est-ce qu'il y avait vraiment le besoin de les séparer? Je crois que la séparation est artificielle.
- Dans Automath, de Bruijn avait déjà identifié les deux lieurs mais dans un système de types beaucoups moins riche que les systèmes du cube. Il ecrivait: $[x : A]B$ pour les deux $\lambda_{x:A}.B$ et $\Pi_{x:A}.B$
- Quelles sont les propriétés des theories des types avec un seul lieur qui represente au même temps le λ et le Π?
- Est ce que le passage des systèmes des types avec deux lieurs à des systèmes avec un seul lieur est assez nécessaire comme le passage des types simples aux types polymorphiques et independents?
- Je crois que dès le début, il fallait pas se limiter aux types simples. Aussi, dès le début, il fallait pas distinguer entre les termes et les types.

Notation

- Soit $\mathcal{V} = \mathcal{V}^* \cup \mathcal{V}^\Box$ où $\mathcal{V}^* \cap \mathcal{V}^\Box = \emptyset$.
  $\mathcal{T} ::= * \mid \Box \mid \mathcal{V} \mid \lambda_{\mathcal{V}:\mathcal{T}}.\mathcal{T} \mid \Pi_{\mathcal{V}:\mathcal{T}}.\mathcal{T} \mid \mathcal{T}\mathcal{T}$
  $\mathcal{T}_\flat ::= * \mid \Box \mid \mathcal{V} \mid \flat_{\mathcal{V}:\mathcal{T}_\flat}.\mathcal{T}_\flat \mid \mathcal{T}_\flat\mathcal{T}_\flat$
  Notons que même que les systèmes modernes n'ont pas adopté l'unification des lieurs λ et Π, ils ont adopté (dans $\mathcal{T}$) un syntax commun dans un certain sense.
- Si $A \in \mathcal{T}$, on définit $\overline{A} \in \mathcal{T}_\flat$ par:
  $\overline{s} \equiv s, \overline{x} \equiv x, \overline{AB} \equiv \overline{A}\,\overline{B}, \overline{\lambda_{x:A}.B} \equiv \overline{\Pi_{x:A}.B} \equiv \flat_{x:\overline{A}}.\overline{B}$.
- Si $A \in \mathcal{T}_\flat$, on définit $A^\lambda \in \mathcal{T}$ par:
  $s^\lambda \equiv s, x^\lambda \equiv x, (AB)^\lambda \equiv A^\lambda B^\lambda$, $(\flat_{x:A}.B)^\lambda \equiv \lambda_{x:A^\lambda}.B^\lambda$
  Aussi, on définit $[A]$ par $\{A' \text{ in } \mathcal{T} \mid \overline{A'} \equiv A\}$.
- Un contexte Γ est de la forme: $x_1 : A_1, \ldots, x_n : A_n$.
  $\mathrm{DOM}\,(\Gamma) = \{x_1, \ldots x_n\}$.
- Dans $\mathcal{T}$ on définit: $\overline{\langle\rangle} \equiv \langle\rangle \qquad \overline{\Gamma, x : A} \equiv \overline{\Gamma}, x : \overline{A}$.
- Dans $\mathcal{T}_\flat$ on définit: $[\Gamma] \equiv \{\Gamma' \mid \overline{\Gamma'} \equiv \Gamma\}$.

- On a trois systèmes $(\mathcal{T}, \to_\beta)$, $(\mathcal{T}, \to_{\beta\Pi})$ et $(\mathcal{T}_\flat, \to_\flat)$ avec les axiomes:
  - $(\lambda_{x:A}.B)C \to_\beta B[x := C]$
  - $(\flat_{x:A}.B)C \to_\flat B[x := C]$
  - $(\Pi_{x:A}.B)C \to_\Pi B[x := C]$
  - Où $\to_{\beta\Pi} = \to_\beta \cup \to_\Pi$
- Church-Rosser: Soit $r \in \{\beta, \beta\Pi, \flat\}$.
  Si $B_1 \; {}_r\!\twoheadleftarrow A \twoheadrightarrow_r B_2$ alors $\exists C$ tel que $B_1 \twoheadrightarrow_r C \; {}_r\!\twoheadleftarrow B_2$.

Le β-cube: R_β = Règles de typage avec deux lieurs, $\to_\beta$ et (appl)

(axiom)	$\langle\rangle \vdash * : \Box$
(start)	$\dfrac{\Gamma \vdash A : s \qquad x^s \not\in \mathrm{DOM}(\Gamma)}{\Gamma, x^s{:}A \vdash x^s : A}$
(weak)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s \qquad x^s \not\in \mathrm{DOM}(\Gamma)}{\Gamma, x^s{:}C \vdash A : B}$
(Π)	$\dfrac{\Gamma \vdash A : s_1 \qquad \Gamma, x{:}A \vdash B : s_2 \qquad (s_1, s_2) \in \mathbf{R}}{\Gamma \vdash \Pi_{x:A}.B : s_2}$
(λ)	$\dfrac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash \Pi_{x:A}.B : s}{\Gamma \vdash \lambda_{x:A}.b : \Pi_{x:A}.B}$
(conv_β)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_\beta B'}{\Gamma \vdash A : B'}$
(appl)	$\dfrac{\Gamma \vdash F : \Pi_{x:A}.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[x{:=}a]}$

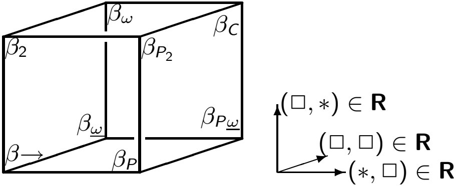

Figure: Le β-cube de Barendregt

Notre exemple dans le système **F** de Girard

- Si $x \notin FV(B)$ on écrit $A \to B$ à la place de $\Pi_{x:A}.B$.
- $\alpha : *, f : \alpha \to \alpha \vdash \lambda_{x:\alpha}.f(f(x)) : \alpha \to \alpha : *$
  (on a besoin de la règle $(*, *)$).
- $\alpha : * \vdash \lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)) : (\alpha \to \alpha) \to (\alpha \to \alpha) : *$
  (on a besoin de la règle $(*, *)$).
- $\vdash \lambda_{\alpha:*}.\lambda_{f:\alpha\to\alpha}.\lambda_{x:\alpha}.f(f(x)) : \Pi_{\alpha:*}.(\alpha \to \alpha) \to (\alpha \to \alpha) : *$
  (on a besoin de la règle $(\square, *)$).

Le π-cube: $R_\pi = R_\beta \setminus (\text{conv}_\beta) \cup (\text{conv}_{\beta\Pi}) =$ Règles de typage avec deux lieurs, $\rightarrow_{\beta\Pi}$ et (appl)

(axiom) (start) (weak) (Π) (λ) (appl)

$$(\text{conv}_{\beta\Pi}) \quad \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_{\beta\Pi} B'}{\Gamma \vdash A : B'}$$

Le π_i-cube: $R_{\pi_i} = R_\pi \setminus$ (appl) $\cup$ (newappl) = Règles de typage avec deux lieurs, $\rightarrow_{\beta\Pi}$ et (newappl)

(axiom) (start) (weak) (Π) (λ)	
(conv$_{\beta\Pi}$)	$\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_{\beta\Pi} B'}{\Gamma \vdash A : B'}$
(newappl)	$\dfrac{\Gamma \vdash F : \Pi_{x:A}.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : (\Pi_{x:A}.B)a}$

Le $\flat$-cube: $R_\flat$ = Règles de typage avec un seul lieur, $\to_\flat$ et (appl$\flat$)

(axiom) (start) (weak)

$$(\flat_1)\quad \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x{:}A \vdash B : s_2 \qquad (s_1, s_2) \in \mathbf{R}}{\Gamma \vdash \flat_{x:A}.B : s_2}$$

$$(\flat_2)\quad \frac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash \flat_{x:A}.B : s}{\Gamma \vdash \flat_{x:A}.b : \flat_{x:A}.B}$$

$$(\mathrm{conv}_\flat)\quad \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B =_\flat B'}{\Gamma \vdash A : B'}$$

$$(\mathrm{appl}\flat)\quad \frac{\Gamma \vdash F : \flat x{:}A.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[x{:=}a]}$$

Le $\flat_i$-cube: $R_{\flat_i} = R_\flat \setminus$ (appl$\flat$) $\cup$ (newappl$\flat$) = Règles de typage avec un seul lieur, $\rightarrow_\flat$ et (newappl$\flat$)

(axiom) (start) (weak) ($\flat_1$) ($\flat_2$) (conv$_\flat$)

(newappl$\flat$) $$\frac{\Gamma \vdash F : \flat x{:}A.B \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : (\flat x : A.B)a}$$

Six propriétés qu'on désire des systèmes des types

Soit r la relation de réduction dans le système en question.

- *Les types sont correctes (TC)*
  Si $\Gamma \vdash A : B$ alors $B \equiv \Box$ ou $\Gamma \vdash B : s$ pour $s \in \{*, \Box\}$.
- *Subject reduction (SR)*
  Si $\Gamma \vdash A : B$ et $A \twoheadrightarrow_r A'$ alors $\Gamma \vdash A' : B$.
- *Préservation des types (PT)*
  Si $\Gamma \vdash A : B$ et $B \twoheadrightarrow_r B'$ alors $\Gamma \vdash A : B'$.
- *Normalisation forte (NF)*
  Si $\Gamma \vdash A : B$ alors $\mathrm{SN}_{\rightarrow_r}(A)$ et $\mathrm{SN}_{\rightarrow_r}(B)$.
- *Les sous-termes sont typables (STT)* Si A est $\vdash$-légale et si C est un sous-terme de A alors C est $\vdash$-légale.
- *Unicité des types*
  - *(UT1)* Si $\Gamma \vdash A_1 : B_1$ et $\Gamma \vdash A_2 : B_2$ et $\Gamma \vdash A_1 =_r A_2$, alors $\Gamma \vdash B_1 =_r B_2$.
  - *(UT2)* If $\Gamma \vdash B_1 : s$, $B_1 =_r B_2$ and $\Gamma \vdash A : B_2$ then $\Gamma \vdash B_2 : s$.

Est ce qu'il ya dans les systèmes avec deux lieurs, des Π-redexes dans les termes typables?

- Lemme:
  - Si $\Gamma \vdash_\beta A : B$ ou $\Gamma \vdash_\pi A : B$ alors Γ, A et B ne contiennent pas des Π-redexes.
  - Si $\Gamma \vdash_{\pi_i} A : B$ alors Γ et A ne contiennent pas des Π-redexes et B est le seul Π-redexe possible dans B.

La corespondence entre les β-, π- et π_i-cubes

- Lemme:
  - $\Gamma \vdash_\beta A : B$ si et seulement si $\Gamma \vdash_\pi A : B$
  - Si $\Gamma \vdash_\beta A : B$ alors $\Gamma \vdash_{\pi_i} A : B$.
  - Si $\Gamma \vdash_{\pi_i} A : B$ alors $\Gamma \vdash_\beta A : [B]_\Pi$
    où $[B]_\Pi$ est la Π-forme normale de B.
- Lemme: Le β-cube et le π-cube satisfont les six propriétés qu'on désire des systèmes des types.

Le π_i-cube

- Le π_i-cube perd trois de ces six propriétés:
  Soit $\Gamma = z : *, x : z$. On a que $\Gamma \vdash_{\pi_i} (\lambda_{y:z}.y)x : (\Pi_{y:z}.z)x$.
  - *On n'a pas TC* $(\Pi_{y:z}.z)x \not\equiv \Box$ et $\Gamma \not\vdash_{\pi_i} (\Pi_{y:z}.z)x : s$.
  - *On n'a pas SR* $(\lambda_{y:z}.y)x \to_{\beta\Pi} x$ mais $\Gamma \not\vdash_{\pi_i} x : (\Pi_{y:z}.z)x$.
  - *On n'a pas UT2* $\vdash_{\pi_i} * : \Box$, $* =_{\beta\Pi} (\Pi_{z:*}.*)\alpha$,
    $\alpha : * \vdash_{\pi_i} (\lambda_{z:*}.*)\alpha : (\Pi_{z:*}.*)\alpha$ and $\not\vdash_{\pi_i} (\Pi_{z:*}.*)\alpha : \Box$
- Mais on a:
  - *On a UT1*
  - *On a STT*
  - *On a PT*
  - *On a NF*
  - *On a une version faible de TC* Si $\Gamma \vdash_{\pi_i} A : B$ et *B n'est pas un Π-redexe* alors ou bien $B \equiv \Box$ ou bien $\Gamma \vdash_{\pi_i} B : s$.
  - *On a une version faible de SR* Si $\Gamma \vdash_{\pi_i} A : B$, *$B$ n'est pas un Π-redexe* et $A \twoheadrightarrow_{\beta\Pi} A'$ alors $\Gamma \vdash_{\pi_i} A' : B$.

Si on élimine la distinction entre λ et Π, il restera toujours possible de déduire le rôle de chaque lieur

- Lemme: Supposons que $\Gamma \vdash_\beta A_1 : B_1$ et $\Gamma \vdash_\beta A_2 : B_2$.
  1. Si $\overline{A_1} \equiv \overline{A_2}$ et $B_1 =_\beta B_2$ alors $A_1 \equiv A_2$ et $B_1 \equiv B_2$.
  2. *Dans un terme $\vdash_\beta$-légale de type s, il y a une manière unique de placer les λs et les Πs*
     Si $B_1 \equiv s_1$, $B_2 \equiv s_2$ et $\overline{A_1} \equiv \overline{A_2}$ alors $A_1 \equiv A_2$ et $s_1 \equiv s_2$.
  3. *Dans un contexte $\vdash_\beta$-légale, il y a une manière unique de placer les λs et les Πs*
     Si Γ_1 et Γ_2 sont $\vdash_\beta$-légales et if $\overline{\Gamma_1} \equiv \overline{\Gamma_2}$ alors $\Gamma_1 \equiv \Gamma_2$.
  4. *Dans un type $\vdash_\beta$-légale, il y a une manière unique de placer les λs et les Πs*
     Si $\overline{B_1} \equiv \overline{B_2}$ alors $B_1 \equiv B_2$.
  5. Si $\overline{A_1} \equiv \overline{A_2}$ et $\overline{B_1} \equiv \overline{B_2}$ alors $A_1 \equiv A_2$ et $B_1 \equiv B_2$.
  6. Si $B_1 \equiv s_1$, $B_2 \equiv s_2$, $\overline{A_1} =_\flat \overline{A_2}$ alors $A_1 =_\beta A_2$.

Un algorithme qui transforme un $\flat$-typage en un β-typage

Soit $\Gamma \vdash_\flat A : B$. On définit $(\Gamma \vdash_\flat A : B)^{-1} \in [\Gamma] \times [A] \times [B]$ par:

$$
\begin{array}{ll}
(\langle\rangle \vdash_\flat * : \Box)^{-1} & = (\langle\rangle, *, \Box) \\
(\Gamma, x^s : A \vdash_\flat * : \Box)^{-1} & = (\Gamma', x^s : A', *, \Box) \text{ où } (\Gamma \vdash_\flat A : s)^{-1} = (\Gamma', A', s) \\
(\Gamma \vdash_\flat x^s : C)^{-1} & = (\Gamma', x^s, C') \text{ où } (\Gamma \vdash_\flat C : s)^{-1} = (\Gamma', C', s) \\
(\Gamma \vdash_\flat \flat_{x:A}.B : C)^{-1} & = \begin{cases} (\Gamma', \Pi_{x:A'}.B', C') & \text{si } C =_\flat s_2 \text{ et i.} \\ (\Gamma', \lambda_{x:A'}.B', C') & \text{si } C =_\flat \flat_{x:A}.D \text{ et ii.} \end{cases} \\
(\Gamma \vdash_\flat Fa : C)^{-1} & = (\Gamma', F'a', C') \text{ où iii.}
\end{array}
$$

Où i, ii, et iii, sont évidents. On écrit simplement la condition i. Pour les autres, voir l'article.

i
- $(\Gamma \vdash_\flat A : s_1)^{-1} = (\Gamma', A', s_1)$ pour un s_1 tel que $(s_1, s_2) \in \mathbf{R}$,
- $(\Gamma, x : A \vdash_\flat B : s_2)^{-1} = (\Gamma'', x : A'', B', s_2)$
- si $C \equiv s_2$ alors $C' \equiv s_2$ sinon
  si $C \not\equiv s_2$ alors $(\Gamma \vdash_\flat C : s)^{-1} = (\Gamma''', C', s)$ pour un certain s.

L'isomorphisme

- Theorème:
  1. si $\Gamma \vdash_\beta A : B$ alors $\overline{\Gamma} \vdash_\flat \overline{A} : \overline{B}$.
  2. si $\Gamma \vdash_\flat A : B$ alors
     - il existe un unique $\Gamma' \in [\Gamma]$ tel que Γ' is $\vdash_\beta$-legal, et
     - il existe un unique $A' \in [A]$, un unique $B' \in [B]$ tel que $\Gamma' \vdash_\beta A' : B'$.

     En plus, Γ', A' et B' sont construits par $^{-1}$ où $(\Gamma \vdash_\flat A : B)^{-1} = (\Gamma', A', B')$.
  3. La vérification des types dans le β-cube est équivalente à la vérification des types dans le $\flat$-cube
  4. La dérivation des types dans le β-cube est équivalente à la dérivation des types dans le $\flat$-cube

Le $\flat$-cube

- Le $\flat$-cube a cinq et demi des six propriétés:
  - *On a TC*
  - *On a SR*
  - *On a STT*
  - *On a PT*
  - *On a NF*
  - *On a UT2*
  - *Bien Sûre on n'a pas UT1 (et on ne le veut pas)*.
    En utilisant $(\Box,\Box)$: $\vdash_\beta \lambda_{x:*}.x : \Pi_{x:*}.*$ et $\vdash_\flat \flat_{x:*}.x : \flat_{x:*}.*$.
    En utilisant $(\Box,*)$: $\vdash_\beta \Pi_{x:*}.x : *$ et $\vdash_\flat \flat_{x:*}.x : *$.
    Notez que: $\flat_{x:*}.* \neq_\flat *$.

On a une multiplicité organisée des types dans le $\flat$-cube

- Soient $\mathsf{SN}_{\to_\flat}(B_1)$ et $\mathsf{SN}_{\to_\flat}(B_2)$. On dit que $B_1 \stackrel{\diamond}{=}_\flat B_2$ ssi $\mathrm{nf}_\flat(B_1) \equiv \flat^{i:1..n_1}_{x_i:F_i}.B$ et $\mathrm{nf}_\flat(B_2) \equiv \flat^{i:1..n_2}_{x_i:F_i}.B$, où $n_1, n_2 \geq 0$.
- *Lemme (MOT)* Si $\Gamma \vdash_\flat A_1 : B_1$ et $\Gamma \vdash_\flat A_2 : B_2$ et $A_1 =_\flat A_2$, alors $B_1 \stackrel{\diamond}{=}_\flat B_2$.
- Alors le $\flat$-cube est élegant, et a toutes les propriétés (on ne veut plus UT1, MOT suffit.

Exemple

- Soit $A \equiv \flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.\flat_{x_3:C}.\flat_{x_4:x_2x_3}.x_2x_3$
  Soit $B \in \{\flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.\flat_{x_3:C}.\flat_{x_4:x_2x_3}.*,$
  $\flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.\flat_{x_3:C}.*,$
  $\flat_{x_1:*}.\flat_{x_2:\flat_{y:C}.*}.*,$
  $\flat_{x_1:*}.*,$
  $*\}.$
  On a que $\vdash_\flat A : B$
- Dans le β-cube on a:

$$
\begin{array}{llll}
\vdash_\beta \lambda_{x_1:*}.\lambda_{x_2:\Pi_{y:C}.*}.\lambda_{x_3:C}.\lambda_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}. & * \\
\vdash_\beta \lambda_{x_1:*}.\lambda_{x_2:\Pi_{y:C}.*}.\lambda_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}. & * \\
\vdash_\beta \lambda_{x_1:*}.\lambda_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}. & * \\
\vdash_\beta \lambda_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & \Pi_{x_1:*}. & * \\
\vdash_\beta \Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3 & : & & *
\end{array}
$$

Notons que seulement $\Pi_{x_1:*}.\Pi_{x_2:\Pi_{y:C}.*}.\Pi_{x_3:C}.\Pi_{x_4:x_2x_3}.x_2x_3$ peut être un type. Les cinq autres termes ne le peuvent pas.

Conclusion

- Si le système de typage peut faire tout le travail (comme on a vu dans le $\flat$-cube), pourquoi séparer le niveau des termes de celui des types?
- Les termes et les types ne sont pas différents. La seule différence est la relation qui existe entre eux. Dans $A : B$, le A à gauche represente un terme, le B à droite represente un type.
- Même si vous ne voulez pas utiliser ce système, son existence a déja une significance.
- Sa présence montre que la division entre termes et types est artificielle. Il y avait jamais un besoin de cette division.
- Il faut explorer l'idée que les preuves et les propositions sont la même chose et ça vaut la peine d'étudier les implications sur la sémantique logique, de programmation et géneral.