



HAL
open science

I-types of system F

Karim Nour

► **To cite this version:**

Karim Nour. I-types of system F. Informatique Théorique et Applications, 2001, 35, pp.223-237.
hal-00380940

HAL Id: hal-00380940

<https://hal.science/hal-00380940v1>

Submitted on 4 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LES I -TYPES DU SYSTÈME \mathcal{F}

I -TYPES OF SYSTEM \mathcal{F}

Karim NOUR¹

Résumé. - *Nous démontrons dans ce papier que les types du système \mathcal{F} habités uniquement par des λI -termes (les I -types) sont à quantificateur positif. Nous présentons ensuite des conséquences de ce résultat et quelques exemples.*

Abstract. - *We prove in this paper that the types of system \mathcal{F} inhabited uniquely by λI -terms (the I -types) have a positive quantifier. We give also consequences of this result and some examples.*

Mathematics Subject Classification : 03B40, 68Q60

Keywords : λI -calculus - system \mathcal{F} - I -type.

1 Introduction

Le λI -calcul est une restriction du λ -calcul où on ne peut abstraire sur une variable que si elle est libre dans le terme. Dans ce calcul, on a l'équivalence entre "la normalisation faible" et "la normalisation forte". H. Barendregt a démontré dans [1] que le λI -calcul est suffisant pour représenter les types de données courants et programmer toutes les fonctions calculables.

J.-Y. Girard a démontré dans [3] que l'on peut aussi représenter les types de données courants dans le système de typage \mathcal{F} . Cependant les fonctions représentables sur ces types sont celles dont les preuves de terminaison se font dans l'arithmétique de Peano du second ordre.

Les types du système \mathcal{F} qui représentent les types de données courants sont tous habités par au moins un λ -terme qui n'est pas un λI -terme. Une question alors se pose : peut-on représenter les types de données courants par des types du système \mathcal{F} habités uniquement par des λI -termes (ces types sont appelés des I -types)?

Le but de cet article est d'étudier les I -types du système \mathcal{F} . On démontre que tout quantificateur du second ordre d'un I -type est positif. On donne ensuite une application de ce résultat sur les types entrées et les types sorties définis dans [2]. On montre aussi que pour vérifier si un type est un I -type, on peut se limiter aux types simples (types sans quantificateurs). On présente enfin quelques exemples de I -types.

¹LAMA - Equipe de Logique - Université de Chambéry - 73376 Le Bourget du Lac - Email nour@univ-savoie.fr

2 Notations et définitions

2.1 Quelques définitions du λ -calcul

Étant donnés des λ -termes t, u, u_1, \dots, u_n , l'application de t à u sera notée $(t)u$, et $(\dots((t)u_1)\dots)u_n$ sera noté $(t)u_1\dots u_n$.

Si t est un λ -terme, on désigne par $Fv(t)$ l'ensemble de ses variables libres.

Si u et v sont des λ -termes, alors on note $\langle u, v \rangle$ le λ -terme $\lambda x(x)uv$ où x ne figure pas dans u et v .

On note $\mathbf{id} = \lambda xx$, $\mathbf{0} = \lambda x\lambda yy$ et $\mathbf{1} = \lambda x\lambda yx$. L'entier de Church \underline{n} est le λ -terme $\lambda x\lambda f(f)\dots(f)x$ ((f) répété n fois).

On note \rightarrow_β (resp. $\rightarrow_{\beta\eta}$) la β -réduction (resp. la $\beta\eta$ -réduction).

Un λ -terme est dit β -normal (resp. $\beta\eta$ -normal) s'il ne contient pas de β -redex (ni de β -redex ni de η -redex).

Un λ -terme est dit *résoluble* si sa réduction de tête termine.

2.2 Le λI -calcul

Définitions 1) L'ensemble des λI -termes (noté ΛI) est définie par induction :

- Si x est une variable, alors $x \in \Lambda I$.
 - Si $u, v \in \Lambda I$, alors $(u)v \in \Lambda I$.
 - Si $u \in \Lambda I$ et x est libre dans u , alors $\lambda xu \in \Lambda I$.
- 2) Un λK -terme est un λ -terme qui n'est pas un λI -terme.

Le λI -calcul possède les propriétés suivantes (voir [1]).

Théorème 2.1 1) Si $t \in \Lambda I$, et $t \rightarrow_{\beta\eta} t'$, alors $t' \in \Lambda I$ et $Fv(t) = Fv(t')$.

2) Un λI -terme est fortement normalisable ssi il est faiblement normalisable.

Notation On note $\bar{0} = \lambda x\lambda f(((x)\mathbf{id})\mathbf{id})f$ et, pour tout $n \in \mathbb{N}$, $\overline{n+1} = \underline{n+1}$.

Le théorème suivant montre qu'en λI -calcul on peut représenter toutes les fonctions partielles récursives (voir [1]).

Théorème 2.2 Pour toute fonction partielle récursive $f : \mathbb{N}^k \rightarrow \mathbb{N}$, il existe un λI -terme \bar{f} tel que :

- $(\bar{f})\overline{n_1}\dots\overline{n_k} \rightarrow_\beta \overline{f(n_1, \dots, n_k)}$ si $f(n_1, \dots, n_k)$ est définie.
- $(\bar{f})\overline{n_1}\dots\overline{n_k}$ est non résoluble sinon.

2.3 Le système \mathcal{F}

Définitions 1) Les *types* du système \mathcal{F} sont construits à partir des variables de type X, Y, Z, \dots en utilisant les opérations suivantes :

- Si E et F sont des types, alors $E \rightarrow F$ est un type.
- Si E est un type, et X est une variable de type, alors $\forall X E$ est un type.

On définit d'une manière usuelle les variables *libres* et *liées* d'un type.

2) Soient A, F deux types et X une variable. Le type $A[F/X]$ est obtenu en remplaçant dans A toute occurrence de la variable X par le type F .

3) Les règles de typage du système \mathcal{F} sont les suivantes :

$$(ax) x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} x_i : A_i \quad (1 \leq i \leq n)$$

$$(\rightarrow_i) \frac{\Gamma, x : A \vdash_{\mathcal{F}} t : B}{\Gamma \vdash_{\mathcal{F}} \lambda x t : A \rightarrow B}$$

$$(\rightarrow_e) \frac{\Gamma \vdash_{\mathcal{F}} u : A \rightarrow B \quad \Gamma \vdash_{\mathcal{F}} v : A}{\Gamma \vdash_{\mathcal{F}} (u)v : B}$$

$$(\forall_i) \frac{\Gamma \vdash_{\mathcal{F}} t : A \quad X \text{ non libre dans } \Gamma}{\Gamma \vdash_{\mathcal{F}} t : \forall X A}$$

$$(\forall_e) \frac{\Gamma \vdash_{\mathcal{F}} t : \forall X A \quad G \text{ est un type}}{\Gamma \vdash_{\mathcal{F}} t : A[G/X]}$$

On écrit $\Gamma \vdash_{\mathcal{F}} t : A$ si t est typable de type A dans le contexte Γ .

Le système \mathcal{F} possède les propriétés suivantes (voir [3] et [4]).

Théorème 2.3 1) *Un type est préservé durant une β -réduction.*

2) *Un λ -terme typable est fortement normalisable.*

Définition Un type D du système \mathcal{F} est dit *propre* ssi si $\forall X E$ est un sous-type de D alors X est libre dans E .

Dans la suite on restreint le système \mathcal{F} aux types propres. Pour cela on modifie légèrement la règle (\forall_e) en demandant à X d'être libre dans A .

Les lemmes suivants seront utilisés dans la suite (voir [3] et [4]).

Définition Soient $Id = \forall X \{X \rightarrow X\}$, $Bool = \forall X \{X \rightarrow (X \rightarrow X)\}$ et $Ent = \forall X \{X \rightarrow [(X \rightarrow X) \rightarrow X]\}$.

Lemme 2.1 *Soit t un λ -terme β -normal.*

1) $\vdash_{\mathcal{F}} t : Id$ ssi $t = \mathbf{id}$.

2) $\vdash_{\mathcal{F}} t : Bool$ ssi $t = \mathbf{0}$ ou $t = \mathbf{1}$.

3) $\vdash_{\mathcal{F}} t : Ent$ ssi il existe $n \in \mathbb{N}$ tel que $t = \underline{n}$ ($n \geq 0$).

Définition Si A et B sont deux types alors on note $A \wedge B$ le type $\forall X \{(A \rightarrow (B \rightarrow X)) \rightarrow X\}$ où X est une variable qui ne figure pas dans A et B .

Lemme 2.2 1) *Si $\Gamma \vdash_{\mathcal{F}} u : A$ et $\Gamma \vdash_{\mathcal{F}} v : B$, alors $\Gamma \vdash_{\mathcal{F}} \langle u, v \rangle : A \wedge B$.*

2) *Si $\Gamma \vdash_{\mathcal{F}} t : A \wedge B$, alors $\Gamma \vdash_{\mathcal{F}} (t)\mathbf{1} : A$ et $\Gamma \vdash_{\mathcal{F}} (t)\mathbf{0} : B$.*

Définition Un *contexte* $C\langle \rangle$ est un terme du λ -calcul avec un trou (une seule occurrence d'une constante spéciale du λ -calcul). On note $C\langle u \rangle$ le résultat de la substitution du trou de $C\langle \rangle$ par u et ceci sans renommage des variables liées de $C\langle \rangle$.

Lemme 2.3 Si dans le typage $\Gamma \vdash_{\mathcal{F}} C\langle u \rangle : A$ on a $\Gamma' \vdash_{\mathcal{F}} u : B$, alors pour tout λ -terme v tel que $\Gamma' \vdash_{\mathcal{F}} v : B$ on a $\Gamma \vdash_{\mathcal{F}} C\langle v \rangle : A$

Preuve Il suffit de remplacer dans le typage $\Gamma \vdash_{\mathcal{F}} C\langle u \rangle : A$ l'arbre de typage de $\Gamma' \vdash_{\mathcal{F}} u : B$ par celui de $\Gamma' \vdash_{\mathcal{F}} v : B$. \spadesuit

Notations Pour simplifier on note la formule $\forall X_1 \dots \forall X_n F$ par $\forall \overline{X} F$ et la formule $A_1 \rightarrow (A_2 \rightarrow \dots (A_n \rightarrow A) \dots)$ par $A_1, A_2, \dots, A_n \rightarrow A$.

3 Les I -types du système \mathcal{F}

Définition Un type clos D du système \mathcal{F} est dit *I -type* ssi si t est un λ -terme clos β -normal tel que $\vdash_{\mathcal{F}} t : D$, alors t est un λI -terme.

Nous allons démontrer qu'un I -type du système \mathcal{F} (habité par au moins un λ -terme) est un type à quantificateur positif. Nous présentons tout d'abord, sur des exemples, la méthode adoptée.

Exemples Soient $E = \forall X \{ \forall Y (Id \rightarrow Y) \rightarrow Id \}$ et $F = \forall X \{ \forall Y (Y \rightarrow Id) \rightarrow Id \}$. Ces types ne sont pas à quantificateur positif car le quantificateur $\forall Y$ occure négativement dans E et F . Nous allons montrer comment fabriquer à partir d'un λI -terme de type E (resp. de type F) un λK -terme de même type. Considérons les deux typages suivants :

$$\frac{\frac{\frac{x : \forall Y (Id \rightarrow Y) \vdash x : \forall Y (Id \rightarrow Y)}{x : \forall Y (Id \rightarrow Y) \vdash x : Id \rightarrow Id} \quad \frac{}{\vdash_{\mathcal{F}} \mathbf{id} : Id}}{x : \forall Y (Id \rightarrow Y) \vdash (x)\mathbf{id} : Id}}{\vdash \lambda x(x)\mathbf{id} : \forall Y (Id \rightarrow Y) \rightarrow Id}}{\vdash \lambda x(x)\mathbf{id} : E}$$

$$\frac{\frac{\frac{x : \forall Y (Y \rightarrow Id) \vdash x : \forall Y (Y \rightarrow Id)}{x : \forall Y (Y \rightarrow Id) \vdash x : Id \rightarrow Id} \quad \frac{}{\vdash_{\mathcal{F}} \mathbf{id} : Id}}{x : \forall Y (Y \rightarrow Id) \vdash (x)\mathbf{id} : Id}}{\vdash \lambda x(x)\mathbf{id} : \forall Y (Y \rightarrow Id) \rightarrow Id}}{\vdash \lambda x(x)\mathbf{id} : F}$$

Les deux typages suivants donnent des λK -termes de type E et F :

$$\begin{array}{c}
\frac{x : \forall Y (Id \rightarrow Y) \vdash x : \forall Y (Id \rightarrow Y)}{x : \forall Y (Id \rightarrow Y) \vdash x : Id \rightarrow (Bool \rightarrow Id)} \quad \frac{}{\vdash \mathbf{id} : Id} \\
\hline
\frac{x : \forall Y (Id \rightarrow Y) \vdash (x)\mathbf{id} : Bool \rightarrow Id}{x : \forall Y (Id \rightarrow Y) \vdash (x)\mathbf{id} \mathbf{0} : Id} \quad \frac{}{\vdash \mathbf{0} : Bool} \\
\hline
\frac{}{\vdash \lambda x(x)\mathbf{id} \mathbf{0} : \forall Y (Id \rightarrow Y) \rightarrow Id} \\
\hline
\vdash \lambda x(x)\mathbf{id} \mathbf{0} : E
\end{array}$$

$$\begin{array}{c}
\frac{x : \forall Y (Y \rightarrow Id) \vdash x : \forall Y (Y \rightarrow Id)}{x : \forall Y (Y \rightarrow Id) \vdash x : (Bool \wedge Id) \rightarrow Id} \quad \frac{}{\vdash \mathbf{0} : Bool} \quad \frac{}{\vdash \mathbf{id} : Id} \\
\frac{}{\vdash_{\mathcal{F}} \langle \mathbf{0}, \mathbf{id} \rangle : Bool \wedge Id} \\
\hline
\frac{x : \forall Y (Y \rightarrow Id) \vdash (x)\langle \mathbf{0}, \mathbf{id} \rangle : Id}{\vdash \lambda x(x)\langle \mathbf{0}, \mathbf{id} \rangle : \forall Y (Y \rightarrow Id) \rightarrow Id} \\
\hline
\vdash \lambda x(x)\langle \mathbf{0}, \mathbf{id} \rangle : F
\end{array}$$

Remarquons que dans le premier typage on a remplacé la variable Y (qui est en position négative dans E) par $Bool \rightarrow Id$ et dans le deuxième typage on a remplacé Y (qui est en position positive dans F) par $Bool \wedge Id$. On peut vérifier facilement qu'en remplaçant Y par $Bool \rightarrow (Bool \wedge Id)$ on trouve aussi des λK -termes de type E et F . Signalons enfin qu'il suffit de remplacer le type $Bool$ par un type quelconque habité par un λK -terme ou par une nouvelle variable de type. Nous allons montrer qu'on peut généraliser cette méthode pour un type quelconque. Ceci nécessite l'introduction de plusieurs notions et la démonstration de plusieurs résultats.

Définition Pour tout type A et pour toute variable X , on définit deux constantes du λ -calcul $\mathcal{U}_{A,X}$ et $\mathcal{V}_{A,X}$. Le $\lambda_{\mathcal{UV}}$ -calcul est obtenu en considérant les règles de réduction suivantes :

$$\begin{array}{c}
(\lambda x u)v \hookrightarrow_{\beta} u[v/x] \\
\frac{}{(\mathcal{U}_{Y,X})t \hookrightarrow_u t \text{ si } Y \neq X} \quad \frac{}{(\mathcal{V}_{Y,X})t \hookrightarrow_v t \text{ si } Y \neq X} \\
(\mathcal{U}_{B \rightarrow C, X})t \hookrightarrow_u \lambda y(\mathcal{U}_{C, X})(t)(\mathcal{V}_{B, X})y \quad (\mathcal{V}_{B \rightarrow C, X})t \hookrightarrow_v \lambda y(\mathcal{V}_{C, X})(t)(\mathcal{U}_{B, X})y \\
(\mathcal{U}_{\forall Y B, X})t \hookrightarrow_u (\mathcal{U}_{B, X})t \quad (\mathcal{V}_{\forall Y B, X})t \hookrightarrow_v (\mathcal{V}_{B, X})t
\end{array}$$

On écrit $t \hookrightarrow t'$ si t' est obtenu à partir de t en appliquant un nombre fini de fois les règles précédentes.

Lemme 3.1 *La clôture réflexive et transitive de la réduction \hookrightarrow_u (resp. de \hookrightarrow_v) est fortement normalisable.*

Preuve On définit, par induction sur les $\lambda_{\mathcal{UV}}$ -termes, une notion de longueur : $N(x) = 0$, $N((u)v) = N(u) + N(v)$, $N(\lambda x u) = N(u)$ et $N(\mathcal{U}_{A, X}) = N(\mathcal{V}_{A, X}) = L(A)$ où $L(A)$ est le nombre des connecteurs logiques de A . Il est clair que si $t \hookrightarrow_u t'$ ou $t \hookrightarrow_v t'$, alors $N(t) > N(t')$. D'où le résultat. ♠

Définitions Soit t un $\lambda_{\mathcal{UV}}$ -terme et E un ensemble de variables.

1) Une occurrence d'une variable x dans t est dite *E-inactive* ssi ou bien $x \in E$ ou bien il existe un sous-terme u de t tel que $x \in Fv(u)$ et il existe une occurrence

d'une variable E -inactive y telle que $((\dots(y)u_1\dots u_i)\lambda y_1\dots\lambda y_m\lambda x u)u_{i+1}\dots u_n$ est un sous-terme de t .

2) Un sous-terme u de t est dit E -inactif ssi $u = (x)u_1\dots u_n$ et x est E -inactive dans t .

3) Un sous-terme u de t est dit E -passif ssi il existe un $\lambda_{\mathcal{UV}}$ -terme E -inactif $(x)u_1\dots u_n$ de t tel que $u_i = \lambda y_1\dots\lambda y_m u$.

Lemme 3.2 Soient u, v des $\lambda_{\mathcal{UV}}$ -termes, E un ensemble de variables et $x \notin E$. Les occurrences des variables E -inactives (resp. les termes E -inactifs, les termes E -passifs) de u et v sont des occurrences des variables E -inactives (resp. des termes E -inactifs, des termes E -passifs) de $u[v/x]$.

Preuve Facile. ♠

Définition Soit E un ensemble de variables. Un $\lambda_{\mathcal{UV}}$ -terme t est dit E -bon ssi chaque occurrence de $\mathcal{U}_{B,X}$ dans t est appliquée à un seul argument w et $(\mathcal{U}_{B,X})w$ est E -passif et chaque occurrence de $\mathcal{V}_{B,X}$ dans t est appliquée à un $\lambda_{\mathcal{UV}}$ -terme E -inactive.

Il est clair qu'un $\lambda_{\mathcal{UV}}$ -terme E -bon a l'une des formes suivantes : $\lambda x u$, $(x)u_1\dots u_n$, $(\lambda x u)vv_1\dots v_n$ ou $(\mathcal{V}_{B,X})(x)u_1\dots u_n$. Le lemme suivant caractérise les $\lambda_{\mathcal{UV}}$ -termes E -bons.

Lemme 3.3 1) $\lambda x u$ est E -bon ssi u est $E - \{x\}$ -bon.

2) $(x)u_1\dots u_n$ est E -bon ssi pour tout $(1 \leq i \leq n)$, u_i est E -bon ou (si $x \in E$) $u_i = \lambda y_1\dots\lambda y_m(\mathcal{U}_{B,X})w$ et w est $E - \{y_1, \dots, y_m\}$ -bon.

3) $(\lambda x u)vv_1\dots v_n$ est E -bon ssi v est E -bon, pour tout $(1 \leq i \leq n)$, v_i est E -bon et u est $E - \{x\}$ -bon.

4) $(\mathcal{V}_{B,X})(x)u_1\dots u_n$ est E -bon ssi $x \in E$ et $(x)u_1\dots u_n$ est E -bon.

Preuve Facile. ♠

Lemme 3.4 Soient u, v des $\lambda_{\mathcal{UV}}$ -termes et E un ensemble de variables. Si u, v sont E -bons, alors $(u)v$ est E -bon.

Preuve Facile. ♠

Lemme 3.5 Soient u, v des $\lambda_{\mathcal{UV}}$ -termes, E un ensemble de variables et $x \notin E$. Si u, v sont E -bons, alors $u[v/x]$ est E -bon.

Preuve On utilise le lemme 3.2. ♠

Théorème 3.1 Si t est un $\lambda_{\mathcal{UV}}$ -terme E -bon et $t \leftrightarrow t'$, alors t' est E -bon.

Preuve Il suffit de faire la preuve pour un seul pas de réduction. On procède par induction sur t et on utilise les lemmes 3.3, 3.4 et 3.5.

– Si $t = \lambda x u$, alors il suffit d'appliquer l'hypothèse d'induction et utiliser le lemme 3.5.

- Si $t = (x)u_1\dots u_n$, alors on a deux cas à voir. Si u_i est E -bon et se réduit à u'_i , alors il suffit d'appliquer l'hypothèse d'induction et utiliser le lemme 3.3. Si $x \in E$, $u_i = \lambda y_1\dots\lambda y_m(\mathcal{U}_{B,X})w$ et w est $E - \{y_1, \dots, y_m\}$ -bon, alors le résultat est évident si on fait la réduction dans w . Si on réduit le \mathcal{U} -redex, alors ceci dépend de B .
- Si $B = Y \neq X$, alors u_i se réduit à $\lambda y_1\dots\lambda y_m w$. Or comme w est $E - \{y_1, \dots, y_m\}$ -bon, alors, en utilisant le lemme 3.3, $\lambda y_1\dots\lambda y_m w$ est E -bon et donc t' aussi.
- Si $B = C \rightarrow D$, alors u_i se réduit à $\lambda y_1\dots\lambda y_m \lambda y(\mathcal{U}_{D,X})(w)(\mathcal{V}_{C,X})y$. Or comme w est $E - \{y_1, \dots, y_m\}$ -bon, alors (par définition) t' est E -bon.
- Si $B = \forall Y C$, alors le résultat est évident.
- Si $t = (\lambda x u)vv_1\dots v_n$, alors le résultat est évident si on fait la réduction dans les termes u, v, v_1, \dots, v_n . Si $t' = (u[v/x])v_1\dots v_n$, alors le résultat provient des lemmes 3.4 et 3.5.
- Si $t = (\mathcal{V}_{B,X})(x)u_1\dots u_n$, alors le résultat est évident si on fait la réduction dans les termes u_1, \dots, u_n . Si on réduit le \mathcal{V} -redex, alors ceci dépend de B .
- Si $B = Y \neq X$, alors $t' = (x)u_1\dots u_n$ qui est E -bon.
- Si $B = C \rightarrow D$, alors t se réduit à $\lambda y(\mathcal{V}_{D,X})(x)u_1\dots u_n(\mathcal{U})y$. Donc (par définition) t' est E -bon.
- Si $B = \forall Y C$, alors le résultat est évident. ♠

Définition Un $\lambda_{\mathcal{U}\mathcal{V}}$ -terme t est dit *bon* ssi il est $Fv(t)$ -bon.

Corollaire 3.1 Si t est un $\lambda_{\mathcal{U}\mathcal{V}}$ -terme bon et $t \hookrightarrow t'$, alors t' est bon.

Preuve D'après le théorème 3.1. ♠

Définition Soit \mathcal{U} et \mathcal{V} deux constantes fixées du λ -calcul. Pour tout type A et pour toute variable X , on définit par induction deux λ -termes $\mathcal{I}'_{A,X}$ et $\mathcal{J}'_{A,X}$ de la manière suivante :

$$\begin{array}{ll}
\mathcal{I}'_{Y,X} = \mathbf{id} \text{ si } Y \neq X & \mathcal{J}'_{Y,X} = \mathbf{id} \text{ si } Y \neq X \\
\mathcal{I}'_{X,X} = \mathcal{U} & \mathcal{J}'_{X,X} = \mathcal{V} \\
\mathcal{I}'_{B \rightarrow C, X} = \lambda x \lambda y (\mathcal{I}'_{C,X})(x)(\mathcal{J}'_{B,X})y & \mathcal{J}'_{B \rightarrow C, X} = \lambda x \lambda y (\mathcal{J}'_{C,X})(x)(\mathcal{I}'_{B,X})y \\
\mathcal{I}'_{\forall Y B, X} = \lambda x (\mathcal{I}'_{B,X})x & \mathcal{J}'_{\forall Y B, X} = \lambda x (\mathcal{J}'_{B,X})x
\end{array}$$

Lemme 3.6 Pour tout type A et pour toute variable X , les λ -termes $\mathcal{I}'_{A,X}$ et $\mathcal{J}'_{A,X}$ sont des λI -termes.

Preuve Par induction sur A . ♠

Définition On associe à chaque $\lambda_{\mathcal{U}\mathcal{V}}$ -terme t un λ -terme noté \widehat{t} de la manière suivante : $\widehat{x} = x$, $\widehat{\lambda x u} = \lambda x \widehat{u}$, $\widehat{(u)v} = (\widehat{u})\widehat{v}$, $\widehat{\mathcal{U}_{A,X}} = \mathcal{I}'_{A,X}$ et $\widehat{\mathcal{V}_{A,X}} = \mathcal{J}'_{A,X}$.

Lemme 3.7 Si u, v sont des $\lambda_{\mathcal{U}\mathcal{V}}$ -termes, alors $u[\widehat{v/x}] = \widehat{u}[\widehat{v/x}]$.

Preuve Par induction sur u . ♠

Lemme 3.8 Soient u, v des $\lambda_{\mathcal{U}\mathcal{V}}$ -termes. Si $u \hookrightarrow v$, alors $\widehat{u} \rightarrow_{\beta} \widehat{v}$.

Preuve Par induction sur u et on utilise le lemme 3.7. \spadesuit

Lemme 3.9 Si t est un $\lambda_{\mathcal{U}\mathcal{V}}$ -terme tel que \widehat{t} est fortement normalisable, alors t est fortement normalisable.

Preuve Sinon, alors il existe une suite $(t_i)_i$ telle que $t = t_0$ et $(t_i \hookrightarrow_{\beta} t_{i+1}$ ou $t_i \hookrightarrow_u t_{i+1}$ ou $t_i \hookrightarrow_u t_{i+1})$. D'après les lemmes 3.1 et 3.9, il existe une suite croissante d'entiers $(n_i)_i$ telle que $n_0 = 0$ et $\widehat{t_{n_i}} \rightarrow_{\beta} \widehat{t_{n_{i+1}}}$. Ce qui contredit le fait que \widehat{t} est fortement normalisable. \spadesuit

Avec les hypothèses du lemme précédent (et en utilisant le lemme 3.8) on a unicité de la forme normale de t .

Donc la suite on fixe une constante du λ -calcul α et une constante de type O .

Définitions 1) Soient $U = \lambda x \lambda d \langle x, \alpha \rangle$ et $V = \lambda x (x) \alpha \mathbf{1}$. Pour tout type A et pour toute variable X , on note $\mathcal{I}_{A,X} = \mathcal{I}'_{A,X}[U/\mathcal{U}, V/\mathcal{V}]$ et $\mathcal{J}_{A,X} = \mathcal{J}'_{A,X}[U/\mathcal{U}, V/\mathcal{V}]$.

2) Pour tout type G du système \mathcal{F} , on note $G^\circ = O \rightarrow (G \wedge O)$.

Lemme 3.10 Pour tout type A et pour toute variable X , on a :

1) $\alpha : O \vdash_{\mathcal{F}} \mathcal{I}_{A,X} : \forall Y \{A[Y/X] \rightarrow A[Y^\circ/X]\}$.

2) $\alpha : O \vdash_{\mathcal{F}} \mathcal{J}_{A,X} : \forall Y \{A[Y^\circ/X] \rightarrow A[Y/X]\}$.

Preuve Par induction sur A .

– Si $A = Y \neq X$, alors $A[Y/X] = A[Y^\circ/X] = Y$, $\mathcal{I}_{A,X} = \mathcal{J}_{A,X} = \mathbf{id}$ et on a $\alpha : O \vdash_{\mathcal{F}} \mathbf{id} : \forall Y \{Y \rightarrow Y\}$.

– Si $A = X$, alors $A[Y/X] = Y$, $A[Y^\circ/X] = Y^\circ$, $\mathcal{I}_{A,X} = U$ et $\mathcal{J}_{A,X} = V$. On a les deux typages suivants :

$$\frac{\alpha : O, x : Y, d : Y \vdash_{\mathcal{F}} \langle x, \alpha \rangle : Y \wedge O}{\alpha : O \vdash_{\mathcal{F}} U : \forall Y \{Y \rightarrow Y^\circ\}}$$

$$\frac{\alpha : O, x : Y^\circ, \vdash_{\mathcal{F}} (x) \alpha \mathbf{1} : Y}{\alpha : O \vdash_{\mathcal{F}} V : \forall Y \{Y^\circ \rightarrow Y\}}$$

– Si $A = B \rightarrow C$, alors $A[Y/X] = B[Y/X] \rightarrow C[Y/X]$, et $A[Y^\circ/X] = B[Y^\circ/X] \rightarrow C[Y^\circ/X]$. Par hypothèse d'induction on a $\alpha : O \vdash_{\mathcal{F}} \mathcal{I}_{C,X} : \forall Y \{C[Y/X] \rightarrow C[Y^\circ/X]\}$ et $\alpha : O \vdash_{\mathcal{F}} \mathcal{J}_{B,X} : \forall Y \{B[Y^\circ/X] \rightarrow B[Y/X]\}$.

Donc

$$\frac{\frac{\alpha : O, x : B[Y/X] \rightarrow C[Y/X], y : B[Y^\circ/X] \vdash_{\mathcal{F}} (\mathcal{J}_{B,X})y : B[Y/X]}{\alpha : O, x : B[Y/X] \rightarrow C[Y/X], y : B[Y^\circ/X] \vdash_{\mathcal{F}} (x)(\mathcal{J}_{B,X})y : C[Y/X]}}{\alpha : O, x : B[Y/X] \rightarrow C[Y/X], y : B[Y^\circ/X] \vdash_{\mathcal{F}} (\mathcal{I}_{C,X})(\mathcal{J}_{B,X})y : C[Y/X]}}$$

$$\alpha : O, \vdash_{\mathcal{F}} \mathcal{I}_{A,X} : \forall Y \{A[Y/X] \rightarrow A[Y^\circ/X]\}$$

De même on démontre que $\vdash_{\mathcal{F}} \mathcal{J}_{A,X} : \forall Y \{A[Y^\circ/X] \rightarrow A[Y/X]\}$.
 – Si $A = \forall ZB$, alors $A[Y/X] = \forall ZB[Y/X]$, et $A[Y^\circ/X] = \forall ZB[Y^\circ/X]$. Par hypothèse d'induction on a $\alpha : O \vdash_{\mathcal{F}} \mathcal{I}_{B,X} : \forall Y \{B[Y/X] \rightarrow C[B^\circ/X]\}$. Donc

$$\frac{\frac{\alpha : O, x : \forall ZB[Y/X] \vdash_{\mathcal{F}} (\mathcal{I}_{B,X})x : B[Y^\circ/X]}{\alpha : O, x : \forall ZB[Y/X] \vdash_{\mathcal{F}} (\mathcal{I}_{B,X})x : \forall ZB[Y^\circ/X]}}{\vdash_{\mathcal{F}} \mathcal{I}_{A,X} : \forall Y \{A[Y/X] \rightarrow A[Y^\circ/X]\}}$$

De même on démontre que $\vdash_{\mathcal{F}} \mathcal{J}_{A,X} : \forall Y \{A[Y^\circ/X] \rightarrow A[Y/X]\}$. ♠

Définitions 1) Soit t un terme β -normal ; t s'écrit $\lambda x_1 \dots \lambda x_n (x) t_1 \dots t_m$. Les sous-termes *essentiels* de t sont, par définition, $(x) t_1 \dots t_m$ et les sous-termes essentiels des t_i ($1 \leq i \leq m$).

2) Soient Γ un contexte, A un type, et t un λ -terme β -normal. On dit que t est un λ -terme η -long de type A dans le contexte Γ ssi $\Gamma \vdash_{\mathcal{F}} t : A$ et dans ce typage tous les sous termes essentiels de t sont typés par des variables de type.

Lemme 3.11 *Si t un λ -terme β -normal tel que $\Gamma \vdash_{\mathcal{F}} t : A$, alors il existe un λ -terme η -long t' de type A dans le contexte Γ tel que $t' \rightarrow_{\eta} t$.*

Preuve Il suffit de remplacer chaque sous-terme essentiel u de t de type $B = \forall \overline{X}_0 (B_1 \rightarrow \dots \forall \overline{X}_{n-1} (B_n \rightarrow \forall \overline{X}_n X) \dots)$ par le λ -terme $\lambda x_1 \dots \lambda x_n (u) x_n \dots x_1$. ♠

Définitions 1) Un sous-terme u de t est dit *en position d'application* ssi il existe un terme v tel que $(u)v$ est un sous-terme de t .

2) Un sous-terme u de t est dit *en position d'argument* s'il n'est pas en position d'application.

Lemme 3.12 *Soient t un λ -terme β -normal contenant α . Si $\alpha : O, \Gamma \vdash_{\mathcal{F}} t : A$, alors α est en position d'argument dans t et donc, pour tout λ -terme β -normal u , $t[u/\alpha]$ est β -normal.*

Preuve Facile. ♠

Définition Un λ -terme t est dit *bon* ssi le $\lambda_{\mathcal{U}\mathcal{V}}$ -terme $t[\mathcal{U}_{X,X}/\mathcal{U}, \mathcal{V}_{X,X}/\mathcal{V}]$ est bon.

Lemme 3.13 *Si t est un λ -terme bon β -normal contenant l'une des constantes \mathcal{U} ou \mathcal{V} , alors $t[U/\mathcal{U}, V/\mathcal{V}]$ est normalisable et sa forme normale contient α .*

Preuve Si t contient \mathcal{U} , alors \mathcal{U} est appliqué à un seul terme a , et donc dans $t[U/\mathcal{U}, V/\mathcal{V}]$ le sous-terme $(\lambda x \lambda d \langle x, \alpha \rangle) a$ se réduit à $\lambda d \langle a, \alpha \rangle$ sans créer des nouveaux redex. Si t contient \mathcal{V} , alors \mathcal{V} est appliqué à un terme de la forme $(x) a_1 \dots a_n$, et donc dans $t[U/\mathcal{U}, V/\mathcal{V}]$ le sous-terme $(\lambda x (x) \alpha \mathbf{1})(x) a_1 \dots a_n$ se réduit à $(x) a_1 \dots a_n \alpha \mathbf{1}$ sans créer des nouveaux redex. Dans les deux cas la forme normale de $t[U/\mathcal{U}, V/\mathcal{V}]$ contient α . ♠

Lemme 3.14 Soient $u_1, \dots, u_n, v_1, \dots, v_m$ des λI -termes, A un type contenant X comme variable libre. Si $T = ((\mathcal{J}_{A,X})(x)u_1 \dots u_n)v_1 \dots v_m$ est typable, alors la forme normale de T contient α .

Preuve Considérons le λ -terme $T' = ((\mathcal{J}'_{A,X})(x)u_1 \dots u_n)v_1 \dots v_m$. D'après le lemme 3.6, T' est un λI -terme. Comme T' est fortement normalisable (sinon T ne le sera pas) et A contient X , alors, d'après le lemme 3.6 et le théorème 2.1, la forme normale T'_1 contient \mathcal{U} ou \mathcal{V} . Soit $T'' = ((\mathcal{V}_{A,X})(x)u_1 \dots u_n)v_1 \dots v_m$. D'après le lemme 3.9, T'' est fortement normalisable, et si on note T''_1 sa forme normale, alors $T'_1 = \widehat{T''_1}$. T'' est un $\lambda_{\mathcal{U}\mathcal{V}}$ -terme bon, alors, d'après le corollaire 3.1, T''_1 est bon ne contenant que des $\mathcal{U}_{X,X}$ et des $\mathcal{V}_{X,X}$, donc T'_1 est bon. Donc, d'après le lemme 3.13, $T'_1[U/\mathcal{U}, V/\mathcal{V}]$ contient α . Or comme $T = T'[U/\mathcal{U}, V/\mathcal{V}]$, alors la forme normale de T contient aussi α . \spadesuit

Définition On distingue trois règles d'élimination du quantifacteur \forall :

$$(\forall_e)_1 \frac{\Gamma \vdash_{\mathcal{F}} \lambda xt : \forall X A}{\Gamma \vdash_{\mathcal{F}} \lambda xt : A[G/X]}$$

$$(\forall_e)_2 \frac{\Gamma \vdash_{\mathcal{F}} (x)t_1 \dots t_n : \forall X A}{\Gamma \vdash_{\mathcal{F}} (x)t_1 \dots t_n : A[G/X]}$$

$$(\forall_e)_3 \frac{\Gamma \vdash_{\mathcal{F}} (\lambda xu)vt_1 \dots t_n : \forall X A}{\Gamma \vdash_{\mathcal{F}} (\lambda xu)vt_1 \dots t_n : A[G/X]}$$

Lemme 3.15 Soit t un λI -terme β -normal. Si dans un typage $\Gamma \vdash_{\mathcal{F}} t : C$ on utilise la règle $(\forall_e)_2$, alors il existe un λ -terme β -normal t' contenant α et $\Gamma, \alpha : O \vdash_{\mathcal{F}} t' : C$.

Preuve Dans le typage de t on a donc

$$\frac{\Delta \vdash_{\mathcal{F}} (x)u_1 \dots u_n : \forall X A}{\Delta \vdash_{\mathcal{F}} (x)u_1 \dots u_n : A[G/X]}$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$\frac{\Delta' \vdash_{\mathcal{F}} (x)u_1 \dots u_n v_1 \dots v_m : B$$

et dans t , $(x)u_1 \dots u_n v_1 \dots v_m$ n'est pas appliqué à un autre λ -terme.

Considérons le typage suivant

$$\frac{\frac{\Delta \vdash_{\mathcal{F}} (x)u_1 \dots u_n : \forall X A}{\Delta \vdash_{\mathcal{F}} (x)u_1 \dots u_n : A[G^\circ/X]} \quad \frac{}{\alpha : O \vdash_{\mathcal{F}} \mathcal{J}_{A,X} : A[G^\circ/X] \rightarrow A[G/X]}}{\Delta, \alpha : O \vdash (\mathcal{J}_{A,X})(x)u_1 \dots u_n : A[G/X]}$$

$$\cdot$$

$$\cdot$$

$$\cdot$$

$$\frac{}{\Delta', \alpha : O \vdash ((\mathcal{J}_{A,X})(x)u_1 \dots u_n)v_1 \dots v_m : B}$$

Si u est la forme normale de $((\mathcal{J}_{A,X})(x)u_1\dots u_n)v_1\dots v_m$, alors $\Delta', \alpha : O \vdash_{\mathcal{F}} u : B$ et, d'après le lemme 3.14, u contient α . Si $t = C\langle(x)u_1\dots u_nv_1\dots v_m\rangle$, alors posons $t' = C\langle u\rangle$. On a t' est un λ -terme β -normal contenant α et, d'après le lemme 2.3, $\Gamma, \alpha : O \vdash_{\mathcal{F}} t' : C$. \spadesuit

Lemme 3.16 *Soient D un I -type et t un λI -terme β -normal. Si $\vdash_{\mathcal{F}} t : D$, alors la règle $(\forall_e)_2$ ne peut pas être utilisée.*

Preuve Sinon, d'après le lemme 3.15, il existe un λI -terme β -normal t' contenant α tel que $\alpha : O \vdash_{\mathcal{F}} t' : D$. D'après le lemme 3.12, α est en position d'argument dans le λ -terme t' , et donc $t'' = t'[1/\alpha]$ est un λK -terme β -normal tel que $\vdash_{\mathcal{F}} t'' : D$. Contradiction. \spadesuit

Définition On définit deux ensembles de type du système $\mathcal{F} : \forall^+$ (ensemble de types \forall -positifs), et \forall^- (ensemble de types \forall -négatifs) de la manière suivante :
- Si $A = X$, alors $A \in \forall^+$, et $A \in \forall^-$;
- Si $T \in \forall^+$, et $T' \in \forall^-$, alors, $T' \rightarrow T \in \forall^+$, et $T \rightarrow T' \in \forall^-$;
- Si $T \in \forall^+$, et X est libre dans T , alors $\forall XT \in \forall^+$.
Donc, T est un type \forall^+ (resp. \forall^-) ssi le quantificateur du second ordre est positif (resp. négatif) dans T .

Lemme 3.17 *Soit t un λI -terme β -normal tel que $Fv(t) = \{x_1, \dots, x_n\}$. Si t est terme η -long de type A dans le context $\Gamma = x_1 : A_1, \dots, x_n : A_n$ sans utiliser la règle $(\forall_e)_2$, alors les $A_i \in \forall^-$ ($1 \leq i \leq n$) et $A \in \forall^+$.*

Preuve Par induction sur t .

- Si t est une variable, alors $n = 1$, $A = X$ et $x_1 : A_1 \vdash_{\mathcal{F}} x_1 : X$. Donc $A_1 = X$ et par conséquent A est \forall^+ et A_1 est \forall^- .
- Si $t = (x_i)t_1\dots t_m$ ($m \geq 1$), alors $A = X$, $A_i = B_1, B_2, \dots, B_m \rightarrow X$, et $\Gamma \vdash_{\mathcal{F}} t_j : B_j$. Comme t est un λI -terme alors, en appliquant l'hypothèse d'induction sur les t_j , on peut déduire que les A_k ($k \neq i$) sont \forall^- et les B_j sont \forall^+ . Donc A_i est aussi \forall^- .
- Si $t = \lambda xu$, alors $A = \forall \overline{X}(B \rightarrow C)$ et $\Gamma, x : B \vdash_{\mathcal{F}} u : C$. Par hypothèse d'induction, on déduit que les A_i et B sont \forall^- et C est \forall^+ . Donc A est \forall^+ . \spadesuit

Définitions 1) Si D est un type du système \mathcal{F} alors on note $\Lambda(D) = \{t \text{ } \beta\eta\text{-normal clos} / \vdash_{\mathcal{F}} t : D\}$.

2) Un type D du système \mathcal{F} est dit *démontrable* ssi $\text{card}(\Lambda(D)) \geq 1$.

Théorème 3.2 *Si D est un I -type démontrable du système \mathcal{F} , alors $D \in \forall^+$.*

Preuve Soit t un λI -terme η -longue de type D . D'après le lemme 3.16, dans le typage de t , on n'a pas utilisé la règle $(\forall_e)_2$. Donc, d'après le lemme 3.17, $D \in \forall^+$. \spadesuit

4 Résultats supplémentaires

4.1 Les types entrées et les types sorties

Commencant tout d'abord par définir les systèmes de typage \mathcal{F}_0 et \mathcal{F}_1 .

Définitions 1) Le système de typage \mathcal{F}_0 est obtenue en éliminant la règle de typage (\forall_e) du système \mathcal{F} .

2) Le système de typage \mathcal{F}_1 est obtenue en remplaçant la règle de typage (\forall_e) du système \mathcal{F} par

$$(\forall_e)^1 \frac{\Gamma \vdash_{\mathcal{F}_1} t : \forall X A \quad Y \text{ est une variable}}{\Gamma \vdash_{\mathcal{F}_1} t : A[Y/X]}$$

Les auteurs ont donné des définitions (par typage) des types entrées et des types sorties du système \mathcal{F} (voir [2]).

Définitions 1) Un type clos E du système \mathcal{F} est dit *type entrée* ssi si $\vdash_{\mathcal{F}} t : E$ alors $\vdash_{\mathcal{F}_0} t : E$. Intuitivement un type entrée est un type dont toutes les démonstrations se font dans le système \mathcal{F}_0 , et donc le problème de typage pour ce type est décidable.

2) Un type clos S du système \mathcal{F} (ne contenant pas une constante de type fixés O) est dit *type sortie* ssi si t est un λ -terme β -normal tel que $\alpha : O \vdash_{\mathcal{F}} t : S$ alors $\alpha \notin Fv(t)$. Intuitivement cela veut dire que les fonctions à valeurs dans un type sortie indépendamment du type de leurs arguments sont les fonctions constantes.

On a les résultats suivants (voir [2] et [5]) :

Théorème 4.1 1) *Un type clos et \forall^+ est un type entrée et sortie.*

2) *Un type entrée est un type sortie.*

On ne sait pas si la réciproque du 2) du théorème précédent est vraie. Nous allons voir que le lemme 3.15 apporte une réponse partielle à cette question :

En effet si un type clos D n'est pas un type entrée alors il existe un λ -terme β -normal clos t tel que $\vdash_{\mathcal{F}} t : D$ et $\not\vdash_{\mathcal{F}_0} t : D$ c.à.d la règle (\forall_e) est utilisée dans le typage. Le lemme 3.15 montre que si c'est la règle $(\forall_e)_2$ qui est utilisée et si t est un λI -terme, alors D n'est pas un type sortie.

4.2 Les types simples

On va montrer qu'on peut limiter l'étude des I -types à une classe de types très simples.

Définitions 1) Les *types* du système \mathcal{S} (appelé système simple) sont construits à partir des variables de type en utilisant uniquement le connecteur \rightarrow .

2) Les règles de typage du système \mathcal{S} sont (ax) , (\rightarrow_i) et (\rightarrow_e) .

3) On écrit $\Gamma \vdash_{\mathcal{S}} t : A$ si t est typable sans le système \mathcal{S} de type A dans le contexte Γ .

Le théorème 2.3 reste valable dans le système \mathcal{S} .

Définition Si A est un type du système \mathcal{F} , alors on note \mathcal{S}_A l'ensemble des types du système \mathcal{S} obtenus en effaçant dans A tous les quantificateurs (on suppose que toutes les variables du type utilisées dans A ont des noms différents).

On a démontré dans [5] le résultat suivant :

Théorème 4.2 Soient A un type \forall^+ du système \mathcal{F} , $A^* \in \mathcal{S}_A$, et t un λ -terme β -normal. Si $\vdash_{\mathcal{F}} t : A$, alors $\vdash_{\mathcal{S}} t : A^*$

Donc pour vérifier si un type clos D de l'ensemble \forall^+ est un I -type, il suffit d'étudier cette question pour une clôture d'un type simple de l'ensemble \mathcal{S}_D . Ceci est plus facile à étudier vu que le problème de typage pour les types simples est décidable.

La réciproque du théorème 4.2 est évidemment fautive si on permet l'utilisation des types non propres. En effet, soient $A = \forall X \{(X \rightarrow X) \rightarrow (X \rightarrow \forall Z X)\}$, $A^* = (X \rightarrow X) \rightarrow (X \rightarrow X)$. On a $\vdash_{\mathcal{S}} \mathbf{id} : A^*$ mais $\not\vdash_{\mathcal{F}} \mathbf{id} : A$. Remarquons que l'exemple précédent ne marche pas si on considère des λ -termes η -longs de type A^* . On va présenter un contre exemple à la réciproque du théorème 4.2 en se limitant aux λ -termes η -longs.

Définition Soient :

$$\begin{aligned} T &= \forall X \{[\forall Y ((Y \rightarrow \forall Z ((X, Y \rightarrow Z) \rightarrow Z)) \rightarrow X) \rightarrow X] \rightarrow X\} \\ T^* &= [(((Y \rightarrow ((X, Y \rightarrow Z) \rightarrow Z)) \rightarrow X) \rightarrow X) \rightarrow X], X \rightarrow X \\ t &= \lambda x \lambda y (x) \lambda z (x) \lambda u (z) \lambda v \lambda w ((w)(u) \lambda d \langle y, v \rangle) v \end{aligned}$$

Théorème 4.3 Le λ -terme t est η -long de type T^* mais $\not\vdash_{\mathcal{F}} t : T$.

Preuve Supposons que $\vdash_{\mathcal{F}} t : T$, alors il existe un typage de t où les variables liées de t sont déclarées de la manière suivantes :

$$\begin{aligned} x &: \forall Y ((Y \rightarrow \forall Z ((X, Y \rightarrow Z) \rightarrow Z)) \rightarrow X) \rightarrow X \\ y &: X \\ z &: (Y \rightarrow \forall Z ((X, Y \rightarrow Z) \rightarrow Z)) \rightarrow X \\ u &: (Y' \rightarrow \forall Z ((X, Y' \rightarrow Z) \rightarrow Z)) \rightarrow X \quad (Y' \neq Y) \\ v &: Y \\ w &: X, Y \rightarrow Z \\ d &: Y' \end{aligned}$$

et le sous-terme $\langle y, v \rangle$ est de type $\forall Z ((X, Y' \rightarrow Z) \rightarrow Z)$. D'où v est typable de type Y' . Contradiction. \spadesuit

4.3 Quelques exemples

Définition Soit $n \in \mathbb{N}$. Un I -type D est dit d'ordre n (resp. infini) ssi $\text{Card}(\Lambda(D)) = n$ (resp. $\Lambda(D)$ est infini).

On va donner des exemples de I -types d'ordre quelconque.

Théorème 4.4 *Id est I -type d'ordre 1.*

Preuve Facile. ♠

Notations 1) Soient A_1, \dots, A_n des types du système \mathcal{F} . On note $A_1 \wedge \dots \wedge A_n$ le type $\forall X \{ (A_1, \dots, A_n \rightarrow X) \rightarrow X \}$ où X ne figure pas dans les types A_1, \dots, A_n .
2) Soient t_1, \dots, t_n des λ -termes. On note $\langle t_1, \dots, t_n \rangle$ le λ -terme $\lambda x(x)t_1 \dots t_n$.

Définition Soit $n \geq 2$. On note :

$$B_n = \forall X \forall Y_1 \dots \forall Y_n \{ (Y_1 \rightarrow X), \dots, (Y_n \rightarrow X) \rightarrow [Y_1, \dots, Y_n \rightarrow (X \wedge Y_1 \wedge \dots \wedge Y_n)] \wedge (Y_1 \rightarrow X) \wedge \dots \wedge (Y_n \rightarrow X) \}$$

et, pour tout $1 \leq i \leq n$, $T_i = \lambda x_1 \dots \lambda x_n \langle \lambda y_1 \dots \lambda y_n \langle (x_i) y_i, y_1, \dots, y_n \rangle, x_1, \dots, x_n \rangle$.

Théorème 4.5 *B_n est un I -type d'ordre n et $\Lambda(B_n) = \{T_1, \dots, T_n\}$.*

Preuve Facile. ♠

Notation Soient x, y, z des variables et n un entier. On note $\lambda u [(x)(y)]^n (z)u$ la forme $\beta\eta$ -normale du λ -terme $\lambda u (x)(y) \dots (x)(y)(z)u$ ($(x)(y)$ est répété n fois).

Définition Soit $B_\infty = \forall X \forall Y \{ (X \rightarrow Y), (Y \rightarrow X) \rightarrow [(X \rightarrow Y) \wedge (Y \rightarrow X)] \}$ et, pour tout $(i, j) \in \mathbb{N}^2$, $T_{i,j} = \lambda x \lambda y \langle \lambda u [(x)(y)]^i (x)u, \lambda u [(y)(x)]^j (y)u \rangle$.

Théorème 4.6 *B_∞ est un I -type infini et $\Lambda(B_\infty) = \{T_{i,j} ; (i, j) \in \mathbb{N}^2\}$.*

Preuve Facile. ♠

D'après le théorème 4.5 le type B_2 est un I -type qui représente les booléens. Il contient cinq quantificateurs. Une question se pose : "Peut-on trouver des I -types plus simples (avec moins de variables ou de quantificateurs) pour les booléens ?" Nous allons montrer que les I -types contenant au plus deux quantificateurs sont au plus d'ordre 1.

Théorème 4.7 *Si D est un I -type démontrable du système \mathcal{F} contenant un seul quantificateur, alors $D = \forall X \{ X \rightarrow X \}$.*

Preuve Soient $D = \forall X \{ A_1, \dots, A_n \rightarrow X \}$ un tel I -type et t un terme η -long minimal tel que $\vdash_{\mathcal{F}} t : D$. Alors $t = \lambda x_1 \dots \lambda x_n u$, $u = (x_i)u_1 \dots u_m$ ($m \geq 0$), et $x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} u : X$. Si $m \neq 0$, alors $x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} u_1 : B_1, \dots, B_k \rightarrow X$. D'après le choix de t , on a $k \geq 1$ et $\lambda x_1 \dots \lambda x_n ((x_i) \lambda y_1 \dots \lambda y_k u) u_2 \dots u_m$ qui n'est pas un λI -terme est de type D . D'où $m = 0$, $u = x_i$ et donc $n = 1$, et

$D = \forall X\{X \rightarrow X\}$. ♠

Notations 1) Si B, A sont des types, alors, pour tout $n \geq 1$, on note $B^n \rightarrow A$ le type $B, \dots, B \rightarrow A$ où B est répété n fois.

2) Si X et Y sont deux variables de types, alors on note les formules $\forall X\forall Y A$ et $\forall Y\forall X A$ par $\forall X, Y A$.

Théorème 4.8 *Si D est un I -type démontrable du système \mathcal{F} contenant deux quantificateurs, alors $D = \forall X, Y\{[(Y \rightarrow Y)^n \rightarrow X] \rightarrow X\}$, $D = \forall X\{(\forall Y(Y \rightarrow Y) \rightarrow X) \rightarrow X\}$, $D = \forall X, Y\{Y, (Y^n \rightarrow X) \rightarrow X\}$ ou $D = \forall X, Y\{(Y^n \rightarrow X), Y \rightarrow X\}$.*

Preuve Soient $D = \forall X\forall Y\{A_1, \dots, A_n \rightarrow X\}$ un I -type démontrable du système \mathcal{F} et t un terme η -long minimal tel que $\vdash_{\mathcal{F}} t : D$. Alors $t = \lambda x_1 \dots \lambda x_n u$, $u = (x_i)u_1 \dots u_k$ ($k \geq 1$), et $x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} u : X$. Les u_i ($1 \leq i \leq k$) ne peuvent pas être de type $B_1, \dots, B_m \rightarrow X$. En effet si c'est le cas, on contredit soit la minimalité de t soit le fait que D est un I -type. Cherchons donc les termes η -longs minimaux v tels que $x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} \lambda y_1 \dots \lambda y_m v : B_1, \dots, B_m \rightarrow Y$ c.à.d. $\Gamma = x_1 : A_1, \dots, x_n : A_n, y_1 : B_1, \dots, y_m : B_m \vdash_{\mathcal{F}} v : Y$. Supposons que $v = (z)v_1 \dots v_l$ ($l \geq 0$). Remarquons d'abord que les v_i ($1 \leq i \leq l$) ne peuvent pas être ni de type $C_1, \dots, C_r \rightarrow Y$ ni de type $D_1, \dots, D_{r'} \rightarrow X$ avec ($r' \geq 1$). Donc ils sont forcément de type X . Soit w un terme η -long minimal tel que $\Gamma \vdash_{\mathcal{F}} w : X$. w est évidemment une variable. On a deux cas à voir :

- Si $w = x_j$, alors $A_j = X$ et $D = \forall X\forall Y\{X \rightarrow X\}$. Contradiction.
- Si $w = y_j$ alors $x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} \lambda y_1 \dots \lambda y_m v[u/y_j] : B_1, \dots, B_m \rightarrow Y$ et D n'est pas un I -type.

Donc v est une variable. De nouveau on a deux cas à voir :

- Si $v = x_j$, alors $A_j = Y$. Dans ce cas tous les u_i ($1 \leq i \leq k$) sont égaux à x_j . Donc $t = \lambda x \lambda y(x)y \dots y$ ou $t = \lambda x \lambda y(y)x \dots x$. D'où $D = \forall X\forall Y\{Y, (Y^n \rightarrow X) \rightarrow X\}$ ou $D = \forall X\forall Y\{(Y^n \rightarrow X), Y \rightarrow X\}$.
- Si $v = y_j$ alors $B_j = Y$. Dans ce cas tous les u_i ($1 \leq i \leq k$) sont égaux à $\lambda z z$. D'où $t = \lambda x(x)\lambda z z \dots \lambda z z$ et $D = \forall X\forall Y\{((Y \rightarrow Y)^n \rightarrow X) \rightarrow X\}$.

Si $D = \forall X\{A_1, \dots, A_i \rightarrow \forall Y(A_{i+1}, \dots, A_n \rightarrow X)\}$, alors, en reprenant la preuve que nous venons de faire, on déduit que $i = 0$ et donc D est l'un des types trouvés précédemment.

Si $D = \forall X\{A_1, \dots, A_k, \dots, A_n \rightarrow X\}$ et A_k contient le deuxième quantificateur, alors soit t est terme η -longue minimal tel que $\vdash_{\mathcal{F}} t : D$. Alors $t = \lambda x_1 \dots \lambda x_n (x_i)u_1 \dots u_m$ et $x_1 : A_1, \dots, x_n : A_n \vdash_{\mathcal{F}} (x_i)u_1 \dots u_m : X$. Les u_i ($1 \leq i \leq m$) ne peuvent pas être de type $B_1, \dots, B_r \rightarrow X$. Donc $i = k$, $m = 1$ et $A_k = \forall Y(C_1, \dots, C_l \rightarrow Y) \rightarrow X$. En reprenant la preuve que nous avons faite précédemment, on déduit que $l = 1$ et $C_1 = Y$. D'où $D = \forall X\{(\forall Y(Y \rightarrow Y) \rightarrow X) \rightarrow X\}$. ♠

Corollaire 4.1 *Si D est un I -type du système \mathcal{F} contenant au plus deux quantificateurs, alors $\text{Card}(\Lambda(D)) \leq 1$.*

Preuve D'après les théorèmes 4.7 et 4.8. ♠

References

- [1] H. Barendregt. *The lambda calculus, its syntax and semantics*. North Holland (1984).
- [2] S. Farkh, *Types de données en logique du second ordre*. Thèse de doctorat, Université de Savoie, France (1998).
- [3] J.-Y. Girard, Y. Lafont, P. Taylor, *Proofs and types*. Cambridge University Press (1986).
- [4] J.-L. Krivine, *Lambda calcul, types et modèles*. Masson, (1990).
- [5] K. Nour, Opérateurs de mise en mémoire et types \forall -positifs. *RAIRO - Inform. Théor. Appl.* **30** (1996) 261-293.