Une réponse négative à la conjecture de E. Tronci pour les systèmes numériques typés

Karim Nour

▶ **To cite this version:**

Karim Nour. Une réponse négative à la conjecture de E. Tronci pour les systèmes numériques typés. Informatique Théorique et Applications, 1998, 31 (6), pp.539-558. hal-00381049

HAL Id: hal-00381049
https://hal.science/hal-00381049v1

Submitted on 5 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.

Une réponse négative à la conjecture de E.Tronci pour les systémes numériques typés

Karim NOUR
LAMA - Equipe de Logique
Université de Savoie
73376 Le Bourget du Lac
e-mail nour@univ-savoie.fr

Résumé *Un système numérique est une suite de λ-termes normaux clos distincts pour laquelle il existe des λ-termes clos pour les fonctions successeur et test à zéro. Un système numérique est dit adéquat ssi il existe un λ-terme clos pour la fonction prédécesseur. Un opérateur de mise en mémoire pour un système numérique est un λ-terme clos qui simule "l'appel-par-valeur" dans le cadre de "l'appel-par-nom". E. Tronci a conjecturé le résultat suivant : un système numérique est adéquat s'il possède un opérateur de mise en mémoire. Nous donnons, dans cet article, une réponse négative à la conjecture de E. Tronci mais uniquement pour les systèmes numériques typable dans le système $\mathcal{F}$. La conjecture de E. Tronci reste sans solution en λ-calcul pur.*

Mots clés : Système numérique ; λ-calcul ; Successeur ; Test à zéro ; Prédécesseur ; Opérateur de mise en mémoire ; Système numérique adéquat ; Appel-par-valeur ; Appel-par-nom ; Système $\mathcal{F}$.

Abstract *A numeral system is a sequence of an infinite different closed normal λ-terms which has closed λ-terms for successor and zero test. A numeral system is said adequate iff it has a closed λ-term for predecessor. A storage operator for a numeral system is a closed λ-term which simulate "call-by-value" in the context of a "call-by-name" strategy. E. Tronci conjectured the following result : a numeral system is adequate if it has a storage operator. This paper gives a negative answer to this conjecture for the numeral systems typable in the J.-Y. Girard type system $\mathcal{F}$. The E. Tronci's conjecture remains open in pure λ-calculus.*

Keywords : Numeral system ; λ-calculus ; Successor ; Zero test ; Predecessor ; Storage operator ; Adequate numeral system ; Call-by-value ; Call-by-name ; Type system $\mathcal{F}$.

1 Introduction

Un système numérique est une suite de λ-termes normaux clos distincts $\mathbf{d} = d_0, d_1, ..., d_n, ...$ pour laquelle il existe des λ-termes clos S_d et Z_d pour les fonctions successeur et test à zéro. Un système numérique est dit adéquat ssi il existe un λ-terme clos P_d pour la fonction prédécesseur. H. Barendregt a démontré dans [1] qu'un système numérique est adéquat ssi toutes les fonctions récursives totales sont représentables dans le système.

La différence entre notre définition d'un système numérique et celle proposée par H. Barendregt (voir [1]) est le fait d'imposer aux λ-termes d_i d'être normaux et distinctes. En effet ces conditions permettent, pour des stratégies de réduction gagnantes, de trouver la valeur exacte d'une fonction numérique totale calculée sur des entiers.

Une des stratégies de réduction gagnantes est la réduction gauche (itération de la réduction de tête notée $\succ$). Mais pour cette stratégie l'argument d'une fonction est calulé le nombre de fois où la fonction l'utilise. Les opérateurs de mise en mémoire ont été introduits par J.-L. Krivine pour remédier à ce défaut.

Un λ-terme clos O_d est dit opérateur de mise en mémoire pour un système numérique $\mathbf{d}$ ssi pour tout $n \in \mathbf{N}$, il existe un λ-terme clos $\tau_n \simeq_\beta d_n$ tel que pour tout $\theta_n \simeq_\beta d_n$, $(O_d\ \theta_n\ f) \succ (f\ \tau_n)$ (où f est une nouvelle variable).

Nous allons justifier cette définition. Soit F un λ-terme (pour une fonction), et θ_n un λ-terme β-équivalent à d_n. Durant la réduction gauche de $(F\ \theta_n)$, θ_n sera réduit chaque fois qu'il arrive en tête. Au lieu de réduire $(F\ \theta_n)$, effectuons la réduction de tête de $(O_d\ \theta_n\ F)$. La réduction de $(O_d\ \theta_n\ F) = \{(O_d\ \theta_n\ f)\}[F/f]$ commence par amener $(O_d\ \theta_n\ f)$ à sa forme normale de tête qui est $(f\ \tau_n)$, et puis réduire $(F\ \tau_n)$. Dans la réduction de $(O_d\ \theta_n\ F)$, θ_n est calculé le premier, et le résultat est donné à F comme argument. O_d a donc mis en mémoire le résultat τ_n, avant de le donner à la fonction F. Donc la réduction de tête $(O_d\ \theta_n\ F) \succ (F\ \tau_n)$ dépend seulement de θ_n et pas de F.

J.-L. Krivine a démontré dans [4] que, dans le système de typage $\mathcal{F}$ de J.-Y. Girard, le type $N^* \to \neg\neg N$ convient pour les opérateurs de mise en mémoire pour le système numérique de Church : où N est le type des entiers de Church, et l'opération $*$ est la simple traduction de Gődel qui associe à chaque formule F la formule F^* obtenue en remplacant dans F chaque variable de type par sa négation.

Nous démontrons dans ce papier que chaque système numérique adéquat possède un opérateur de mise en mémoire. E. Tronci a conjecturé qu'un système numérique est adéquat s'il possède un opérateur de mise en mémoire.

Nous donnons, ensuite, une réponse négative à la conjecture de E. Tronci mais uniquement pour les systèmes numériques typable dans le système $\mathcal{F}$. Nous construisons donc un type clos E, une suite de λ-termes normaux clos distincts $\mathbf{e} = e_0, e_1, ..., e_n, ...$, et des λ-termes clos S_e, Z_e, et O_e tels que :

– Si t est un λ-terme normal clos, alors $\vdash_{\mathcal{F}} t : E$ ssi $t = e_i$ où $i \in \mathbf{N}$.

– $\vdash_{\mathcal{F}} S_e : E \to E$ et $(S_e\ e_n) \simeq_\beta e_{n+1}$ pour tout $n \in \mathbf{N}$.

– $\vdash_{\mathcal{F}} Z_e : E \to B$ (B est le type des Booléens du système $\mathcal{F}$), $(Z_e\ e_0) \simeq_\beta \lambda x\lambda y x$ et $(Z_e\ e_{n+1}) \simeq_\beta \lambda x\lambda y y$ pour tout $n \in \mathbf{N}$.

– $\vdash_{\mathcal{F}} O_e : E^* \rightarrow \neg\neg E$, et, pour tout $n \in \mathbf{N}$, il existe un λ-terme clos $\tau_n \simeq_\beta e_n$ tel que pour tout $\theta_n \simeq_\beta e_n$, $(O_d\ \theta_n\ f) \succ (f\ \tau_n)$.

– Il n'existe pas un λ-terme clos P_e tel que $\vdash_{\mathcal{F}} P_e : E \rightarrow E$ et $(P_e\ e_{n+1}) \simeq_\beta e_n$ pour tout $n \in \mathbf{N}$.

La conjecture de E. Tronci reste sans solution en λ-calcul pur.

2 Notations et définitions

2.1 Le λ-calcul pur

Notations :

1) La β-équivalence est notée $u \simeq_\beta v$.

2) Si u et v sont deux λ-termes, alors on note $< u, v >$ le λ-terme $\lambda x(x\ u\ v)$.

3) On note T (pour True) le λ-terme $\lambda x \lambda y x$ et F (pour False) le λ-terme $\lambda x \lambda y y$.

4) Pour tous λ-termes u, v, on définit $(u^n\ v)$ par induction : $(u^0\ v) = v$ et $(u^{n+1}\ v) = (u\ (u^n\ v))$. Pour chaque entier n, on définit *l'entier de Church* $\underline{n} = \lambda x \lambda f(f^n\ x)$.

5) La notation $\sigma(t)$ représent le résultat d'une substitution simultanée σ sur les variables libres de t après un rénommage de ses variables liées.

6) On note $\Theta = (U\ U)$ où $U = \lambda x \lambda f(f\ (x\ x\ f))$. Le λ-terme Θ est appelé *le point fixe de Turing.*

Définitions : Un λ-terme t soit il possède un *redex de tête* [i.e. $t = \lambda x_1 ... \lambda x_n(\lambda x u\ v\ v_1 ... v_m)$, le redex de tête est $(\lambda x u\ v)$], soit il est en *forme normale de tête* [i.e. $t = \lambda x_1 ... \lambda x_n(x\ v_1 ... v_m)$]. La notation $u \succ v$ signifie que v est obtenue à partir de u après quelques pas de réductions de tête. Un λ-terme est dit *résoluble* si sa réduction de tête termine.

Les résultats suivants sont bien connus (voir [3] et [4]).

Théorème 1
1) Si t est β-équivalent à une forme normale de tête, alors t est résoluble.
2) Si $u \succ v$, alors, pour toute substitution σ, $\sigma(u) \succ \sigma(v)$.
3) Si $u \succ v$, alors, pour toute suite $w_1, ..., w_n$, il existe un λ-terme w tel que $(u\ w_1 ... w_n) \succ w$ et $(v\ w_1 ... w_n) \succ w$.

Définition : On définit sur les λ-termes une relation d'équivalence $\sim$ par : $u \sim v$ ssi il existe un λ-terme t, tel que $u \succ t$, et $v \succ t$.

Donc, si t est résoluble, alors $u \sim t$ ssi u est résoluble, et possède la même forme normale de tête que t. Si u est une forme normale de tête, alors $t \sim u$ signifie que u est la forme

normale de tête de t.

D'après le théorème 1, on obtient les résultats suivants (voir [4]).

Théorème 2
1) Si $u \sim v$, alors, pour toute substitution σ, $\sigma(u) \sim \sigma(v)$.
2) Si $u \sim v$, alors, pour toute suite $w_1, ..., w_n$, $(u\ w_1 ... w_n) \sim (v\ w_1 ... w_n)$.

2.2 Le système $\mathcal{F}$

Définition : Les types du système $\mathcal{F}$ sont construits à partir des variables de type $X, Y, Z, ...$ et une constante $\perp$ (pour l'absurde) en utilisant les opérations suivantes :

– Si U et V sont des types, alors $U \to V$ est un type.

– Si V est un type, et X est une variable de type, alors $\forall X V$ est un type.

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

Définition : Soient t un λ-terme, A un type, et $\Gamma = x_1 : A_1, ..., x_n : A_n$ un contexte. On définit par les règles suivantes la notion "t est de type A dans Γ" ; cette notion est notée $\Gamma \vdash_{\mathcal{F}} t : A$.

$$(1)\ \Gamma \vdash_{\mathcal{F}} x_i : A_i\ (1 \leq i \leq n)$$

$$(2)\quad \frac{\Gamma, x : A \vdash_{\mathcal{F}} t : B}{\Gamma \vdash_{\mathcal{F}} \lambda x t : A \to B} \qquad (3)\quad \frac{\Gamma \vdash_{\mathcal{F}} u : A \to B \quad \Gamma \vdash_{\mathcal{F}} v : A}{\Gamma \vdash_{\mathcal{F}} (u)v : B}$$

$$(4)\quad \frac{\Gamma \vdash_{\mathcal{F}} t : A}{\Gamma \vdash_{\mathcal{F}} t : \forall X A}\ (*) \qquad (5)\quad \frac{\Gamma \vdash_{\mathcal{F}} t : \forall X A}{\Gamma \vdash_{\mathcal{F}} t : A[G/X]}\ (**)$$

Avec les conditions suivantes :
(*) X n'est pas libre dans Γ.
(**) G est un type.

Le système $\mathcal{F}$ possède les propriétés suivantes (voir [2] et [3]).

Théorème 3
1) Un type est préservé durant une β-réduction.
2) Un λ-terme typable est fortement normalisable.

Le type $F_1 \to (F_2 \to (... \to (F_n \to G)...))$ est noté $F_1, F_2, ..., F_n \to G$ et le type $F \to \perp$ est noté $\neg F$.

Les lemmes 1 et 2 seront très utiles pour nos démonstrations. Le lemme 1 (resp. le lemme 2) a été démontré dans [5] (resp. dans [2] et [3]).

Lemme 1
1) Soit X une variable de type. Si $\Gamma \vdash_{\mathcal{F}} t : X$, alors t ne commence pas par λ.
2) Si $\Gamma \vdash_{\mathcal{F}} \lambda x t : A \to B$, alors $\Gamma, x : A \vdash_{\mathcal{F}} t : B$.
3) Si $\Gamma, x : A \to B \vdash_{\mathcal{F}} (x\ u_1 ... u_n) : C$, alors $\Gamma, x : A \to B \vdash_{\mathcal{F}} u_1 : A$.
4) Si $\Gamma, x : A_1, ..., A_m \to X \vdash_{\mathcal{F}} (x\ u_1 ... u_n) : C$, alors X est libre dans C.

Lemme 2 *Si $x_1 : A_1, ..., x_n : A_n \vdash_{\mathcal{F}} t : A$, alors, pour toute variable X et tout type G, $x_1 : A_1[G/X], ..., x_n : A_n[G/X] \vdash_{\mathcal{F}} t : A[G/X]$.*

Dans le système $\mathcal{F}$ on a la possibilité de définir les types de données. Soit $B = \forall X\{X, X \to X\}$ (*le type des Booléens*) et $N = \forall X\{X, (X \to X) \to X\}$ (*le type des entiers*). On a les résultats suivants (voir [2] et [3]).

Théorème 4 *Soit t un λ-terme normal clos.*
1) $\vdash_{\mathcal{F}} t : B$ ssi $t = T$ ou $t = F$.
2) $\vdash_{\mathcal{F}} t : N$ ssi il existe $n \in \mathbf{N}$ tel que $t = \underline{n}$.

Définition : On note encore $\mathcal{F}$ le système logique sousjacent au système de typage $\mathcal{F}$, et on écrit $\Gamma \vdash_{\mathcal{F}} A$ si A est démontrable à partir des formules de Γ en utilisant les règles du système logique $\mathcal{F}$.

Il est claire que : $A_1, ..., A_n \vdash_{\mathcal{F}} A$ ssi il existe un λ-terme t tel que $x_1 : A_1, ..., x_n : A_n \vdash_{\mathcal{F}} t : A$. Ce résultat est connu sous le nom de "la corresponce du Curry-Howard".

2.3 Le système $\mathcal{F}_{\mathcal{C}}$ et la traduction de Gődel

Définition : On ajoute au système logique $\mathcal{F}$ la règle :

$$(0) \quad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}$$

Cette règle axiomatise la logique classique au dessus de la logique intuitionniste.
On note $\mathcal{F}_{\mathcal{C}}$ ce nouveau système et on écrit $\Gamma \vdash_{\mathcal{F}_{\mathcal{C}}} A$ si A est démontrable à partir des formules de Γ dans le système $\mathcal{F}_{\mathcal{C}}$. On a le résultat suivant (voir [2]).

Théorème 5 *Le système $\mathcal{F}_{\mathcal{C}}$ est non contradictoire (i.e. $\not\vdash_{\mathcal{F}_{\mathcal{C}}} \forall X X$).*

Définition : Pour chaque formule A de $\mathcal{F}_{\mathcal{C}}$, on définit la formule A^* par :

- Si $A = \bot$, alors $A^* = A$;
- Si $A = X$, alors $A^* = \neg X$;
- Si $A = B \to C$, alors $A^* = B^* \to C^*$;
- Si $A = \forall X B$, alors $A^* = \forall X B^*$.

A^* est appelée *la traduction de Gődel* de A.

On a le résultat suivant (voir [3]).

Théorème 6 *Si* $\vdash_{\mathcal{F}_c} A$, *alors* $\vdash_{\mathcal{F}} A^*$.

3 Les systèmes numériques

3.1 Les systèmes numériques en λ-calcul pur

Définition : Un *système numérique* est une suite de λ-termes normaux clos distincts $\mathbf{d} = d_0, d_1, ..., d_n, ...$ pour laquelle il existe des λ-termes clos S_d et Z_d tels que :

$$(S_d\ d_n) \simeq_\beta d_{n+1} \text{ pour tout } n \in \mathbf{N}$$
et
$$(Z_d\ d_0) \simeq_\beta T$$
$$(Z_d\ d_{n+1}) \simeq_\beta F \text{ pour tout } n \in \mathbf{N}$$

Les λ-termes S_d et Z_d sont appelés *successeur* et *test à zéro* pour $\mathbf{d}$.

Chaque système numérique peut être considérer comme un codage des entiers en λ-calcul et donc on peut représenter les fonctions numériques totales de la manière suivante.

Définition : Une fonction numérique totale $\phi : \mathbf{N}^p \to \mathbf{N}$ is dite *λ-définissable dans le système numérique* $\mathbf{d}$ ssi il existe un λ-terme F_ϕ tel que pour tout $n_1, ..., n_p \in \mathbf{N}$

$$(F_\phi\ d_{n_1}...d_{n_p}) \simeq_\beta d_{\phi(n_1,...,n_p)}$$

Définition : Un système numérique $\mathbf{d}$ is dit *adéquat* ssi il existe un λ-terme clos P_d tel que

$$(P_d\ d_{n+1}) \simeq_\beta d_n \text{ pour tout } n \in \mathbf{N}.$$

Le λ-terme P_d est appelé *prédécesseur* pour $\mathbf{d}$.

H. Barendregt a démontré que (voir [1]) :

Théorème 7 *Un système numérique* $\mathbf{d}$ *est adéquat ssi toutes les fonctions numériques récursives totales sont λ-définissables dans* $\mathbf{d}$.

Exemples :
1) Un exemple simple d'un système numérique adéquat est le *système numérique de Church* $\underline{\mathbf{n}} = \underline{0}, \underline{1}, ..., \underline{n}, ...$ Il est facile de vérifier que :

$\underline{S} = \lambda n \lambda x \lambda f (f\ (n\ f\ x))$,

$\underline{Z} = \lambda n (n\ T\ \lambda x F)$,

$\underline{P} = \lambda n(n\ U\ <\underline{0},\underline{0}>\ T)$ où $U = \lambda x < (\underline{S}\ (x\ T)), (x\ F) >$.

sont des λ-termes pour le successeur, le test à zéro, et le prédécesseur pour $\underline{\mathbf{n}}$.
2) Nous avons donné dans [6] un exemple d'un système numérique non adéquat. □

Définition : Soient $\mathbf{d}$ un système numérique et O_d un λ-terme clos. On dit que O_d est un *opérateur de mise en mémoire* pour $\mathbf{d}$ ssi pour tout $n \in \mathbf{N}$, il existe un λ-terme clos $\tau_n \simeq_\beta d_n$, tel que, pour tout $\theta_n \simeq_\beta d_n$, $(O_d\ \theta_n f) \succ (f\ \tau_n)$ où f est une nouvelle variable.

Exemple : Soit $O_N = \lambda n \lambda f(n\ f\ J\ \underline{0})$ où $J = \lambda x \lambda y(x\ (\underline{S}\ y))$. Il est facile de vérifier que pour tout $\theta_n \simeq_\beta \underline{n}$, $(O_N\ \theta_n\ f) \succ (f\ (\underline{S}^n\ \underline{0}))$. Donc O_N est un opérateur de mise en mémoire pour $\underline{\mathbf{n}}$. □

Remarque : J.-L. Krivine autorise, dans sa définition des opérateurs de mise en mémoire, le λ-terme τ_n de contenir des variables libres qui peuvent être remplacées par des λ-termes qui ne dépendent que de θ_n. Avec cette définition on garde aussi tous les résultats de ce papier. □

Théorème 8 *Chaque système numérique adéquat possède un opérateur de mise en mémoire.*

Preuve Soit $\mathbf{d}$ un système numérique adéquat.
Soit $O_d = (\Theta\ H_d)$ où $H_d = \lambda h \lambda n \lambda f((Z_d\ n)\ (f\ d_0)\ (h\ (P_d\ n)\ \lambda x(f\ (S_d\ x))))$.
Démontrons (par récurrence sur i) que, pour tout $i \in \mathbf{N}$ et pour tout $\theta_i \simeq_\beta d_i$, $(O_d\ \theta_i\ f) \sim (f\ (S_d{}^i\ d_0))$.

- Pour $i = 0$,

$$\begin{aligned}(O_d\ \theta_0\ f) &\sim (H_d\ (\Theta\ H_d)\ \theta_0\ f)\\ &\sim ((Z_d\ \theta_0)\ (f\ d_0)((\Theta\ H_d)\ (P_d\ \theta_0)\ \lambda x(f\ (S_d\ x))))\end{aligned}$$

  Comme $(Z_d\ d_0) \simeq_\beta T$, alors $(Z_d\ \theta_0) \succ T$, et, d'après le théorème 1, $(O_d\ \theta_0\ f) \sim (f\ d_0)$.

- Supposons le résultat vrai pour i, et prouvons le pour $i+1$.

$$\begin{aligned}(O_d\ \theta_{i+1}\ f) &\sim (H_d\ (\Theta\ H_d)\ \theta_{i+1}\ f)\\ &\sim ((Z_d\ \theta_{i+1})\ (f\ d_0)\ ((\Theta\ H_d)\ (P_d\ \theta_{i+1}))\ \lambda x(f\ (S_d\ x))))\end{aligned}$$

  Comme $(Z_d\ d_{i+1}) \simeq_\beta F$, alors $(Z_d\ \theta_{i+1}) \succ F$, et, d'après le théorème 1,
  $(O_d\ \theta_{i+1}\ f) \sim (O_d\ (P_d\ \theta_{i+1})\ \lambda x(f\ (S_d\ x)))$. Mais $(P_d\ \theta_{i+1}) \simeq_\beta d_i$, alors, par hypothèse d'induction, $(O_d\ (P_d\ \theta_{i+1})\ f) \sim (f\ (S_d{}^i d_0))$, et

$$\begin{aligned}(O_d\ (P_d\ \theta_{i+1})\ \lambda x(f\ (S_d\ x))) &\sim (\lambda x(f\ (S_d\ x))\ (S_d{}^i d_0))\\ &\sim (f\ (S_d{}^{i+1} d_0))\end{aligned}$$

  D'où, pour tout $i \in \mathbf{N}$ et pour tout $\theta_i \simeq_\beta d_i$, $(O_d\ \theta_i\ f) \succ (f\ (S_d{}^i d_0))$. □

E. Tronci a conjecturé le résultat suivant :

Conjecture *Un système numérique est adéquat s'il possède un opérateur de mise en mémoire.*

Nous donnons dans ce papier une réponse négative à cette conjecture mais uniquement pour les systèmes numériques typable dans le système $\mathcal{F}$.

3.2 Les systèmes numériques typés

Définition : Un *système numérique typé* est une paire $\mathcal{D} =< D, \mathbf{d} >$ où D est un type clos du système $\mathcal{F}$, et $\mathbf{d} = d_0, d_1, ..., d_n, ...$ est une suite de λ-termes normaux clos tels que :

– Si t est un λ-terme normal, alors $\vdash_{\mathcal{F}} t : D$ ssi il existe $i \in \mathbf{N}$ tel que $t = d_i$.

– Il existe des λ-termes clos S_d et Z_d tels que:

* $\vdash_{\mathcal{F}} S_d : D \to D$ et $(S_d\ d_n) \simeq_\beta d_{n+1}$ pour tout $n \in \mathbf{N}$;
* $\vdash_{\mathcal{F}} Z_d : D \to B$ et $(Z_d\ d_n) \simeq_\beta \begin{cases} T & \text{si } n = 0 \\ F & \text{si } n \geq 1 \end{cases}$.

Les λ-termes S_d et Z_d sont appelés *successeur* et *test à zéro* pour $\mathcal{D}$.

Définition : Un système numérique typé $\mathcal{D}$ est dite *adéquat* ssi il existe un λ-terme clos P_d tel que $\vdash_{\mathcal{F}} P_d : D \to D$ et $(P_d\ d_{n+1}) \simeq_\beta d_n$ pour tout $n \in \mathbf{N}$. Le λ-terme P_d est appelé *prédécesseur* pour $\mathcal{D}$.

Exemple : Il est facile de vérifier que $\mathcal{N} =< N, \underline{\mathbf{n}} >$ est un système numérique typé. □

Définitions :
1) Soient D, E deux types clos. On dit que $D \subseteq E$ ssi pour tout λ-terme clos t, si $\vdash_{\mathcal{F}} t : D$, alors $\vdash_{\mathcal{F}} t : E$.
2) Soit $\mathcal{D} =< D, \mathbf{d} >$ un système numérique typé tel que $D \subseteq D^*$. Soit O_d un λ-terme clos. On dit que O_d est un *opérateur de mise en mémoire* pour $\mathcal{D}$ ssi $\vdash_{\mathcal{F}} O_d : D^* \to \neg\neg D$, et pour tout $n \in \mathbf{N}$, il existe un λ-terme clos $\tau_n \simeq_\beta d_n$ et $\vdash_{\mathcal{F}} \tau_n : D$ tel que, pour tout $\theta_n \simeq_\beta d_n$, $(O_d\ \theta_n f) \succ (f\ \tau_n)$ où f est une nouvelle variable.

Exemple : On peut vérifier que $N \subseteq N^*$ et $\vdash_{\mathcal{F}} O_N : N^* \to \neg\neg N$. Donc O_N est un opérateur de mise en mémoire pour $\mathcal{N}$. □

4 Le contre exemple

Soit $P = \forall X \forall Y \{((X \to Y) \to X) \to X\}$ (P est la loi de Pierce) et $Q = P \to \forall X X$.

Lemme 3
1) $\vdash_{\mathcal{F}_c} P$.

2) Il existe un λ-terme clos t_P tel que $\vdash_{\mathcal{F}} t_P : P^*$.

Preuve
1) C'est un résultat connu. Faisons la démonstration.

$$\begin{aligned}
\neg X, X, \neg Y \vdash_{\mathcal{F}_c} \bot &\implies \neg X, X \vdash_{\mathcal{F}_c} \neg\neg Y \\
&\implies \neg X, X \vdash_{\mathcal{F}_c} Y \\
&\implies \neg X, (X \to Y) \to X \vdash_{\mathcal{F}_c} X \to Y \\
&\implies \neg X, (X \to Y) \to X \vdash_{\mathcal{F}_c} X \\
&\implies \neg X, (X \to Y) \to X \vdash_{\mathcal{F}_c} \bot \\
&\implies (X \to Y) \to X \vdash_{\mathcal{F}_c} \neg\neg X \\
&\implies (X \to Y) \to X \vdash_{\mathcal{F}_c} X \\
&\implies \vdash_{\mathcal{F}_c} P.
\end{aligned}$$

2) D'après 1) et le théorème 6, on a $\vdash_{\mathcal{F}} P^*$, donc il existe un λ-terme t_P tel que $\vdash_{\mathcal{F}} t_P : P^*$. Un exemple d'un tel λ-terme est $t_P = \lambda x \lambda y (x\ \lambda z \lambda \alpha (z\ y)\ y)$. En effet :

$$\begin{aligned}
z : \neg X, y : X, \alpha : Y \vdash_{\mathcal{F}} (z\ y) : \bot &\implies y : X \vdash_{\mathcal{F}} \lambda z \lambda \alpha (z\ y) : \neg X \to \neg Y \\
&\implies x : (\neg X \to \neg Y) \to \neg X, y : X \vdash_{\mathcal{F}} (x\ \lambda z \lambda \alpha (z\ y)\ y) : \bot \\
&\implies \vdash_{\mathcal{F}} t_p : P^*.
\end{aligned}$$

□

Lemme 4
1) $\nvdash_{\mathcal{F}} Q \to P$.
2) $\nvdash_{\mathcal{F}_c} (Q \to P) \to Q$.

Preuve
1) Un contexte Γ est dit *bon* ssi Γ est de la forme $[\alpha : Q, \{x_i : (X_i \to Y_i) \to X_i\}_{1 \le i \le n}, \{y_j : X_j\}_{1 \le j \le m}]$ où :

- $X_i \neq X_j$ $(1 \le i < j \le n)$;
- $Y_i \neq Y_j$ $(1 \le i < j \le m)$;
- $X_i \neq Y_j$ $(1 \le i \le n)$ et $(1 \le j \le m)$.

Il suffit de démontrer que pour tout contexte bon Γ il n'existe pas de λ-terme u tel que $\Gamma \vdash_{\mathcal{F}} u : P$. Nous démontrons ceci par induction sur u.
u ne peut pas être une variable. Si $u = (z\ u_1 ... u_m)$ $(m \ge 1)$, alors $z = \alpha$, et $\Gamma \vdash_{\mathcal{F}} u_1 : P$. Ce qui est impossible par hypothèse d'induction. Donc $u = \lambda x v$, et $\Gamma, x : (X \to Y) \to X \vdash_{\mathcal{F}} v : X$ où X, Y sont des nouvelles variables différentes. v ne peut pas être ni une variable ni un λ-terme qui commence par λ. Donc $v = (z\ v_1 ... v_m)$ $(m \ge 1)$ et $z \neq x_i, y_j$. Il reste, donc, deux cas à voir :

- Si $z = \alpha$, alors $\Gamma, x : (X \to Y) \to X \vdash_{\mathcal{F}} v_1 : P$. Ce qui est impossible par hypothèse d'induction.

– Si $z = x$, alors $m = 1$, et $\Gamma, x : (X \to Y) \to X \vdash_{\mathcal{F}} v_1 : X \to Y$. v_1 ne peut pas être une variable. Donc on a de nouveau deux cas à voir :

– Si $v_1 = (z\ v'_1 ... v'_k)$ $(k \geq 1)$, alors $z = \alpha$, et $\Gamma, x : (X \to Y) \to X \vdash_{\mathcal{F}} v'_1 : P$. Ce qui est impossible par hypothèse d'induction.

– Si $v_1 = \lambda y w$, alors $\Gamma, x : (X \to Y) \to X, y : X \vdash_{\mathcal{F}} w : Y$. w ne peut pas être ni une variable ni un λ-terme qui commence par λ. Donc $w = (z\ w_1 ... w_r)$ $(r \geq 1)$ et $z \neq x, y, x_i, y_j$. Donc $z = \alpha$, et $\Gamma, x : (X \to Y) \to X, y : X \vdash_{\mathcal{F}} w_1 : P$. Ce qui est impossible par hypothèse d'induction.

2) Si $\vdash_{\mathcal{F}_c} (Q \to P), P \to \forall X X$, alors $\vdash_{\mathcal{F}_c} \forall X X$ (puisque $\vdash_{\mathcal{F}_c} P$). Ce qui contredit le théorème 5. □

Lemme 5 *Soit w un λ-terme normal. Si $\alpha : Q \to P, x : X, f : X \to X \vdash_{\mathcal{F}} w : X$, alors il existe un $n \in \mathbf{N}$ tel que $w = (f^n\ x)$.*

Preuve Par induction sur w.
Le λ-terme w ne peut pas commencer par un λ. Si w est une variable, alors $w = x$. Donc $w = (y\ w_1 ... w_m)$ $(m \geq 1)$, et on a deux possibilités pour la variable y.

– Si $y = \alpha$, alors $\alpha : Q \to P, x : X, f : X \to X \vdash_{\mathcal{F}} w_1 : Q$, et $Q \to P, X, X \to X \vdash_{\mathcal{F}} Q$. Soit U une formule close démontrable dans le système logique $\mathcal{F}$. D'après le lemme 2, on a $\{Q \to P, X, X \to X\}[U/X] \vdash_{\mathcal{F}} Q[U/X]$, donc $Q \to P \vdash_{\mathcal{F}} Q$ (puisque U et $U \to U$ sont démontrables). Ce qui contredit 2) du lemme 4.

– Si $y = f$, alors $m = 1$ et $\alpha : Q \to P, x : X, f : X \to X \vdash_{\mathcal{F}} w_1 : X$. Par hypothèse d'induction, il existe un $n \in \mathbf{N}$ tel que $w_1 = (f^n\ x)$, donc $w = (f^{n+1}\ x)$. □

Soit $D = (Q \to P) \to N$ et, pout tout $n \in \mathbf{N}$, $d_n = \lambda \alpha \underline{n}$.

Lemme 6 *Soit t un λ-terme normal clos. $\vdash_{\mathcal{F}} t : D$ ssi il existe un $n \in \mathbf{N}$ tel que $t = d_n$.*

Preuve $\Longleftarrow$) Facile à vérifier.
$\Longrightarrow$) Comme t est clos, alors $t = \lambda \alpha u$ et $\alpha : Q \to P \vdash_{\mathcal{F}} u : N$. u ne peut pas être une variable et si $u = (\alpha\ u_1 ... u_m)$ $(m \geq 1)$, alors $\alpha : Q \to P \vdash_{\mathcal{F}} u_1 : Q$, donc $Q \to P \vdash_{\mathcal{F}} Q$. Ce qui contredit 2) du lemme 4. Donc $u = \lambda x v$, et $\alpha : Q \to P, x : X \vdash_{\mathcal{F}} v : (X \to X) \to X$. v ne peut pas être une variable, donc on a deux cas à voir.

– Si $v = (y\ v_1 ... v_m)$ $(m \geq 1)$, alors $y = \alpha$, $\alpha : Q \to P, x : X \vdash_{\mathcal{F}} v_1 : Q$, et $Q \to P, X \vdash_{\mathcal{F}} Q$. Soit U une formule close démontrable dans le système logique $\mathcal{F}$. D'après le lemme 2, on a $\{Q \to P, X\}[U/X] \vdash_{\mathcal{F}} Q[U/X]$, donc $Q \to P \vdash_{\mathcal{F}} Q$. ΛCe qui contredit 2) du lemme 4.

– Si $v = \lambda f w$, alors $\alpha : Q \to P, x : X, f : X \to X \vdash_{\mathcal{F}} w : X$. Donc, d'après le lemme 5, il existe un $n \in \mathbf{N}$ tel que $w = (f^n\ x)$ et $t = d_n$. □

Lemme 7 Soit $S_d = \lambda n \lambda \alpha(\underline{S}\ (n\ \alpha))$.
1) Pour tout $n \in \mathbf{N}$, $(S_d\ d_n) \simeq_\beta d_{n+1}$.
2) Pour tout $n \in \mathbf{N}$, $({S_d}^n\ d_0) \simeq_\beta d_n$.

Preuve Facile à vérifier. □

Lemme 8 *Soit* $O_d = \lambda n(n\ T_P\ \hat{d_0}\ \hat{S_d})$ *où*
$T_P = \lambda \alpha t_P$, $\hat{d_0} = \lambda f(f\ d_0)$, *et* $\hat{S_d} = \lambda x \lambda y(x\ \lambda z(y\ (S_d\ z)))$.
O_d *est un opérateur de mise en mémoire pour le système numérique typé* $< D, \mathbf{d} >$.

Preuve On va démontrer que :
1) $D \subseteq D^*$ et $\vdash_{\mathcal{F}} O_d : D^* \rightarrow \neg\neg D$.
2) Pour tout $\theta_n \simeq_\beta d_n$, $(O_d\ \theta_n\ f) \succ (f\ ({S_d}^n\ d_0))$.

1) Il est facile de vérifier que $D \subseteq D^*$.
On a :

$$\vdash_{\mathcal{F}} d_0 : D \implies \vdash_{\mathcal{F}} \hat{d_0} : \neg\neg D$$

et

$$\begin{aligned} \vdash_{\mathcal{F}} S_d : D \rightarrow D &\implies y : \neg D, z : D \vdash_{\mathcal{F}} (y\ (S_d\ z)) : \bot \\ &\implies x : \neg\neg D, y : \neg D \vdash_{\mathcal{F}} (x\ \lambda z(y\ (S_d\ z))) : \bot \\ &\implies \vdash_{\mathcal{F}} \hat{S_d} : \neg\neg D \rightarrow \neg\neg D. \end{aligned}$$

De plus, d'après le lemme 3, on a $\vdash_{\mathcal{F}} t_P : P^*$, donc $\vdash_{\mathcal{F}} T_P : (Q \rightarrow P)^* = P^* \rightarrow Q^*$.
D'où

$$\begin{aligned} n : D^* \vdash_{\mathcal{F}} (n\ T_P) : N^* &\implies n : D^* \vdash_{\mathcal{F}} (n\ T_P) : \neg\neg D, (\neg\neg D \rightarrow \neg\neg D) \rightarrow \neg\neg D \\ &\implies \vdash_{\mathcal{F}} O_d : D^* \rightarrow \neg\neg D. \end{aligned}$$

2) Soit $\theta_n \simeq_\beta d_n$.

– Si $n = 0$, alors $\theta_n \succ d_0$.

– Si $n \neq 0$, alors $\theta_n \succ \lambda\alpha\lambda x \lambda g(g\ t_{n-1})$, $t_{n-k} \succ (g\ t_{n-k-1})$ $(1 \leq k \leq n-1)$, et $t_0 \succ x$.

Si $n = 0$, alors

$$\begin{aligned} (O_d\ \theta_n\ f) &\sim (\hat{d_0}\ f) \\ &\sim (f\ d_0). \end{aligned}$$

Si $n \neq 0$, alors

$$\begin{aligned} (O_d\ \theta_n\ f) &\sim (\hat{S_d}\ t_{n-1}[\hat{S_d}/g, \hat{d_0}/x]\ f) \\ &\sim (t_{n-1}[\hat{S_d}/g, \hat{d_0}/x]\ \lambda z(f\ (S_d\ z))). \end{aligned}$$

On définit deux suites de λ-termes $(\tau_i)_{1 \leq i \leq n}$:

$$\tau_1 = \lambda z(f\ (S_d\ z))$$
$$\text{et } \tau_{k+1} = \lambda z(\tau_k\ (S_d\ z)) \text{ pour tout } (1 \leq k \leq n-1)$$

Démontrons (par récurrence sur k) que, pour tout $(1 \leq k \leq n)$, on a :

$$(O_d\ \theta_n\ f) \sim (t_{n-k}[\hat{S_d}/g, \hat{d_0}/x]\ \tau_k)$$

– Pour $k = 1$, le résultat est vrai.

– Supposons le résultat vrai pour k, et démontrons le pour $k+1$.

$$\begin{array}{rcl}(O_d\ \theta_n\ f) & \sim & (t_{n-k}[\hat{S_d}/g, \hat{d_0}/x]\ \tau_k) \\ & \sim & (\hat{S_d}\ t_{n-k-1}[\hat{S_d}/g, \hat{d_0}/x]\ \tau_k) \\ & \sim & (t_{n-k-1}[\hat{S_d}/g, \hat{d_0}/x]\ \lambda z(\tau_k\ (S_d\ z))) \\ & = & (t_{n-k-1}[\hat{S_d}/g, \hat{d_0}/x]\ \tau_{k+1}).\end{array}$$

Donc, en particulier, pour $k = n$ on a :

$$\begin{array}{rcl}(O_d\ \theta_n\ f) & \sim & (t_0[\hat{S_d}/g, \hat{d_0}/x]\ \tau_k) \\ & \sim & (\hat{d_0}\ \tau_n) \\ & \sim & (\tau_n\ d_0).\end{array}$$

Démontrons (par récurrence sur k) que, pour tout $(1 \leq k \leq n)$, on a :

$$\tau_k \sim \lambda z(f\ ({S_d}^k\ z))$$

– Pour $k = 1$, le résultat est vrai.

– Supposons le résultat vrai pour k, et démontrons le pour $k+1$.

$$\begin{array}{rcl}\tau_{k+1} & = & \lambda z(\tau_k\ (S_d\ z)) \\ & \sim & \lambda z(\lambda z(f\ ({S_d}^k\ z))\ (S_d\ z)) \\ & \sim & \lambda z(f\ ({S_d}^{k+1}\ z)).\end{array}$$

Donc, en particulier, pour $k = n$ on a : $\tau_n \sim \lambda z(f\ ({S_d}^n\ z))$.
Et

$$\begin{array}{rcl}(O_d\ \theta_n\ f) & \sim & (\lambda z(f\ ({S_d}^n\ z))\ d_0) \\ & \sim & (f\ ({S_d}^n\ d_0)).\end{array}$$

D'où $(O_d\ \theta_n\ f) \succ (f\ ({S_d}^n\ d_0))$. □

Lemme 9 Soit $O_B = \lambda n(n\ \lambda f(f\ T)\ \lambda f(f\ F))$.
1) $\vdash_{\mathcal{F}} O_B : B^* \rightarrow \neg\neg B$.
2) Pour tout $\epsilon = T$ *ou* F *et pour tout* $\theta_\epsilon \simeq_\beta \epsilon$, $(O_B\ \theta_\epsilon\ f) \succ (f\ \epsilon)$.

Preuve
1) On a :

$$\vdash_{\mathcal{F}} T : B \implies \vdash_{\mathcal{F}} \lambda f(f\ T) : \neg\neg B$$

et

$$\vdash_{\mathcal{F}} F : B \implies \vdash_{\mathcal{F}} \lambda f(f\ F) : \neg\neg B.$$

Donc

$$n : B^* \vdash_{\mathcal{F}} n : \neg\neg B, \neg\neg B \to \neg\neg B \implies \vdash_{\mathcal{F}} O_B : B^* \to \neg\neg B.$$

2) Si $\theta_\epsilon \simeq_\beta \epsilon$, alors $\theta_\epsilon \succ \epsilon$, et donc

$$\begin{aligned}(O_B\ \theta_\epsilon\ f) &\sim (\theta_\epsilon\ \lambda f(f\ T)\ \lambda f(f\ F)\ f)\\ &\sim (\epsilon\ \lambda f(f\ T)\ \lambda f(f\ F)\ f)\\ &\sim (f\ \epsilon).\end{aligned}$$

D'où $(O_B\ \theta_\epsilon\ f) \succ (f\ \epsilon)$. □

Lemme 10 *Soit t un λ-terme normal clos. $\vdash_{\mathcal{F}} t : D \to B$ ssi $t = \lambda\alpha T$ ou $t = \lambda\alpha F$.*

Preuve $\Longleftarrow$) Facile à vérifier.
$\Longrightarrow$) Comme t est clos, alors $t = \lambda\alpha u$ et $\alpha : D \vdash_{\mathcal{F}} u : B$. u ne peut pas être une variable, donc on a deux cas à voir.

– Si $u = (\alpha\ u_1 ... u_m)$ $(m \geq 1)$, alors $\alpha : D \vdash_{\mathcal{F}} u_1 : Q \to P$, donc $\vdash_{\mathcal{F}} Q \to P$ (car D est démontrable dans le système logique $\mathcal{F}$). Ce qui contredit 1) du lemme 4.

– Si $u = \lambda x v$, alors $\alpha : D, x : X \vdash_{\mathcal{F}} v : X \to X$. v ne peut pas être une variable, donc on a de nouveau deux cas à voir.

– Si $v = (z\ v_1 ... v_m)$ $(m \geq 1)$, alors $y = \alpha$, $\alpha : D, x : X \vdash_{\mathcal{F}} v_1 : Q\ P$, et $D, X \vdash_{\mathcal{F}} Q \to P$. Soit U une formule close démontrable dans le système logique $\mathcal{F}$. D'après le lemme 2, on a $\{D, X\}[U/X] \vdash_{\mathcal{F}} \{Q \to P\}[U/X]$, donc $\vdash_{\mathcal{F}} Q \to P$. Ce qui contredit 1) du lemme 4.

– Si $v = \lambda y w$, alors $\alpha : D, x : X, y : X \vdash_{\mathcal{F}} w : X$. w ne peut pas commencer par un λ et si w est une variable, alors $w = x$ ou $w = y$, donc $t = \lambda\alpha T$ or $t = \lambda\alpha F$. Il reste donc le cas où $w = (\alpha\ w_1 ... w_m)$ $(m \geq 1)$. Dans ce cas on a $\alpha : D, x : X, y : X \vdash_{\mathcal{F}} w_1 : Q \to P$, et $D, X \vdash_{\mathcal{F}} Q \to P$. Soit U une formule close démontrable dans le système logique $\mathcal{F}$. D'après le lemme 2, on a $\{D, X, X\}[U/X] \vdash_{\mathcal{F}} \{Q \to P\}[U/X]$, donc $\vdash_{\mathcal{F}} Q \to P$. Ce qui contredit 1) du lemme 4. □

Lemme 11 Soit t un λ-terme normal.
1) Si $x : B, D \to X, y : X \vdash_{\mathcal{F}} t : B$, alors $t = T$ ou $t = F$.

2) Si $x : B, D \to X, y : X \vdash_{\mathcal{F}} t : D$*, alors il existe* $n \in \mathbf{N}$ *tel que* $t = d_n$*.*

Preuves Même preuve que celles des lemmes 5 et 6. □

Soit $E = \forall X\{((B, D \to X), X \to X\}$.

Pour tous λ-termes u, v, on note $\ll u, v \gg$ le λ-terme $\lambda x \lambda y(x\ u\ v)$.

Lemme 12 *Soit* t *un* λ*-terme normal clos.* $\vdash_{\mathcal{F}} t : E$ *ssi* $(t = F)$ *ou il existe* $n \in \mathbf{N}$ *tel que* $(t = \ll b, d_n \gg$ *où* $b = T$ *ou* $F)$*.*

Preuve $\Longleftarrow$) Facile à vérifier.
$\Longrightarrow$) Soit t un λ-terme normal clos tel que $\vdash_{\mathcal{F}} t : E$. Alors $t = \lambda x u$ et $x : B, D \to X \vdash_{\mathcal{F}} u : X \to X$. u ne peut pas être une variable, donc on a deux cas à voir.

- Si $u = (x\ u_1 ... u_m)$ $(m \geq 1)$, alors $x : B, D \to X \vdash_{\mathcal{F}} (x\ u_1) : D \to X$. Ce qui est impossible.

- Si $u = \lambda y v$, alors $x : B, D \to X, y : X \vdash_{\mathcal{F}} v : X$. v ne peut pas commencer par un λ, donc on a de nouveau deux cas à voir.

  - Si v est une variable, alors $v = y$ et $u = F$.

  - Si $v = (z\ v_1 ... v_m)$ $(m \geq 1)$, alors $z = x$, $n = 2$, $x : B, D \to X, y : X \vdash_{\mathcal{F}} v_1 : D$, et $x : B, D \to X, y : X \vdash_{\mathcal{F}} v_2 : B$. Donc, d'après le lemme 11, il existe $n \in \mathbf{N}$ tel que $t = \ll b, d_n \gg$ où $b = T$ ou $b = F$. □

Théorème 9 *Il existe un système numérique typé non adéaquat qui possède un opérateur de mise en mémoire.*

Preuve Soit $\mathcal{E} =< E, \mathbf{e} >$ où :

$$\begin{aligned} e_0 &= F \\ e_{2n+1} &= \ll F, d_n \gg \quad (n \geq 0) \\ e_{2n+2} &= \ll T, d_n \gg \quad (n \geq 0) \end{aligned}$$

Le test à zéro

Soit $Z_e = \lambda n(n\ \lambda x \lambda y F\ T)$.

Typage de Z_e

On a :

$$x : B, y : D \vdash_{\mathcal{F}} F : B \implies \vdash_{\mathcal{F}} \lambda x \lambda y F : B, D \to B$$

donc

$$\begin{aligned} n : E \vdash_{\mathcal{F}} n : (B, D \to B), B \to B &\implies n : E \vdash_{\mathcal{F}} (n\ \lambda x \lambda y F\ T) : B \\ &\implies \vdash_{\mathcal{F}} Z_e : E \to B. \end{aligned}$$

Fonctionnement de Z_e

Si $n = 0$, alors :

$$\begin{aligned} (Z_e\ e_n) &\simeq_\beta (F\ \lambda x \lambda y F\ T) \\ &\simeq_\beta T. \end{aligned}$$

Si $n \neq 0$, alors :

$$\begin{aligned} (Z_e\ e_{n+1}) &\simeq_\beta (e_{n+1}\ \lambda x \lambda y F\ T) \\ &\simeq_\beta (\lambda x \lambda y F\ b\ d_m) \\ &\simeq_\beta F. \end{aligned}$$

Le successeur

Soit $S_e = \lambda n((Z_e\ n)\ e_1\ ((n\ T\ T) \ll F, (S_d\ (n\ F\ d_0)) \gg \ll T, (n\ F\ d_0) \gg))$.

Typage de S_e

On a :

$$\begin{aligned} n : E \vdash_{\mathcal{F}} n : (B, D \to D), D \to D &\implies n : E \vdash_{\mathcal{F}} (n\ F\ d_0) : D \\ &\implies n : E \vdash_{\mathcal{F}} \ll T, (n\ F\ d_0) \gg : E \end{aligned}$$

et

$$\begin{aligned} n : E \vdash_{\mathcal{F}} n : (B, D \to D), D \to D &\implies n : E \vdash_{\mathcal{F}} (S_d\ (n\ F\ d_0)) : D \\ &\implies n : E \vdash_{\mathcal{F}} \ll F, (S_d\ (n\ F\ d_0)) \gg : E. \end{aligned}$$

Donc

$$\begin{aligned} n : E \vdash_{\mathcal{F}} n : (B, D \to B), B \to B &\implies n : E \vdash_{\mathcal{F}} (n\ T\ T) : B \\ &\implies n : E \vdash_{\mathcal{F}} (n\ T\ T) : E, E \to E \\ &\implies n : E \vdash_{\mathcal{F}} ((n\ T\ T) \ll F, (S_d\ (n\ F\ d_0)) \gg \\ &\qquad \ll T, (n\ F\ d_0) \gg) : E. \end{aligned}$$

D'où

$$\begin{aligned} n : E \vdash_{\mathcal{F}} (Z_e\ n) : B &\implies n : E \vdash_{\mathcal{F}} (Z_e\ n) : E, E \to E \\ &\implies \vdash_{\mathcal{F}} S_e : E \to E. \end{aligned}$$

Fonctionnement de S_e

On a trois cas :

$$\begin{array}{rcl}(S_e\ e_0) & \simeq_\beta & (T\ e_1\ ((e_0\ T\ T) \ll F, (S_d\ (e_0\ F\ d_0)) \gg \ll T, (e_0\ F\ d_0) \gg)) \\ & \simeq_\beta & e_1.\end{array}$$

$$\begin{array}{rcl}(S_e \ll F, d_n \gg) & \simeq_\beta & (F\ e_1\ ((\ll F, d_n \gg\ T\ T) \ll F, (S_d\ (e_0\ F\ d_0)) \gg \\ & & \ll T, (\ll F, d_n \gg\ F\ d_0) \gg)) \\ & \simeq_\beta & ((\ll F, d_n \gg\ T\ T) \ll F, (S_d\ (e_0\ F\ d_0)) \gg \\ & & \ll T, (\ll F, d_n \gg\ F\ d_0) \gg) \\ & \simeq_\beta & (F \ll F, (S_d\ (e_0\ F\ d_0)) \gg \ll T, (\ll F, d_n \gg\ F\ d_0) \gg) \\ & \simeq_\beta & \ll T, (\ll F, d_n \gg\ F\ d_0) \gg \\ & \simeq_\beta & \ll T, d_n \gg.\end{array}$$

$$\begin{array}{rcl}(S_e \ll T, d_n \gg) & \simeq_\beta & (F\ e_1\ ((\ll T, d_n \gg\ T\ T) \ll F, (S_d\ (\ll F, d_n \gg\ F\ d_0)) \gg \\ & & \ll T, (\ll T, d_n \gg\ F\ d_0) \gg)) \\ & \simeq_\beta & ((\ll T, d_n \gg\ T\ T) \ll F, (S_d\ (\ll F, d_n \gg\ F\ d_0)) \gg \\ & & \ll T, (\ll T, d_n \gg\ F\ d_0) \gg) \\ & \simeq_\beta & (T \ll F, (S_d\ (\ll T, d_n \gg\ F\ d_0)) \gg \ll T, (\ll T, d_n \gg\ F\ d_0) \gg) \\ & \simeq_\beta & \ll F, (S_d\ (\ll T, d_n \gg\ F\ d_0)) \gg \\ & \simeq_\beta & \ll F, (S_d\ d_n) \gg \\ & \simeq_\beta & \ll F, d_{n+1} \gg.\end{array}$$

L'opérateur de mise en mémoire

Il est facile de vérifier que $E \subseteq E^*$.

Soit $O_e = \lambda n(n\ \hat{S_e}\ \hat{e_0})$ où $\hat{S_e} = \lambda x \lambda y \lambda z((O_B\ x)\ \lambda u((O_d\ y)\ \lambda v(z \ll u, v \gg)))$ et $\hat{e_0} = \lambda f(f\ e_0)$.

Typage de O_e

On a :

$$\vdash_{\mathcal{F}} e_0 : E \implies \vdash_{\mathcal{F}} \hat{e_0} : \neg\neg E.$$

D'autre part, en utilisant les lemmes 8 et 9, on a :

$$\begin{array}{rcl}u : B, v : D \vdash_{\mathcal{F}} \ll u, v \gg : E & \implies & u : B, z : \neg E \vdash_{\mathcal{F}} \lambda v(z \ll u, v \gg) : \neg D \\ & \implies & y : D^*, z : \neg E \vdash_{\mathcal{F}} \lambda u((O_d\ y)\ \lambda v(z \ll u, v \gg)) : \neg B\end{array}$$

$$\begin{aligned} \Longrightarrow \quad & x : B^*, y : D^* \vdash_{\mathcal{F}} \lambda z((O_B\ x)\ \lambda u((O_d\ y) \\ & \lambda v(z\ \ll u, v \gg))) : \neg\neg E \\ \Longrightarrow \quad & \vdash_{\mathcal{F}} \hat{S_e} : B^*, D^* \to \neg\neg E. \end{aligned}$$

D'où

$$n : E^* \vdash_{\mathcal{F}} n : (B^*, D^* \to \neg\neg E), \neg\neg E\ \neg\neg E \quad \Longrightarrow \quad \vdash_{\mathcal{F}} O_e : E^* \to \neg\neg E.$$

Fonctionnement de O_e

Soit $\theta_n \simeq_\beta e_n$, alors :

- Si $n = 0$, alors $\theta_n \succ e_0$.
- Si $n \neq 0$, alors $\theta_n \succ \lambda x \lambda y (x\ \alpha_n\ \beta_n)$ où $\alpha_n \simeq_\beta \epsilon$ et $\beta_n \simeq_\beta d_m$ si $e_n = \ll \epsilon, d_m \gg$.

Si $n = 0$, alors

$$\begin{aligned} (O_e\ \theta_n\ f) \quad & \sim \quad (e_0\ \hat{S_e}\ \hat{e_0}\ f) \\ & \sim \quad (\hat{e_0}\ f) \\ & \sim \quad (f\ e_0). \end{aligned}$$

Si $n \neq 0$, alors

$$\begin{aligned} (O_e\ \theta_n\ f) \quad & \sim \quad (\hat{S_e}\ \alpha_n\ \beta_n\ f) \\ & \sim \quad ((O_B\ \alpha_n)\ \lambda u((O_d\ \beta_n)\ \lambda v(f\ \ll u, v \gg))). \end{aligned}$$

D'après le Lemma 9, on a : pour tout λ-terme U, $((O_B\ \alpha_n)\ U) \sim (U\ \epsilon)$.

Donc

$$\begin{aligned} (O_e\ \theta_n\ f) \quad & \sim \quad (\lambda u((O_d\ \beta_n)\ \lambda v(f\ \ll u, v \gg))\ \epsilon) \\ & \sim \quad ((O_d\ \beta_n)\ \lambda v(f\ \ll \epsilon, v \gg)). \end{aligned}$$

D'après le Lemma 8, on a : pour tout λ-terme V, $((O_d\ \beta_n)\ V) \sim (V\ ({S_d}^m\ d_0))$.

Donc

$$\begin{aligned} (O_e\ \theta_n\ f) \quad & \sim \quad (\lambda v(f\ \ll \epsilon, v \gg)\ ({S_d}^m\ d_0)) \\ & \sim \quad (f\ \ll \epsilon, ({S_d}^m\ d_0) \gg). \end{aligned}$$

D'où $(O_e\ \theta_n\ f) \succ (f\ \ll \epsilon, ({S_d}^m\ d_0) \gg)$

L'inéxistance d'un prédécesseur

Supposons qu'il existe un λ-terme normal clos P_e pour le prédécesseur.
Soit $P' = \lambda n(P_e \ll F, n \gg T\ F)$.

On a

$$\begin{aligned} n : D \vdash_{\mathcal{F}} \ll F, n \gg : E &\implies n : D \vdash_{\mathcal{F}} (P_e \ll F, n \gg) : E \\ &\implies n : D \vdash_{\mathcal{F}} (P_e \ll F, n \gg) : (B, D \to B), B \to B \\ &\implies n : D \vdash_{\mathcal{F}} (P_e \ll F, n \gg T\ F) : B \\ &\implies \vdash_{\mathcal{F}} P' : D \to B. \end{aligned}$$

Donc, d'après le lemme 10, on obtient $P' = \lambda \alpha T$ ou $P' = \lambda \alpha F$.
Mais on a :

$$\begin{aligned} (P'\ d_0) &\simeq_\beta (P_e\ e_1\ T\ F) \\ &\simeq_\beta (e_0\ T\ F) \\ &\simeq_\beta F \end{aligned}$$

et

$$\begin{aligned} (P'\ d_1) &\simeq_\beta (P_e\ e_3\ T\ F) \\ &\simeq_\beta (e_2\ T\ F) \\ &\simeq_\beta (T\ T\ d_0) \\ &\simeq_\beta T. \end{aligned}$$

D'où une contradiction. □

Remarque : Il est facile de vérifier que le λ-terme
$P_e = \lambda n((Z_e\ n)\ e_0\ ((n\ T\ T) \ll F, (n\ F\ d_0) \gg (\underline{Z}\ (n\ F\ d_0\ T)) \ll T, (\lambda \alpha (\underline{P}\ (n\ F\ d_0\ \alpha))) \gg F)))$
est un prédécesseur (non typable dans le système $\mathcal{F}$ de type $E \to E$) pour le système numérique **e**.

5 Conclusion

Suite à cette étude, deux questions se posent :

- Est-il vrai que chaque système numérique typé adéquat possède un opérateur de mise en mémoire? En effet l'opérateur de mise en mémoire qu'on a construit pour un système numérique adéquat quelconque (voir la preuve du théorème 6) utilise un opérateur de point fixe et donc il est non typable dans le système $\mathcal{F}$.
- Quelles sont les fonctions qu'on peut représenter dans un système numérique typé adéquat?

References

[1] H. Barendregt. *The lambda calculus, its syntax and semantics.*
North Holland, 1984

[2] J.-Y. Girard, Y. Lafont, P. Taylor. *Proofs and types.*
Cambridge University Press, 1986.

[3] J.-L. Krivine. *Lambda calcul, types et modèles.*
Masson, 1990

[4] J.-L. Krivine. *Opérateurs de mise en mémoire et traduction de Gődel*
Archive for Mathematical Logic 30 (1990), pp. 241-267.

[5] K. Nour. *Opérateurs de mise en mémoire en lambda-calcul pure et typé*
Thèse de Doctorat, Université de Chambéry, 1993.

[6] K. Nour. *An example of a non adequate numeral system.*
CRAS. Paris, 323, Série I (1996), pp. 439-442.

[7] K. Nour. *A conjecture on numeral system.*
Notre Dame of Formal Logic, vol. 38 (1997), pp. 270-275.