

K. NOUR

## Entiers intuitionnistes et entiers classiques en $\lambda C$ -calcul

*Informatique théorique et applications*, tome 29, n° 4 (1995),  
p. 293-313

[http://www.numdam.org/item?id=ITA\\_1995\\_\\_29\\_4\\_293\\_0](http://www.numdam.org/item?id=ITA_1995__29_4_293_0)

© AFCET, 1995, tous droits réservés.

L'accès aux archives de la revue « Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

## ENTIERS INTUITIONNISTES ET ENTIERS CLASSIQUES EN $\lambda$ C-CALCUL (\*)

par K. NOUR <sup>(1)</sup>

Communiqué par R. CORI

---

Résumé. – Dans cet article, on présente une preuve syntaxique de la caractérisation opérationnelle des entiers classiques (des termes clos typables, dans un système de typage classique, de type  $N[s^n(0)]$ ) de J. L. Krivine. On donne ensuite des conditions sur le système de typage classique afin d'obtenir les entiers intuitionnistes.

Abstract. – In this paper, we give a syntactical proof of the J. L. Krivine's operational characterization of classical integers (a closed terms typable, in a classical type system, by the type  $N[s^n(0)]$ ). We give later conditions on the classical type system in order to obtain the intuitionistic integers.

### INTRODUCTION

Dans cet article, nous considérons le  $\lambda$ -calcul avec la réduction de tête (appelée « appel par nom ») muni d'un système de typage basé sur la logique intuitionniste du second ordre : le système  $\mathcal{AF}2$  de J. L. Krivine.

Ce système se distingue du système  $\mathcal{F}$  de J. Y. Girard par sa capacité à exprimer les spécifications exactes des programmes, ce qui permet d'obtenir un programme calculant une fonction en écrivant une démonstration de sa totalité.

Comment programme-t-on en  $\mathcal{AF}2$  ?

- On exprime tout d'abord les types de données par des formules. Par exemple, pour le type des entiers naturels, on introduit une formule

---

(\*) Reçu en décembre 1993, accepté en septembre 1994.

(1) LAMA-Équipe de Logique, Université de Savoie – Campus Scientifique, 73376 Le Bourget-du-Lac Cedex, France.  
email nour@univ-savoie.fr

$N[x]$  signifiant «  $x$  est un entier » défini par :  $N[x] = \forall X \{ \forall y [X(y) \rightarrow X(sy)] \rightarrow [X(0) \rightarrow X(x)] \}$  ( $X$  est un symbole de relation unaire,  $0$  est une constante pour le zéro, et  $s$  est un symbole de fonction unaire pour le successeur). Ce type signifie que  $x$  est un entier si et seulement si  $x$  appartient au plus petit ensemble contenant zéro et stable par le successeur.

La représentation des données en  $\lambda$ -calcul s'obtient par démonstration à partir de la définition logique du type de données. Ainsi la représentation de l'entier  $n$  est obtenue en démontrant l'énoncé  $N[s^n(0)]$ . L'élément clé qui fait fonctionner les choses est l'unicité de la représentation des données. Pour les entiers, on démontre facilement que les entiers de Church sont les seuls  $\lambda$ -termes normaux de type  $N[s^n(0)]$   $n \geq 0$ .

- On exprime ensuite les spécifications du programme par un système d'équations définissant la fonction à calculer.

- On démontre enfin le théorème énonçant la totalité de la fonction : dans le cas où la fonction  $f$  est définie de  $\mathbf{N}^m \rightarrow \mathbf{N}$ , on obtient un programme pour  $f$  en démontrant le théorème :

$$\forall x_1 \dots \forall x_m \{ N[x_1], \dots, N[x_m] \rightarrow N[f(x_1, \dots, x_m)] \}$$

(ceci signifie que si  $x_1, \dots, x_m$  sont des entiers, alors  $f(x_1, \dots, x_m)$  en un entier).

J. L. Krivine a ajouté une nouvelle constante, notée  $C$ , au  $\lambda$ -calcul pour deux raisons :

- Pour modéliser des techniques de programmation utilisées dans les langages impératifs : (l'« affection » – l'instruction « exit » ainsi que ces généralisations – la boucle « while » – l'instruction « input » – ...).

Comme nous considérons la stratégie de l'appel par nom, chaque terme est considéré comme un programme où l'exécution est la réduction de tête du terme. Pour ceci la réduction associée à la nouvelle constante  $C$  est définie seulement si elle est en tête.

- Dans le système de typage nous déclarons  $C$  de type  $\forall X \{ \neg X \rightarrow X \}$  qui axiomatise la logique classique comme une extension de la logique intuitionniste. Ainsi, notre système est étendu à la logique classique par la « correspondance de Curry-Howard » usuelle.

La méthode détaillée ci-dessus pour programmer des fonctions en  $\lambda$ -calcul typé est mal adaptée en logique classique. Ceci provient du fait qu'en logique classique on perd la propriété de l'unicité de la représentation des données.

Par exemple :  $\theta_1 = \lambda f \lambda x (C) \lambda y (y) (f) (C) \lambda z (y) (f) x$  est un  $\lambda C$ -terme normal de type  $N [s(0)]$  (voir p. 303). De plus il est difficile *a priori* de connaître la valeur d'un entier classique (un  $\lambda C$ -terme normal  $\theta_n$  de type  $N [s^n(0)]$   $n \geq 0$ ).

On est amené alors à se poser les questions suivantes :

- Peut-on caractériser les entiers classiques ?
- Peut-on connaître la « valeur » d'un entier classique ?

Pour résoudre ces problèmes, M. Parigot a proposé une solution basée sur la notion d'opérateur de mise en mémoire typé. Il a démontré (dans son système classique) que si  $T$  est un  $\lambda$ -terme clos typable dans  $\mathcal{AF}2$  d'un certain type, et si  $\theta_n$  est un entier classique de valeur  $n$ , alors  $(T) \theta_n \lambda x x$  se réduit à  $\tau_n \simeq \beta n$  (voir [6]).

Ceci donne un moyen pour calculer la valeur d'un entier classique.

J. L. Krivine a démontré que ce résultat est valable dans son système. La démonstration repose sur un lemme clé caractérisant d'une manière fonctionnelle les entiers classiques (voir [2]).

Dans ce papier, nous présentons une démonstration *syntactique* de la caractérisation opérationnelle des entiers classiques de J. L. Krivine. Cette démonstration utilise des propriétés syntaxiques du système de typage et la notion de typage par des obstacles. Ce dernier typage s'obtient par des restrictions sur le système de typage destinées à empêcher la construction des termes typables d'un certain type. Cette démonstration est purement combinatoire et donc, en un certain sens, elle explique mieux le résultat de J. L. Krivine. Nous montrons ensuite comment on peut utiliser ce résultat pour connaître la valeur d'un entier classique.

Les entiers, que nous obtenons dans le système classique de J. L. Krivine, se comportent d'une manière un peu compliquée. Si nous analysons les raisons pour lesquelles nous avons ce genre de phénomènes, nous nous apercevons que ceci provient essentiellement du fait que l'absurde, qui représente dans ce système le type du calcul achevé, ne contient pas des informations sur la partie de l'entier que l'on est en train de calculer.

Pour avoir cette dernière propriété, nous ajoutons un symbole de relation unaire  $\perp$  pour l'absurde. Pour tout formule  $A$ , et pour tout entier  $n$ , nous notons  $\neg_n A$  la formule  $A \rightarrow \perp(n)$ . Nous déclarons la constante  $C$  de type  $\forall X \{ \forall n (\neg_n \neg_n X) \rightarrow X \}$  (la vraie règle de typage concernant la constante  $C$  est : si  $A$  est une formule, et  $t$  un  $\lambda C$ -terme de type  $\neg_n \neg_n A$  pour tout entier  $n$ , alors  $(C)t$  est de type  $A$ ). Nous démontrons alors que, dans ce système de typage, nous obtenons des entiers classiques qui se

comportent d'une manière analogue aux entiers de Church. Nous appelons ces entiers les entiers intuitionnistes.

L'axiome  $\forall X \{ \forall n (\neg_n \neg_n X) \rightarrow X \}$  peut être interprété de la façon suivante. Nous considérons  $\perp(n)$  comme étant la contradiction à l'instant  $n$ , et  $\neg_n A$  comme étant la formule  $A$  fautive à l'instant  $n$ . De fait, l'axiome signifie que nous pouvons affirmer qu'une formule est vraie si elle n'est pas fautive à chaque instant. Nous pouvons appeler cette logique « la logique persistante » <sup>(1)</sup>.

Nous présentons enfin deux systèmes de typage obtenus par des restrictions sur le système classique de J. L. Krivine. Nous démontrons, grâce au système de typage avec plusieurs faux, que dans ces nouveaux systèmes nous obtenons les entiers intuitionnistes. De plus nous démontrons que ces deux systèmes sont des systèmes intuitionnistes mais ayant la propriété d'utiliser la constante  $C$  selon certaines conditions.

L'article est organisé de la manière suivante :

- Les parties 1 et 2 sont consacrées à des préliminaires sur le  $\lambda C$ -calcul pur et typé.

- Dans la partie 3, nous rappelons quelques propriétés syntaxiques du système de typage  $\mathcal{AF}2$ . Ces propriétés jouent un rôle essentiel dans les démonstrations.

- Dans la partie 4, nous présentons la notion du typage avec des obstacles.

- Dans la partie 5, nous présentons la démonstration syntaxique de la caractérisation opérationnelle des entiers classiques de J. L. Krivine. Nous montrons comment on peut utiliser ce résultat pour connaître la valeur d'un entier classique.

- Dans la partie 6, nous présentons un système de typage « classique » contenant plusieurs « faux ». Dans ce système de typage nous montrons que les seuls  $\lambda C$ -termes typables de type  $N[s^n(0)]$  sont les entiers intuitionnistes.

- Dans la partie 7, nous donnons des conditions sur le système de typage classique afin d'obtenir les entiers intuitionnistes. Nous montrons qu'avec ces restrictions nous obtenons la logique intuitionniste.

Une question naturelle se pose à la suite de cette étude, à savoir : *peut-on trouver un système de typage classique dans lequel les entiers se comportent d'une manière uniforme ?* Jusqu'à présent, cette question est encore ouverte.

---

<sup>(1)</sup> Ce système logique sera étudié dans un autre travail.

## 1. LE $\lambda C$ -CALCUL

Ajoutons une constante  $C$  au  $\lambda$ -calcul pur, et considérons les deux règles de réduction :

1)  $(\lambda x u) t t_1 \dots t_n \rightarrow (u [t/x]) t_1 \dots t_n$  pour tous  $u, t, t_1, \dots, t_n$  des termes du  $\lambda C$ -calcul;

2)  $(C) t t_1 \dots t_n \rightarrow (t) \lambda x (x) t_1 \dots t_n$  pour tous  $t, t_1, \dots, t_n$  des termes du  $\lambda C$ -calcul, et  $x$  est une variable qui ne paraît pas dans  $t_1, \dots, t_n$ .

La règle 1) est la  $\beta$ -réduction habituelle. On n'autorise pas ici le passage au contexte, et donc tout  $\lambda C$ -terme admet au plus une réduction.

Pour tous  $t, t'$  deux  $\lambda C$ -termes, on note  $t \succ_c t'$ , si  $t'$  est obtenu à partir de  $t$  après l'application d'un nombre fini des règles précédentes.

Il est facile de vérifier que : si  $t \succ_c t'$ , alors

$$t [t_1/x_1, \dots, t_n/x_n] \succ_c t' [t_1/x_1, \dots, t_n/x_n].$$

Un  $\lambda C$ -terme  $t$  est dit  $C$ -résoluble si et seulement si  $t \succ_c (f) t_1 \dots t_n$  où  $f$  est une variable.

## 2. LE SYSTÈME DE TYPAGE $C2$

On considère le calcul des prédicats classiques du second ordre, écrit avec les symboles suivants :

– Les seuls symboles logiques  $\perp$  (un symbole de relation 0-aire pour l'absurde),  $\rightarrow$  et  $\forall$ ;

– Un ensemble dénombrable  $\mathcal{V}_1$  de variables du premier ordre;

– Un ensemble dénombrable  $\mathcal{V}_2$  de variables du second ordre;

– Des symboles de fonction  $n$ -aire ( $n = 0, 1, \dots$ );

– Des symboles de relation  $n$ -aire ( $n = 0, 1, \dots$ ).

Les termes et les formules sont construits de la façon usuelle. On note  $\mathcal{T}$  l'ensemble des termes et  $\mathcal{F}$  l'ensemble des formules.

La formule  $F_1 \rightarrow (F_2 \rightarrow (\dots \rightarrow (F_n \rightarrow G) \dots))$  est notée  $F_1, F_2, \dots, F_n \rightarrow G$ , et la formule  $F \rightarrow \perp$  est notée  $\neg F$ .

Une  $\perp$ -substitution est une application  $S : \mathcal{V}'_1 \cup \mathcal{V}'_2 \rightarrow \mathcal{T} \cup \mathcal{F}$  vérifiant :

–  $\mathcal{V}'_i$  est un sous ensemble fini de  $\mathcal{V}_i$   $i = 1$  ou  $2$ ;

–  $S(\mathcal{V}'_1) \subset \mathcal{T}$  et  $S(\mathcal{V}'_2) \subset \mathcal{F}$ ;

–  $S(\perp) = \perp$ .

Elle possède une extension canonique (qui est notée aussi  $S$ )  $S : \mathcal{T} \cup \mathcal{F} \rightarrow \mathcal{T} \cup \mathcal{F}$ , définie par la manière usuelle. Si  $\mathcal{V}'_1 \cup \mathcal{V}'_2$  est l'ensemble  $\{x_1, \dots, x_n, X_1, \dots, X_m\}$ ,  $S(x_i) = t_i$   $1 \leq i \leq n$ , et  $S(X_i) = F_i$   $1 \leq i \leq m$ , la substitution  $S$  est aussi noté  $[t_1/x_1, \dots, t_n/x_n, F_1/X_1, \dots, F_m/X_m]$ .

Si  $X$  une variable de relation unaire,  $t$  et  $t'$  deux termes, alors la formule  $\forall X [X_t \rightarrow X'_t]$  est notée  $t = t'$ , et est dite *équation fonctionnelle* ou *formule équationnelle*. Un cas particulier de l'équation  $t = t'$  est une formule de la forme  $t[u_1/x_1, \dots, u_n/x_n] = t'[u_1/x_1, \dots, u_n/x_n]$  ou  $t'[u_1/x_1, \dots, u_n/x_n] = t[u_1/x_1, \dots, u_n/x_n]$ ,  $u_1, \dots, u_n$  étant des termes du langage.

On décrit un système de  $\lambda$ -calcul typé appelé *arithmétique fonctionnelle classique du second ordre* (en abrégé  $C_2$ ) dont les types sont les formules du langage. Étant donné un  $\lambda C$ -terme  $t$ , un type  $A$ , et un contexte  $\Gamma = x_1 : A_1, \dots, x_n : A_n$ , on définit au moyen des règles suivantes la notion «  $t$  est typable de type  $A$  dans le contexte  $\Gamma$  », et on écrit :  $\Gamma \vdash_{C_2} t : A$ .

(0)  $\Gamma \vdash_{C_2} C : \forall X \{ \neg X \rightarrow X \}$ .

(1)  $\Gamma \vdash_{C_2} x_i : A_i$   $1 \leq i \leq n$ .

(2) Si  $\Gamma, x : B \vdash_{C_2} t : C$ , alors  $\Gamma \vdash_{C_2} \lambda x t : B \rightarrow C$ .

(3) Si  $\Gamma \vdash_{C_2} u : A \rightarrow B$ , et  $\Gamma \vdash_{C_2} v : A$ , alors  $\Gamma \vdash_{C_2} (u)v : B$ .

(4) Si  $\Gamma \vdash_{C_2} t : A$ , et  $x$  ne figure pas dans  $\Gamma$ , alors  $\Gamma \vdash_{C_2} t : \forall x A$ .

(5) Si  $\Gamma \vdash_{C_2} t : \forall x A$ , alors, pour tout terme  $u$ ,  $\Gamma \vdash_{C_2} t : A[u/x]$ .

(6) Si  $\Gamma \vdash_{C_2} t : A$ , et  $X$  ne figure pas dans  $\Gamma$ , alors  $\Gamma \vdash_{C_2} t : \forall X A$ .

(7) Si  $\Gamma \vdash_{C_2} t : \forall X A$ , alors, pour toute formule  $G$ ,  $\Gamma \vdash_{C_2} t : A[G/X]$ .

(8) Si  $\Gamma \vdash_{C_2} t : A[u/x]$ , et  $u = v$  est un cas particulier d'une équation, alors  $\Gamma \vdash_{C_2} t : A[v/x]$ .

THÉORÈME 1 : Si  $\Gamma \vdash_{C_2} t : \perp$ , et  $t \succ_c t'$ , alors  $\Gamma \vdash_{C_2} t' : \perp$ .

Preuve : Voir [1].  $\square$

THÉORÈME 2 : Si  $\Gamma \vdash_{C_2} t : \perp$ , alors  $t$  est  $C$ -résoluble.

Preuve : Voir [1].  $\square$

LEMME 1 : Si  $\Gamma \vdash_{C_2} t : A$ , alors, pour toute  $\perp$ -substitution  $S$ ,  $S(\Gamma) \vdash_{C_2} t : S(A)$ , et on utilise les mêmes règles de typage.

Preuve : Par récurrence sur la structure de la démonstration de  $\Gamma \vdash_{C_2} t : A$ .  $\square$

**THÉORÈME 3 :** *Si  $\Gamma \vdash_{C2} t : A$ , et  $A$  est une formule atomique, alors  $t$  est  $C$ -résoluble.*

*Preuve :* Si  $\Gamma \vdash_{C2} t : A = X(t_1, \dots, t_n)$ , alors d'après le lemme 1,  $\Gamma[\perp/X] \vdash_{C2} t : A[\perp/X] = \perp$ . Donc, d'après le théorème 2,  $t$  est  $C$ -résoluble.  $\square$

Il est clair que : si on enlève la règle (0), alors on obtient le système de typage  $\mathcal{AF}2$ , et que :  $\Gamma \vdash_{C2} t : A$  si et seulement si  $\Gamma, C : \forall X \{ \neg X \rightarrow X \} \vdash_{\mathcal{AF}2} t : A$ .

### 3. TYPAGE NORMAL DANS LE SYSTÈME $\mathcal{AF}2$

Dans le système de typage  $\mathcal{AF}2$  on note :

( $\forall I_1$ ) la règle d'introduction des variables du premier ordre.

( $\forall I_2$ ) la règle d'introduction des variables du second ordre.

( $\forall E_1$ ) la règle d'élimination des variables du premier ordre.

( $\forall E_2$ ) la règle d'élimination des variables du second ordre.

(Eq) la règle concernant les équations.

Dans un typage, une  $\forall$ -coupure est la succession dans l'ordre de deux règles ( $\forall I_j$ ) et ( $\forall E_j$ )  $j = 1, 2$ .

Un typage est dit *normal* s'il ne contient pas de  $\forall$ -coupures.

**THÉORÈME 4 :** *Tout typage peut être supposé normal.*

*Preuve :* Voir [3] ou [4].  $\square$

**THÉORÈME 5 :** *Si dans un typage on est passé de  $\Gamma \vdash_{\mathcal{AF}2} t : A$  à  $\Gamma \vdash_{\mathcal{AF}2} t : B$ , alors on peut supposer qu'on a commencé par des applications de ( $\forall E_j$ )  $j = 1, 2$ , ensuite de (Eq), et enfin de ( $\forall I_j$ )  $j = 1, 2$ .*

*Preuve :* Voir [3] ou [4].  $\square$

On définit sur les types de  $\mathcal{AF}2$  une relation binaire  $<$  de la manière suivante :

–  $\forall x A < A[u/x]$ , si  $u$  est un terme;

–  $\forall X A < A[F/X]$ , si  $F$  est une formule.

Soit  $\leq$  la clôture réflexive et transitive de  $<$ .



Soit  $E$  un système d'équations. On définit sur les types de  $\mathcal{AF}2$  une relation binaire  $\sim'_E$  de la manière suivante :

$A \sim'_E B$  si  $A = C[u/x]$ ,  $B = C[v/x]$ , et  $u = v$  est un cas particulier d'une équation de  $E$ .

Soit  $\sim_E$  de la clôture réflexive et transitive de  $\sim'_E$ .

On écrit  $a \sim_E a'$ , si  $a = a'$  est une conséquence du système d'équations  $E$ .

Pour simplifier, on écrit, dans la suite,  $\sim'$  et  $\sim$  sans préciser le système d'équations utilisé.

**COROLLAIRE 1 :** 1) Soit  $A$  une formule atomique. Si  $\Gamma \vdash_{\mathcal{AF}2} t : A$ , alors  $t$  ne commence pas par  $\lambda$ . Autrement dit, si  $\Gamma \vdash_{\mathcal{AF}2} \lambda xt : B$ , alors  $B$  contient au moins une flèche.

2) Si  $\Gamma, x : A \vdash_{\mathcal{AF}2} (x) u_1 \dots u_n : B$ , alors :

$n = 0$ ,  $A \leq C$ ,  $C \sim C'$ ,  $B = \forall \xi_1 \dots \xi_m C'$ , et  $\xi_1, \dots, \xi_m$  ne figurent pas dans  $\Gamma$  et  $A$ , ou

$n > 0$ ,  $A \leq C_1 \rightarrow B_1$ ,  $B'_i \leq C_{i+1} \rightarrow B_{i+1}$   $1 \leq i \leq n-1$ ,  $B'_n \leq B_{n+1}$ ,  $B = \forall \xi_1 \dots \xi_m B'_{n+1}$  où  $B_i \sim B'_i$   $1 \leq i \leq n+1$ ,  $\Gamma, x : \vdash_{\mathcal{AF}2} u_i : C_i$   $1 \leq i \leq n$ , et  $\xi_1, \dots, \xi_m$  ne figurent pas dans  $\Gamma$  et  $A$ .

*Preuve :* On utilise les théorèmes 4 et 5.  $\square$

#### 4. TYPAGE AVEC DES OBSTACLES

Soit  $O$  un symbole particulier de relation unaire. Une formule  $O(a)$  est appelée *obstacle*.

Soit  $A$  une formule. On dit que  $A$  se termine par  $O$  si et seulement si  $A$  est obtenu par les règles suivantes :

- $O(a)$  se termine par  $O$ ;
- Si  $A$  se termine par  $O$ , alors  $B \rightarrow A$  se termine par  $O$  pour tout type  $B$ ;
- Si  $A$  se termine par  $O$ , alors  $\forall \xi A$  se termine par  $O$  pour toute variable  $\xi$ .

Le système de typage  $\mathcal{C}2O$  est le système de typage  $\mathcal{C}2$  où on remplace les règles (2) et (7) par les règles :

(2') Si  $\Gamma, x : A \vdash_{\mathcal{C}2O} t : B$ , et  $B$  ne se termine pas par  $O$ , alors  $\Gamma \vdash_{\mathcal{C}2O} \lambda xt : A \rightarrow B$ .

(7')  $\Gamma \vdash_{\mathcal{C}2O} t : \forall X A$ , et  $G$  ne se termine pas par  $O$ , alors  $\Gamma \vdash_{\mathcal{C}2O} t : A[G/X]$ .

On définit sur les types de  $\mathcal{C}2O$  la relation binaire  $<_O$  de la manière suivante :

- $\forall x A <_O A[u/x]$ , si  $u$  est un terme;

–  $\forall X A <_O A [F/X]$ , si  $F$  est une formule qui ne se termine pas par  $O$ .  
 Soit  $\leq_O$  la clôture réflexive et transitive de  $<_O$ .

Il est clair que dans ce système de typage, on peut obtenir un résultat analogue au corollaire 1.

THÉORÈME 6 : Si  $\Gamma \vdash_{C2O} t : \perp$ , et si  $t \succ_c t'$ , alors  $\Gamma \vdash_{C2O} t' : \perp$ .

*Preuve* : Même preuve que celle du théorème 1.  $\square$

THÉORÈME 7 : Si  $\Gamma \vdash_{C2O} t : A$ , et si  $A$  est une forme atomique, alors  $t$  est  $C$ -résoluble.

*Preuve* : Ceci provient de fait que : chaque typage de  $C2O$  est un typage de  $C2$ .  $\square$

LEMME 2 : Si  $\Gamma \vdash_{C2O} t : O(a)$ , et si  $t \succ_c t'$ , alors  $t = t'$ .

*Preuve* : Supposons que  $t$  et  $t'$  sont distincts. Dans ce cas,  $t$  se réduit en au moins une étape en  $t'$ . Le  $\lambda C$ -terme  $t$  est donc réductible, on distingue deux cas :

– Si  $t = (\lambda x u) v v_1 \dots v_m$ , alors  $\Gamma \vdash_{C2O} \lambda x u : A$ ,  $A \leq_O F \rightarrow G$ ,  $G' \leq_O F_1 \rightarrow G_1$ ,  $G'_j \leq_O F_{j+1} \rightarrow G_{j+1}$   $1 \leq j \leq m - 1$ ,  $G_m \sim O(a)$ ,  $G_j \sim G'_j$   $1 \leq j \leq m - 1$ ,  $\Gamma \vdash_{C2O} v : F$ , et  $\Gamma \vdash_{C2O} v_j : F_j$   $1 \leq j \leq m$ .  $\Gamma \vdash_{C2O} \lambda x u : A$  et  $A \leq_O F \rightarrow G$ , donc  $\Gamma, x : F \vdash_{C2O} u : G$ , et  $G$  ne se termine pas par  $O$ , donc  $G_j$   $1 \leq j \leq m$  ne se termine pas par  $O$ , ce qui est contradictoire car  $G_m \sim O(a)$ .

– Si  $t = (C) v v_1 \dots v_m$ , alors il existe une formule  $A$  qui ne se termine pas par  $O$ , telle que :  $A' \leq_O F_1 \rightarrow G_1$ ,  $G'_j \leq_O F_{j+1} \rightarrow G_{j+1}$   $1 \leq j \leq m - 1$ ,  $G_m \sim O(a)$ ,  $A \sim A'$ ,  $G_j \sim G'_j$   $1 \leq j \leq m$ ,  $\Gamma \vdash_{C2O} v : \neg A$ , et  $\Gamma \vdash_{C2O} v_j : F_j$   $1 \leq j \leq m$ .  $A$  ne se termine pas par  $O$ , donc  $G_j$   $1 \leq j \leq m$  ne se termine pas par  $O$ , ce qui est contradictoire car  $G_m \sim O(a)$ .  $\square$

LEMME 3 : Si

$$\Gamma = y_1 : A_1, \dots, y_n : A_n, x_1 : O(a_1), \dots, x_m : O(a_m) \vdash_{C2O} t : O(a),$$

et si les  $A_i$   $1 \leq i \leq n$  ne se terminent pas par  $O$ , alors  $t$  est l'un des  $x_i$ , et  $a_i \sim a$   $1 \leq i \leq m$ .

*Preuve* : D'après le théorème 7, on a  $t \succ_c (f) t_1 \dots t_r$ , et, d'après le lemme 2,  $t = (f) t_1 \dots t_r$ . Donc  $\Gamma \vdash_{C2O} (f) t_1 \dots t_r : O(a)$ .

Si  $f = x_i$   $1 \leq i \leq m$ , alors  $r = 0$ ,  $t = x_j$ , et  $O(a_i) \sim O(a)$ , donc  $a_i \sim a$ .

Si  $f = y_j$   $1 \leq j \leq k$ , alors  $A_j \leq_O F_1 \rightarrow G_1$ ,  $G'_k \leq_O F_{k+1} \rightarrow G_{k+1}$   $1 \leq k \leq r - 1$ ,  $G_r \sim O(a)$ ,  $G_k \sim G'_k$   $1 \leq k \leq r$ , et  $\Gamma \vdash_{C_2O} t_k : F_k$   $1 \leq k \leq r$ . Comme  $A_j$  ne se termine pas par  $O$ , alors les  $G_k$   $1 \leq k \leq r$  ne se terminent pas par  $O$ , ce qui est contradictoire car  $C_r \sim O(a)$ .  $\square$

*Remarque :* Le lemme 3 signifie qu'on ne peut pas construire, à partir des variables déclarées par des obstacles, un  $\lambda C$ -terme typé par un obstacle.

## 5. ENTIERS CLASSIQUES

Le type des entiers est la formule  $N[x] = \forall X \{ \forall y (X(y) \rightarrow X(sy)), X(0) \rightarrow X(x) \}$  où  $X$  est une variable de second ordre unaire,  $0$  est une constante (pour le zéro), et  $s$  est un symbole de fonction unaire (pour le successeur).

Un système d'équations  $E$  est dit *adéquat* pour le type des entiers si et seulement si :

- $s(a) \not\sim 0$ ;
- Si  $s(a) \sim s(b)$ , alors  $a \sim b$ .

Dans la suite, on suppose que le système d'équations est adéquat pour le type des entiers.

Pour simplifier, notons  $n$  le terme  $s^n(0)$ .

**THÉORÈME 8 :** Si  $\Gamma \vdash_{\mathcal{AF}_2} \theta_n : N[n]$ , alors  $\theta_n \simeq \beta \underline{n}$  (ou  $\theta_n \simeq \beta \lambda x x$  si  $n = 1$ ).

*Preuve :* Voir [1].  $\square$

**COROLLAIRE 2 :** Si  $\Gamma \vdash_{\mathcal{AF}_2} \theta_n : N[n]$ , alors pour toutes variables distinctes  $g, a, x_0, \dots, x_n$  on a :  $(\theta_n) g a x_0 \succ (g) t_1 x_0$ ,  $(t_i) x_i \succ (g) t_{i+1} x_i$   $1 \leq i \leq n - 1$ , et  $(t_n) x_n \succ (a) x_n$ .

*Preuve :* Ceci résulte directement du théorème 8.  $\square$

Un *entier classique*  $\theta_n$  est un  $\lambda C$ -terme tel que  $\Gamma \vdash_{C_2} \theta_n : N[n]$   $n \geq 0$ . On dit aussi que  $\theta_n$  est un entier classique de *valeur*  $n$ .



donc

$$\Gamma_1 = g : \forall y \{ [O(y) \rightarrow \perp] \rightarrow [O(sy) \rightarrow \perp] \},$$

$$a : O(0) \rightarrow \perp, x_0 : O(n) \vdash_{C_2O} (\theta_n) gax_0 : \perp,$$

donc, d'après les théorèmes 6 et 7,  $(\theta_n) gax_0$  est  $C$ -résoluble, et trois cas sont à voir :

– Si  $(\theta_n) gax_0 \succ_c (x_0) t_1 \dots t_r$ , alors  $r = 0$ , et il existe un terme  $a$ , tel que  $O(a) \sim \perp$ . Ce qui est impossible.

– Si  $(\theta_n) gax_0 \succ_c (a) t_1 \dots t_r$ , alors  $r = 1$ , et  $\Gamma_1 \vdash_{C_2O} t_1 : O(0)$ . Donc, d'après le lemme 3,  $t_1 = x_0$ . Posons  $I(x_0) = 0$ .

– Si  $(\theta_n) gax_0 \succ_c (g) t_1 \dots t_r$ , alors  $r = 2$ ,  $\Gamma_1 \vdash_{C_2O} t_1 : O(a) \rightarrow \perp$ ,  $\Gamma_1 \vdash_{C_2O} t_2 : O(s(a'))$ , et  $a \sim a'$ . D'après le lemme 3, on a  $t_2 = x_0$ , et  $s(a') \sim n$ , donc  $a \sim n - 1$ . Donc  $(\theta_n) gax_1 \succ_c (g) t_1 x_0$ , et  $\Gamma_1 \vdash_{C_2} t_1 : O(n - 1) \rightarrow \perp$ . Posons  $I(x_0) = n$ .

Montrons que : si  $\Gamma_i = g : \forall y \{ [O(y) \rightarrow \perp] \rightarrow [O(sy) \rightarrow \perp] \}$ ,  $a : O(0) \rightarrow \perp$ ,  $x_0 : O(I(x_0))$ ,  $\dots$ ,  $x_i : O(I(x_i)) \vdash_{C_2O} (t_i) x_i : \perp$ , alors :  $[(t_i) x_i \succ_c (g) t_{i+1} x_{ri}]$ , et  $\Gamma_i \vdash_{C_2O} t_{i+1} : O(I(x_{ri}) - 1) \rightarrow \perp$  ou  $[(t_i) x_i \succ_c (a) x_{ri}]$ , et  $I(x_{ri}) = 0$ .

$\Gamma_i \vdash_{C_2} (t_i) x_i : \perp$ , donc, d'après les théorèmes 6 et 7,  $(t_i) x_i$  est  $C$ -résoluble, et trois cas sont à voir :

– Si  $(t_i) x_i \succ_c (x_j) u_1 \dots u_r$   $0 \leq j \leq i$  alors  $r = 0$ , et il existe un terme  $a$ , tel que  $O(a) \sim \perp$ . Ce qui est impossible.

– Si  $(t_i) x_i \succ_c (a) u_1 \dots u_r$ , alors  $r = 1$ , et  $\Gamma_i \vdash_{C_2O} u_1 : O(0)$ . Donc, d'après le lemme 3,  $u_1 = x_{ri}$ , et  $I(x_{ri}) = 0$ .

– Si  $(t_i) x_i \succ_c (g) u_1 \dots u_r$ , alors  $r = 2$ ,  $\Gamma_i \vdash_{C_2O} u_1 : O(a) \rightarrow \perp$ ,  $\Gamma_i \vdash_{C_2O} u_2 : O(s(a'))$ , et  $a \sim a'$ . Donc, d'après le lemme 3,  $u_2 = x_{ri}$ , et  $s(a') \sim I(x_{ri})$ , donc  $a \sim I(x_{ri}) - 1$ . Donc  $(t_i) x_i \succ_c (g) t_{i+1} x_{ri}$ , et  $\Gamma_i \vdash_{C_2O} t_{i+1} : O(I(x_{ri}) - 1) \rightarrow \perp$ .

La construction s'arrête toujours sur  $(a) x_{ri}$ . En effet, sinon le  $\lambda C$ -terme  $(((\theta_n) \lambda xx) \lambda xx) x_0$  devient non  $C$ -résoluble, ce qui est contradictoire, car  $x_0 : \perp \vdash_{C_2} (((\theta_n) \lambda xx) \lambda xx) x_0 : \perp$ . En effet  $\vdash_{C_2} \theta_n : N[n]$ , donc  $\vdash_{C_2O} \theta_n : \forall y \{ [\perp \rightarrow \perp] \rightarrow [\perp \rightarrow \perp] \}$ ,  $[\perp \rightarrow \perp]$ ,  $\perp \rightarrow \perp$ . Or  $\vdash_{C_2} \lambda xx : \perp \rightarrow \perp$ , et  $\vdash_{C_2} \lambda xx : \forall y \{ [\perp \rightarrow \perp] \rightarrow [\perp \rightarrow \perp] \}$ , donc  $x_0 : \perp \vdash_{C_2} (((\theta_n) \lambda xx) \lambda xx) x_0 : \perp$ .  $\square$

**Conséquence du théorème 9**

*Soit  $\theta$  un entier classique. Comment peut-on calculer la valeur de  $\theta$ ?*

*Première méthode*

D'après le théorème 9, il existe  $m \geq 0$ , des variables distinctes  $g, a, x_0, x_1, \dots, x_m$  telles que :

$$(\theta_n) g a x_0 \succ_c (g) t_1 x_{r_0}; \quad (t_i) x_i \succ_c (g) t_{i+1} x_{r_i} \quad 0 \leq i < m;$$

$$(t_m) x_m \succ_c (a) x_{r_m}.$$

Soit  $J : \{x_0, \dots, x_m\} \rightarrow N$ , telle que  $J(x_0) = 0$  et  $J(x_{i+1}) = I(x_{r_i}) + 1$  pour  $0 \leq i \leq m$ .

*Montrons que  $\theta$  est de valeur  $J(x_{r_m})$ .*

Soit  $n$  la valeur de  $\theta$ . D'après le théorème 9, il existe une application  $I : \{x_0, \dots, x_m\} \rightarrow N$ , telle que :  $I(x_0) = n, I(x_{r_m}) = 0$ , et  $I(x_{i+1}) = I(x_{r_i}) - 1$  pour  $0 \leq i < m$ .

Il est facile de vérifier que  $J(x_i) = n - I(x_i) \quad 0 \leq i \leq m$ . Donc, en particulier,  $J(x_{r_m}) = n$ .

*Deuxième méthode*

On peut également trouver la valeur de  $\theta$  en utilisant des  $\lambda$ -termes particuliers. Notons par  $N^*[x]$  la formule

$$\forall X \{ \forall y (\neg X(y) \rightarrow \neg X(sy)), \neg X(0) \rightarrow \neg X(x) \}.$$

J. L. Krivine a démontré, en utilisant le théorème 9, que si  $\vdash_{\mathcal{AF}2} T : \forall x \{ N^*[x] \rightarrow \neg N[x] \}$ , alors pour tout  $n \geq 0$ , il existe  $\tau_n \simeq \beta n$  (ou  $\tau_n \simeq \beta \lambda x x$  si  $n = 1$ ), tel que pour tout entier classique  $\theta$  de valeur  $n$ , il existe une substitution  $\sigma$  telle que  $(T)\theta \lambda x x \succ_c \sigma(\tau_n)$  (voir [2]).

On peut améliorer le théorème 9.

**THÉORÈME 10 :** *Si  $\theta_n$  est un entier classique de valeur  $n$ , alors il existe  $m \geq 0$ , des variables distinctes  $g, a, x_0, x_1, \dots, x_m$ , et une application  $I : \{x_0, \dots, x_m\} \rightarrow N$ , telles que pour toutes variables distinctes  $y_1, \dots, y_r$ ,*

on a :

$$\begin{aligned} & (\theta_n) gax_0 y_1 \dots y_r \succ_c (g) t_1 x_{r0} y_1 \dots y_r; \\ & (t_i) x_i y_1 \dots y_r \succ_c (g) t_{i+1} x_{ri} y_1 \dots y_r \quad 0 \leq i \leq m; \\ & (t_m) x_m y_1 \dots y_r \succ_c (a) x_{r_m} y_1 \dots y_r \end{aligned}$$

où  $I(x_0) = n$ ,  $I(x_{r_m}) = 0$ , et  $I(x_{i+1}) = I(x_{ri}) - 1$  pour  $0 \leq i \leq m$ .

*Preuve* : Soient  $O_1, \dots, O_r$  des nouveaux symboles de relation 0-aire. Continuons à noter  $C2O$  le système de typage  $C2$  où on remplace les règles (2) et (7) par les règles :

(2'') Si  $\Gamma, x : A \vdash_{C2O} t : B$ , et  $B$  ne se termine pas par  $O$  ou  $O_i$   $1 \leq i \leq r$ , alors  $\Gamma \vdash_{C2O} \lambda x t : A \rightarrow B$ .

(7'') Si  $\Gamma \vdash_{C2O} t : \forall X A$ , et  $G$  ne se termine pas par  $O$  ou  $O_i$   $1 \leq i \leq r$ , alors  $\Gamma \vdash_{C2O} t : A[G/X]$ .

Il est facile de démontrer que :

– Si  $\Gamma = y_1 : A_1, \dots, y_n : A_n, x_1 : O(a_1), \dots, x_m : O(a_m) \vdash_{C2O} t : O(a)$ , et si les  $A_i$   $1 \leq i \leq n$  ne se terminent pas par  $O$ , alors  $t$  et l'un des  $x_i$ , et  $a_i \sim a$   $1 \leq i \leq m$ .

– Si  $\Gamma = y_1 : A_1, \dots, y_n : A_n, x_1 : O_{i1}, \dots, x_m : O_{im} \vdash_{C2O} t : O_{ij}$ , et si les  $A_i$   $1 \leq i \leq n$  ne se terminent pas par  $O_i$   $1 \leq i \leq r$ , alors  $t$  est l'un des  $x_j$   $1 \leq j \leq m$ .

Si  $\vdash_{C2} \theta_n : N[n]$ , alors

$$\vdash_{C2O} \theta_n : \{\forall y \{[O(y), O_1, \dots, O_r \rightarrow \perp] \rightarrow [O(sy), O_1, \dots, O_r \rightarrow \perp]\}, [O(0), O_1, \dots, O_r \rightarrow \perp], O(n), O_1, \dots, O_r \rightarrow \perp\},$$

donc

$$\begin{aligned} g : \forall y \{[O(y), O_1, \dots, O_r \rightarrow \perp] \rightarrow [O(sy), O_1, \dots, O_r \rightarrow \perp]\}, \\ a : [O(0), O_1, \dots, O_r \rightarrow \perp], x_0 : O(n), \\ y_1 : O_1, \dots, y_r : O_r \vdash_{C2O} (\theta_n) gax_0 y_1 \dots y_r : \perp. \end{aligned}$$

On reprend la démonstration du théorème 9.  $\square$

Un entier classique  $\theta_n$  (de valeur  $n$ ) est dit *entier intuitionniste* (de valeur  $n$ ) si et seulement si pour toutes variables distinctes  $g, a, x_0, \dots, x_n$  on a :  $(\theta_n) gax_0 \succ (g) t_1 x_0$ ,  $(t_i) x_i \succ (g) t_{i+1} x_i$   $1 \leq i \leq n-1$ , et  $(t_n) x_n \succ (a) x_n$  (c'est-à-dire  $\theta_n$  se comporte comme l'entier de Church  $\underline{n}$ ).

Dans les parties 6 et 7, nous présentons des systèmes de typage qui permettent d'obtenir les entiers intuitionnistes.

**6. LOGIQUE CLASSIQUE FAIBLE**

On ajoute à notre langage un symbole particulier de relation unaire  $\perp \neq O$  (pour l'absurde). Pour toute formule  $A$ , et pour tout terme  $a$ , on note  $\neg_a A$  la formule  $A \rightarrow \perp(a)$ .

On décrit un système de  $\lambda$ -calcul typé appelé *arithmétique fonctionnelle classique faible du second ordre* (abrégé en  $\mathcal{CF}2$ ) dont les types sont les formules du langage. Étant donné un  $\lambda C$ -terme  $t$ , un type  $A$ , et un contexte  $\Gamma = x_1 : A_1, \dots, x_n : A_n$ , on définit au moyen des règles suivantes la notion «  $t$  est typable de type  $A$  dans le contexte  $\Gamma$  », et on écrit  $\Gamma \vdash_{\mathcal{CF}2} t : A$ .

- Les règles (1), ..., (8) du système  $\mathcal{AF}2$ .
- (0') Si  $\Gamma \vdash_{\mathcal{CF}2} t : \neg_n \neg_n A$  pour tout entier  $n$ , alors  $\Gamma \vdash_{\mathcal{CF}2} (C) t : A$ .  
La règle (0') remplace la règle (0) du système  $\mathcal{C}2$ .

**THÉORÈME 11 :**  $\Gamma \vdash_{\mathcal{CF}2} t : \perp(n)$ , et  $t \succ_c t'$ , alors  $\Gamma \vdash_{\mathcal{CF}2} t' : \perp(n)$ .

*Preuve :* Il suffit de prouver que : si  $\Gamma \vdash_{\mathcal{CF}2} (C) v v_1 \dots v_r : \perp(n)$ , alors  $\Gamma \vdash_{\mathcal{CF}2} (v) \lambda x (x) v_1 \dots v_r : \perp(n)$ . Si  $\Gamma \vdash_{\mathcal{CF}2} (C) v v_1 \dots v_r : \perp(n)$ , alors il existe une formule  $A$  telle que :  $A' \leq B_1 \rightarrow C_1, C'_i \leq B_{i+1} \rightarrow C_{i+1} \ 1 \leq i \leq r-1, C'_r \leq \perp(n), A \sim A', C_i \sim C'_i \ 1 \leq i \leq r, \Gamma \vdash_{\mathcal{CF}2} v : \neg_m \neg_m A$  pour tout entier  $m$ , et  $\Gamma \vdash_{\mathcal{CF}2} v_i : C_i \ 1 \leq i \leq r$ , donc  $\Gamma \vdash_{\mathcal{CF}2} v : \neg_n \neg_n A$ . On a  $\Gamma, x : A \vdash_{\mathcal{CF}2} (x) v_1 \dots v_r : \perp(n)$ , donc  $\Gamma \vdash_{\mathcal{CF}2} \lambda x (x) v_1 \dots v_r : \neg_n A$ , et  $\Gamma \vdash_{\mathcal{CF}2} (v) \lambda x (x) v_1 \dots v_r : \perp(n)$ .  $\square$

**THÉORÈME 12 :** Si  $\Gamma \vdash_{\mathcal{CF}2} t : A$ , et si  $A$  est une formule atomique, alors  $t$  est  $C$ -résoluble.

*Preuve :* Même preuve que celle du théorème 3.  $\square$

Le système de typage  $\mathcal{CF}2O$  est le système de typage  $\mathcal{CF}2$  où on remplace les règles (2) et (7) par les règles :

- (2') Si  $\Gamma, x : A \vdash_{\mathcal{CF}2O} t : B$ , et  $B$  ne se termine pas par  $O$ , alors  $\Gamma \vdash_{\mathcal{CF}2O} \lambda x t : A \rightarrow B$ .
- (7') Si  $\Gamma \vdash_{\mathcal{CF}2O} t : \forall X A$ , et  $G$  ne se termine pas par  $O$ , alors  $\Gamma \vdash_{\mathcal{CF}2O} t : A [G/X]$ .

On définit sur les types de  $\mathcal{CF}2$  la relation binaire  $<_O$  de la manière suivante :

- $\forall x A <_O A [u/x]$ , si  $u$  est un terme;
  - $\forall X A <_O A [F/X]$ , si  $F$  est une formule qui ne se termine pas par  $O$ .
- Soit  $\leq_O$  la clôture réflexive et transitive de  $<_O$ .



THÉORÈME 13 : Si  $\Gamma \vdash_{\mathcal{CF}2O} t : \perp(n)$ , et  $t \succ_c t'$ , alors  $\Gamma \vdash_{\mathcal{CF}2O} t' : \perp(n)$ .

*Preuve* : Même preuve que celle du théorème 11.  $\square$

THÉORÈME 14 : Si  $\Gamma \vdash_{\mathcal{CF}2O} t : A$ , et si  $A$  est une formule atomique, alors  $t$  est  $C$ -résoluble.

*Preuve* : Ceci provient de fait que : chaque typage de  $\mathcal{CF}2O$  est un typage de  $\mathcal{CF}2$ .  $\square$

LEMME 4 : Si  $\Gamma \vdash_{\mathcal{CF}2O} t : O(a)$ , et  $t \succ_c t'$ , alors  $t = t'$ .

*Preuve* : Supposons que  $t$  et  $t'$  sont distincts. Dans ce cas,  $t$  se réduit en au moins une étape en  $t'$ . Le  $\lambda C$ -terme  $t$  est donc réductible, on distingue deux cas :

– Si  $t = (\lambda xu) v v_1 \dots v_m$ , alors  $\Gamma \vdash_{\mathcal{CF}2O} \lambda xu : A$ ,  $A \leq_O F \rightarrow G$ ,  $G' \leq_O F_1 \rightarrow G_1$ ,  $G'_j \leq_O F_{j+1} \rightarrow G_{j+1}$   $1 \leq j \leq m-1$ ,  $G_m \sim O(a)$ ,  $G_j \sim G'_j$   $1 \leq j \leq m-1$ ,  $\Gamma \vdash_{\mathcal{CF}2O} v : F$ , et  $\Gamma \vdash_{\mathcal{CF}2O} v_j : F_j$   $1 \leq j \leq m$ .  $\Gamma \vdash_{\mathcal{CF}2O} \lambda xu : A$ , et  $A \leq_O F \rightarrow G$ , donc  $\Gamma, x : F \vdash_{\mathcal{CF}2O} u : G$ , et  $G$  ne se termine pas par  $O$ , donc  $G_j$   $1 \leq j \leq m$  ne se termine pas par  $O$ , ce qui est contradictoire car  $G_m \sim O(a)$ .

– Si  $t = (C) v v_1 \dots v_m$ , alors il existe une formule  $A$  qui ne se termine pas par  $O$ , telle que :  $A' \leq_O F_1 \rightarrow G_1$ ,  $G'_j \leq_O F_{j+1} \rightarrow G_{j+1}$   $1 \leq j \leq m-1$ ,  $G_m \sim O(a)$ ,  $A \sim A'$ ,  $G_j \sim G'_j$   $1 \leq j \leq m-1$ ,  $\Gamma \vdash_{\mathcal{C}2O} v : \neg_n \neg_n A$  pour tout entier  $n$ , et  $\Gamma \vdash_{\mathcal{CF}2O} v_j : F_j$   $1 \leq j \leq m$ .  $A$  ne se termine pas par  $O$ , donc  $G_j$   $1 \leq j \leq m$  ne se termine pas par  $O$ , ce qui est contradictoire car  $G_m \sim O(a)$ .  $\square$

LEMME 5 : Si

$$\Gamma = y_1 : A_1, \dots, y_n : A_n, x_1 : O(a_1), \dots, x_m : O(a_m) \vdash_{\mathcal{CF}2O} t : O(a),$$

et si les  $A_i$   $1 \leq i \leq n$  ne se terminent pas par  $O$ , alors  $t$  est l'un des  $x_i$ , et  $a_i \sim a$   $1 \leq i \leq m$ .

*Preuve* : D'après le théorème 14, on a  $t \succ_c (f) t_1 \dots t_r$ , donc, d'après le lemme 4,  $t = (f) t_1 \dots t_r$ . Donc  $\Gamma \vdash_{\mathcal{CF}2O} (f) t_1 \dots t_r : O(a)$ .

Si  $f = x_i$   $1 \leq i \leq m$ , alors  $r = 0$ ,  $t = x_i$ , et  $O(a_i) \sim O(a)$ , donc  $a_i \sim a$ .

Si  $f = y_j$   $1 \leq j \leq n$ , alors  $A_j \leq_O F_1 \rightarrow G_1$ ,  $G'_k \leq_O F_{k+1} \rightarrow G_{k+1}$   $1 \leq k \leq r-1$ ,  $G_r \sim O(a)$ ,  $G_k \sim G'_k$   $1 \leq k \leq r-1$ , et  $\Gamma \vdash_{\mathcal{CF}2O} t_k : F_k$   $1 \leq k \leq r$ . Comme  $A_j$  ne se termine pas par  $O$ ,

alors les  $G_k$   $1 \leq k \leq r$  ne se terminent pas par  $O$ , ce qui est contradictoire car  $C_r \sim O(a)$ .  $\square$

**THÉORÈME 15 :** *Si  $\vdash_{\mathcal{CF}_2} \theta_n : N[n]$ , alors  $\theta_n$  est un entier intuitionniste.*

*Preuve :* Montrons que pour toutes variables distinctes  $g, y, x_1, \dots, x_n$ , on a  $(\theta_n)gyx_0 \succ_c (g)t_1x_0$ ,  $(t_i)x_i \succ_c (g)t_{i+1}x_i$   $1 \leq i \leq n-1$ , et  $(t_n)x_n \succ_c (y)x_n$ .

Si  $\vdash_{\mathcal{CF}_2} \theta_n : N[n]$ , alors

$$\vdash_{\mathcal{CF}_2O} \theta_n : \{ \forall y \{ [O(y) \rightarrow \perp(y)] \rightarrow [O(sy) \rightarrow \perp(sy)] \}, \\ [O(0) \rightarrow \perp(0)], O(n) \rightarrow \perp(n) \},$$

donc

$$\Gamma_1 = g : \forall y \{ [O(y) \rightarrow \perp(y)] \rightarrow [O(sy) \rightarrow \perp(sy)] \}, \\ y : O(0) \rightarrow \perp(0), x_0 : O(n) \vdash_{\mathcal{CF}_2O} (\theta_n)gyx_0 : \perp(n),$$

donc, d'après les théorèmes 13 et 14,  $(\theta_n)gyx_0$  est  $C$ -résoluble, et trois cas sont à voir :

- Si  $(\theta_n)gyx_0 \succ_c (x_0)t_1 \dots t_r$ , alors  $r = 0$ , et il existe  $a, b$  deux termes, tels que  $O(a) = \perp(b)$ . Ce qui est impossible.

- Si  $(\theta_n)gyx_0 \succ_c (y)t_1 \dots t_r$ , alors  $r = 1$ , et  $0 \sim n$ . Donc  $n = 0$ , et  $\Gamma_1 \vdash_{\mathcal{CF}_2O} t_1 : O(0)$ , donc, d'après le lemme 5,  $t_1 = x_0$ .

- Si  $(\theta_n)gyx_0 \succ_c (g)t_1 \dots t_r$ , alors  $r = 2$ ,  $\Gamma \vdash_{\mathcal{CF}_2O} t_1 : O(a) \rightarrow \perp(a)$ ,  $\Gamma_1 \vdash_{\mathcal{CF}_2O} t_2 : O(s(a'))$ ,  $a \sim a'$ , et  $s(a) \sim n$ . Donc  $a \sim n-1$ , et d'après le lemme 5,  $t_2 = x_0$ . Donc  $(\theta_n)gyx_1 \succ_c (g)t_1x_0$ , et  $\Gamma_1 \vdash_{\mathcal{CF}_2} t_1 : O(n-1) \rightarrow \perp(n-1)$ .

Montrons que : si  $\Gamma_i = g : \forall y \{ [O(y) \rightarrow \perp(y)] \rightarrow [O(sy) \rightarrow \perp(sy)] \}$ ,  $y : O(0) \rightarrow \perp(0)$ ,  $x_0 : O(n)$ ,  $\dots$ ,  $x_i : O(n-i) \vdash_{\mathcal{CF}_2O} (t_i)x_i : \perp(n-i)$ , alors : [si  $n = i$ ,  $(t_i)x_i \succ_c (y)x_i$ ] et [si  $n \neq i$ ,  $(t_i)x_i \succ_c (g)t_{i+1}x_i$ , et  $\Gamma_i \vdash_{\mathcal{CF}_2O} t_{i+1} : O(n-i-1) \rightarrow \perp(n-i-1)$ ].

$\Gamma_i \vdash_{\mathcal{CF}_2O} (t_i)x_i : \perp(n-i)$ , donc, d'après les théorèmes 13, 14,  $(t_i)x_i$  est  $C$ -résoluble, et trois cas sont à voir :

- Si  $(t_i)x_i \succ_c (x_j)u_1 \dots u_r$   $0 \leq j \leq i$ , alors  $r = 0$ , et il existe  $a, b$  deux termes, tels que  $O(a) = \perp(b)$ . Ce qui est impossible.

- Si  $(t_i)x_i \succ_c (y)u_1 \dots u_r$ , alors  $r = 1$ , et  $0 \sim n-i$ . Donc  $n = i$ , et  $\Gamma_i \vdash_{\mathcal{CF}_2O} u_1 : O(0)$ , donc, d'après le lemme 5,  $u_1 = x_i$ , et  $(t_i)x_i \succ_c (y)x_i$ .

- Si  $(t_i)x_i \succ_c (g)u_1 \dots u_r$ , alors  $r = 2$ ,  $\Gamma_i \vdash_{\mathcal{CF}_2O} u_1 : O(a) \rightarrow \perp(a)$ ,  $\Gamma_i \vdash_{\mathcal{CF}_2O} u_2 : O(s(a'))$ ,  $a \sim a'$ , et  $s(a) \sim n-i$ . Donc  $a \sim n-i-1$ , et

d'après le lemme 5,  $u_2 = x_i$ . Donc  $(t_i) x_i \succ_c (g) t_{i+1} x_i$ , et  $\Gamma_i \vdash_{CF_2O} t_{i+1} : O(n - i - 1) \rightarrow \perp(n - i - 1)$ .

Donc de proche en proche, on construit la suite  $(t_i)_{0 \leq i \leq n-1}$ .  $\square$

## 7. APPLICATIONS

Nous présentons dans cette partie deux restrictions du système  $C2$  qui permettent d'obtenir les entiers intuitionnistes.

### 7.1. Le système $CR2$

On suppose maintenant que le langage contient un ensemble dénombrable de symboles de relation 0-aire  $\perp_0, \perp_1, \dots, \perp_n, \dots$

Notons  $\neg_i A$  la formule  $A \rightarrow \perp_i$  ;  $i \geq 0$ .

En remplaçant la règle (0) du système  $C2$  par les règles :

$$(i, j) \quad \Gamma \vdash_{C2} C : \forall X \{ \neg_i \neg_j X \rightarrow X \} \quad i, j \geq 0$$

on obtient un système de typage équivalent à  $C2$ .

Nous continuons à noter  $C2$  ce nouveau système de typage.

Le système  $CR2$  (pour *arithmétique fonctionnelle classique restreint du second ordre*) est le système  $C2$  où on remplace les règles  $(i, j)$  du système  $C2$  par les règles :

$(i, j)'$  Si  $\Gamma \vdash_{CR2} t : \neg_i \neg_j A$  ;  $i, j \geq 0$ , et  $\perp_i, \perp_j$  ne figurent ni dans  $A$ , ni dans les formules de  $\Gamma$ , alors  $\Gamma \vdash_{CR2}(C) t : A$ .

La signification intuitive des règles  $(i, j)'$  est la suivante :

Si  $\Gamma \vdash_{CR2} t : \neg \neg A$ , et si les absurdes de la double négation de  $\neg \neg A$  ne figurent pas dans  $A$ , et ne proviennent pas des hypothèses, alors  $\Gamma \vdash_{CR2}(C) t : A$ .

Une  $\perp$ -substitution spéciale est une application  $S : \mathcal{V}'_1 \cup \mathcal{V}'_2 \rightarrow \mathcal{T} \cup \mathcal{F}$  vérifiant :

- $\mathcal{V}'_i$  est un sous ensemble fini de  $\mathcal{V}_i$  ;  $i = 1$  ou  $2$ ;
- $S(\mathcal{V}'_1) \subset \mathcal{T}$  et  $S(\mathcal{V}'_2) \subset \mathcal{F}$ ;
- $S(\perp_i) = \perp(n_i)$  où  $n_i$  est un entier.

Elle possède une extension canonique (qui est notée aussi  $S$ )  $S : \mathcal{T} \cup \mathcal{F} \rightarrow \mathcal{T} \cup \mathcal{F}$ , définie par la manière usuelle.

LEMME 6 : Si  $\Gamma \vdash_{CR2} t : A$ , alors pour toute  $\perp$ -substitution spéciale  $S$ , on a  $S(\Gamma) \vdash_{CF2} t : S(A)$ .

*Preuve* : Par récurrence sur le nombre  $n$  des règles  $(i, j)'$  utilisées dans le typage  $\Gamma \vdash_{CR2} t : A$ .

Si  $n = 0$ , c'est évident.

Sinon, considérons la dernière fois où on a utilisé une des règles  $(i, j)'$ .

On a  $\Gamma \vdash_{CR2} u : \neg_i \neg_j B$ ,  $i, j \geq 0$ ,  $\perp_i, \perp_j$  ne figurent ni dans  $B$ , ni dans les formules de  $\Gamma$ , et  $\Gamma \vdash_{CR2}(C) u : B$ . Par hypothèse de récurrence, on a, pour toute  $\perp$ -substitution spéciale  $S$ ,  $S(\Gamma) \vdash_{CF2} u : S(\neg_i \neg_j B)$ . Pour chaque  $\perp$ -substitution spéciale, et pour chaque  $k \geq 0$ , on note  $S_k$  une  $\perp$ -substitution spéciale vérifiant :  $S_k(\perp_i) = S_k(\perp_j) = \perp(k)$ , et  $S_k(B) = S(B)$  si  $B$  ne contient pas  $\perp_i, \perp_j$ . Donc, pour toute  $\perp$ -substitution spéciale  $S$ , et pour tout  $k \geq 0$ ,  $S_k(\Gamma) \vdash_{CF2} u : S_k(\neg_i \neg_j B)$ , donc pour toute  $\perp$ -substitution spéciale  $S$ , et pour tout  $k \geq 0$ , on a  $S(\Gamma) \vdash_{CF2} u : \neg_k \neg_k S(B)$ , donc  $S(\Gamma) \vdash_{CF2}(C) t : S(B)$ . Donc pour toute  $\perp$ -substitution spéciale  $S$ , on a  $S(\Gamma) \vdash_{CF2} u : S(A)$ .  $\square$

THÉORÈME 16 : Si  $\vdash_{CR2} \theta_n : N[n]$ , alors  $\theta_n$  est un entier intuitionniste.

*Preuve* : Montrons que pour toutes variables distinctes  $g, y, x_1, \dots, x_n$ , on a  $(\theta_n)gyx_0 \succ_c (g)t_1x_0$ ,  $(t_i)x_i \succ_c (g)t_{i+1}x_i$   $1 \leq i \leq n - 1$ , et  $(t_n)x_n \succ_c (y)x_n$ .

Si  $\vdash_{CR2} \theta_n : N[n]$ , alors, d'après le lemme 6,  $\vdash_{CF2} \theta_n : N[n]$ , donc d'après le théorème 15, pour toutes variables distinctes  $g, y, x_1, \dots, x_n$ , on a  $(\theta_n)gyx_0 \succ_c (g)t_1x_0$ ,  $(t_i)x_i \succ_c (g)t_{i+1}x_i$   $1 \leq i \leq n - 1$ , et  $(t_n)x_n \succ_c (y)x_n$ .  $\square$

Continuons à noter  $\mathcal{AF2}$  et  $\mathcal{CR2}$  les systèmes logiques qui correspondent aux systèmes de typage.

THÉORÈME 17 : Si  $A_1, \dots, A_n \vdash_{CR2} A$ , alors  $A_1, \dots, A_n \vdash_{\mathcal{AF2}} A$ .

*Preuve* : Par récurrence sur le nombre  $n$  des règles  $(i, j)'$  utilisées dans la démonstration  $A_1, \dots, A_n \vdash_{CR2} A$ .

Si  $n = 0$ , c'est évident.

Sinon, considérons la dernière fois où on a utilisé une des règles  $(i, j)'$ .

On a  $\Gamma \vdash_{CR2} \neg_i \neg_j B$ ,  $i, j \geq 1$ ,  $\Gamma \vdash_{CR2} t : B$ ,  $\perp_i, \perp_j$  ne figurent ni dans  $B$ , ni dans les formules de  $\Gamma$ , et  $\Gamma \vdash_{CR2} B$ . Par hypothèse de récurrence on a  $\Gamma \vdash_{\mathcal{AF2}} \neg_i \neg_j B$ , donc  $\Gamma = \Gamma[B/\perp_i, B/\perp_j] \vdash_{\mathcal{AF2}} \{\neg_i \neg_j B\}$

$[B/\perp_i, B/\perp_j] = (B \rightarrow B) \rightarrow B$ , donc  $\Gamma \vdash_{\mathcal{AF}2} B$ , et on continue la démonstration.  $\square$

## 7.2. Le système $\mathcal{AF}2+$

Ajoutons une constante  $\mathcal{A}$  au  $\lambda$ -calcul pur, et considérons les deux règles de réduction :

1)  $(\lambda x u) t t_1 \dots t_n \rightarrow (u [t/x]) t_1 \dots t_n$  pour tous  $u, t, t_1, \dots, t_n$  des termes du  $\lambda \mathcal{A}$ -calcul ;

2)  $(\mathcal{A}) t t_1 \dots t_n \rightarrow t$  pour tous  $t, t_1, \dots, t_n$  des termes du  $\lambda \mathcal{A}$ -calcul.

Pour tous  $t, t'$  deux  $\lambda \mathcal{A}$ -termes, on note  $t \succ_{\mathcal{A}} t'$ , si  $t'$  est obtenu à partir de  $t$  après l'application d'un nombre fini des règles précédentes.

Le système  $\mathcal{AF}2+$  est le système  $\mathcal{AF}2$  où on ajoute la règle de typage suivante :

( $\mathcal{A}$ ) Si  $\Gamma \vdash_{\mathcal{AF}2+} t : \perp$ , alors, pour tout type  $A$ ,  $\Gamma \vdash_{\mathcal{AF}2+} (\mathcal{A}) t : A$ .

Signalons que la règle ( $\mathcal{A}$ ) n'a rien à voir avec la logique classique, puisque la formule  $\forall X \{ \perp \rightarrow X \}$  (où  $X$  est une variable de relation 0-aire) est valide en logique intuitionniste.

Le système  $\mathcal{AF}2+$  peut être vu comme un sous système du système  $\mathcal{C}2$ .

En effet, posons  $\mathcal{A} = \lambda x (C) \lambda y x$ .

*Fonctionnement de  $\mathcal{A}$  :*

$$(\mathcal{A}) t t_1 \dots t_n \succ_c (C) \lambda y t t_1 \dots t_n \succ_c (\lambda y t) \lambda x (x) t_1 \dots t_n \succ_c t.$$

*Typage de  $\mathcal{A}$  :*

$$x : \perp \vdash_{\mathcal{C}2} \lambda y x : \neg X \Rightarrow x : \perp \vdash_{\mathcal{C}2} (C) \lambda y x : X, \Rightarrow \vdash_{\mathcal{C}2} \mathcal{A} : \forall X \{ \perp \rightarrow X \}.$$

Donc si  $\Gamma \vdash_{\mathcal{C}2} t : \perp$ , alors, pour tout type  $A$ ,  $\Gamma \vdash_{\mathcal{C}2} (\mathcal{A}) t : A$ .

De plus, on a la caractérisation suivante.

**THÉORÈME 18 :** Si  $\vdash_{\mathcal{C}2} T : \forall X \{ \perp \rightarrow X \}$ , alors pour tout  $n \geq 0$ , et pour tous  $\lambda \mathcal{C}$ -termes  $t, t_1, \dots, t_n$ , on a  $(T) t t_1 \dots t_n \succ_c t$ .

*Preuve :* Voir [5].  $\square$

Dans le système  $\mathcal{CF}2$ , notons  $\perp$  le type  $\forall x \perp(x)$ .

On a :

$$x : \perp \vdash_{\mathcal{CF}2} x : \perp(n) \Rightarrow x : \perp \vdash_{\mathcal{CF}2} \lambda y x : \neg_n \neg_n X \text{ pour tout } n \geq 0 \Rightarrow x : \perp \vdash_{\mathcal{CF}2} (C) \lambda y x : X \Rightarrow \vdash_{\mathcal{CF}2} \mathcal{A} : \forall X \{ \perp \rightarrow X \}.$$

Donc si  $\Gamma \vdash_{\mathcal{CF}_2} t : \perp$ , alors, pour tout type  $A$ ,  $\Gamma \vdash_{\mathcal{CF}_2} (\mathcal{A})t : A$ .

LEMME 7 : Si  $\Gamma \vdash_{\mathcal{AF}_2} t : A$ , alors  $\Gamma [\perp/\perp] \vdash_{\mathcal{CF}_2} t [\mathcal{A}/\mathcal{A}] : A [\perp/\perp]$ .

*Preuve* : Par récurrence sur le nombre  $n$  des règles  $(\mathcal{A})$  utilisées dans le typage  $\Gamma \vdash_{\mathcal{AF}_2} t : A$ .

Si  $n = 0$ , c'est évident.

Sinon, considérons la dernière fois où on a utilisé  $(\mathcal{A})$ .

On a  $\Gamma \vdash_{\mathcal{AF}_2} u : \perp$ , et  $\Gamma \vdash_{\mathcal{CR}_2} (\mathcal{A})u : A$  pour tout type  $A$ . Par hypothèse de récurrence, on a  $\Gamma [\perp/\perp] \vdash_{\mathcal{CF}_2} u [\mathcal{A}/\mathcal{A}] : \perp$ , donc  $\Gamma [\perp/\perp] \vdash_{\mathcal{CR}_2} \{(\mathcal{A})u\} [\mathcal{A}/\mathcal{A}] : A [\perp/\perp]$  pour tout type  $A$ , et on continue le typage.  $\square$

THÉORÈME 19 : Si  $\Gamma \vdash_{\mathcal{AF}_2} \theta_n : N[n]$ , alors  $\theta_n$  est un entier intuitionniste.

*Preuve* : Montrons que pour toutes variables distinctes  $g, y, x_1, \dots, x_n$ , on a  $(\theta_n)gyx_0 \succ_c (g)t_1x_0$ ,  $(t_i)x_i \succ_c (g)t_{i+1}x_i$   $1 \leq i \leq n-1$ , et  $(t_n)x_n \succ_c (y)x_n$ .

Si  $\Gamma \vdash_{\mathcal{AF}_2} \theta_n : N[n]$ , alors, d'après le lemme 7,  $\Gamma \vdash_{\mathcal{CF}_2} \theta_n [\mathcal{A}/\mathcal{A}] : N[n]$ , donc d'après le théorème 15, pour toutes variables distinctes  $g, y, x_1, \dots, x_n$ , on a :  $(\theta_n [\mathcal{A}/\mathcal{A}])gyx_0 \succ_c (g)t_1 [\mathcal{A}/\mathcal{A}]x_0$ ,  $(t_i [\mathcal{A}/\mathcal{A}])x_i \succ_c (g)t_{i+1} [\mathcal{A}/\mathcal{A}]x_i$   $1 \leq i \leq n-1$ , et  $(t_n [\mathcal{A}/\mathcal{A}])x_n \succ_c (y)x_n$ . Donc pour toutes variables distinctes  $g, y, x_1, \dots, x_n$ , on a  $(\theta_n)gyx_0 \succ_{\mathcal{A}} (g)t_1x_0$ ,  $(t_i)x_i \succ_{\mathcal{A}} (g)t_{i+1}x_i$   $1 \leq i \leq n-1$ , et  $(t_n)x_n \succ_{\mathcal{A}} (y)x_n$ .  $\square$

#### REMERCIEMENT

Je tiens à remercier M. J. L. Krivine pour ses précieux conseils qui m'ont permis de réaliser ce travail.

#### RÉFÉRENCES

1. J. L. KRIVINE, *Lambda calcul, types et modèle*, Masson, Paris, 1990.
2. J. L. KRIVINE, Classical logic, storage operators and second order lambda-calculus, à paraître dans *Annals of pure and applied logic*, 1994.
3. K. NOUR, *Opérateurs de mise en mémoire en lambda-calcul pur et typé*, Thèse de doctorat, Université de Savoie, 1993.
4. K. NOUR, Opérateurs de mise en mémoire et types  $\forall$ -positifs, soumis à *Informatique Théorique et Applications*, 1993.
5. K. NOUR, Quelques résultats sur  $\lambda C$ -calcul, *C. R. Acad. Sci.*, Paris, t. 320, série I, 1995, p. 259-262.
6. M. PARIGOT,  $\lambda\mu$ -calculus: an algorithmic interpretation of classical natural deduction. *LNCS*, 1992, 624, p. 190-201.