INFORMATIQUE THÉORIQUE ET APPLICATIONS

KARIM NOUR

Opérateurs de mise en mémoire et types ∀-positifs

Informatique théorique et applications, tome 30, n° 3 (1996), p. 261-293

<http://www.numdam.org/item?id=ITA_1996__30_3_261_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 30, n° 3, 1996, pp. 261-293)

OPÉRATEURS DE MISE EN MÉMOIRE ET TYPES ∀-POSITIFS (*)

par Karim NOUR ([1])

Communiqué par J. GALLIER

Résumé. – *En 1990, J.-L. Krivine a introduit la notion d'opérateurs de mise en mémoire pour simuler « l'appel par valeur » dans le cadre de « l'appel par nom ». J.-L. Krivine a démontré qu'en utilisant la traduction de Gödel de la logique classique en logique intuitionniste, on peut trouver un type très simple du système $\mathcal{AF}2$ pour les opérateurs de mise en mémoire. Dans cet article nous étudions les types ∀-positifs (le quantificateur universel du second ordre apparaît positivement dans les types), et les transformations de Gödel (une généralisation de la traduction de Gödel classique) du système $\mathcal{AF}2$. Nous généralisons, en utilisant des méthodes purement syntaxiques, le théorème de J.-L. Krivine sur ces types et pour ces transformations.*

Mots clés : *Opérateur de mise en mémoire, traduction de Gödel, logique classique, logique intuitionniste, type ∀-positif, transformation de Gödel.*

Abstract. – *In 1990, J.-L. Krivine introduced the notion of storage operator to simulate the "call by value" in a context of a "call by name". J.-L. Krivine has shown that using Gödel translation from classical to intuitionistic logic, we can find a very simple type of $\mathcal{AF}2$ type system for the storage operators. In this paper we study the ∀-positive types (the universel second order quantifier appears positively in the types), and the Gödel transformations (a generalization of classical Gödel translation) of the system $\mathcal{AF}2$. We generalize, by using a pure syntaxic methods, the J.-L. Krivine theorem about these types and for these transformations.*

Keywords: *Storage operator, Gödel translation, Classical logic, Intuitionistic logic, ∀-positive type, Gödel transformation.*

INTRODUCTION

Dans cet article, nous considèrons le λ-calcul avec la réduction gauche (itération de la réduction de tête notée $\succ$). Cette stratégie a beaucoup d'avantages: le plus convaincant est qu'elle aboutit toujours si le λ-terme auquel nous l'appliquons est normalisable. De plus, elle semble être plus économique, puisque nous n'évaluons un λ-terme que s'il faut le faire pour la suite du calcul. Mais l'inconvénient majeur de cette stratégie est que l'argument d'une fonction est recalculé à chaque utilisation. Ce qui explique le fait que cette évaluation n'est pas vraiment utilisée. Nous allons tenter de remédier à ce défaut.

(*) Article reçu en janvier 1994; accepté en novembre 1995.

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

Soient F un λ-terme (une fonction), $\underline{N}$ l'ensemble des entiers de Church, et $\underline{n} \in \underline{N}$. Pour réduire, par la réduction gauche, le λ-terme $(F)\,\theta_n\,(\theta_n \simeq_\beta \underline{n})$, il y a un risque de réduire θ_n plusieurs fois (autant de fois que F l'utilise). On aimerait trouver un moyen de transformer $(F)\,\theta_n$ en $(F)\,\underline{n}$. Nous voulons également que cette transformation ne dépende que de θ_n (et pas de F). Autrement dit, nous cherchons des termes clos T du λ-calcul qui vérifient les deux propriétés suivantes :

– Pour tout F, pour tout $\underline{n} \in \underline{N}$, et pour tout $\theta_n \simeq_\beta \underline{n}$, $(T)\,\theta_n\,F \succ (F)\,\underline{n}$;

– Le temps calcul de $(T)\,\theta_n\,F$ ne dépend que de θ_n.

D'où une première définition : un terme clos T du λ-calcul est dit **opérateur de mise en mémoire** pour $\underline{N}$ si et seulement si pour tout $\underline{n} \in \underline{N}$, et pour tout $\theta_n \simeq_\beta \underline{n}$, $(T)\,\theta_n\,f \succ (f)\,\underline{n}$ (f est une nouvelle variable).

Il est clair qu'un opérateur de mise en mémoire satisfait les propriétés voulues. En effet :

– Puisque $(T)\,\theta_n\,f \succ (f)\,\underline{n}$, alors la variable f n'est pas arrivée en tête au cours de la réduction, donc nous pouvons remplacer f par n'importe quel λ-terme.

– Le temps calcul de $(T)\,\theta_n f$ ne dépend que de θ_n.

Nous avons démontré (*voir* [11]) qu'il n'est pas possible d'obtenir la forme normale de θ_n. On est donc amené à changer la définition : un terme clos T du λ-calcul est dit **opérateur de mise en mémoire** pour $\underline{N}$ si et seulement si pour tout $\underline{n} \in \underline{N}$, il existe un λ-terme clos $\tau_n \simeq_\beta \underline{n}$ (par exemple $\tau_n = (\underline{s})^n\,\underline{0}$ où $\underline{s}$ est un λ-terme pour le successeur) tel que pour tout $\theta_n \simeq_\beta \underline{n}$, $(T)\,\theta_n\,f \succ (f)\,\tau_n$ (f est une nouvelle variable).

Si nous prenons $T_1 = \lambda\,n\,((n)\,\lambda\,f\,(f)\,\underline{0})\,\lambda\,x\lambda\,y\,(x)\,\lambda\,z\,(y)\,(\underline{s})\,z$ et $T_2 = \lambda\,n\,\lambda\,f\,(((n)\,f)\,\lambda\,x\,\lambda\,y\,(x)\,(\underline{s})\,y)\,\underline{0}$, alors, il est facile de vérifier que pour tout $\theta_n \simeq_\beta \underline{n}$, $(T_1)\,\theta_n\,f \succ (f)\,(\underline{s})^n\,\underline{0}$ et $(T_2)\,\theta_n\,f \succ (f)\,(\underline{s})^n\,\underline{0}$. Donc T_1 et T_2 sont des opérateurs de mise en mémoire pour $\underline{N}$.

Soit $N\,[x] = \forall\,X\,\{X\,(0), \forall\,y\,[X\,(y) \to X\,(sy)] \to X\,(x)\}$ le type entier du système $\mathcal{AF}2$ (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.

Si nous essayons de typer un opérateur de mise en mémoire T pour $\underline{N}$, nous trouvons naturellement le type $\forall x\,\{N\,[x] \to [(N\,[x] \to O) \to O]\}$. Vérifions simplement que ce type convient. Si nous avons $\vdash_{\mathcal{AF}2} \tau_n : N\,[s^n\,(0)]$, et si nous déclarons la variable f de type $N\,[s^n\,(0)] \to O$, alors

$f : N[s^n(0)] \to O \vdash_{\mathcal{AF}2} (f)\,\tau_n : O$ (O est un symbole de relation 0-aire qui représente un type quelconque), et donc nous devons typer $(T)\,\theta_n\,f$ de type O. Si nous avons $\vdash_{\mathcal{AF}2} \theta_n : N[s^n(0)]$, alors il est naturel de considérer que des opérateurs de mise en mémoire comme T doivent être typés de type $\forall x\,\{N[x] \to [(N[x] \to O) \to O]\}$.

Il est facile de vérifier que

$$\vdash_{\mathcal{AF}2} T_1,\ T_2 : \forall x\,\{N[x] \to [(N[x] \to O) \to O]\}.$$

Le type $\forall x\,\{N[x] \to [(N[x] \to O) \to O]\}$ ne caractérise pas les opérateurs de mise en mémoire. En effet, si nous prenons $t = \lambda n\,\lambda f\,(f)\,n$, alors :

– Nous avons $n : N[x],\ f : N[x] \to O \vdash_{\mathcal{AF}2} (f)\,n : O$, donc

$$\vdash_{\mathcal{AF}2} t : \forall x\,\{N[x] \to [(N[x] \to O) \to O]\}.$$

– Pour tout $\theta_n \simeq_\beta \underline{n}$, $(t)\,\theta_n\,f \succ (f)\,\theta_n$, donc t n'est pas un opérateur de mise en mémoire pour $\underline{N}$.

Ceci résulte du fait que le type $\forall x\,\{N[x] \to [(N[x] \to O) \to O]\}$ ne tient pas compte de l'indépendance de τ_n par rapport au θ_n. Pour résoudre ce problème, nous allons interdir l'utilisation du premier $N[x]$ de $\forall x\,\{N[x] \to [(N[x] \to O) \to O]\}$ ainsi que ses sous types dans la démonstration du second. Nous remplaçons donc le premier $N[x]$ par un nouveau type $N^*[x]$ qui satisfait les trois propriétés suivantes :

– $\vdash_{\mathcal{AF}2} \underline{n} : N^*[s^n(0)]$ (par exemple, nous prenons

$$N^*[x] = \forall X\,\{F(X,\,0),\ \forall y\,[F(X,\,y) \to F(X,\,sy)],\ \to F(X,\,x)\})\,;$$

– Si $\nu : N^*[s^n(0)],\ x_i : \forall y\,[F(G_i,\,y) \to F(G_i,\,sy)]$,

$$y_j : F(H_j,\,a_j) \to F(H_j,\,s(a_j)),$$

$z_k : F(K_k,\,b_k) \vdash_{\mathcal{AF}2} t : N[s^n(0)]$, alors $\vdash_{\mathcal{AF}2} t' : N[s^n(0)]$, où t' est la forme normale de t ;

– Il existe un λ-terme T tel que

$$\vdash_{\mathcal{AF}2} T : \forall x\,\{N^*[x] \to [(N[x] \to O) \to O]\}.$$

Une solution simple de la deuxième propriété revient à imposer à $F(X,\,a)$ de se terminer par une nouvelle constante. En effet, comme $N[x]$ ne contient pas cette constante, nous ne pouvons pas utiliser les variables : ν, x_i, y_j, z_k

dans le typage de t'. Pour vérifier les deux autres propriétés, nous suggèrons la proposition suivante :

$$N^*[x] = \forall X \{X(0) \to O, \forall y [X(y) \to O, X(sy) \to O], X(x) \to O\}.$$

Il est facile de vérifier que

$$\vdash_{\mathcal{AF}2} T_1, T_2 : \forall x \{N^*[x] \to [(N[x] \to O) \to O]\}.$$

Pour chaque formule F de $\mathcal{AF}2$, nous notons $\neg F$ la formule $F \to O$, et F^* la formule obtenue en remplaçant chaque formule atomique A de F par $\neg A$ (F^* est dite la tranduction de Gödel de F).

J. L. Krivine a montré que le type $\forall x \{N^*[x] \to \neg\neg N[x]\}$ caractérise les opérateurs de mise en mémoire. Mais le λ-terme τ_n obtenu peut contenir des variables qui sont substituées par des λ-termes dépendant de θ_n. Ceci ne pose pas de problème. En effet, τ_n est β-équivalent à $\underline{n}$, donc la réduction gauche de $\tau_n [u_1/x_1, \ldots, u_n/x_n]$ est un calcul équivalent à la réduction gauche de τ_n. Le calcul est donc indépendant des λ-termes $u_1, \ldots, u_n$ qui ne sont jamais évalués.

Nous sommes amené de nouveau à modifier la définition : un terme clos T du λ-calcul est dit **opérateur de mise en mémoire** pour $\underline{N}$ si et seulement si pour tout $\underline{n} \in \underline{N}$, il existe un λ-terme $\tau_n \simeq_\beta \underline{n}$ tel que pour tout $\theta_n \simeq_\beta \underline{n}$, il existe une substitution S telle que $(T)\theta_n f \succ (f) S(\tau_n)$ (f est une nouvelle variable).

Dans le cas où τ_n est clos, nous disons que T est un **opérateur propre de mise en mémoire**. Ces opérateurs sont étudiés dans [10].

Le théorème de J. L Krivine s'énonce de la manière suivante :

si $\vdash_{\mathcal{AF}2} T : \forall x \{N^*[x] \to \neg\neg N[x]\}$, *alors* T *est un opérateur de mise en mémoire pour* $\underline{N}$.

Dans cet article, nous étudions les types D, et les transformations * pour lesquels nous avons le résultat suivant (une généralisation naturelle du théorème précédent) :

si $\vdash_{\mathcal{AF}2} T : D^* \to \neg\neg D$ alors pour tout $\vdash_{\mathcal{AF}2} t : D$, il existe deux λ-termes τ et τ' vérifiant $\tau \simeq_\beta \tau'$ et $\vdash_{\mathcal{AF}2} \tau' : D$ tels que pour tout $\theta_t \simeq_\beta t$, il existe une substitution S telle que $(T)\theta_n f \succ (f) S(\tau_n)$ (f est une nouvelle variable).

Nous prouvons ([1]) que, pour avoir ce résultat, il suffit de supposer que :

– Le quantificateur universel du second ordre apparait positivement dans D (nous disons que D est un type $\forall$-positif ([2])).

– La transformation * vérifie les trois propriétés suivantes :

• Si A est un type atomique, alors A^* se termine par O ;

• $(A \to B)^* = A^* \to B^*$;

• $(\forall \xi\, A)^* = \forall \xi\, A^*$.

Le but de cet article est de démontrer le théorème énoncé ci-dessus. Il est organisé de la manière suivante :

• Dans les parties 1 et 2, nous présentons les outils de base du λ-calcul pur et typé.

• Dans la partie 3, nous donnons quelques propriétés du système de typage $\mathcal{AF}2$. Nous définissons ensuite un système de typage équivalent au système $\mathcal{AF}2$ (ils typent les mêmes λ-termes de mêmes types). Ce nouveau système va nous aider pour faire quelques preuves.

• Dans la partie 4, nous introduisons les types $\forall$-positifs et les types $\forall$-négatifs ainsi que leurs propriétés. Nous prouvons, en particulier, que si un λ-terme normal clos est typable d'un type $\forall$-positif, alors nous pouvons supposer qu'au cours de ce typage, toutes les variables sont déclarées de types $\forall$-négatifs, tous ses sous termes essentiels sont typable de types $\forall$-positifs, et ceci en utilisant uniquement une version faible de la spécification des variables du second ordre.

• Dans la partie 5, nous étudions les types qui se terminent par un symbole particulier de relation 0-aire. En utilisant ces types, nous introduisons les transformations de Gödel (une généralisation de la traduction de Gödel), ainsi que leurs propriétés. Nous démontrons, en particulier, que si * est une transformation de Gödel, et si un λ-terme normal clos t est typable d'un type $\forall$-positif A, alors t est typable de type A^*.

• Dans la partie 6, nous exposons la définition des opérateurs de mise en mémoire et en donnons des exemples. Nous présentons ensuite le lambda-calcul dirigé (lambda-calcul avec des contextes dirigés). Ce lambda-calcul permet d'obtenir une définition équivalente pour les opérateurs de mise en mémoire plus pratique pour leur étude. Nous donnons ici quelques résultats

([1]) J.-L. Krivine a prouvé indépendamment le même résultat (*voir* [7]).

([2]) Ces types ont été étudiés par plusieurs auteurs (en particulier R. Labib-Sami), et ont des propriétés remarquables (*voir* [8]).

(sans démonstration) sur le lambda-calcul dirigé. Pour plus de détails, vous pouver consulter [9] et [13].

• Dans la partie 7, nous présentons la preuve du théorème général. Pour éclaircir les idées de la preuve, nous la présentons aussi sur un cas particulier.

N.B. Les définitions que nous allons présenter possèdent quelques inintéressantes difficultés techniques. Nous n'allons pas définir formellement l'ensemble de sous-termes d'un λ-terme. Nous supposons, par exemple, que – comme nous travaillons modulo α-équivalence – les λ-termes $(x)\,x$, $(y)\,y$, $(z)\,z$, ... sont des sous-termes du λ-terme $\lambda\,x\,(x)\,x$. Un λ-terme peut avoir plusieurs sous-termes égaux ; nous supposons que nous pouvons distinguer entre eux. Ces problèmes peuvent être résolu en utilisant la notation de de-Bruijn. Nous n'allons pas le faire ici.

1. NOTIONS DE BASE DU λ-CALCUL PUR

Les notations sont standard (*voir* [1] et [5]). Nous supposons une familiarité avec les notions de base du λ-calcul pur. Nous donnons, ci-dessous, quelques notations particulières, et des théorèmes essentiels du λ-calcul qui seront utilisés dans la suite.

– On note Λ l'ensemble des termes du λ-calcul, dites aussi λ-**termes**.

– Étant donnés des λ-termes t, u, u_1, ..., $u_n \in \Lambda$, l'application de t à u sera notée $(t)\,u$ ou tu, et $(\ldots((t)\,u_1)\ldots)\,u_n$ sera noté $(t)\,u_1\ldots u_n$ ou $tu_1\ldots u_n$.

– Si t est normalisable, on note t^β sa forme normale.

– Pour tout λ-terme normal t, on associe un ensemble de λ-termes $STE\,(t)$, appelé ensemble des sous-termes essentiels de t, de la manière suivante :

Si $t = \lambda\,x_1\ldots\lambda\,x_n\,(x)\,t_1\ldots t_m$, alors $STE\,(t) = \{t\} \cup \bigcup_{1\leq i\leq m} STE\,(t_i)$.

– Pour tout λ-terme t, on note $Fv\,(t)$ l'ensemble de ses variables libres.

– La notation $t \succ t'$ signifie que t' est obtenu à partir de t par quelques réductions de tête, et on note $n\,(t,\,t')$ le nombre de pas effectué pour passer de t à t'.

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

THÉORÈME 1.1 : *Si $t \succ t'$, alors pour tout u_1, ..., $u_n \in \Lambda$:*

1) *Il existe $v \in \Lambda$ tel que $(t)\,u_1\ldots u_n \succ v$, $(t')\,u_1\ldots u_n \succ v$, et $n\,((t)\,u_1\ldots u_n,\,v) = n\,((t')\,u_1\ldots u_n,\,v) + n\,(t,\,t')$.*

2) $t[u_1/x_1, \ldots, u_n/x_n] \succ t'[u_1/x_1, \ldots, u_n/x_n]$, *et* $n(t[u_1/x_1, \ldots, u_n/x_n], t'[u_1/x_1, \ldots, u_n/x_n]) = n(t, t')$.

Preuve : Voir [6]. ♦

Remarque : Le théorème 1.1 montre que pour effectuer la réduction de tête de $(t)\, u_1 \ldots u_n$ (resp. $t[u_1/x_1, \ldots, u_n/x_n]$) il est équivalent (même résultat, et même nombre de pas) d'effectuer un certain nombre de pas dans la réduction de tête de t, puis de faire la réduction de tête de $(t')\, u_1 \ldots u_n$ (resp. $t'[u_1/x_1, \ldots, u_n/x_n]$).

COROLLAIRE 1.2 : *Soient* $t, u_1, \ldots, u_n, v_1, \ldots, v_m \in \Lambda$. *Si* $(t[u_1/x_1, \ldots, u_n/x_n])\, v_1 \ldots v_m$ *est résoluble, alors* t *est résoluble.*

Preuve : Il suffit d'appliquer le théorème 1.1. ♦

COROLLAIRE 1.3 : *Soient* $t \simeq_\beta u$ *tel que* u *ne contient pas les variables* $x_1, \ldots, x_n$. *Alors la réduction gauche de* $t[u_1/x_1, \ldots, u_n/x_n]$ *est un calcul équivalent à la réduction gauche de* t. *Ce calcul est donc indépendant des* λ*-termes* $u_1, \ldots, u_n$ *qui ne sont jamais évalués.*

Preuve : Voir [6]. ♦

2. NOTIONS DE BASE DU λ-CALCUL TYPÉ

2.1 Le système de typage $\mathcal{F}$ ([3])

Les types du système $\mathcal{F}$ sont les formules construites à l'aide d'un ensemble dénombrable de variables propositionnelles, et les connecteurs $\rightarrow$ et $\forall$. Étant donnés un terme t du λ-calcul, un type A, et un contexte $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, on définit au moyen des règles suivantes la notion « t est typable de type A dans le contexte Γ » et on écrit $\Gamma \vdash_{\mathcal{F}} t : A$.

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

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

(3) Si $\Gamma \vdash_{\mathcal{F}} u : B \rightarrow C$, et $\Gamma \vdash_{\mathcal{F}} v : B$, alors $\Gamma \vdash_{\mathcal{F}} (u)\, v : C$.

(4) Si $\Gamma \vdash_{\mathcal{F}} t : A$ et X n'est pas libre dans Γ, alors $\Gamma \vdash_{\mathcal{F}} t : \forall X A$.

(5) Si $\Gamma \vdash_{\mathcal{F}} t : \forall X A$, alors, pour tout type G, $\Gamma \vdash_{\mathcal{F}} t : A[G/X]$.

([3]) Ce système est dû à J.-Y. Girard (*voir* [2]).

THÉORÈME 2.1 (théorème de conservation): *Si* $\Gamma \vdash {}_{\mathcal{F}}t : A$, *et* $t \to {}_{\beta}t'$, *alors* $\Gamma \vdash {}_{\mathcal{F}}t' : A$.

Preuve: Voir [2]. ◆

THÉORÈME 2.2 (théorème de normalisation forte): *Si* $\Gamma \vdash {}_{\mathcal{F}}t : A$, *alors* t *est fortement normalisable.*

Preuve: Voir [2]. ◆

2.2 Le système de typage $\mathcal{AF}2$ ([4])

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

– Les seuls symboles logiques $\to$ et $\forall$;

– Un ensemble dénombrable de variables d'individu: $x, y, \ldots$ (appelées aussi variables du premier ordre);

– Un ensemble dénombrable de variables de relation n-aire $(n = 0, 1, \ldots)$: $X, Y, \ldots$ (appelées aussi variables du second ordre);

– Des symboles de fonction n-aire $(n = 0, 1, \ldots)$ sur les individus;

– Des symboles de relation n-aire $(n = 0, 1, \ldots)$ sur les individus.

Un symbole de fonction 0-aire est appelé symbole de constante. Une variable de relation 0-aire est aussi appelée variable propositionnelle.

• Les **termes** sont construits de la façon suivante:

– Chaque variable d'individu, et chaque symbole de constante est un terme;

– Si f est un symbole de fonction n-aire et $t_1, \ldots, t_n$ sont des termes, alors $f(t_1, \ldots, t_n)$ est un terme.

Les **formules** sont construites de la façon suivante:

– Si A est un symbole ou variable de relation n-aire et $t_1, \ldots, t_n$ sont des termes, alors $A(t_1, \ldots, t_n)$ est une formule dite **formule atomique**;

– Si A, B sont des formules, alors $A \to B$ est une formule;

– Si A est une formule, alors $\forall xA$ et $\forall XA$ sont des formules, x (resp. X) étant une variable d'individu (resp. de relation).

On définit les notions de variables **libres** et **liées** de manière usuelle. Une formule est dite **close**, si elle ne contient pas de variables libres.

([4]) Ce système est dû à J.-L. Krivine (*voir* [5]).

• La formule $F_1 \to (F_2 \to (\ldots \to (F_n \to G)\ldots))$ est notée $F_1, F_2, \ldots, F_n \to G$. La négation d'une formule F est notée $F^{\perp}$.

• Si X est une variable de relation unaire, t et t' deux termes, alors la formule $\forall X\,[X\,t \to X\,t']$ 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 $t\,[u_1/x_1, \ldots, u_n/x_n] = t'\,[u_1/x_1, \ldots, u_n/x_n]$ ou $t'\,[u_1/x_1, \ldots, u_n/x_n] = t\,[u_1/x_1, \ldots, u_n/x_n]$, $u_1, \ldots, u_n$ étant des termes du langage. Dans la suite on note $\mathbf{E}$ un système d'équations fonctionnelles.

• On décrit un système de λ-calcul typé appelé **Arithmétique Fonctionnelle du second ordre** (abrégé en $\mathcal{AF}2$) dont les types sont les formules du langage. Dans l'écriture des λ-termes typés de ce système, nous emploierons les mêmes symboles pour les variables du λ-calcul, et les variables d'individu du langage. Étant donnés un terme t du λ-calcul, un type A, et un contexte $\Gamma = x_1 : A_1, \ldots, x_n : A_n$, on définit au moyen des règles suivantes la notion « t est typable de type A dans le contexte Γ » et on écrit $\Gamma \vdash_{\mathcal{AF}2} t : A$.

(1) $\Gamma \vdash_{\mathcal{AF}2} x_i : A_i\ 1 \leq i \leq n$.

(2) Si $\Gamma, x : B \vdash_{\mathcal{AF}2} t : C$, alors $\Gamma \vdash_{\mathcal{AF}2} \lambda\, x t : B \to C$.

(3) Si $\Gamma \vdash_{\mathcal{AF}2} u : B \to C$, et $\Gamma \vdash_{\mathcal{AF}2} v : B$, alors $\Gamma \vdash_{\mathcal{AF}_2} (u)\, v : C$.

(4) Si $\Gamma \vdash_{\mathcal{AF}2} t : A$, et x n'est pas libre dans Γ, alors $\Gamma \vdash_{\mathcal{AF}2} t : \forall\, x\, A$.

(5) Si $\Gamma \vdash_{\mathcal{AF}2} t : \forall\, x\, A$, alors, pour tout terme u, $\Gamma \vdash_{\mathcal{AF}2} t : A\,[u/x]$.

(6) Si $\Gamma \vdash_{\mathcal{AF}2} t : A$, et X n'est pas libre dans Γ, alors $\Gamma \vdash_{\mathcal{AF}2} t : \forall\, X A$.

(7) Si $\Gamma \vdash_{\mathcal{AF}2} t : \forall\, X A$, alors, pour toute formule G, $\Gamma \vdash_{\mathcal{AF}2} t : A\,[G/X\,(x_1, \ldots, x_n)]$ (*).

(8) Si $\Gamma \vdash_{\mathcal{AF}2} t : A\,[u/x]$, alors $\Gamma \vdash_{\mathcal{AF}2} t : A\,[v/x]$, $u = v$ étant un cas particulier d'une équation de $\mathbf{E}$.

(*) $A\,[G/X\,(x_1, \ldots, x_n)]$ est obtenue en replaçant dans A chaque formule atomique $X\,(t_1, \ldots, t_n)$ par $G\,[t_1/x_1, \ldots, t_n/x_n]$.

THÉORÈME 2.3 (théorème de conservation): *Si* $\Gamma \vdash_{\mathcal{AF}2} t : A$, *et* $t \to_{\beta} t'$, *alors* $\Gamma \vdash_{\mathcal{AF}2} t' : A$.

Preuve: Voir [5]. ♦

A chaque variable X de relation, on associe de façon biunivoque une variable X° du système $\mathcal{F}$. A chaque formule A du système $\mathcal{AF}2$, on associe la formule A° du système $\mathcal{F}$ obtenue en supprimant les variables du

premier ordre. Si $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ est un contexte du système $\mathcal{AF}2$, alors on note Γ° le contexte $x_1 : A_1^\circ, \ldots, x_n : A_n^\circ$ du système $\mathcal{F}$.

THÉORÈME 2.4: *Si* $\Gamma \vdash_{\mathcal{AF}2} t : A$, *alors* $\Gamma^\circ \vdash_{\mathcal{F}} t : A^\circ$. *Donc* t *est typable dans* $\mathcal{AF}2$ *si et seulement si* t *est typable dans* $\mathcal{F}$.

Preuve: Par induction sur la longueur de la dérivation $\Gamma \vdash_{\mathcal{AF}2} t : A$. ♦

THÉORÈME 2.5 (théorème de normalisation forte): *Si* $\Gamma \vdash_{\mathcal{AF}2} t : A$, *alors* t *est fortement normalisable.*

Preuve: On utilise les théorèmes 2.2 et 2.4. ♦

Notations

– Soient A, G deux formules, X une variable de relation n-aire. On écrit, dans la suite $A[G/X]$ au lieu de $A[G/X(x_1, \ldots, x_n)]$.

– Pour simplifier, on utilise parfois les notations suivantes:

On note $\mathbf{G}$ la suite finie de termes et/ou de formules $G_1, \ldots, G_n$.

Soient $\mathbf{G} = G_1, \ldots, G_n$, $\mathbf{F} = F_1, \ldots, F_m$ deux suites finies de termes et/ou de formules, $\boldsymbol{\xi} = \xi_1, \ldots, \xi_n$ une suite finie de variables du premier et/ou du second ordre telle que: G_i est un terme si ξ_i est une variable du premier ordre, et G_i est une formule si ξ_i est une variable du second ordre, et F, A deux termes et/ou deux formules.

On note $F[\mathbf{G}/\boldsymbol{\xi}]$, , le terme et/ou la formule $F[G_1/\xi_1, \ldots, G_n/\xi_n]$, et $\mathbf{F}[\mathbf{G}/\boldsymbol{\xi}]$ la suite $F_1[\mathbf{G}/\boldsymbol{\xi}], \ldots, F_m[\mathbf{G}/\boldsymbol{\xi}]$.

La suite $G_1, \ldots, G_n, F_1, \ldots, F_m$ est notée $\mathbf{G}, \mathbf{F}$ et

$$A[G_1/\xi_1, \ldots, G_n/\xi_n, F_1/\delta_m, \ldots, F_n/\delta_m]$$

est notée $A[\mathbf{G}/\boldsymbol{\xi}, \mathbf{F}/\boldsymbol{\delta}]$ etc.

– La formule $\forall \xi_1 \ldots \forall \xi_n F$ est notée $\forall \boldsymbol{\xi} F$. L'écriture « $\boldsymbol{\xi}$ n'est pas libre dans A » signifie que pour tout $1 \leq i \leq n$, ξ_i n'est pas libre dans A.

3. TYPAGE NORMAL

Définitions

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

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

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

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

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

$(\mathbf{Eq})$ la règle concernant les équations.

– Dans un typage, une ∀-**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 ∀-coupures.

– La hauteur d'un typage est le nombre de règles utilisées dans ce typage.

LEMME 3.1 :

1) *Si* $\Gamma \vdash {}_{\mathcal{AF}2}t : A$, *alors, pour toute suite de termes* $\mathbf{u}$, $\Gamma[\mathbf{u}/\mathbf{x}] \vdash {}_{\mathcal{AF}2}t : A[\mathbf{u}/\mathbf{x}]$, *et on utilise les mêmes règles de typage.*

2) *Si* $\Gamma \vdash {}_{\mathcal{AF}2}t : A$, *alors, pour toute suite de formules* $\mathbf{F}$, $\Gamma[\mathbf{F}/\mathbf{X}] \vdash {}_{\mathcal{AF}2}t : A[\mathbf{F}/\mathbf{X}]$, *et on utilise les mêmes règles de typage.*

Preuve : Voir [5]. ♦

COROLLAIRE 3.2 : *Tout typage peut être supposé normal.*

Preuve : Par induction sur le couple (n, m) où n est le nombre de ∀-coupures, et m est la hauteur minimale des ∀-coupures.

Si $n = 0$, alors le typage est normal.

Si $n \neq 0$, alors considérons une ∀-coupure de hauteur minimale :

$\Gamma \vdash {}_{\mathcal{AF}2}t : A$

$\Gamma \vdash {}_{\mathcal{AF}2}t : \forall \xi A$ où ξ n'est pas libre dans Γ

$\Gamma \vdash {}_{\mathcal{AF}2}t : A[G/\xi]$.

En utilisant le lemme 3.1, on élimine cette ∀-coupure pour obtenir $\Gamma[G/\xi] = \Gamma \vdash {}_{\mathcal{AF}2}t : A[G/\xi]$, et (n, m) diminue strictement. ♦

LEMME 3.3 :

1) *Dans un typage, on peut remplacer la succession de n fois* $(\forall I_j)$ *et de m fois* (Eq), *par la succession de m fois* (Eq) *et de n fois* $(\forall I_j)$ $j = 1, 2$.

2) *Dans un typage, on peut remplacer la succession de n fois* (Eq) *et de m fois* $(\forall I_2)$, *par la succession de m fois* $(\forall I_2)$ *et de n fois* (Eq).

3) *Dans un typage, on peut remplacer la succession de n fois (Eq) et de m fois* $(\forall E_j)$, *par la succession de m fois* $(\forall E_j)$ *et de n fois* (Eq) $j = 1, 2$.

Preuve :

1) Par induction sur n et m. Le cas de base est (par exemple pour $j = 1$)

$\Gamma \vdash_{\mathcal{AF}2t} : A[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : \forall x\, A[u/y]$ où x n'est pas libre dans Γ

$\Gamma \vdash_{\mathcal{AF}2t} : \forall x\, A[v/y]$ où $u = v$ est un cas particulier d'une équation de $\mathbf{E}$.

Donc

$\Gamma \vdash_{\mathcal{AF}2t} : A[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : A[v/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : \forall x\, A[v/y]$.

2) Par induction sur n et m. Le cas de base est

$\Gamma \vdash_{\mathcal{AF}2t} : A[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : A[v/y]$ où $u = v$ est un cas particulier d'une équation de $\mathbf{E}$

$\Gamma \vdash_{\mathcal{AF}2t} : \forall XA[v/y]$ où X n'est pas libre dans Γ.

Donc

$\Gamma \vdash_{\mathcal{AF}2t} : A[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : \forall XA[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : \forall XA[v/y]$.

3) Par induction sur n et m. Le cas de base est (par exemple pour $j = 1$)

$\Gamma \vdash_{\mathcal{AF}2t} : \forall x\, A[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : \forall x\, A[v/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : A[v/y][w/x]$ où $u = v$ est un cas particulier d'une équation de $\mathbf{E}$, et w est un terme.

Puisqu'on peut supposer que y n'est pas libre dans w, et x n'est pas libre dans u et v, on obtient :

$\Gamma \vdash_{\mathcal{AF}2t} : \forall x\, A[u/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : A[v/y][w/x] = A[w/x][v/y]$

$\Gamma \vdash_{\mathcal{AF}2t} : A[w/x][v/y] = A[v/y][w/x]$. ◆

Remarques

1) Dans un typage, on ne peut pas toujours remplacer la succession de (Eq) et de $(\forall I_1)$, par la succession de $(\forall I_1)$ et de (Eq). En effet, soit $\mathbf{E}$ l'équation $f(x) = 0$ où 0 est une constante, et f un symbole de fonction unaire. On a :

$x : X0 \vdash_{\mathcal{AF}2x} : X0$

$x : X0 \vdash_{\mathcal{AF}2x} : Xf(x)$

$x : X0 \vdash_{\mathcal{AF}2x} : \forall xXf(x)$

mais on ne peut pas renverser l'ordre d'application des règles de typage.

2) Dans un typage, on ne peut pas toujours remplacer la succession de $(\forall E_j)$ et de (Eq), par la succession de (Eq) et de $(\forall E_j)$ $j = 1, 2$. En effet, soit **E** l'équation $f(g(x)) = 0$ où 0 est une constante, et f et g sont deux symboles de fonction unaire. On a:

$$x : \forall x X f(x) \vdash_{\mathcal{AF}2} x : \forall x X f(x)$$

$$x : \forall x X f(x) \vdash_{\mathcal{AF}2} x : X f(g(x))$$

$$x : \forall x X f(x) \vdash_{\mathcal{AF}2} x : X0$$

et

$$x : \forall X X g(x) \vdash_{\mathcal{AF}2} x : \forall X X g(x)$$

$$x : \forall X X g(x) \vdash_{\mathcal{AF}2} x : X f(g(x))$$

$$x : \forall X X g(x) \vdash_{\mathcal{AF}2} x : X0$$

mais dans les deux typage, on ne peut pas renverser l'ordre d'application des règles de typage.

COROLLAIRE 3.4: *Si dans un typage normal 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: Par induction sur le nombre de règles utilisées pour passer de $\Gamma \vdash_{\mathcal{AF}2} t : A$ à $\Gamma \vdash_{\mathcal{AF}2} t : B$. On regarde la dernière règle utilisée, et on utilise le lemme 3.3. ◆

DÉFINITIONS: On définit sur les types de $\mathcal{AF}2$ les deux relations binaires $<$ et $\sim'$ de la manière suivante:

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

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

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

Soient $\leq$ et $\sim$ les clôtures réflexives et transitives respectives de $<$ et $\sim'$.

Remarques

1) La relation $\sim$ est une relation d'équivalence.

2) La relation $\leq$ n'est pas une relation d'ordre. En effet, elle n'est pas antisymétrique :

Si X est une variable de relation unaire, et a et b sont deux termes distincts du langage, alors $\forall XXa \leq \forall XXb \leq \forall XXa$.

LEMME 3.5 : *Si* $A \leq B$, *alors* $A = \forall \xi_1 \ldots \forall \xi_n C$, *et*

$$B = C[G_1/\xi_1] \ldots [G_n/\xi_n].$$

Preuve : On distingue deux cas :

– Si $A = \forall \xi_1 \ldots \forall \xi_r (F \to G)$, alors $A \leq B$ implique que

$$B = \forall \xi_{n+1} \ldots \forall \xi_r (F \to G)[G_1/\xi_1] \ldots [G_n/\xi_n]$$

et on pose $C = \forall \xi_{n+1} \ldots \forall \xi_r (F \to G)$.

– Si $A = \forall \xi_1 \ldots \forall \xi_r X(t_1, \ldots, t_k)$, alors on distingue deux sous cas :

– Si pour tout $1 \leq i \leq r$ on a,

$\xi_i \neq X$, alors $A \leq B$ implique que

$$B = \forall \xi_{n+1} \ldots \forall \xi_r X(t_1, \ldots, t_k)[G_1/\xi_1] \ldots [G_n/\xi_n],$$

et on pose $C = \forall \xi_1 \ldots \forall \xi_n X(t_1, \ldots, t_k)$.

– S'il existe $1 \leq i \leq r$ tel que $\xi_i = X$, alors posons $C = X(t_1, \ldots, t_k)$.

On a $B = C[\xi_1/\xi_1] \ldots [\xi_{i-1}/\xi_{i-1}][B/\xi_i][\xi_{i+1}/\xi_{i+1}] \ldots [\xi_r/\xi_r]$. ♦

D'après les corollaires 3.2, et 3.4, on obtient un système de typage équivalent au système $\mathcal{AF}2$ (ils typent les mêmes λ-termes de mêmes types), défini à partir des trois règles de typage :

$$\frac{x : A \quad A \leq B \quad B \sim B'}{x : \forall \xi B'} \tag{1}$$

$$\frac{\begin{array}{c}\overline{x : A} \\ \vdots \end{array}}{\begin{array}{c} u : B \quad A \sim A' \quad B \sim B' \\ \hline \lambda x u : \forall \xi (A' - B') \end{array}} \tag{2}$$

$$\frac{u : A \to B \quad v : A \quad B \leq C \quad C \sim C'}{(u)v : \forall \xi C'} \tag{3}$$

où ξ n'est pas libre dans les hypothèses des règles.

On note par $\mathcal{AF}2\mathcal{N}$ (pour $\mathcal{AF}2$ normal), ce nouveau système de typage. L'écriture $\Gamma \vdash_{\mathcal{AF}2\mathcal{N}} t : A$ signifie que dans le système de typage $\mathcal{AF}2\mathcal{N}$, et dans le contexte Γ, le λ-terme t est typable de type A. On a donc $\Gamma \vdash_{\mathcal{AF}2} t : A$ si et seulement si $\Gamma \vdash_{\mathcal{AF}2\mathcal{N}} t : A$.

Remarques

1) Si on restreint ce système de typage aux types du système $\mathcal{F}$, on obtient exactement le système de Giannini et Ronchi Della Rocca (*voir* [3]).

2) Il est clair qu'on peut supposer que, chaque fois qu'on effectue un $A \leq B$, on détruit tous les quantificateurs qui sont au début de A.

COROLLAIRE 3.6 :

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

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

$n = 0$, $A \leq C$, $C \sim C'$, $B = \forall \boldsymbol{\xi}\, C'$, et $\boldsymbol{\xi}$ n'est pas libre dans Γ et A,

ou

$A \leq C_1 \to B_1$, $B'_i \leq C_{i+1} \to B_{i+1}$ $1 \leq i \leq n-1$, $B'_n \leq B_{n+1}$, $B = \forall \boldsymbol{\xi}\, B'_{n+1}$ où $B_i \sim B'_i$ $1 \leq i \leq n+1$, $\Gamma,\ x : A \vdash_{\mathcal{AF}2} u_i : C_i$ $1 \leq i \leq n$, et $\boldsymbol{\xi}$ n'est pas libre dans Γ et A.

3) *Soit O une formule atomique. Si $\Gamma,\ x : A \to O \vdash_{\mathcal{AF}2} (x)\, u_1 \ldots u_n : C$, alors $n = 1$, et $\Gamma,\ x : A \to O \vdash_{\mathcal{AF}2} u_1 : A$.*

Preuve :

1) C'est une conséquence directe du système de typage $\mathcal{AF}2\mathcal{N}$.

2) Par induction sur n.

Si $n = 0$, ou $n = 1$, alors c'est évident.

Si $n > 1$, alors $\Gamma,\ x : A \vdash_{\mathcal{AF}2} (x)\, u_1 \ldots u_{n-1} : C'_n \to B'_n$, $\Gamma, x : A \vdash_{\mathcal{AF}2} u_n : C'_n$, $B'_n \leq B_{n+1}$, $B = \forall \boldsymbol{\xi}\, B'_{n+1}$ où $B_{n+1} \sim B'_{n+1}$, et $\boldsymbol{\xi}$ n'est pas libre dans Γ et A. Par hypothèse d'induction, on a $A \leq C_1 \to B_1$, $B'_i \leq C_{i+1} \to B_{i+1}$ $1 \leq i \leq n-2$, $B'_{n-1} \leq B_n$, et $C'_n \to B'_n = B'_n$ où $B_i \sim B'_i$ $1 \leq i \leq n$, et $\Gamma,\ x : A \vdash_{\mathcal{AF}2} u_i : C_i$ $1 \leq i \leq n-1$. Donc $B_n = C_n \to B_n$ où $B_n \sim B'_n$, $C_n \sim C'_n$, et $\Gamma,\ x : A \vdash_{\mathcal{AF}2} u_n : C_n$.

3) Si $n = 0$, ou $n > 1$, alors, d'après 2), O est non atomique, ce qui est contradictoire. Donc $n = 1$, et, d'après 2), $\Gamma, x : A \to O \vdash_{\mathcal{AF}2} u_1 : A$. ◆

4. TYPES ∀-POSITIFS

DÉFINITION : Soient $\mathcal{A}$ l'ensemble des types atomiques du système du typage $\mathcal{AF}2$, $\mathcal{V}_1$ l'ensemble des variables du premier ordre, et $\mathcal{V}_2$ l'ensemble des variables du second ordre. On définit deux ensembles de types $\mathcal{T}^+$ (ensemble des **types ∀-positifs**), et $\mathcal{T}^-$ (ensemble des **types ∀-négatifs**) de la manière suivante :

- Si $A \in \mathcal{A}$, alors $A \in \mathcal{T}^+$, et $A \in \mathcal{T}^-$;
- Si $T^+ \in \mathcal{T}^+$, et $T^- \in \mathcal{T}^-$, alors $T^- \to T^+ \in \mathcal{T}^+$, et $T^+ \to T^- \in \mathcal{T}^-$;
- Si $x \in \mathcal{V}_1$, et $T^+ \in \mathcal{T}^+$, alors $\forall\, x\, T^+ \in \mathcal{T}^+$;
- Si $X \in \mathcal{V}_2$, et $T^+ \in \mathcal{T}^+$, alors $\forall\, X T^+ \in \mathcal{T}^+$;
- Si $x \in \mathcal{V}_1$, et $T^- \in \mathcal{T}^-$, alors $\forall\, x T^- \in \mathcal{T}^-$;
- Si $X \in \mathcal{V}_2$, $T^- \in \mathcal{T}^-$, et X n'est pas libre dans T^-, alors $\forall X T^- \in \mathcal{T}^-$.

Remarques

1) $\mathcal{T}^+$ (resp. $\mathcal{T}^-$) est l'ensemble des types où tous les quantificateurs « actifs » du second ordre sont en positions positifs (resp. négatifs).

2) $T^+ \in \mathcal{T}^+$ si et seulement si $T^+ = \forall\, \boldsymbol{\xi}\, T'$ où T' est un type atomique, ou $T' = T' \to T'^+$, $T'^+ \in \mathcal{T}^+$, $T'^- \in \mathcal{T}^-$.

3) $T^- \in \mathcal{T}^-$ si et seulement si $T^- = \forall\, \boldsymbol{\xi}\, T'$ où T' est un type atomique, ou $T' = T' \to T'^-$, $T'^- \in \mathcal{T}^-$, $T'^+ \in \mathcal{T}^+$, et $\boldsymbol{\xi} = \xi_1, \ldots, \xi_n$, alors ξ_i est, soit une variable du premier ordre, soit une variable du second ordre qui n'est pas libre dans T' $1 \leq i \leq n$.

LEMME 4.1 :

1) Si $T \in \mathcal{T}^+$ *(resp.* $T \in \mathcal{T}^-$*), et* $T \sim T'$, *alors* $T' \in \mathcal{T}^+$ *(resp.* $T' \in \mathcal{T}^-$*).*

2) *Si* $T^- \in \mathcal{T}^-$, *et* $T^- \leq C \to D$, *alors* $C \in \mathcal{T}^+$, *et* $D \in \mathcal{T}^-$.

Preuve : Évident. ♦

DÉFINITION : On note $\mathcal{AF}20$, le système de typage $\mathcal{AF}2$ où on remplace la règle de spécification des variables du second ordre par la règle :

$(\forall\, \mathbf{E}_0)$ Si $\Gamma \vdash_{\mathcal{AF}20} t : \forall\, X A$, et Y est une variable de même arité que X, alors $\Gamma \vdash_{\mathcal{AF}20} t : A[Y/X]$.

Remarque : Il est clair que si $\Gamma \vdash_{\mathcal{AF}20} t : T^-$, $T^- \in \mathcal{T}^-$, et $T^- \leq T$, alors $\Gamma \vdash_{\mathcal{AF}20} t : T$.

THÉORÈME 4.2 : *Soient* $A_1, \ldots, A_n \in \mathcal{T}^-$, $A \in \mathcal{T}^+$, t *un* λ*-terme normal, et* $\Gamma = x_1 : A_1, \ldots, x_n : A_n$. *Si* $\Gamma \vdash_{\mathcal{AF}2} t : A$, *alors* $\Gamma \vdash_{\mathcal{AF}20} t : A$, *et au cours de ce typage, chaque variable est déclarée d'un type* $\forall$*-négatif, et chaque* $u \in STE(t)$ *est typable d'un type* $\forall$*-positif.*

Preuve : Par induction sur t.

– Si $t = x_i$ $1 \leq i \leq n$, alors $A_i \leq C$, $C \sim C'$, $A = \forall\, \boldsymbol{\xi}\, C'$, et $\boldsymbol{\xi}$ n'est pas libre dans Γ. Comme $A_i \in \mathcal{T}^-$, alors $\Gamma \vdash_{\mathcal{AF}20} t : A$.

– Si $t = \lambda x u$, alors Γ, $x : B \vdash_{\mathcal{AF}20} u : C$, $B \sim B'$, $C \sim C'$, $A = \forall\, \boldsymbol{\xi}\, (B' \to C')$, et $\boldsymbol{\xi}$ n'est pas libre dans Γ. Comme $A \in \mathcal{T}^+$, , alors $B' \in \mathcal{T}^-$, et $C' \in \mathcal{T}^+$, donc, d'après le lemme 4.1, $B \in \mathcal{T}^-$, et $C \in \mathcal{T}^+$. Par hypothèse d'induction on a Γ, $x : B \vdash_{\mathcal{AF}20} u : C$, donc $\Gamma \vdash_{\mathcal{AF}20} t : A$.

– Si $t = (x_j)\, u_1 \ldots u_m$ $1 \leq j \leq n$, alors $A_j \leq C_1 \to B_1$, $B'_i \leq C_{i+1} \to B_{i+1}$ $1 \leq i \leq m-1$, $B'_m \leq B_{m+1}$, $A = \forall\, \boldsymbol{\xi}\, B'_{m+1}$ où $B_i \sim B'_i$ $1 \leq i \leq m+1$, Γ, $x : A \vdash_{\mathcal{AF}2} u : C_i$ $1 \leq i \leq m$, et $\boldsymbol{\xi}$ n'est pas libre dans Γ. Comme $A_j \in \mathcal{T}^-$, alors, d'après le lemme 4.1, $C_i \in \mathcal{T}^+$ $1 \leq i \leq m$, et $B_i \in \mathcal{T}^-$ $1 \leq i \leq m+1$. Par hypothèse d'induction on a Γ, $x : A \vdash_{\mathcal{AF}20} u_i : C_i$, donc comme $A_j \in \mathcal{T}^-$ et $B_j \in \mathcal{T}^-$ $1 \leq j \leq m+1$, on déduit que $\Gamma \vdash_{\mathcal{AF}20} t : A$. ◆

Remarque : Le théorème 4.2 signifie que si un λ-terme normal clos est typable d'un type $\forall$-positif, alors on peut supposer qu'au cours de ce typage, toutes les variables sont déclarées de types $\forall$-négatifs, tous ses sous termes essentiels sont typables de types $\forall$-positifs, et ceci en utilisant uniquement la version faible de spécification des variables du second ordre $(\forall\, E_0)$.

5. TRANSFORMATION DE GÖDEL

DÉFINITION : Soient O un symbole de relation 0-aire, A un type de $\mathcal{AF}2$. On dit que A **se termine par O** si et seulement si A est obtenu par les règles suivantes :

– O se termine par O.

– Si A se termine par O, alors $B \to A$ 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 ξ.

Autrement dit T se termine par O si et seulement si $T = \forall\, \boldsymbol{\xi_0}\, C_0$, $C_i = B_{i+1} \to \forall\, \boldsymbol{\xi_{i+1}}\, C_{i+1}$ $0 \leq i \leq r-1$, $C_r = B_{r+1} \to \forall\, \boldsymbol{\xi_{r+1}}\, O$.

LEMME 5.1 :

1) *Si* A *se termine par* O, *et* $A \sim A'$, *alors* A' *se termine par* O.

2) *Si A se termine par O, et $A \leq A'$, alors A' se termine par O.*

Preuve : Évident. ◆

LEMME 5.2 : *Soient t un λ-terme normal, $A_1, \ldots, A_n \in \mathcal{T}^-$, $A \in \mathcal{T}^+$, O ne figure pas dans les types $A_1, \ldots, A_n$, A, et $B_1, \ldots, B_m$ des types qui se terminent par O.*

Si $\Gamma = x_1 : A_1, \ldots, x_n : A_n, y_1 : B_1, \ldots, y_m : B_m \vdash_{\mathcal{AF}2} t : A$, alors $x_1 : A_1, \ldots, x_n : A_n \vdash_{\mathcal{AF}2} t : A$.

Preuve : Par induction sur t.

– Si t est une variable, alors $t = x_i$ $1 \leq i \leq n$ ou $t = y_i$ $1 \leq i \leq m$.

– Si $t = x_i$, alors c'est évident.

– Si $t = y_i$, alors $A = \forall\, \xi\, C'$, ξ n'est pas libre dans Γ, $B_i \leq C$, et $C \sim C'$. Comme B_i se termine par O, alors, d'après le lemme 5.1, A se termine par O, donc O figure dans A, ce qui est contradictoire.

– Si $t = \lambda\, x_{n+1}\, t'$, alors Γ, $x_{n+1} : A_{n+1} \vdash_{\mathcal{AF}2} t' : D$, $A = \forall\, \xi\, (A'_{n+1} \to D')$, ξ n'est pas libre dans Γ, $A_{n+1} \sim A'_{n+1}$, et $D \sim D'$. Comme $A \in \mathcal{T}^+$, alors $A'_{n+1} \in \mathcal{T}^-$, et $D' \in \mathcal{T}^+$, donc, d'après le lemme 4.1, $A_{n+1} \in \mathcal{T}^-$, et $D \in \mathcal{T}^+$. Comme O ne figure pas dans A, alors O ne figure pas dans A'_{n+1}, et O ne figure pas dans D', donc O ne figure pas dans A_{n+1}, et O ne figure pas dans D. Par hypothèse d'induction, on a $x_1 : A_1, \ldots, x_n : A_n$, $x_{n+1} : A_{n+1} \vdash_{\mathcal{AF}2} t : D$, donc $x_1 : A_1, \ldots, x_n : A_n \vdash_{\mathcal{AF}2} t : A$.

– Si $t = (x)\, u_1 \ldots u_k$ $k \geq 1$, , alors deux cas sont à voir :

  – Si $x = y_i$ $1 \leq i \leq m$, alors, d'après le corollaire 3.6, $B_i \leq C_1 \to D_1$, $D'_j \leq C_{j+1} \to D_{j+1}$ $1 \leq j \leq k-1$, $D'_k \leq D_{k+1}$, $A = \forall\, \xi\, D'_{k+1}$ où $D_j \sim D'_j$ $1 \leq j \leq k+1$, $\Gamma \vdash_{\mathcal{AF}2} u_j : C_j$ $1 \leq j \leq k$, et ξ n'est pas libre dans Γ. Comme B_i se termine par O alors, d'après le lemme 5.1, D_j $1 \leq j \leq k+1$ se termine par O, donc A se termine par O, donc O figure dans A, ce qui est contradictoire.

  – Si $x = x_i$ $1 \leq i \leq n$ alors, d'après le corollaire 3.6, $A_i \leq C_1 \to D_1$, $D'_j \leq C_{j+1} \to D_{j+1}$ $1 \leq j \leq k-1$, $D'_k \leq D_{k+1}$, $A = \forall\, \xi\, D'_{k+1}$ où $D_j \sim D'_j$ $1 \leq j \leq k+1$, $\Gamma \vdash_{\mathcal{AF}2} u_j : C_j$ $1 \leq j \leq k$, et ξ n'est pas libre dans Γ. Comme $A_i \in \mathcal{T}^+$ donc, d'après le lemme 4.1, on a $C_j \in \mathcal{T}^-$ $1 \leq j \leq k$ et $D_i \in \mathcal{T}^+$ $1 \leq i \leq k+1$. Comme O ne figure pas dans A_i, alors O ne figure pas dans C_j $1 \leq j \leq k$. En appliquant l'hypothèse inductive sur les u_j $1 \leq j \leq k$, on obtient $x_1 : A_1, \ldots, A_n \vdash_{\mathcal{AF}2} u_j : C_j$ $1 \leq j \leq k$, et donc $x_1 : A_1, \ldots, x_n : A_n \vdash_{\mathcal{AF}2} t : A$. ◆

DÉFINITION : À chaque variable de relation X, on associe un ensemble fini non vide de variables de relation $V_X = \{X_1, \ldots, X_r\}$ de même arité que X tel que : si $X \neq Y$, alors $V_X \cap V_Y = \varnothing$. À chaque variable de relation n-aire X, et à chaque liste de variables d'individu $x_1, \ldots, x_n$ on associe une formule qui se termine par O notée F_X où les variables libres de F_X sont parmi $x_1, \ldots, x_n$ et les éléments de V_X.

Pour chaque formule A, on définit la formule A^* de la manière inductive suivante :

- Si $A = O$, alors $A^* = O$.
- Si $A = X(t_1, \ldots, t_n)$, alors $A^* = F_X[t_1/x_1, \ldots, t_n/x_1]$.
- Si $A = B \to C$, alors $A^* = B^* \to C^*$.
- Si $A = \forall x\, B$, alors $A^* = \forall x\, B^*$.
- Si $A = \forall X B$, alors $A^* = \forall X_1 \ldots \forall X_r B^*$ où $V_X = \{X_1, \ldots, X_r\}$.

A^* est appelée la **transformation de Gödel** de A.

Remarques

1) La transformation de Gödel, ainsi définie, n'est pas formellement correcte car elle ne passe pas à l'α-équivalence. Il serait facile (mais fastidieux) de modifier la définition pour éviter ce problème.

2) En supposant que $A^\perp = A \to O$, J. L. Krivine a pouvé qu'une transformation de Gödel est une traduction de Gödel c.à.d. elle vérifie le résultat suivant :

Si $A_1, \ldots, A_n \vdash A$ est prouvable en logique classique du second ordre, alors $A_1^*, \ldots, A_n^* \vdash A^*$ est prouvable en logique intuitionniste du second ordre.

LEMME 5.3 : *Pour tout type D, et pour toute transformation de Gödel *, D^* se termine par O.*

Preuve : Par induction sur D. ♦

LEMME 5.4 : *Si D est un type de $\mathcal{AF}2$, et $\mathbf{u}$ une suite de termes, alors $D[\mathbf{u}/\mathbf{x}]^* = D^*[\mathbf{u}/\mathbf{x}]$.*

Preuve : Par induction sur D. ♦

LEMME 5.5 :

1) *Si $A \sim A'$, alors $A^* \sim A'^*$.*

2) *Si* $A \in \mathcal{T}^-$, *et* $A \leq A'$, *alors* $A^* \leq A'^*$.

Preuve : Il suffit d'utiliser le lemme 5.4. ◆

LEMME 5.6 : *Soient* x *une variable du premier ordre,* X *une variable de second ordre, et* A *un type de* $\mathcal{AF}2$.

1) x *est libre dans* A *si et seulement si* x *est libre dans* A^*.

2) *Si* X *n'est pas libre dans* A, *et* $Y \in V_X$, *alors* Y *n'est pas libre dans* A^*.

Preuve : Par induction sur A. ◆

DÉFINITION : Si Γ est le contexte $x_1 : A_1, \ldots, x_n : A_n$, alors on note Γ^* le contexte $x_1 : A_1^*, \ldots, x_n : A_n^*$.

LEMME 5.7 : *Si* $\Gamma \vdash {}_{\mathcal{AF}2}t : A$, *alors* $\Gamma^* \vdash {}_{\mathcal{AF}2}t : A^*$, *et on utilise les mêmes règles de typage (l'introduction (resp. l'élimination) d'une variable du second ordre* X *est remplacé par* r *fois l'introduction (resp. l'élimination) de variables* $X_1, \ldots, X_r$ *si* $V_X = \{X_1, \ldots, X_r\}$).

Preuve : Par induction sur le typage de t et on utilise les lemmes 5.4 et 5.6. ◆

COROLLAIRE 5.8 : *Soit* $D \in \mathcal{T}^+$, *et* t *un* λ*-terme normal clos. Si* $\vdash {}_{\mathcal{AF}2}t : D$, *alors* $\vdash {}_{\mathcal{AF}2}t : D^*$.

Preuve : Ceci résulte directement du théorème 4.2, et du lemme 5.7. ◆

6. OPÉRATEURS DE MISE EN MÉMOIRE

6.1 Définitions et exemples

Définitions

– Un λ-terme t est dit **essentiel** s'il est β-équivalent à un λ-terme normal clos.

– Soient T un λ-terme clos, t un λ-terme essentiel. On dit que T est un **opérateur de mise en mémoire** (en abrégé **o.m.m.**) **pour t** si et seulement si il existe un λ-terme $\tau_t \simeq {}_\beta t$, tel que pour tout $\theta_t \simeq {}_\beta t$, $(T)\,\theta_t\, f \succ (f)\,\tau_t\,[\theta_1/x_1, \ldots, \theta_n/x_n]$, où f est une nouvelle variable, $Fv\,(\tau_t) = \{x_1, \ldots, x_n, f\}$, et $\theta_1, \ldots, \theta_n$ sont des λ-termes qui dépendent de θ_t.

– Soient T un λ-terme clos, D un ensemble de λ-termes essentiels. On dit que T est un **o.m.m. pour D** si et seulement si c'en est un pour chaque élément de D.

Remarque : Soient F un terme quelconque du λ-calcul, et θ_t un λ-terme β-équivalent à $t \in D$. Au lieu d'évaluer $(F)\,\theta_t$, considérons $(T)\,\theta_t\,F$ que nous réduisons par la gauche. Comme $(T)\,\theta_t\,F = \{(T)\,\theta_t\,f\}\,[F/f]$, alors, d'après le théorème 1.1, ce calcul revient à réduire $(T)\,\theta_t\,f$ à sa forme normale de tête U, puis à réduire $U\,[F/f]$. On trouve donc, par un calcul indépendant de F, $\{(f)\,\tau_t\,[\theta_1/x_1, \ldots, \theta_n/x_n]\}\,[F/f] = (F)\,\tau_t\,[\theta'_1/x_1, \ldots, \theta'_n/x_n,\ F/f]$ où $\theta'_i = \theta_i\,[F/f]$. D'après le corollaire 1.4, la réduction gauche de $(F)\,\tau_t\,[\theta'_1/x_1, \ldots, \theta'_n/x_n,\ F/f]$ est un calcul équivalent à celle de $(F)\,\tau_t$. Tout se passe donc comme si θ_t était calculé en premier (sous la forme τ_t), et le résultat est passé comme argument à F.

Exemples: Pour chaque $n \geq 0$, on définit l'entier de Church $\underline{n} = \lambda x \lambda f\,(f)^n\,x$. Soit $\underline{N}$ l'ensemble de ces entiers. Désignons par $\underline{s} = \lambda n \lambda x \lambda f\,((n)(f)\,x)\,f$, il est facile de vérifier que $\underline{s}$ est un λ-terme pour le successeur.

Posons $T_1 = \lambda\nu\,(\nu)\,\delta G$ où $G = \lambda x \lambda y\,(x)\,\lambda z\,(y)\,(\underline{s})\,z$ et $\delta = \lambda f\,(f)\,\underline{0}$;

$$T_2 = \lambda\nu\,\lambda f\,(\nu)\,f F \underline{0} \text{ où } F = \lambda x \lambda y\,(x)\,(\underline{s})\,y.$$

Il est facile de vérifier que T_1 et T_2 sont des o.m.m. pour $\underline{N}$.

Soit $N\,[x] = \forall\,X\,\{X\,(0), \forall\,y\,[X\,(y) \to X\,(sy)] \to X\,(x)\}$ le type entier du système $\mathcal{AF}2$ (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.

Pour chaque formule F de $\mathcal{AF}2$, nous notons $\neg F$ la formule $F \to O$, et F^* la formule obtenue en remplaçant chaque formule atomique A de F par $\neg A$. * est une transformation de Gödel.

Typage de T_1

On a $\vdash_{\mathcal{AF}2} \underline{0} : N\,[0]$, donc $\vdash_{\mathcal{AF}2} \delta : \neg\neg N\,[0]$.

Comme $n : N\,[y] \vdash_{\mathcal{AF}2} n : X\,(s0), \forall\,y\,\{X\,(sy) \to X\,(ssy)\} \to X\,(sx)$ et $f : \forall\,y\,\{X\,(y) \to X\,(sy)\} \vdash_{\mathcal{AF}2} f : \forall\,y\,\{X\,(sy) \to X\,(ssy)\}$, alors $n : N\,[y]$, $f : \forall\,y\,\{X\,(y) \to X\,(sy)\}$, $x : X\,(0) \vdash_{\mathcal{AF}2} ((n)\,(f)\,x)\,f : X\,(sx)$, d'où $\vdash_{\mathcal{AF}2} \underline{s} : \forall\,y\,\{N\,[y] \to N\,[sy]\}$.

On a $x : \neg\neg N[y]$, $y : \neg N[sy]$, $z : N[y] \vdash {}_{\mathcal{AF}2}(y)(\underline{s})\, z : O$, donc $x : \neg\neg N[y]$, $y : \neg N[sy] \vdash {}_{\mathcal{AF}2}(x)\, \lambda z\, (y)(\underline{s})\, z : O$, et $\vdash {}_{\mathcal{AF}2} G : \forall y \{\neg\neg N[y] \to \neg\neg N[sy]\}$.

Or $\nu : N^*[x] \vdash {}_{\mathcal{AF}2}\nu : \neg\neg N[0]$, $\forall y \{\neg\neg N[y] \to \neg\neg N[sy]\} \to \neg\neg N[x]$, d'où $\vdash {}_{\mathcal{AF}2} T_1 : \forall x \{N^*[x] \to \neg\neg N[x]\}$.

Typage de T_2

Soit $F(x, y) = \forall X \{X\, y, \forall y (X\, y \to X\, sy) \to X\, x\}$.

On a $\vdash {}_{\mathcal{AF}2}\underline{0} : F(x, x)$ et $F(x, 0) = N[x]$.

Comme $n : F(x, sy)$, $f : \forall y (X\, y \to X\, sy)$, $x : X\, y \vdash {}_{\mathcal{AF}2}((n)(f)\, x)\, f : X\, x$, donc $\vdash {}_{\mathcal{AF}2}\underline{s} : \forall y \{F(x, sy) \to F(x, y)\}$.

On a $x : \neg F(x, y)$, $y : F(x, sy) \vdash {}_{\mathcal{AF}2}(x)(\underline{s})\, y : O$, donc $\vdash {}_{\mathcal{AF}2} F : \forall y \{\neg F(x, y) \to \neg F(x, sy)\}$.

Or $\nu : N^*[x] \vdash {}_{\mathcal{AF}2}\nu : \neg F(x, 0)$, $\forall y \{\neg F(x, y) \to \neg F(x, sy)\} \to \neg F(x, x)$, d'où $\vdash {}_{\mathcal{AF}2} T_2 : \forall x \{N^*[x] \to \neg\neg N[x]\}$.

J.-L. Krivine a montré, dans [6], que le type $\forall x \{N^*[x] \to \neg\neg N[x]\}$ caractérise les o.m.m.

THÉORÈME 6.1 : *Si* $\vdash {}_{\mathcal{AF}2} T : \forall x \{N^*[x] \to \neg\neg N[x]\}$, *alors* T *est un o.m.m. pour* $\underline{N}$.

Preuve : Voir [6]. ◆

6.2 Lambda-calcul dirigé

La définition des opérateurs de mise en mémoire est mal adaptée pour étudier leurs propriétés. En effet, c'est un énoncé *a priori* $\prod_4^0 (\forall t \exists \tau_t \forall \theta_t \exists s\, A[T, t, \tau_t, \theta_t, s])$. Le λ-calcul dirigé est une extension du λ-calcul ordinaire destiné à suivre la trace d'un λ-terme normal clos t durant une réduction de tête. L'idée de base de ce calcul est de souligner les termes provenant de t, et d'interdir l'effectuation des substitutions portant sur ces sous termes. Nous avons démontré (théorème 6.2), en utilisant le λ-calcul dirigé, que la définition des opérateurs de mise en mémoire est en faite équivalente à un énoncé $\prod_1^0$ (τ_t peut se calculer en fonction de t et σ en fonction de θ_t).

Définitions

• Si Λ est l'ensemble des λ-termes ayant V comme ensemble variables, alors l'ensemble des termes du **λ -calcul dirigé** (dites aussi **$\lambda[\ ]$-termes**), noté par $\mathbf{\Lambda[\ ]}$, est défini de la manière suivante :

– Si $x \in V$, alors $x \in \Lambda[\ ]$;

– Si $x \in V$, et $u \in \Lambda[\ ]$, alors $\lambda x u \in \Lambda[\ ]$;

– Si $u \in \Lambda[\ ]$, et $v \in \Lambda[\ ]$, alors $(u)\, v \in \Lambda[\ ]$;

– Si $t \in \Lambda$ est un λ-terme normal tel que $Fv(t) \subseteq \{x_1, \ldots, x_n\}$, $a_1, \ldots, a_n \in \Lambda[\ ]$, et i est un entier, alors $[t]_i \langle a_1/x_1, \ldots, a_n/x_n \rangle \in \Lambda[\ ]$.

Un $\lambda[\ ]$-terme de la forme $[t]_i \langle a_1/x_1, \ldots, a_n/x_n \rangle$ est dit **une boîte de numéro i dirigé par** t. Cette notation représente, intuitivement, le λ-terme t où les variables $x_1, \ldots, x_n$ sont remplacées par $a_1, \ldots, a_n$.

• On note $\langle \mathbf{a}/\mathbf{x} \rangle$ la substitution $\langle a_1/x_1, \ldots, a_n/x_n \rangle$. La substitution $\langle a_1/x_1, \ldots, a_n/x_n,\ b_1/y_1, \ldots, b_m/y_m \rangle$ est notée $\langle \mathbf{a}/\mathbf{x},\ \mathbf{b}/\mathbf{y} \rangle$, et la substitution $\langle a_1\,[\mathbf{u}/\mathbf{y}]/x_1, \ldots, a_n\,[\mathbf{u}/\mathbf{y}]/x_n \rangle$ est notée $\langle \mathbf{a}\,[\mathbf{u}/\mathbf{y}]/\mathbf{x} \rangle$.

• Pour tout $u, u_1, \ldots, u_m \in \Lambda[\ ]$, on étend les définitions de $Fv(u)$, et $u\,[\mathbf{u}/\mathbf{y}]$ par :

– $Fv([t]_i \langle \mathbf{a}/\mathbf{x} \rangle) = Fv(\mathbf{a}) = \bigcup_{1 \le i \le n} Fv(a_i)$.

– $[t]_i \langle \mathbf{a}/\mathbf{x} \rangle\,[\mathbf{u}/\mathbf{y}] = [t]_i \langle \mathbf{a}\,[\mathbf{u}/\mathbf{y}]/\mathbf{x} \rangle$, après un renommage des variables liées de $\mathbf{a}$ qui sont libres dans $u_1, \ldots, u_m$.

• Un $\lambda[\ ]$-terme t est dit **bon** si et seulement si il existe un λ-terme u, tel que $t = u\,[[t_1]_{i1} \langle \mathbf{a_1}/\mathbf{x_1} \rangle/y_1, \ldots, [t_m]_{im} \langle \mathbf{a_m}/\mathbf{x_m} \rangle/y_m]$, et pour tout $1 \le i \le m$ si $\mathbf{a_i} = a_{1,i}, \ldots, a_{ni,i}$, alors $a_{j,i}$ est bon $1 \le j \le n_i$.

Exemples : Le $\lambda[\ ]$-terme $[x_1]_0 \langle x/x_1 \rangle$ est bon, mais le $\lambda[\ ]$-terme $\lambda x\,[x_1]_0 \langle x/x_1 \rangle$ n'est pas un $\lambda[\ ]$-terme bon. En effet, la variable x est devenue liée, et alors on ne peut pas trouver un λ-terme u, tel que $\lambda x\,[x_1]_0 \langle x/x_1 \rangle = u\,[[t]_0 \langle \mathbf{a}/\mathbf{x} \rangle/y]$.

• Pour tout $\lambda[\ ]$-terme bon t, on définit un ensemble de boîtes $[\ ](t)$, par induction sur t :

Si $t = u\,[[t_1]_{i1} \langle \mathbf{a_1}/\mathbf{x_1} \rangle/y_1, \ldots, [t_m]_{im} \langle \mathbf{a_m}/\mathbf{x_m} \rangle/y_m]$, alors

$[\ ](t) = \{[t_1]_{i1} \langle \mathbf{a_1}/\mathbf{x_1} \rangle, \ldots, [t_m]_{im} \langle \mathbf{a_m}/\mathbf{x_m} \rangle\} \bigcup_{1 \le i \le m} [\ ](\mathbf{a_i})$

$[\ ](t)$ est l'ensemble des boîtes de t.

• Un **β-redex** est un $\lambda[\ ]$-terme de la forme $(\lambda x u)\, v$; $u\,[v/x]$ est son contracté.

Un **[ ]-redex** R est un $\lambda[\ ]$-terme de la forme $([\lambda y_1 \dots \lambda y_r\,(y)\,u_1 \dots u_s]_i\,\langle \mathbf{a}/\mathbf{x}\rangle)\,t_1 \dots t_r$; son contracté est défini par :

$$\begin{cases} \text{si } y = y_j\ 1 \le j \le r, \text{ alors } R = (t_j)\,[u_1]_{i1}\,\langle \mathbf{a}/\mathbf{x},\ \mathbf{t}/\mathbf{y}\rangle \dots [u_r]_{ir}\,\langle \mathbf{a}/\mathbf{x},\ \mathbf{t}/\mathbf{y}\rangle \\ \text{si } y = x_j\ 1 \le j \le n, \text{ alors } R = (a_j)\,[u_1]_{i1}\,\langle \mathbf{a}/\mathbf{x},\ \mathbf{t}/\mathbf{y}\rangle \dots [u_r]_{ir}\,\langle \mathbf{a}/\mathbf{x},\ \mathbf{t}/\mathbf{y}\rangle \end{cases}$$

où

$$i_j = \begin{cases} 0 \text{ si pour tout } i \ge 0, \text{ on a } [u_j]_i\,\langle \mathbf{a}/\mathbf{x},\ \mathbf{t}/\mathbf{y}\rangle \notin [\ ](t) \\ \max(i \text{ tel que } [u_j]_i\,\langle \mathbf{a}/\mathbf{x},\ \mathbf{t}/\mathbf{y}\rangle \in [\ ](t)) + 1 \text{ sinon} \end{cases}$$

Si t' est le $\lambda[\ ]$-terme obtenu à partir de t par contraction de son redex de tête (β-redex ou [ ]-redex), on dit que : t se réduit à t' par une **β_0-réduction de tête** (resp. par une **[ ]$_0$-réduction de tête**, resp. par une **β[ ]$_0$-réduction de tête**), et on écrit $t \succ_{\beta 0} t'$ (resp. $t \succ_{[\ ]0} t'$, resp. $t \succ_{\beta[\ ]0} t'$), si c'est un β-redex (resp. un [ ]-redex, resp. un β-redex ou un [ ]-redex). On dit que t se réduit par une **β-réduction de tête** (resp. **[ ]-réduction de tête**, resp. **β[ ]-réduction de tête**) à t', et on écrit $t \succ_\beta t'$ (resp. $t \succ_{[\ ]} t'$, resp. $t \succ_{\beta[\ ]} t'$) si et seulement si t' est obtenu à partir de t par une serie de β_0-réduction de tête (resp. [ ]$_0$-réduction de tête, resp. β[ ]$_0$-réduction de tête).

Il est facile de vérifier que (*voir* [9] et [13]) :

1) *Si t est un $\lambda[\ ]$-terme bon, et si $t \succ_{\beta[\ ]} (f)\,u_1 \dots u_r$, alors $u_1, \dots, u_r$ sont bons.*

2) *Si t est un $\lambda[\ ]$-terme bon, u un λ-terme clos, et $t[u/x] \succ_{\beta[\ ]} (f)\,u_1 \dots u_r$, alors $t[[u]_i/x] \succ_{\beta[\ ]} (f)\,u'_1 \dots u'_r$.*

• On associe à chaque boîte $[t]_i\,\langle \mathbf{a}/\mathbf{x}\rangle$ une constante $x_{t,i,\mathbf{a}}$ (une variable sur laquelle on n'a pas le droit d'abstraire), et on associe à chaque $\lambda[\ ]$-terme t un λ-terme $c(t)$ par la manière suivante :

– Si $t = x$, alors $c(t) = x$.

– Si $t = \lambda x u$, alors $c(t) = \lambda x c(u)$.

– Si $t = (u)\,v$, alors $c(t) = (c(u))\,c(v)$.

– Si $t = [u]_j\,\langle \mathbf{b}/\mathbf{y}\rangle$, alors $c(t) = x_{u,j,c(\mathbf{b})}$.

Il est facile de vérifier que (*voir* [9] et [13]) :

1) *Si t est un λ-terme, u un λ-terme clos, et $t[[u]_0/x] \succ_{\beta[\ ]} (f)\,u_1 \dots u_n$, alors il existe une substitution S telle que $t[u/x] \succ (f)\,S(c(u_1)) \dots S(c(u_n))$.*

2) *Si t est un $\lambda[\ ]$-terme bon, et si $t \succ_\beta t'$, alors $c(t) \succ c(t')$.*

3) *Si* t *est un* $\lambda[\ ]$*-terme bon,* $t \succ_{[\,]_0} t'$, *et*

$$c(t) = \lambda x_1 \ldots \lambda x_n (x_{u,i,a_1,\ldots,a_r})\, v_1 \ldots v_m,$$

alors $t = \lambda x_1 \ldots \lambda x_n\, ([u]_i\, \langle \mathbf{b}/\mathbf{y} \rangle)\, u_1 \ldots u_m$ *où* $c(b_j) = a_j$ $1 \leq j \leq r$, *et* $c(u_j) = v_j$ $1 \leq j \leq m$.

THÉORÈME 6.2 : *Soient* t *un normal* λ*-terme clos, et* T *un* λ*-terme clos.*

T est un o.m.m. pour t si et seulement si il existe un λ-terme $\tau_t \simeq_\beta t$, tel que $T\,[t]_0 f \succ_{\beta[\,]} (f)\,\tau_t\,[[t_1]_{i1}\,\langle \mathbf{a_1}/\mathbf{x_1} \rangle / y_1, \ldots, [t_m]_{im}\,\langle \mathbf{a_m}/\mathbf{x_m} \rangle / y_m]$.

Preuve : Voir [9] et [13]. ♦

7. OPÉRATEURS DE MISE EN MÉMOIRE POUR DES TYPES

DÉFINITION : Soient T un λ-terme clos, D, E deux types clos du système de typage ST ($= \mathcal{AF}2$ ou $\mathcal{F}$). On dit que T est un **o.m.m. pour le couple de types (D, E)** si et seulement si pour tout λ-terme t vérifiant $\vdash_{ST} t : D$, il existe un λ-terme essentiel τ vérifiant $\vdash_{ST} \tau^\beta : E$ tel que pour tout $\theta_t \simeq_\beta t$, on a $(T)\,\theta_t\, f \succ (f)\,\tau\,[\theta_1/x_1, \ldots, \theta_n/x_n]$ où f est une nouvelle variable, $Fv(\tau) = \{x_1, \ldots, x_n, f\}$ et $\theta_1, \ldots, \theta_n$ sont des λ-termes qui dépendent de θ_t.

Dans le cas où $D = E$, on dit que T est un **o.m.m. pour le type D**.

THÉORÈME 7.1 : *T est un o.m.m. pour le couple (D, E) si et seulement si pour tout λ-terme normal t vérifiant $\vdash_{ST} t : D$, il existe un $\lambda[\ ]$-terme τ vérifiant $c(\tau)$ est essentiel et $\vdash_{ST} c(\tau)^\beta : E$ tel que $T\,[t]_0 f \succ_{\beta[\,]} (f)\,\tau$.*

Preuve : Même preuve que celle du théorème 6.2. ♦

THÉORÈME 7.2 : *Soient D, E deux types clos $\forall$-positifs du système $\mathcal{AF}2$ tels que O ne figure pas dans E. Si $\vdash_{\mathcal{AF}2} T : D^* \to \neg\neg E$, alors T est un o.m.m. pour le couple (D, E).*

Preuve : C'est une conséquence directe du théorème suivant :

THÉORÈME 7.3 : *Soient D, E deux types clos $\forall$-positifs du système F tels que O ne figure pas dans E. Si $\vdash_{\mathcal{F}} T : D^* \to \neg\neg E$, alors T est un o.m.m. pour le couple (D, E).*

En effet :

LEMME 7.4 :

1) *Si* $T^+ \in \mathcal{T}^+$, *alors* $T^{+\circ} \in \mathcal{T}^+$.

2) *Si* $T^- \in \mathcal{T}^-$, *alors* $T^{-\circ} \in \mathcal{T}^-$.

Preuve: Par induction simultanée sur les types ∀-positifs et les types ∀-négatifs. ◆

LEMME 7.5: *Pour toute transformation de Gödel* * *du système* $\mathcal{AF}2$, *il existe une transformation de Gödel* $^{*\prime}$ *du système F vérifiant la propriété suivante: pour tout type D de* $\mathcal{AF}2$ *on a* $D^{*\circ} = D^{\circ *\prime}$.

Preuve: $^{*\prime}$ est la restriction de * sur les types de $\mathcal{F}$. ◆

Soit t un λ-terme normal tel que $\vdash_{\mathcal{F}} t : D$. Si $\vdash_{\mathcal{AF}2} T : D^* \to \neg\neg E$, alors, d'après le théorème 2.4, $\vdash_{\mathcal{F}} T : D^{*\circ} \to \neg\neg E^\circ$. D'après le lemme 7.5, il existe une transformation de Gödel $^{*\prime}$ telle que $\vdash_{\mathcal{F}} T : D^{\circ *\prime} \to \neg\neg E^\circ$, donc, d'après le lemme 7.4, et le théorème 7.3, il existe un $\lambda[\ ]$-terme τ vérifiant $c(\tau)$ est essentiel, et $\vdash_{\mathcal{F}} c(\tau)^\beta : E$ tel que $T\,[t]_0\, f \succ_{\beta[\ ]} (f)\,\tau$. Donc $Ttf \succ (f)\,c(\tau)\ [t_1/x_1, \ldots, t_n/x_n]$. D'après le corollaire 5.8, on a $f : \neg E \vdash_{\mathcal{AF}2} Ttf : O$, donc $f : \neg E \vdash_{\mathcal{AF}2} (f)\,c(\tau)\ [t_1/x_1, \ldots, t_n/x_n]$, donc $f : \neg E \vdash_{\mathcal{AF}2} (f)\,c(\tau)^\beta : O$, et, d'après le corollaire 3.6, $\vdash_{\mathcal{AF}2} c(\tau)^\beta : E$. ◆

Remarque: La condition « $D, E \in \mathcal{T}^+$ » est nécessaire pour avoir le théorème 7.3. En effet soient $D = \forall X\,\{\forall Y\,(Y \to X) \to X\}$, $t = \lambda x\,(x)\,\lambda yy$, $T = \lambda\nu\,(\nu)\,\lambda x\lambda f\,(f)\,\lambda y\,(y)\,x$, et * la traduction de Gödel.

– D n'est ni un type ∀-positif ni un type ∀-négatif.

– On a $x : \forall Y\,(Y \to X) \vdash_{\mathcal{F}} x : (X \to X) \to X$, et $\vdash_{\mathcal{F}} \lambda yy : X \to X$, donc $x : \forall Y\,(Y \to X) \vdash_{\mathcal{F}} (x)\,\lambda yy : X$, et alors $\vdash_{\mathcal{F}} t : D$.

– On a

$$\nu : D^* = \forall X\,\{\forall Y\,(\neg Y \to \neg X) \to \neg X\}$$
$$\vdash_{\mathcal{F}} \nu : \forall Y\,(\neg Y \to \neg\neg D) \to \neg\neg D.$$

Comme $x : \neg Y$, $f : \neg D$, $y : \forall Y\,(Y \to X) \vdash_{\mathcal{F}} y : \neg Y \to X$, alors $x : \neg Y$, $f : \neg D \vdash_{\mathcal{F}} \lambda y\,(y)\,x : D$, donc $\vdash_{\mathcal{F}} \lambda x\,\lambda f\,(f)\,\lambda y\,(y)\,x : \forall Y\,(\neg Y \to \neg\neg D)$; Il en résulte que $\vdash_{\mathcal{F}} T : D^* \to \neg\neg D$.

– $T\,[t]_0\, f \succ_\beta (([t]_0)\,\lambda x\,\lambda f\,(f)\,\lambda y\,(y)\,x)\,f \succ_{[\ ]0}$

$$((\lambda x\,\lambda f\,(f)\,\lambda y\,(y)\,x)\,[\lambda yy]_0)\,f \succ_\beta (f)\,\lambda y\,(y)\,[\lambda yy]_0$$

et $c(\lambda y\,(y))\,[\lambda yy]_0) = \lambda y\,(y)\,x_{\lambda yy,0}$. Donc T n'est pas un o.m.m. pour le type D.

Avant de faire la preuve du théorème 7.3, nous la regardons dans un cas particulier.

7.1 Preuve du théorème 7.3 dans un cas particulier

Posons $\mathbb{N} = \forall X \{X, X \to X, \to X\}$, et désignons par * la traduction de Gödel.

Prouvons que: Si $\vdash {}_{\mathcal{F}}T : \mathbb{N}^* \to \neg\neg\mathbb{N}$, alors T est un o.m.m. pour $\mathbb{N}$.

Soit t un λ-terme normal tel que $\vdash {}_{\mathcal{F}}t : \mathbb{N}$, alors $t = \underline{n}\, n \geq 0$.

Faisons la preuve pour un $\underline{n} \geq 0$

Il est facile de vérifier que $\vdash {}_{\mathcal{F}}\underline{n} : \mathbb{N}^*$ (ceci peut être déduit du corollaire 5.8). Donc $f : \neg\mathbb{N} \vdash {}_{\mathcal{F}}T\,\underline{n}f : O$, et $T\,\underline{n}f$ est résoluble. D'après le corollaire 3.6, on a $T\,\underline{n}f \succ (f)\,\delta$, donc $T\,[\underline{n}]_0\, f \succ {}_{\beta[\ ]}(f)\,\tau$, et il suffit de prouver que $c(\tau)$ est essentiel et $\vdash {}_{\mathcal{F}}c(\tau)^{\beta} : \mathbb{N}$.

LEMME 7.7:

1) *Si* $\Gamma' = \Gamma,\ x : \mathbb{N}^* \vdash {}_{\mathcal{F}}(x)\,u_1 \ldots u_n : O$ *alors* $n = 3$, *et il existe un type* F *tel que:*

$- \Gamma' \vdash {}_{\mathcal{F}}u_1 : \neg F.$

$- \Gamma' \vdash {}_{\mathcal{F}}u_2 : \neg F \to \neg F.$

$- \Gamma' \vdash {}_{\mathcal{F}}u_3 : F.$

2) *Si* $\Gamma,\ x : \neg F \vdash {}_{\mathcal{F}}(x)\,u : O$, *alors* $\Gamma,\ x : \neg F \vdash {}_{\mathcal{F}}u : F$.

Preuve:

1) D'après le corollaire 3.6 on a $\mathbb{N}^* \leq A_1 \to B_1$, $B_i \leq A_{i+1} \to B_{i+1}$ $1 \leq i \leq n-1$, $B_n \leq O$, et $\Gamma' \vdash {}_{\mathcal{F}}u_i : A_i$ $1 \leq i \leq n$. Comme $\mathbb{N}^* \leq A_1 \to B_1$, alors il existe un type F tel que $A_1 = \neg F$, et $B_1 = (\neg F \to \neg F) \to \neg F$. De plus $B_1 \leq A_2 \to B_2$, donc $A_2 = \neg F \to \neg F$, et $B_2 = \neg F$. Finalement $B_2 \leq A_3 \to B_3$, donc $A_3 = F$, et $B_3 = O$, et, d'après le corollaire 3.6, on s'arrête, et $n = 3$.

2) Même preuve que 1). ♦

DÉFINITION: Un contexte $\Gamma = f : \neg\mathbb{N},\ x_{u1,i1,a1,b1} : A_1, \ldots, x_{us,is,as,bs} : A_s$ est dit **bon** si et seulement si pour tout $1 \leq j \leq s$ on a l'un de deux cas suivant:

$- u_j = \underline{n},\ i_j = 0,\ a_j,\ b_j = \varnothing$, et $A_j = \mathbb{N}^*$.

$- u_j = (x_1)^m\, x_2$ $0 \leq m \leq n-1$, et il existe un type F tel que $A_j = \neg F$, $\Gamma \vdash {}_{\mathcal{F}}a_j : \neg F$, et $\Gamma \vdash {}_{\mathcal{F}}b_j : \neg F \to \neg F$.

Donc en particulier chaque type A_j $1 \leq j \leq s$ se termine par O.

$T\,[\underline{n}]_0 f \succ_{\beta[\ ]} (f)\,\tau$, donc il existe une suite $t_0 = T\,[\underline{n}]_0 f$, $t_1, \ldots, t_{r-1}$, $t_r = (f)\,\tau$ telle que:

– $t_{k-1} \succ_{\beta} t_k$ ou $t_{k-1} \succ_{[\ ]0} t_k$ $1 \leq k \leq r$:

– si $t_{k-1} \succ_{\beta} t_k$, alors $k = r$ ou $k < r$ et $t_k \succ_{[\ ]0} t_{k+1}$.

Pour simplifier, notons m le λ-terme $(x_1)^m\, x_2$ $1 \leq m \leq n$.

LEMME 7.8: *Pour tout* $0 \leq k \leq r$, *il existe un contexte bon* Γ_k *tel que* $\Gamma_k \vdash_{\mathcal{F}} c(t_k) : O$, *et si* $t_k \succ_{[\ ]0} t_{k+1}$, *alors* $t_k = ([\underline{n}]_0)\,u_1\,u_2\,u_3$ *ou* $t_k = ([m]_i\,\langle v_1/x_1,\ v_2/x_2\rangle)\,v_3$ $0 \leq m \leq n-1$.

Preuve: Procédons par induction sur k.

Pour $k = 0$, posons $\Gamma_0 = f : \neg\mathbb{N}$, $x_{\underline{n},0} : \mathbb{N}^*$. Comme $c(t_0) = T\,x_{\underline{n},0}\,f$, alors $\Gamma_0 \vdash_{\mathcal{F}} c(t_0) : O$, donc $T\,x_{\underline{n},0}\,f$ est résoluble. D'après le corollaire 3.6 et le lemme 7.7, on a $T\,x_{\underline{n},0}\,f \succ (x_{\underline{n},0})\,v_1\,v_2\,v_3$ ou $T\,x_{\underline{n},0} f \succ (f)\,\delta$, donc $T\,[\underline{n}]_0\,f$ est résoluble et $t_1 = ([\underline{n}]_0)\,u_1\,u_2\,u_3$ ou $t_1 = (f)\,\tau$.

Supposons que la propriété est vraie pour k et prouvons la pour $k+1$.

– Si $t_k \succ_{\beta} t_{k+1}$, alors $c(t_k) \succ c(t_{k+1})$, et on pose $\Gamma_{k+1} = \Gamma_k$.

– Si $t_k \succ_{[\ ]0} t_{k+1}$, alors $t_n = ([\underline{n}]_0)\,u_1\,u_2\,u_3$ ou $t_k = ([m]_i\,\langle u_2/x_1,\ u_1/x_2\rangle)\,u_3$ $0 \leq m \leq n$.

• Si $t_k = ([\underline{n}]_0)\,u_1\,u_2\,u_3$, alors $\Gamma_k \vdash_{\mathcal{F}} (x_{\underline{n},0})\,c(u_1)\,c(u_2)\,c(u_3) : O$, donc $x_{\underline{n},0} : \mathbb{N}^* \in \Gamma_k$, et, d'après le lemme 7.7, il existe un type F tel que $\Gamma_k \vdash_{\mathcal{F}} c(u_1) : \neg F$, $\Gamma_k \vdash_{\mathcal{F}} c(u_2) : \neg F \to \neg F$, et $\Gamma_k \vdash_{\mathcal{F}} c(u_3) : F$. Deux cas sont à voir:

– Si $n > 0$, alors $t_{k+1} = (u_1)\,[n-1]_j\,\langle u_2/x_1, u_1/x_2\rangle)\,u_3$, et $c(t_{k+1}) = ((c(u_1))\,x_{n-1,j,c(u2),c(u1)})\,c(u_3)$ donc, si on pose $\Gamma_{k+1} = \Gamma_k$, $x_{n-1,j,c(u2),c(u1)} : \neg F$, on obtient $\Gamma_{k+1} \vdash_{\mathcal{F}} c(t_{k+1}) : O$.

– Si $n = 0$, alors $t_{k+1} = (u_1)\,u_3$, et $c(t_{k+1}) = (c(u_1))\,c(u_3)$, donc, si on pose $\Gamma_{k+1} = \Gamma_k$, on obtient $\Gamma_{k+1} \vdash_{\mathcal{F}} c(t_{k+1}) : O$.

• Si $t_k = ([m]_i\,\langle v_1/x_1,\ v_2/x_2\rangle\,v_3$ $0 \leq m \leq n$, alors $\Gamma_k \vdash_{\mathcal{F}} (x_{m,i,c(v1),c(v2)})\,c(v_3) : O$, donc $x_{m,i,c(v1),c(v2)} : \neg F \in \Gamma_k$, $\Gamma_k \vdash_{\mathcal{F}} c(v_1) : F$, $\Gamma_k \vdash_{\mathcal{F}} c(u_2) : \neg F \to \neg F$, et, d'après le lemme 7.7, $\Gamma_k \vdash_{\mathcal{F}} c(v_3) : F$. Deux cas sont à voir:

– Si $m > 0$, alors $t_{k+1} = ((v_1)\,[m-1]_j\,\langle v_1/x_1,\ v_2/x_2\rangle)\,v_3$, et $c(t_{k+1}) = ((c(v_1))\,x_{n-1,j,c(v1),c(v2)})\,c(v_3)$, donc, si on pose $\Gamma_{k+1} = \Gamma_k$, $x_{m-1,j,c(v1),c(v2)} : \neg F$, on obtient $\Gamma_{k+1} \vdash_{\mathcal{F}} c(t_{k+1}) : O$.

– Si $m = 0$, alors $t_{k+1} = (v_1)\, v_3$, et $c(t_{k+1}) = (c(v_2))\, c(v_3)$, donc, si on pose $\Gamma_{k+1} = \Gamma_k$, on obtient $\Gamma_{k+1} \vdash_{\mathcal{F}} c(t_{k+1}) : O$.

Il est clair que Γ_{k+1} est bon.

$\Gamma_{k+1} \vdash_{\mathcal{F}} c(t_{k+1}) : O$ donc $c(t_{k+1})$ est résoluble. D'après le corollaire 3.6, et le lemme 7.7, on a $c(t_{k+1}) \succ (x_{\underline{n},0})\, v_1\, v_2\, v_3$ ou $c(t_{k+1}) \succ (x_{m,j,v1,v2})\, v_3$ ou $c(t_{k+1}) \succ (f)\, \delta$, donc, si $t_k \succ {}_{\beta}[\,]_0 t_{k+1}$, $t_{k+1} = ([\underline{n}]_0)\, u_1\, u_2\, u_3$ ou $t_{k+1} = ([m]_j\, \langle u_1/x_1,\, u_2/x_2\rangle)\, u_3$. ◆

Donc il existe un contexte bon Γ_r tel que $\Gamma_r \vdash_{\mathcal{F}} c(t_r) = (f)\, c(\tau) : O$, et, d'après le corollaire 3.6, $\Gamma_r \vdash_{\mathcal{F}} c(\tau) : \mathbb{N}$. Donc $c(\tau)$ est fortement normalisable, et $\Gamma_r \vdash_{\mathcal{F}} c(\tau)^{\beta} : \mathbb{N}$. D'où, d'après le lemme 5.2, $c(\tau)$ est essentiel, et $\vdash_{\mathcal{F}} c(\tau)^{\beta} : \mathbb{N}$. ◆

7.2 Preuve générale du théorème 7.3

Soit t un λ-terme normal tel que $\vdash_{\mathcal{F}} t : D$. D'après le corollaire 5.8, on a $\vdash_{\mathcal{F}} t : D^*$. Si $\vdash_{\mathcal{F}} T : D^* \to \neg\neg E$, alors $f : \neg E \vdash_{\mathcal{F}} T\, t\, f : O$, donc $T\, t\, f$ est résoluble. D'après le corollaire 3.6, $T\, t\, f \succ (f)\, \delta$, donc $T\, [t]_0 f \succ {}_{\beta}[\,](f)\, \tau$. Il reste à prouver que $c(\tau)$ est essentiel, et $\vdash_{\mathcal{F}} c(\tau)^{\beta} : E$. Prenons un typage normal de $\vdash_{\mathcal{F}} t : D$, et pour tout $u \in STE(t)$, désignons par D_u le type de u, et pour toute variable x de t, désignons par A_x le type déclaratif de x. D'après le théorème 4.2, on a $D_u \in \mathcal{T}^+$, $A_x \in \mathcal{T}^-$, et on peut associer à chaque λ-terme $u = \lambda x_1 \ldots \lambda x_n\, (x)\, u_1 \ldots u_m \in STE(t)$ deux listes d'équations (L_1 et L_2) qui proviennent du typage de u:

$$L_1 = \begin{cases} A^*_x = \forall\, \mathbf{X_0}\, B^*_0 \\ B^*_i = D^*_{u_{i+1}} \to \forall\, \mathbf{X_{i+1}}\, B^*_{i+1} \qquad 0 \le i \le m-1 \end{cases}$$

$$L_2 = \begin{cases} D^*_u = \forall\, \mathbf{Y_0}\, C^*_0 \\ C^*_i = A^*_{x_{i+1}} \to \forall\, \mathbf{Y_{i+1}}\, C^*_{i+1} \qquad 0 \le i \le n-2 \\ C^*_{n-1} = A^*_{x_n} \to \forall\, \mathbf{Y_n} \forall\, \mathbf{X_m}\, B^*_m \\ \\ \text{ou } n = 0 \text{ et } D^*_u = \forall\, \mathbf{Y_n} \forall\, \mathbf{X_m}\, B^*_m. \end{cases}$$

où $\mathbf{X_i}$ n'est pas libre dans B^*_j $0 \le i, j \le m$, $\mathbf{Y_i}$ n'est pas libre dans $A^*_{x_j}$ $j < i$, et n'est pas libre dans A_x pour tout x $1 \le i \le n$.

DÉFINITION: Un contexte $\Gamma = f : \neg E,\, x_{u1,i1,\mathbf{a1}} : A_1,\, \ldots,\, x_{ul,il,\mathbf{al}} : A_1$ est dit **bon** si et seulement si pour tout $1 \le i \le l$, $u_i \in STE(t)$, $A_i = D^*_{u_i}\, [\mathbf{F}/\mathbf{X}]$, et si a représente la variable x_j dans la suite $\mathbf{a_i}$, alors $\Gamma \vdash_{\mathcal{F}} a : A^*_{x_j}\, [\mathbf{F}/\mathbf{X}]$.

Donc, d'après le lemme 5.3, chaque type A_i $1 \leq i \leq l$ se termine par O.

LEMME 7.9: *Si* $u = \lambda x_1 \ldots \lambda x_n (x) u_1 \ldots u_m$, Γ' *est un contexte bon et* $\Gamma' = \Gamma$, $x_{u,i,\mathbf{a}} : D_u^* [\mathbf{F}/\mathbf{X}] \vdash \mathcal{F}(x_{u,i,\mathbf{a}}) v_1 \ldots v_n \; w_1 \ldots w_l : O$, *alors il existe un contexte bon* Γ'' *tel que* $\Gamma'' \vdash \mathcal{F}(a) x_{u1,i,\mathbf{a},\mathbf{v}} \ldots x_{u_m,i_m,\mathbf{a},\mathbf{v}} w_1 \ldots w_l : O$ *où*

$$a = \begin{cases} v_i \text{ si } x = x_i & 1 \leq i \leq n \\ a_j \text{ si } x = y_i \text{ est libre dans } u. \end{cases}$$

Preuve: Faisons la preuve dans le cas où $n \neq 0$. D'après le corollaire 3.6, on a $D_u^* [\mathbf{F}/\mathbf{X}] \leq D_1 \to E_1$, $E_i \leq D_{i+1} \to E_{i+1}$ $1 \leq i \leq n-1$, $E_n \leq D_1' \to E_1'$, $E_j' \leq D_{j+1}' \to E_{j+1}'$ $1 \leq j \leq l-1$, $E_l' \leq O$, $\Gamma' \vdash \mathcal{F} v_i : D_i$ $1 \leq i \leq n$, et $\Gamma' \vdash \mathcal{F} w_i : D_i'$ $1 \leq i \leq l$.

LEMME 7.10: *Il existe des suites de formules* $\mathbf{F_0}, \ldots, \mathbf{F_{n-1}}$ *telles que:*

$$D_i = A_{x_i}^* [\mathbf{F}/\mathbf{X}, \mathbf{F_0}/\mathbf{Y_0}, \ldots, \mathbf{F_{i-1}}/\mathbf{Y_{i-1}}] \qquad 1 \leq i \leq n,$$

$$E_i = \forall \mathbf{Y_i} C_i^* [\mathbf{F}/\mathbf{X}, \mathbf{F_0}/\mathbf{Y_0}, \ldots, \mathbf{F_{i-1}}/\mathbf{Y_{i-1}}] \qquad 1 \leq i \leq n-1,$$

et

$$E_n = \forall \mathbf{Y_n} \forall \mathbf{X_m} B_m^* [\mathbf{F}/\mathbf{X}, \mathbf{F_0}/\mathbf{Y_0}, \ldots, \mathbf{F_{i-1}}/\mathbf{Y_{i-1}}].$$

Preuve: Par induction sur n, et on utilise la liste L_2. ♦

Soient $\mathbf{F_n}$ la suite de formules utilisées pour remplacer les variables $\mathbf{Y_n}$ en passant de E_n à $D_1' \to E_1'$. Pour toute formule F notons $F^{\$}$, la formule $F [\mathbf{F}/\mathbf{X}, \mathbf{F_0}/\mathbf{Y_0}, \ldots, \mathbf{F_n}/\mathbf{Y_n}]$.

On a $D_i = A_{x_i}^{*\$}$ et $\Gamma' \vdash \mathcal{F} a : A_x^{*\$}$. Donc d'après la liste L_1, on a:

$$\Gamma'' = \Gamma', \; x_{u1,i1,\mathbf{a},\mathbf{v}} : D_{u_1}^{*\$}, \ldots, x_{u_m,i_m,\mathbf{a},\mathbf{v}} : D_{u_m}^{*\$}$$

$\vdash \mathcal{F}(a) x_{u1,i1,\mathbf{a},\mathbf{v}} \ldots x_{u_m,i_m,\mathbf{a},\mathbf{v}} : \forall \mathbf{X_m} B_m^{*\$}$, donc

$$\Gamma'' \vdash \mathcal{F}(a) x_{u1,i1,\mathbf{a},\mathbf{v}} \ldots x_{u_m,i_m,\mathbf{a},\mathbf{v}} w_1 \ldots w_l : O.$$

Il est facile de vérifier que Γ'' est un contexte bon. ♦

$T [t]_0 f \succ_{\beta[\,]} (f) \tau$, donc il existe une suite $t_0 = T [t]_0 f$, $t_1, \ldots, t_{r-1}$, $t_r = (f) \tau$ telle que:

- $t_{k-1} \succ_\beta t_k$ ou $t_{k-1} \succ_{[\,]0} t_k$ $1 \leq k \leq r$:
- si $t_{k-1} \succ_\beta t_k$, alors $k = r$ ou $k < r$ et $t_k \succ_{[\,]0} t_{k+1}$.

LEMME 7.11: *Pour tout* $0 \leq k \leq r$, *il existe un contexte bon* Γ_k *tel que* $\Gamma_k \vdash \mathcal{F} c(t_k) : O$.

Preuve : Procédons par induction sur k

Pour $k = 0$, posons $\Gamma_0 = f : \neg E$, $x_{t,0} : D^*$. Comme on a $c(t_0) = T\, x_{t,0} f$, donc $\Gamma_0 \vdash_{\mathcal{F}} c(t_0) : O$.

Supposons que la propriété est vraie pour k et prouvons la pour $k+1$.

– Si $t_k \succ_{\beta} t_{k+1}$, alors $c(t_k) \succ c(t_{k+1})$, et on pose $\Gamma_{k+1} = \Gamma_k$.

– Si $t_k \succ_{[\,]_0} t_{k+1}$, alors, comme $\Gamma_k \vdash_{\mathcal{F}} c(t_k) : O$, $c(t_k)$ est résoluble. D'après le corollaire 3.6, on a $c(t_k) \succ (x_{u,i,\mathbf{a}})\, v_1 \dots v_r$, et d'après la liste L_2, r est plus grand que le nombre des abstractions se trouvant au début de u, donc $t_k = ([\lambda x_1 \dots \lambda x_n (x)\, u_1 \dots u_m]_i \langle \mathbf{a}/\mathbf{y} \rangle)\, v_1 \dots v_n\, w_1 \dots w_1$, et $t_{k+1} = (a)\, [u_1]_{i_1} \langle \mathbf{a}/\mathbf{y},\, \mathbf{v}/\mathbf{x} \rangle \dots [u_m]_{i_m} \langle \mathbf{a}/\mathbf{y},\, \mathbf{v}/\mathbf{x} \rangle\, w_1 \dots w_1$ où

$$a = \begin{cases} v_i \text{ si } x = x_i & 1 \le i \le n \\ a_j \text{ si } x = y_j \text{ est libre dans } u. \end{cases}$$

D'où $c(t_k) = (x_{\lambda x_1 \dots \lambda x_n (x)\, u_1 \dots u_m,\, i,\, c(\mathbf{a})})\; c(v_1) \dots c(v_n)\, c(w_1) \dots c(w_1)$, et $c(t_{k+1}) = (c(a))\, x_{u_1, i_1, c(\mathbf{a}), c(\mathbf{v})} \cdots x_{u_m, i_m, c(\mathbf{a}), c(\mathbf{v})} c(w_1) \dots c(w_1)$, et, d'après le lemme 7.9, il existe un contexte bon Γ_{k+1}, tel que $\Gamma_{k+1} \vdash_{\mathcal{F}} c(t_{k+1}) : O$. ◆

Donc il existe un contexte bon Γ_r tel que $\Gamma_r \vdash_{\mathcal{F}} c(t_r) = (f)\, c(\tau) : O$, donc, d'après le corollaire 3.6, on a $\Gamma_r \vdash_{\mathcal{F}} c(\tau) : E$. Donc $c(\tau)$ est fortement normalisable, et $\Gamma_r \vdash_{\mathcal{F}} c(\tau)^{\beta} : E$. Donc, d'après le lemme 5.2, on a $c(\tau)$ est essentiel, et $\vdash_{\mathcal{F}} c(\tau)^{\beta} : E$. D'où la preuve du théorème 7.3. ◆

CONCLUSION

Le résultat de J.-L. Krivine (théorème 6.1) concernant la caractérisation des o.m.m. pour les entiers de Church par un type du système $\mathcal{AF}2$ suggère plusieurs questions :

– Pouvons nous généraliser ce résultat pour d'autres types ?

– Pourquoi avons nous besoin d'une traduction de Gödel ?

Le théorème 7.2 donne une réponse sur la première question. En effet, nous avons montré que le résultat de J.-L. Krivine reste valable pour les types où le quantificateur universel du second ordre apparaît positivement, et pour des généralisations de la traduction de Gödel classique. Le théorème 7.2 permet aussi de généraliser le résultat de J.-L. Krivine pour tous les types de données (les booléens, les listes d'objets, les arbres d'objets...). En effet, J.-L. Krivine a présenté, dans [6], une méthode pour définir les types de

données par des types $\forall$-positifs du système de typage $\mathcal{AF}2$. Et donc nous obtenons pour ces types le théorème ([5]) suivant:

Soit $D[x]$ un type de données, $\Lambda(D)$ l'ensemble de représentants de ce type en λ-calcul, et * une transformation de Gödel. Si $\vdash_{\mathcal{AF}2} T : \forall x \{D^*[x] \rightarrow \neg\neg D[x]\}$, alors T est un o.m.m. pour $\Lambda(D)$.

Le théorème 7.2 a été démontré, dans ce papier, dans le cadre des systèmes de typage $\mathcal{F}$ et $\mathcal{AF}2$. Il serait utile d'obtenir des résultats identiques pour d'autres systèmes de typage, par exemple le système $\mathcal{TTR}$ de M. Parigot et le système $\mathcal{F}w$ de J.-Y. Girard.

Une direction vers laquelle s'oriente aussi notre travail actuel est le rôle des opérateurs de mise en mémoire en logique classique. En effet, M. Parigot et J.-L. Krivine ont trouvé des algorithmes permettant d'extraire des programmes (des termes du λ-calcul) à partir des preuves en logique classique. Si θ_n est un λ-terme de type $N[s^n 0]$ dans le systeme $\mathcal{AF}2$, on sait que θ_n est β-équivalent à l'entier de Church $\underline{n}$. Ce n'est plus vrai si θ_n est un « entier classique » (*i.e.* provient d'une preuve en logique classique de $N[s^n 0]$). Mais M. Parigot et J.-L. Krivine ont prouvé que: Si $\vdash_{\mathcal{AF}2} T : \forall x \{N^*[x] \rightarrow \neg\neg N[x]\}$, alors, pour tout entier classique θ_n, $(T)\,\theta_n\,\lambda xx \simeq_\beta \underline{n}$. Les opérateurs de mise en mémoire peuvent donc servir aussi pour « afficher » les résultats provenant d'un programme issus d'une preuve en logique classique. Ce résultat est très intéressant. En effet, pour la première fois, nous observons une intervention de la logique classique, et nous espérons ainsi mieux expliquer la présence de la traduction de Gödel dans le type caractérisant les opérateurs de mise en mémoire.

REMERCIEMENTS

Je tiens à remercier J.-L. Krivine et R. David pour leur aide qui m'a permis de réaliser ce travail.

RÉFÉRENCES

1. H. Barendregt, *The lambda calculus: Its Syntax and Semantics*, North Holland, 1984.
2. J. Y. Girard, Y. Lafont et P. Taylor, *Proofs and types*, Cambridge University Press, 1986.
3. P. Giannini et S. Ronchi Della Rocca. *Characterization of typing in polymorphic type discipline*. LICS, Edinbourg, 1988, p. 61-70.

([5]) J.-L. Krivine a cité ce théorème (sans le démontrer) dans [4] et [6].

4. J.-L. KRIVINE, Lambda calcul, évaluation paresseuse et mise en mémoire, *Informatique Théorique et Application,* 1991, *25*, 1, p. 67-84.
5. J.-L. KRIVINE, *Lambda calcul, types et modèle*, Masson, Paris, 1990.
6. J.-L. KRIVINE, Opérateurs de mise en mémoire et traduction de Gödel, *Archive for Mathematical Logic*, 1990, *30*, p. 241-267.
7. J.-L. KRIVINE, Mise en mémoire (preuve générale). Manuscrit,1991.
8. R. LABIB-SAMI. Typer avec (ou sans) types auxilières, Manuscrit, 1986.
9. K. NOUR, Opérateurs de mise en mémoire en lambda-calcul pur et typé, Thèse de doctorat, Université de Savoie, 1993.
10. K. NOUR, Opérateurs propres de mise en mémoire, *C. R. Acad. Sci.*, 1993, *317*, Série I, p. 1-6.
11. K. NOUR, Strong storage operators and data types, *Archive for Mathematical Logic*, 1995, *34*, p. 65-78.
12. K. NOUR, Preuve syntaxique d'un théorème de J.-L. Krivine sur les opérateurs de mise en mémoire, *C. R. Acad. Sci.*, 1994, *318,* Série I, p. 201-204.
13. K. NOUR et R. DAVID, Storage operators and directed lambda-calculus, *Journal of Symbolic Logic*, 1995, *60*, n° 4, p. 1054-1086.