RAIRO. INFORMATIQUE THÉORIQUE

PATRICK SALLÉ

Une généralisation de la théorie des types en λ-calcul

RAIRO. Informatique théorique, tome 14, nº 2 (1980), p. 143-167

<http://www.numdam.org/item?id=ITA_1980__14_2_143_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 14, n° 2, 1980, p. 143 à 167)

UNE GÉNÉRALISATION DE LA THÉORIE DES TYPES EN λ-CALCUL (Première partie) (*)

par Patrick SALLE ([1])

Communiqué par J.-F. PERROT

Résumé. — *Nous décrivons une généralisation de la théorie des types de Curry qui étend l'affectation de type à l'ensemble du λ-calcul. Dans cette théorie deux expressions β-η convertibles ont le même ensemble de types et une expression typée possède suivant la valeur de son type une forme normale ou une forme normale gauche.*

Abstract. — *We present a generalization of Curry's type theory which extends type assignment to the whole of λ-calculus. In our theory, any two expressions that are β-η convertible have the same set of types. Also, the value of the type of any typed expression indicates whether the expression possesses a normal form or a head normal form.*

INTRODUCTION

Nous proposons une théorie des types sur le λ-calcul qui est un outil pour les preuves d'arrêt et d'équivalence des programmes. Dans cette théorie un système de déduction composé d'axiomes et de règles d'inférences fournit un moyen mécanique d'obtention des types d'une expression.

L'intérêt d'une théorie des types associée à un système formel (ou à un langage de programmation) est de fournir un moyen de vérifier statiquement qu'une expression syntactiquement bien formée du système vérifie également certaines propriétés sémantiques. Dès 1920, Curry [5] reconstruit la logique mathématique à partir de la logique combinatoire et, la théorie des types qu'il propose lui permet d'éliminer les expressions sans interprétation, c'est-à-dire, de son point de vue, sans signification. Depuis, le λ-calcul s'est révélé un bon formalisme pour exprimer la sémantique des langages de programmation tant pour les structures de contrôle que pour les structures de données (Böhm [2], Landin [10], Morris [11], Nolin [12], Robinet [13, 14, 15], Durieux, Salle [20]). Tout programme peut être traduit de manière automatique en une λ-expression.

(*) Reçu juin 1978, révisé octobre 1979.

([1]) Université Paul-Sabatier, Laboratoire Langages et Systèmes informatiques, Toulouse.

L'intérêt d'un type est alors de prouver qu'une λ-expression admet une forme normale pour établir que le programme correspondant se termine et fournit un résultat. Toutefois si on considère un programme représenté par une expression P, appliqué à une donnée représentée par l'expression X, le fait que PX ait une forme normale, c'est-à-dire que le calcul de P sur la donnée X se termine n'implique pas en général que l'expression P seule ait également une forme normale. Or dans les théories classiques, seules certaines expressions ayant une forme normale ont un type. On en déduit que P et PX ont rarement un type.

Notre but est d'étendre la théorie des types de manière à pouvoir :

— affecter un type significatif à des expressions n'ayant qu'une forme normale gauche comme le combinateur de point fixe Y;

— affecter un type à un plus grand nombre d'expressions ayant une forme normale mais ayant des sous-expressions sans forme normale.

D'une manière générale une théorie des types est basée sur la définition d'un ensemble de types et d'un ensemble de règles d'affectation. L'ensemble des types est le plus petit ensemble engendré à partir d'un ensemble de types de base fini ou dénombrable et de deux règles de construction :

1° chaque type de base est un type;

2° si α et β sont deux types alors $(\alpha)\beta$ est un type.

Suivant les théories un choix particulier est effectué sur les types de base et leur signification, de plus des relations peuvent structurer cet ensemble. Le type $(\alpha)\beta$ représente intuitivement l'ensemble des applications de l'ensemble de type α dans l'ensemble de type β. Pour chaque théorie il faut définir les règles d'affectation qui permettent d'associer types et expressions du λ-calcul.

Ainsi Morris [11] et Sanchis [16] reprennent l'hypothèse de Curry [5] de « stratification de l'univers » selon laquelle toute expression a au plus un type. Les règles d'affectation sont les suivantes :

(*a*) deux occurences de la même variable apparaissant libres dans une expression ont le même type;

(*b*) si x est de type α et M de type β alors $\lambda x.M$ est de type $(\alpha)\beta$;

(*c*) si une combinaison MN est de type β et si N est de type α alors M est de type $(\alpha)\beta$.

Dans cette théorie toute expression typée a une forme normale, la réciproque est fausse et de nombreuses formes normales telles que xx n'ont pas de type. Il est montré [11] que cette théorie dans laquelle l'affectation d'un type est toujours décidable peut s'appliquer à des langages de programmation mais hélas seulement dans des cas triviaux. En effet :

— si F est une fonction $F(F(x))$ n'a de type que si l'image de F est du même type que son domaine. Si $F(x)=$partie entière (x), x réel, $F(F(x))$ n'a pas de type;

— le combinateur de point fixe Y nécessaire pour exprimer la sémantique des calculs itératifs et récursifs n'a pas de type puisque n'ayant pas à lui seul de forme normale;

— SI condition ALØRS A SINØN B n'a de type que si A et B sont des expressions correspondant toutes les deux à des calculs qui se terminent.

M. Coppo, M. Dezani [6] ont construit une théorie à partir de deux types de base 0 et 1. Le type 0 correspond à la propriété d'avoir une forme normale et 1 à celle de conserver une forme normale après application à un nombre quelconque d'arguments en forme normale. Les auteurs introduisent deux axiomes d'équivalence et une relation d'ordre partiel notée $\sqsubseteq$ qui structurent l'ensemble des types et qui sont justifiés par leurs travaux précédents [3]. L'hypothèse de stratification est abandonnée et les règles d'affectation de type sont les règles (b) et (c) auxquelles s'adjoignent :

(a') chaque variable libre a un type unique (chaque occurence de la variable aura donc un type en accord avec la règle d);

(d) si une expression F a le type τ et si $\tau' \sqsubseteq \tau$ alors F a le type τ'.

Dans cette théorie il est montré que toute expression typée a une forme normale et que l'ensemble des expressions typées est plus grand que celui obtenu par la théorie de Morris. Toutefois :

— les expressions n'ayant qu'une forme normale gauche comme Y ne sont pas typées;

— les expressions ayant des sous-expressions sans forme normale n'ont pas de type.

Plus récemment les mêmes auteurs [7] ont étendu leur théorie en affectant à chaque expression un ensemble fini de types non comparables. Dans cette théorie, deux expressions du λI-calcul qui sont α-β convertibles ont le même ensemble de types, et une expression a une forme normale si et seulement si elle a un type.

Nous proposons tout d'abord une théorie qui rend possible l'affectation d'un type à des expressions du λ-calcul ayant une forme normale mais contenant des sous-expressions sans forme normale [18]. Pour cela nous introduisons, en plus des deux types de base 0 et 1 qui conservent leur signification, un type de base ω, de caractère universel qui peut être affecté à toute λ-expression. Son but est de tenir compte du fait que dans les λ-expressions externes au λ-I-calcul il existe des β-réductions qui font disparaître des sous-expressions de l'expression initiale. Cet « effet cachant » existe notamment dans l'instruction SI ALORS SINON. La théorie ainsi construite étend l'affectation des types à des expressions n'ayant qu'une forme normale gauche et permet d'engendrer des types qui rendent compte de manière plus précise des propriétés des λ-expressions. Son intérêt est de permettre l'affectation d'un type par déduction formelle à une expression représentant la sémantique d'un programme. Suivant la valeur de ce type cette expression possède une forme normale ou une forme normale gauche. Après avoir étudié la structure de l'ensemble des types T engendré par les types de base 0, 1 et ω nous définissons les règles d'affectation de type et nous étudions leurs propriétés. Toute forme normale possède le type 0, toute expression a le type ω et l'affectation d'un type τ à une forme normale est décidable (prop. 2.3 et th. 1). L'étude de l'ensemble des formes normales ayant le type 1 permet de montrer que cet ensemble est $\mathcal{N}_\omega$ [3] d'où l'on déduit que la combinaison de deux formes normales dont l'une possède le type 1 a également une forme normale (prop. 5). Ce théorème sert de base aux démonstrations des théorèmes 2, toute expression ayant un type distinct de ω a une forme normale gauche, et 3, toute expression ayant le type 0 a une forme normale de même type. Nous donnons ensuite des contre-exemples des réciproques de ces théorèmes et nous analysons les limites de la théorie.

Nous étendons ensuite cette théorie en affectant à chaque expression un ensemble fini de types [19]. Cette extension permet de donner un type significatif à toutes les expressions résolubles (Barendregt [1]) c'est-à-dire ayant une forme normale gauche. Nous présentons pour ce faire un nouvel ensemble de types T' de structure semblable à celle de T en modifiant les règles de construction et nous donnons de nouvelles règles d'affectation adéquates. Nous montrons (th. 4) que deux expressions α-β-η convertibles ont le même ensemble de types. Les théorèmes 5 et 6 prouvent la conservation dans la théorie étendue des résultats des théorèmes 2 et 3. Nous déduisons de ces propriétés le théorème 7 qui établit l'équivalence pour une λ-expression entre avoir un type et être résoluble. Suivant la valeur de ce type l'expression possède une forme normale ou une forme normale gauche. La décidabilité de l'affectation n'est montrée que pour les λ-expressions en forme normale.

Citons enfin quelques développements et applications possibles de cette théorie. Coppo et Dezani [8] ont construit récemment une théorie semblable où l'ensemble des types d'une expression n'est conservé que par α-β-conversion. Nous avons montré [9] que ces deux théories induisaient des relations d'équivalence entre expressions qui sont exactement l'une, l'équivalence dans le modèle $P\omega$, l'autre une équivalence opérationnelle extensionnelle. Ce dernier résultat est très important car il fournit une approche différente de la preuve d'égalité dans $P\omega$. On peut envisager de développer de la même manière des théories associées aux différents modèles du λ-calcul. Des études sont également en cours sur la décidabilité de l'affectation de type et l'extension des résultats aux λ-calculs appliqués avec constantes et δ-règles. Enfin un lieu a été établi entre le λ-calcul étiqueté et certaines théories de type [11].

1. DÉFINITIONS PRÉLIMINAIRES

DÉFINITION 1 : $\mathcal{N}_\omega$ est l'ensemble des formes normales telles que si $N \in \mathcal{N}_\omega$, $\forall n$ entier, $\forall X_1, \ldots, X_n$ formes normales $(\ldots((NX_1)\, X_2)\ldots X_n)$ a une forme normale.

$\mathcal{N}_\omega$ ne contient que des termes non fermés et des constantes.

DÉFINITION 2 : $\mathcal{N}_{-\omega}$ est l'ensemble des expressions telles que si $N \in \mathcal{N}_{-\omega}$, $\forall n$ entier, $\forall X_1, \ldots, X_n$ quelconques, $(\ldots((NX_1)\, X_2)\ldots X_n)$ n'a pas de forme normale.

C'est l'ensemble des termes non résolubles, « unsolvable terms » Barendregt [1], Wadsworth [21].

DÉFINITION 3 : Une forme normale gauche est une expression de la forme $M = \lambda x_1 \ldots \lambda x_n . zM_1 \ldots Mp$ avec $n \geqq 0$, $p \geqq 0$ et où :

– z est appelée variable de tête de M et peut être égale à l'un des x_i;

– M_i est une expression quelconque quel que soit i. C'est la i-ième composante de M;

– $\lambda x_1 \ldots \lambda x_n$ sont les abstractions initiales de M.

DÉFINITION 4 : La longueur d'une λ-expression F notée $|F|$ est définie par :

1° $|x| = 1$, $\forall x$ variable;

2° $|XY| = |X| + |Y|$, $\forall X$ et Y λ-expressions;

3° $|\lambda x . Y| = |Y| + 1$, $\forall Y$ λ-expression.

2. L'ENSEMBLE DES TYPES T

Il est construit à partir de trois types de base 0, 1, ω dont nous donnons les motivations sémantiques.

2.1. Les types de base

ω est le type universel que possède toute λ-expression (prop. 3).

0, toute forme normale a le type 0. D'autre part si F a le type 0 alors F a une forme normale (prop. 4, th. 3).

1, toute expression de $\mathcal{N}_\omega$ a le type 1 et si une expression a le type 1 alors elle a une forme normale qui appartient à $\mathcal{N}_\omega$ (lemme 2 et 3).

Axiomes d'équivalence

A_0 : $0 = (1)\, 0$ qui correspond à la propriété suivante :

Si N a une forme normale, si $X \in \mathcal{N}_\omega$ alors XN a une forme normale (Bohm, Dezani [3]);

A_1 : $1=(0)\ 1$ qui correspond à la définition de $\mathcal{N}_\omega$;

A_ω : $\omega=(\tau)\ \omega$, $\forall \tau \in T$ qui correspond à la définition de $\mathcal{N}_{-\omega}$.

REMARQUE : $(\tau_1)\ (\tau_2)\ldots(\tau_{n-1})\ \tau_n$ est une abréviation pour $(\tau_1)((\tau_2)(\ldots((\tau_{n-1})\tau_n)\ldots))$.

L'emploi du type 1 se justifie :

– par sa complémentarité avec le type 0 et son rôle dans la définition d'une base (§ 3);

– par le fait que dans un λ-calcul appliqué il y a des constantes qui appartiennent à $\mathcal{N}_\omega$ et que dans cet article il n'est pas tenu compte de l'interprétation sémantique de ces constantes avec des δ-règles.

2.2. Structure de l'ensemble T

T est l'ensemble engendré par les règles (1) et (2) données dans l'introduction à partir des types de base $0, 1, \omega$.

DÉFINITION 5 : Deux types σ et τ sont dits équivalents ($\sigma=\tau$) si et seulement s'ils peuvent être réduits au même type par un nombre fini d'applications de A_0, A_1 et A_ω.

Exemple 1 : $((0)\ \omega)\ (0)\ 1$ et $(\omega)\ ((0)\ (0)\ 1)$ sont des types et sont équivalents à $(\omega)\ 1$.

La relation = partitionne T en classes d'équivalences. On montre que :

(P1) l'équivalence de deux types est décidable;

(P2) chaque classe d'équivalence possède un représentant de longueur minimale (nombre minimal d'occurrences de $0, 1, \omega$).

DÉFINITION 6 : La longueur d'un type τ, notée $\|\tau\|$ est la longueur du représentant de longueur minimale de la classe d'équivalence de τ.

Exemple 2 :

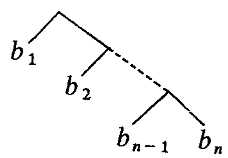
$$\|((0)0)1\|=3, \qquad \|((0)1)0\|=1 \qquad \text{et} \qquad \|(\tau_1)\ldots(\tau_n)\omega\|=1,$$

(P3) $$\forall \tau \in T, \quad \forall n>0, \quad \exists \tau_1, \ldots, \tau_n \in T$$

tels que

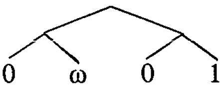
$$\tau=(\tau_1)(\tau_2)\ldots(\tau_{n-1})\tau_n.$$

Les preuves de ces propriétés se font en considérant qu'il existe une bijection entre un type $(\tau_1)\ldots(\tau_{n-1})\ \tau_n$ où τ_i est un type quelconque et l'arbre



où b_i désigne le sous-arbre associé à τ_i dont les feuilles ont pour étiquette des types de base.

Exemple 3 : Le type ((0) ω) (0) 1 est associé à l'arbre



(P4) si $\tau=(\tau_1)\tau_2$ alors soit τ est équivalent à un type atomique soit $\|\tau_1\|<\|\tau\|$ et $\|\tau_2\|<\|\tau\|$.

Dans la suite de l'article nous ne considérerons que le représentant minimal de chaque classe d'équivalence.

Définition 7 : Les relations $\sqsubseteq$ et $\sqsubset$ sont définies sur T comme suit :

– $\omega \sqsubseteq 0 \sqsubseteq 1$;

– soient $(\sigma_1)\tau_1$ et $(\sigma_2)\tau_2$ deux types alors $(\sigma_1)\tau_1 \sqsubseteq (\sigma_2)\tau_2$ si et seulement si $\tau_1 \sqsubseteq \tau_2$ et $\sigma_2 \sqsubseteq \sigma_1$;

– $\tau_1 \sqsubset \tau_2$ si et seulement si $\tau_1 \sqsubseteq \tau_2$ et $\tau_1 \neq \tau_2$.

(P5) $\sqsubseteq$ est une relation d'ordre partiel.

(P6) Les relations $=$, $\sqsubset$, $\sqsubseteq$, $\neq$ sont toujours décidables.

Exemple 4 : $(\omega)\,1 \sqsupset 1$, $((\omega)\,1)\,0 \sqsubset 0$, $(\omega)\,0 \sqsupset 0$, $(\omega)\,0 \not\sqsubseteq 1$, mais $((\omega)\,0)\,(\omega)\,0$ est incomparable à 0 et à 1.

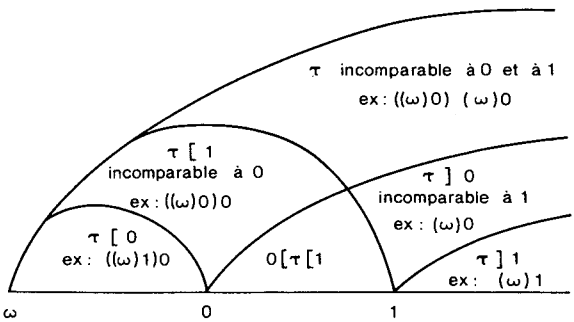
Par récurrence sur la longueur des types on montre que :

(P7) L'ensemble des types compris entre 0 et 1 est exactement l'ensemble des types n'ayant pas d'occurence de ω.

On retrouve l'ensemble des types de [6] :

(P8) T muni de la relation $\sqsubseteq$ a une structure de treillis.

On peut schématiser sa structure de la manière suivante :



3. RÈGLES D'AFFECTATION

L'affectation de type se fait par un système de déduction comportant un schéma d'axiome et quatre règles de déduction. Il s'agit du système décrit dans [6] et modifié pour tenir compte de l'introduction de ω. Le type d'une expression est défini par rapport à une base qui précise les types affectés aux variables libres.

On supposera par la suite que toutes les variables liées d'une expression sont distinctes et distinctes des variables libres ce qui peut toujours s'obtenir par α-conversion.

NOTATION : L'expression F possède le type τ sera noté τF.

DÉFINITION 8 : Une base $\mathscr{B}$ est un ensemble d'affectations de la forme σx ou x est une variable libre, σ un type inclus dans 1 au sens large.

REMARQUE : On utilise une base en l'absence de toute interprétation sémantique des variables libres qui est le cadre fixé pour cet article.

DÉFINITION 9 : Une base étendue $\mathscr{E}$ est une base où les types affectés peuvent être quelconques.

REMARQUE : Le type d'une expression ne se définit que par rapport à une base. Les bases étendues ne sont introduites que pour donner une signification au calcul des types des sous-expressions propres d'une expression.

Le système de déduction est le suivant :

Axiome A_p si $\mathscr{B}$ est une base et si $\sigma x \in \mathscr{B}$ alors $\mathscr{B} \vdash \sigma x$;
Règle R_c : si $\mathscr{B} \vdash (\sigma)\,\tau\, F$ et $\mathscr{B} \vdash \sigma\, G$ alors $\mathscr{B} \vdash \tau\,(FG)$;
Règle R_a : si $\mathscr{B}$, $\sigma x \vdash \tau F$ et si x n'apparaît pas dans $\mathscr{B}$ alors $\mathscr{B} \vdash (\sigma)\,\tau\,(\lambda x . F)$;
Règle R_e : si $\mathscr{B} \vdash \sigma F$ et si $\sigma = \tau$ alors $\mathscr{B} \vdash \tau F$;
Règle R_i : si $\mathscr{B} \vdash \sigma F$ et si $\tau \sqsubset \sigma$ alors $\mathscr{B} \vdash \tau F$.

Les règles A_p, R_c et R_a sont classiques en théorie des types et les règles R_e et R_i donnent un sens aux relations $=$ et $\sqsubset$. Il est aisé de voir que le type d'une λ-expression n'est pas en général unique et que si elle possède un type τ alors elle possède tous les types τ' inclus dans τ.

Les règles d'affectation par rapport à une base étendue $\mathscr{E}$ sont les mêmes que par rapport à une base $\mathscr{B}$. L'introduction des bases étendues est rendue nécessaire par la remarque suivante. Lors de l'application de la règle R_a, $\mathscr{B}$, $\sigma x \vdash \tau F$, σ peut être quelconque. Le type de F est donc un type défini par rapport à une base étendue $\mathscr{E} = \mathscr{B} \cup \sigma x$ alors que celui de $\lambda x . F$ est défini par rapport à la base $\mathscr{B}$. Plus généralement dans une expression F typée par rapport à une base $\mathscr{B}$ les variables libres de F ont un type défini dans $\mathscr{B}$ comme étant inclus dans 1 et les variables liées ont un type quelconque. Dans une sous-expression propre H de F telle que $F = C[H]$ (où $C[\quad]$ désigne le contexte de H) il peut donc y avoir des variables libres $x_1, \ldots, x_n$ qui sont liées dans F. Le type de H est défini par rapport à la base quelconque $\mathscr{E} = \mathscr{B} \cup \sigma_1 x_1 \cup \ldots \cup \sigma_n x_n$ mais ce type n'a de signification pour H que dans le contexte $C[\quad]$.

Exemple 5 : $\Delta = \lambda x . xx$,

$$\left.\begin{array}{l} R_c \ \underline{(\sigma)\tau x \vdash \tau(xx)} \\ R_a \dfrac{(\sigma)\tau x \vdash \tau(xx)}{\vdash ((\sigma)\tau)\tau(\lambda x . xx)} \end{array}\right\} \quad \text{si} \quad (\sigma)\tau \sqsupseteq \sigma.$$

Dans la base vide Δ possède tous les types de la forme $((\sigma)\tau)\tau$ avec $(\sigma)\tau \sqsupseteq \sigma$ et

en particulier les types $(1)\,1$ et $((0)\,0)\,0$ qui contiennent le type 0. Δ est en forme normale.

Exemple 6 : On déduit d'après R_c que $\Delta\Delta$ à tous les types de la forme τ si $((\sigma)\tau)\tau \sqsupseteq (\sigma)\tau$ ce qui peut également s'écrire $\tau \sqsupseteq \tau$ et $(\sigma)\tau \sqsubseteq \sigma$. Mais $(\sigma)\tau \sqsupseteq \sigma$ ce qui implique que $(\sigma)\tau = \sigma$ qui ne se vérifie que pour $\sigma = \tau = \omega$. $\Delta\Delta$ n'a que le type ω et n'a pas de forme normale.

Exemple 7 : $A = \lambda y . y(\Delta\Delta)$. Cette expression n'a pas de type dans les théories classiques

$$\begin{array}{c} R_c \underline{(\omega)\tau y \vdash (\omega)\tau y, \ \vdash \ \omega(\Delta\Delta)}, \\ R_a \dfrac{(\omega)\tau y \vdash \tau(y(\Delta\Delta))}{\vdash ((\omega)\tau)\tau(\lambda y . (y(\Delta\Delta)))}. \end{array}$$

On voit qu'on peut affecter un type distinct de ω à une expression en forme normale gauche et que ce type n'est pas comparable à 0. On rappelle que $y(\Delta\Delta)$ n'a le type τ que par rapport à une base étendue, ici $(\omega)\tau y$ et que ce type n'est significatif que dans son contexte.

Exemple 8 : $B = A\,(\lambda x\, \lambda z . z)$. En prenant $\tau = (\sigma)\sigma$ on obtient :

$$\begin{array}{c} R_a \underline{\sigma z \vdash \sigma z}, \\ R_a \underline{\vdash (\sigma)\sigma(\lambda z . z)}, \\ R_c \dfrac{\vdash (\omega)(\sigma)\sigma(\lambda x . \lambda z . z), \ \vdash ((\omega)(\sigma)\sigma)(\sigma)\sigma A}{\vdash (\sigma)\sigma(A(\lambda x \lambda z . z))} \end{array}$$

B peut avoir n'importe quel type $(\sigma)\sigma$ notamment $(0)\,0$. B a une forme normale qui est $\lambda z . z$.

4. PROPRIÉTÉS DE L'AFFECTATION DE TYPES

Nous montrons tout d'abord une propriété importante sur les types des sous-expressions d'une λ-expression typée qui servira dans de nombreuses démonstrations puis nous établissons la propriété d'invariance des types par β-réduction. Nous donnons également la preuve que toute expression a le type ω, que toute forme normale a le type 0 et que l'affectation d'un type τ quelconque à une forme normale est décidable. Des propriétés semblables sont obtenues par M. Dezani et M. Coppo [6] sur un ensemble de types construit à partir des deux types de base 0 et 1.

PROPOSITION 1: *Soit F une λ-expression différente d'une simple variable telle que $\mathscr{B} \vdash \tau F$ alors :*

– *si $F \equiv XY$ il existe un type σ tel que $\mathscr{B} \vdash (\sigma)\, \tau\, X$ et $\mathscr{B} \vdash \sigma\, Y$;*

– *si $F \equiv \lambda x . Y$ alors il existe σ et ρ, tels que $\tau = (\sigma)\,\rho$ et que $\mathscr{B}$, $\sigma x \vdash \rho\, Y$.*

Remarquons toutefois que $\mathscr{B}$, σx ne constitue une base que si $\sigma \sqsubseteq 1$ et que Y n'a le type ρ qu'à cette condition.

Remarquons également que le théorème est valable pour une base étendue.

Démonstration : On part de la constatation que plusieurs applications successives des règles R_e et R_e peuvent être remplacées par une seule application de R_e ou de R_i de par la transitivité des relations $=$ et $\sqsubseteq$. D'autre part comme F n'est pas une simple variable la dernière règle appliquée dans la déduction de $\mathscr{B} \vdash \tau F$ ne peut être A_p. Nous considèrerons alors les deux cas.

$F \equiv XY$. La dernière règle appliquée ne peut être que R_c, R_e ou R_i. Dans le premier cas la proposition est démontrée. Dans les deux autres cas l'avant-dernière déduction ne peut être que R_c c'est-à-dire :

$$\cfrac{\cfrac{\mathscr{B} \vdash (\sigma)\, e\, X,\ \mathscr{B} \vdash \sigma\, Y}{\mathscr{B} \vdash e\,(XY)}\, R_c}{\mathscr{B} \vdash \tau\,(XY)}\, R_e \text{ ou } R_i$$

qui peut se transformer en

$$\cfrac{\cfrac{\mathscr{B} \vdash (\sigma)\, e\, X}{\mathscr{B} \vdash (\sigma)\, \tau\, X,\ \mathscr{B} \vdash \sigma\, Y}\, R_e \text{ ou } R_i}{\mathscr{B} \vdash \tau\,(XY)}\, R_c$$

puisque si $\rho = \tau$ alors $(\sigma)\,\rho = (\sigma)\,\tau$ et si $\tau \sqsubseteq \rho$ alors $(\sigma)\,\tau \sqsubseteq (\sigma)\,\rho$.

$F \equiv \lambda x.Y$. La dernière règle appliquée ne peut être que R_a, R_e ou R_i et dans les deux derniers cas l'avant-dernière réduction ne peut être que R_a :

$$\frac{\dfrac{\mathscr{B},\ \sigma x \vdash \rho Y}{\mathscr{B} \vdash (\sigma)\rho(\lambda x.Y)} R_a,}{\mathscr{B} \vdash \tau(\lambda x.Y)} R_e \text{ ou } R_i.$$

Dans le cas de R_e on a $\tau = (\sigma)\rho$ et la proposition est prouvée. Dans le cas de R_i puisque $\tau \sqsubseteq (\sigma)\rho$ alors il existe σ_1, ρ_1 tels que $\tau = (\sigma_1)\rho_1$ et que $\sigma \sqsubseteq \sigma_1$ et $\rho_1 \sqsubseteq \rho$. Or si $\mathscr{B},\ \sigma x \vdash \rho Y$ alors $\mathscr{B},\ \sigma_1 x \vdash \rho Y$ puisque l'on peut déduire σx de $\sigma_1 x$ par application de R_i. De même $\mathscr{B},\ \sigma_1 x \vdash \rho_1 Y$ puisque $\rho_1 \sqsubseteq \rho$ ce qui peut se résumer par

$$\frac{\mathscr{B},\ \sigma_1 x \vdash \sigma_1 Y}{\mathscr{B} \vdash (\sigma_1)\rho_1(\lambda x.Y)} R_a \qquad \text{avec} \quad \tau = (\sigma_1)\rho_1$$

ce qui prouve la proposition :

PROPOSITION 2 : *Si une λ-expression Y se réduit en une λ-expression Y' ($Y \rhd Y'$) par une suite de β-réductions et si $\mathscr{B} \vdash \sigma Y$ alors $\mathscr{B} \vdash \sigma Y'$. La proposition est également valable dans le cas d'une base étendue.*

La preuve se fait en deux parties. Nous montrons d'abord que si R est un β-radical et si $\mathscr{B} \vdash \tau R$ alors on a également $\mathscr{B} \vdash \tau R'$ ou R' est le radical R réduit. En effet soit $R \equiv (\lambda x.F)G$. Par la proposition 1 il existe un type σ tel que dans une déduction de $\mathscr{B} \vdash \tau R$ nous ayons $\mathscr{B} \vdash (\sigma)\tau\ (\lambda x.F)$, $\mathscr{B} \vdash \sigma G$ et que x n'apparaisse pas dans $\mathscr{B}$. Toujours d'après la proposition 1 il existe donc une déduction de $\mathscr{B},\ \sigma x \vdash \tau F$. Par définition $R' \equiv [G/x]F$ et pour montrer que $\mathscr{B} \vdash \tau R'$ il suffit de remplacer dans la déduction, $\mathscr{B},\ \sigma x \vdash \tau F$ l'affectation σx qui n'est pas dans $\mathscr{B}$ par une déduction de $\mathscr{B} \vdash \sigma G$.

Il suffit alors pour montrer la proposition de considérer le cas où Y se réduit à Y' par réduction d'un seul β-radical. Si R est ce β-radical alors dans la déduction de $\mathscr{B} \vdash \tau Y$ il existe une déduction de $\mathscr{E} \vdash \sigma\ R$ où $\mathscr{E}$ est obtenue en ajoutant à $\mathscr{B}$ les affectations des types des variables libres de R qui sont liées dans Y. Si R' est le radical R réduit on a démontré qu'il existe une déduction de $\mathscr{E} \vdash \sigma R'$. On obtient alors une déduction de $\mathscr{B} \vdash \tau Y'$ en remplaçant dans $\mathscr{B} \vdash \tau Y$ la déduction de $\mathscr{E} \vdash \sigma R$ par celle de $\mathscr{E} \vdash \sigma R'$.

La proposition suivante justifie le caractère universel du type ω.

PROPOSITION 3 : *Toute λ-expression a le type ω.*

Démonstration par récurrence sur $|N|$:

– $|N| = 1$, $N \equiv x$, il suffit de prendre $\mathscr{B} = \{\omega x\}$;

— si la propriété est vraie pour toute expression de longueur inférieure à n alors si $|N|=n$ deux cas se présentent :

$N \equiv \lambda x.M$, $|M|<n$. Il existe une déduction de $\mathscr{B}$, $\sigma x \vdash \omega M$ par hypothèse de récurrence et donc de $\mathscr{B} \vdash (\sigma)\omega(\lambda x.M)$ de par la règle R_a. Or $(\sigma)\omega=\omega$, $\forall \sigma$;

$N \equiv XY$, $|X|<n$ et $|Y|<n$. Par hypothèse de récurrence $\exists \mathscr{B}$, $\mathscr{B} \vdash \omega X$ et $\mathscr{B} \vdash \omega Y$ et d'après $R_c \mathscr{B} \vdash \omega(XY)$ car $(\omega)\omega=\omega$.

Nous montrons ensuite que 0 est le type que possède toute forme normale.

PROPOSITION 4 : *Si N est une forme normale et $\mathscr{B}$ une base qui affecte* 1 *à toutes les variables libres de N alors $\mathscr{B} \vdash ON$.*

Démonstration par récurrence sur $|N|$:

— $|N|=1$, $N \equiv x$ variable libre

$$\frac{\mathscr{B} \vdash 1x}{\mathscr{B} \vdash Ox} \frac{A_p}{R_i},$$

— supposons la propriété vraie pour toute expression N de longueur inférieure à n et soit N, tel que $|N|=n$. Deux cas se présentent :

$-N \equiv xN_1 \ldots N_p \quad (p>0)$.

Comme $|N_i|<n$, $\forall_i$, $1 \leqq i \leqq p$ il existe une déduction de $\mathscr{B} \vdash 0 N_i$. x apparaît libre dans N, donc $\mathscr{B} \vdash 1x$ et comme $1=(0) \ldots (0)1$ nous obtenons une déduction de $\mathscr{B} \vdash 1N$ par p applications de la règle R_c et de $\mathscr{B} \vdash ON$ par une application de R_i;

— $N \equiv \lambda x.M$, x est libre dans M, $|M|<|N|$ par hypothèse de récurrence $\mathscr{B}$, $1x \vdash OM$. Par la règle R_a $\mathscr{B} \vdash (1)\,0\,(\lambda x.M)$. Or $(1)0=0$.

THÉORÈME 1 : *Si N est en forme normale alors $\mathscr{B} \vdash \tau N$ est décidable.*

Démonstration par récurrence sur $|N|$:

$|N|=1$, $N \equiv x$. La variable libre x a un type σ dans $\mathscr{B}$, et $\tau \sqsubseteq \sigma$ est une relation décidable.

$|N|=n$ et supposons le théorème vrai pour toutes les λ-expressions de taille inférieure à n,

— $N \equiv xN_1 \ldots N_p$. La variable libre x a un type σ dans $\mathscr{B}$ et d'après (P3) σ peut se mettre sous la forme $(\sigma_1) \ldots (\sigma_p)\,\sigma_{p+1}$. Comme $|N_i|<n$, $\forall i$, $1 \leqq i \leqq p$, $\mathscr{B} \vdash \sigma_i N_i$ est décidable, de même que $\mathscr{B} \vdash \sigma_{p+1} N$ et $\sigma_{p+1} \sqsupseteq \tau$. $\mathscr{B} \vdash \tau N$ est donc décidable;

— $N \equiv \lambda x.M$, τ peut se mettre sous la forme $(\tau_1)\tau_2$ d'après (P3). Par hypothèse de récurrence $\mathscr{B}$, $\tau_1 x \vdash \tau_2 M$ est décidable car $|M|<|N|$.

REMARQUE : Le théorème 1 est valable pour une λ-expression quelconque si l'on convient d'affecter le type ω à tous les β-radicaux.

5. PROPRIÉTÉS DES FORMES NORMALES POSSÉDANT LE TYPE 1

Pour étudier ces propriétés nous ferons appel à des résultats de [3] sur l'ensemble $\mathcal{N}_\omega$, obtenues en considérant les variables remplaçables des λ-expressions. Nous montrons que l'ensemble $\mathcal{N}_\omega$ coïncide exactement avec l'ensemble des formes normales ayant le type 1. D'autre part nous montrons que la combinaison de deux formes normales dont l'une possède le type 1 est une expression qui a également une forme normale. Ce dernier résultat servira de base à la démonstration des propriétés des expressions ayant le type 0.

DÉFINITION 10 : Dans une forme normale $N \equiv \lambda y_1 \ldots y_n.(x N_1 \ldots N_p)$:

(*a*) $y_1 \ldots y_n$ sont des variables remplaçables;

(*b*) les variables libres sont non remplaçables;

(*c*) si $X \equiv z Y_1 \ldots Y_m$ est un sous-terme de N et si z est remplaçable (non remplaçable) alors les variables liées dans les abstractions initiales de $Y_i (i=1, m)$ sont remplaçables (non remplaçables).

Exemple 9 : $N \equiv \lambda x_1 \lambda x_2.(u \lambda x_3.(x_2 \lambda x_4.(x_3 x_4)))$:
– $x_1 x_2$ et x_4 sont remplaçables;
– u et x_3 sont non remplaçables.

Nous donnons une nouvelle définition de $\mathcal{N}_\omega$ dont l'équivalence avec la définition 1 a été prouvée dans [3].

DÉFINITION 11 : $N \in \mathcal{N}_\omega$ ssi :
(*a*) la variable de tête de l'expression est libre;
(*b*) si la variable de tête d'un sous-terme propre X de N est remplaçable alors toutes les variables de tête des composantes de X sont non remplaçables.

Dans l'exemple 9, $N \in \mathcal{N}_\omega$.

LEMME 1 : *Si N est une forme normale et si $\mathcal{B} \vdash 1 N$ (respectivement $\mathcal{B} \vdash \tau\ N\ \tau \sqsupseteq 1$ alors il existe une déduction de ce type dans laquelle le type* 0 (*respectivement un type inclus dans* 0) *a été affecté aux variables remplaçables de N.*

Nous prouvons ce résultat par induction sur le nombre k d'applications de la règle (*c*) de la définition 10.

(1) $k=0$ c'est-à-dire que nous considérons une variable remplaçable liée dans les abstractions initiales de N.

Soit $N \equiv \lambda y_1 \ldots \lambda y_l \lambda x . \overline{N}$. Par ${}_{l+1}$ applications du théorème 1 à partir de $\mathscr{B} \vdash 1 N$ on obtient $\mathscr{B}, O y_1, O y_2 \ldots O y_l, O x \vdash 1 \overline{N}$. D'autre part si $\mathscr{B} \vdash \tau N$ alors il existe un type $(\tau_1) \ldots (\tau_{l+1}) \tau_{l+2} = \tau$ tel que

$$\mathscr{B}, \tau_1 y_1, \ldots, \tau_l y_l, \qquad \tau_{l+1} x \vdash \tau_{l+2} \overline{N}$$

et

$$\tau \sqsupseteq 1, \quad \tau_{l+2} \sqsupset 1 \qquad \text{et} \qquad \tau_i \sqsubseteq 0, \qquad \forall i = 1, \quad l+1.$$

(2) Supposons vraie la propriété pour tout $k<n$. Soit $X \equiv z Y_1 \ldots Y_p$ un sous-terme propre de N, z une variable remplaçable et x une variable liée dans les abstractions initiales de Y_i, $i \geqq 1$.

Par hypothèse d'induction nous avons $O z$ (respectivement τz avec $\tau \sqsubseteq 0$) dans une déduction du type de N. Distinguons les deux cas :

(*a*) $O z$ alors $0 = \underbrace{(1) \ldots (1)}_{p \text{ fois}} 0$ et par les règles R_e et R_i nous devons avoir $\mathscr{B}' \cup \mathscr{E} \vdash 1 \; Y_i$ pour une certaine base $\mathscr{B}'$ et où

$$\mathscr{E} = \{ \sigma_1 x_1, \ldots, \sigma_s x_s | \; x_i \text{ est une variable libre de } Y_i \text{ et liée dans } N \};$$

(*b*) τz alors $\tau = (\tau_1) \ldots (\tau_p) \tau_{p+1} \sqsubseteq 0$ ce qui donne $\tau_i \sqsupseteq 1$ pour $1 \leqq i \leqq p$ et $\tau_{p+1} \sqsubseteq 0$.

$\mathscr{B}' \cup \mathscr{E} \vdash \tau_i Y_i$ pour les mêmes raisons que précédemment.

Si $Y_i \equiv \lambda z_1 \ldots \lambda z_k \lambda x . \overline{Y}_i$ alors par $k+1$ applications du théorème 1 on obtiendra respectivement :

$$\mathscr{B}' \cup \mathscr{E}, \qquad O z_1, \ldots, O z_k, \qquad O x \vdash 1 \overline{Y}_i$$

et

$$\mathscr{B}' \cup \mathscr{E}, \qquad \sigma_1 z_1, \ldots, \sigma_k z_k, \qquad \sigma_{k+1} x \vdash \tau_i \overline{Y}_i,$$

avec $\sigma_{k+1} \sqsubseteq 0$.

LEMME 2 : *Si N est une forme normale et si $\mathscr{B} \vdash \tau N$ avec $\tau \sqsupseteq 1$ alors $N \in \mathscr{N}_\omega$.*

Démonstration : Si N est une variable libre la preuve est évidente sinon nous allons montrer que les conditions (*a*) et (*b*) de la définition 11 sont vérifiées.

Condition (*a*) *:* La variable de tête est libre.

$N \equiv \lambda x_1 \ldots \lambda x_n . (z N_1 \ldots N_m)$ par n applications du théorème 1 on trouve que $\tau = (\tau_1) \ldots (\tau_n) \tau_{n+1}$ et que

$$\mathscr{B}, \tau_1 x_1, \ldots, \tau_n x_n \vdash \tau_{n+1} (z N_1 \ldots N_m).$$

$\tau \sqsupseteq 1$ implique que $\tau_{n+1} \sqsupseteq 1$ et $\tau_i \sqsubseteq 0$, $\forall i=1, n$. z doit avoir un type $(\nu_1) \ldots (\nu_m) \tau_{n+1}$ qui ne peut être $\sqsubseteq 0$. z ne peut coïncider avec un quelconque des x_i et est donc libre.

Condition (*b*) : Si une variable de tête z d'un sous-terme $Z \equiv z\, Y_1 \ldots Y_p$ est remplaçable alors toutes les variables de tête des composantes $Y_1 \ldots Y_p$ sont non remplaçables.

Supposons donc que y variable de tête de Y_i soit remplaçable, z étant remplaçable par le lemme 1 il existe une déduction de $\mathscr{B} \vdash \tau\, N$ dans laquelle $\sigma_1\, z$ et $\sigma_2\, y$ avec $\sigma_1 \sqsubseteq 0$ et $\sigma_2 \sqsubseteq 0$. On en déduit que Y_i a un type $\tau_i \sqsupseteq 1$ ce qui est incompatible avec le fait que y ait un type $\sigma_2 \sqsubseteq 0$ [la preuve de cette incompatibilité se fait de manière semblable à la preuve de la condition (*a*)].

Nous omettrons la preuve du lemme 3 semblable à celle du lemme 1.

LEMME 3 : *Si* $N \in \mathscr{N}_\omega$ *alors il existe une déduction de* $\mathscr{B} \vdash 1\, N$.

Il y a donc équivalence entre $\mathscr{N}_\omega$ et l'ensemble des formes normales ayant le type 1. On remarque d'autre part que pour toute forme normale ayant un type $\sqsupseteq 1$ il existe effectivement une déduction lui attribuant le type 1.

PROPOSITION 5 : *Si* M *et* N *sont des formes normales et s'il existe une base telle que* $\mathscr{B} \vdash 1\, N$ *alors* NM *et* MN *possèdent des formes normales.*

Démonstration : $N \in \mathscr{N}_\omega$ et par définition de $\mathscr{N}_\omega$ NM possède une forme normale. En ce qui concerne MN on démontre le résultat par récurrence sur $|M|$:

– $|M|=1$, $MN \equiv x\, N$ en forme normale;

– supposons la propriété vraie $\forall M$, $|M|<n$ et soit M de longueur n alors : si $M \equiv z\, M_1 \ldots M_k$, MN est en forme normale, si $M \equiv \lambda x\, \lambda y_1 \ldots \lambda y_s . z\, M_1 \ldots M_k$ alors $\forall i$, $\lambda x\, M_i$ est une forme normale et $|\lambda x\, M_i|<n$.

Par hypothèse de récurrence $(\lambda x . M_i)\, N$ a une forme normale M_i', $\forall i$.

Deux cas se présentent :

– $z \not\equiv x$, MN a la forme normale $\lambda y_1 \ldots \lambda y_s . z\, M_1' \ldots M_k'$;

– $z \equiv x$, $MN \rhd \lambda y_1 \ldots y_s . N M_1' \ldots M_k'$ or $N \in \mathscr{N}_\omega$;

M_i' sont des formes normales et par k applications de la première partie de la proposition MN se réduit à une forme normale.

6. PROPRIÉTÉS DES EXPRESSIONS POSSÉDANT UN TYPE DIFFÉRENT DE ω

Dans cette partie nous montrons que la propriété pour une λ-expression d'avoir un type distinct de ω est une condition suffisante (mais non nécessaire) pour qu'elle possède une forme normale gauche. La démonstration de cette proposition est faite pour un type calculé par rapport à une base étendue, mais celle-ci est vraie *a fortiori* pour une base normale.

Nous allons maintenant nous intéresser aux expressions F ayant un type dans une base étendue $\mathscr{E}$ et possédant la propriété suivante :

(1) pour toute sous-expression Z de F telle qu'il existe une base étendue $\mathscr{E}' \supset \mathscr{E}$ et $\psi \not\equiv \omega$ tels que $\mathscr{E}' \vdash \psi Z$ alors Z est en forme normale gauche (f. n. g.). Toute sous-expression n'ayant que le type ω est quelconque ainsi que toutes ses sous-expressions).

PROPOSITION 6 : *Si $\mathscr{E} \vdash \tau\,(FX)$, $\tau \not\equiv \omega$ et si F et X sont des λ-expressions qui possèdent la propriété* (1), *alors FX possède une forme normale gauche qui possède la propriété* (1).

Démonstration : D'après le théorème 1 si $\mathscr{E} \vdash \tau\,(FX)$, alors il existe σ tel que $\mathscr{E} \vdash (\sigma)\tau F$ et $\mathscr{E} \vdash \sigma X$. D'autre part $\tau \not\equiv \omega \Rightarrow (\sigma)\tau \not\equiv \omega$.

La preuve du théorème se fait par récurrence sur $\|(\sigma)\|$, $\|\tau\|$ et $F|$ auparavant nous montrons les quatre lemmes suivants.

LEMME 5 : *Si $\mathscr{E} \vdash \tau\,(FX)$, $\tau \not\equiv \omega$ et si F et X sont des expressions qui vérifient* (1), *si $F \equiv z\,G_1 \ldots G_p$ alors FX a une forme normale gauche qui possède la propriété* (1).

Démonstration : $FX \equiv z\,G_1 \ldots G_p X$ forme normale gauche, G_i, $\forall\, i=1, p$ et X vérifient (1) on en déduit que FX vérifie (1).

Dans la suite de la démonstration de la proposition 6 nous ne considèrerons plus que les expressions de la forme $F \equiv \lambda x . \overline{F}$.

LEMME 6 : *Si F et X vérifient les hypothèses de la proposition* 6 *et si $F \equiv \lambda x . \overline{F}$, $|\overline{F}| = 1$ alors FX a une f. n. g. qui vérifie* (1).

Démonstration : $|\overline{F}| = 1$, $\overline{F} \equiv z$ deux cas se présentent :

— $z \not\equiv x$, $FX \rhd z$ forme normale qui vérifie (1);

— $z \equiv x$, $FX \rhd X$ d'après la proposition 2 X est de type $\tau \not\equiv \omega$ et par hypothèse il s'agit donc d'une forme normale gauche qui vérifie (1).

Dans la suite de la démonstration nous ne considèrerons que des expressions $\overline{F}$ de la forme $\lambda y_1 \ldots \lambda y_k . z\ G_1 \ldots G_p$. En appliquant la proposition 1 il existe une base étendue $\mathscr{E}' \equiv \mathscr{E}$, $\varphi_1\ y_1, \ldots, \varphi_k y_k$ telle que

$\tau=(\varphi_1)\ldots(\varphi_k)\psi$ et que $\mathscr{E}', \sigma x \vdash \psi(z\, G_1 \ldots G_p)$. On en déduit également qu'il existe une déduction de $\mathscr{E}'$, $\sigma x \vdash \nu_i G_i$, $\forall i=1, p$ et que le type de z est $(\nu_1)\ldots(\nu_p)\psi$. Remarquons que $\tau \not\equiv \omega \Rightarrow \psi \not\equiv \omega$. D'autre part par la règle R_a il existe une déduction de $\mathscr{E}' \vdash (\sigma)\ \nu_i\ (\lambda x.G_i)$ et de $\mathscr{E}' \vdash \nu_i\ ((\lambda x.G_i)X)$, $\forall i=1, p$. De même on peut montrer que si H est une sous-expression propre de F ayant un type ν dans une base étendue $\mathscr{E}''$, $\sigma x \supset \mathscr{E}$ deux cas se produisent. Ou bien $x \notin H$ et $(\lambda x.H)\ X \triangleright H$ de même type ν ou bien $x \in H$ alors par la règle R_a $\mathscr{E}'' \vdash (\sigma)\nu\ (\lambda x.H)$ et par la règle R_c $\mathscr{E}'' \vdash \nu\ ((\lambda x.H)X)$.

LEMME 7 : *Si F et X vérifient les hypothèses de la proposition* 6 *et si de plus F et X ont un type de longueur* 1 *alors FX a une forme normale.*

Démonstration : $\mathscr{E} \vdash \tau\ (FX)$, $\mathscr{E} \vdash (\sigma)\ \tau\ F$, $\|(\sigma)\ \tau\|=1$, $\tau \not\equiv \omega$ alors soit $(\sigma)\ \tau=0$ soit $(\sigma)\ \tau=1$.

Nous allons montrer que si une expression vérifie (1) et a un type 0 ou 1 alors elle est en forme normale. En effet raisonnons sur la taille d'une expression F :

– $|F|=1$ $F \equiv x$ quel que soit le type de F, elle est en forme normale;

– supposons le résultat vrai pour toute expression de longueur inférieure à n et soit F, $|F|=n$, $F \equiv \lambda x_1 \ldots \lambda x_k.z\, G_1 \ldots G_p$. De par la proposition 1 si F a le type 0 (respect 1) alors G_i, $\forall i$, $i=1, p$ a le type 1 (respect 0). Par hypothèse de récurrence G_i est en forme normale.

On déduit de ce résultat que F et X sont en forme normale et par la proposition 5 que FX a une forme normale.

LEMME 8 : *Si F et X vérifient les hypothèses de la proposition* 6 *et si de plus F a un type de longueur* 2 *alors FX a une forme normale gauche et vérifie* (1) *(donc une forme normale puisque $\tau=0$ ou* 1).

Démonstration : Compte tenu du lemme 7 nous devons démontrer ce lemme dans les cas suivants :

– le type de F est $(\tau)\ \tau$ avec $\tau=0$ ou 1;

– le type de F est $(\omega)\ \tau$ avec $\tau=0$ ou 1.

Raisonnons par récurrence sur la taille de $\overline{F}$ et grâce au lemme 6 supposons la propriété vraie, $\forall \overline{F}$, $|\overline{F}|<m$. Soit alors $\overline{F}$, $|\overline{F}|=m$. F ayant un type $(\sigma)\tau$ par le théorème 1, $\overline{F}$ a le type τ, 0 ou 1. $\overline{F}$ est en forme normale et toute sous-expression propre de $\overline{F}$ a le type 0 ou 1. En particulier G_i d'où l'on déduit qu'il existe une base $\mathscr{E}'$ telle que $\mathscr{E}' \vdash (\sigma)\ \nu_i\ (\lambda x.G_i)$, $\nu_i=0$ ou 1, $\forall_i=1, p$. D'autre part $|\lambda x.G_i|<m$, G_i en forme normale vérifie (1), $\|(\sigma)\ \nu_i\| \leqq 2$ et par hypothèse de récurrence $(\lambda x.G_i)\ X \triangleright G_i'$ où G_i' est une forme normale gauche de type 0 ou 1 vérifiant la condition (1), c'est-à-dire une forme normale.

Deux cas se présentent suivant que :

– $z \not\equiv x$, $FX \triangleright \lambda y_1 \ldots \lambda y_k . z\, G'_1 \ldots G'_p$ forme normale;

– $z \equiv x$, $FX \triangleright \lambda y_1 \ldots \lambda y_k . X\, G'_1 \ldots G'_p$.

Dans ce cas σ le type X est égal au type de z différent de ω. Comme $\|\sigma\| = 1$ on déduit que X a le type 0 ou 1. D'autre part le type de FX est τ, 0 ou 1, il en est de même de XG'_1, $XG'_1\, G'_2$, ..., $XG'_1 \ldots G'_p$. Si X a le type 0 (respectivement 1) alors G'_i a le type 1, $\forall i$, $i = 1$, p. X et G'_1 sont en forme normale et d'après la proposition 5 $XG'_1 \triangleright X_1$ forme normale de même type que X. En appliquant p fois la proposition 5 on montre que FX se réduit à une forme normale.

Démonstration de la proposition 6 : Nous avons montré dans les lemmes précédents que la proposition 6 est vraie pour toutes les expressions F et X vérifiant les hypothèses de la proposition 6 et de plus les trois conditions suivantes :

$\|\sigma\| = 1$, $\|\tau\| = 1$ et toutes les sous-expressions propres de F et X ont un type de longueur inférieure ou égale à $\|\tau\|$ dans une base étendue $\mathscr{E}' \supset \mathscr{E}$.

1re *étape :* Nous montrons par récurrence que le théorème est vrai pour toutes les expressions F et X vérifiant les hypothèses de la proposition 6 et la condition $\|\sigma\| = 1$.

Pour cela supposons qu'il soit vrai dans tous les cas suivants :

– $\|\sigma\| = 1$, $\|\tau\| < k$ et toutes les sous-expressions de F et X ont un type de longueur inférieure à k dans une base $\mathscr{E}' \supset \mathscr{E}$. Montrons la proposition pour $k+1$ par récurrence sur la taille de F;

– $|\overline{F}| = 1$ les conditions d'application du lemme 6 sont vérifiées;

– supposons la proposition vraie pour tout $\overline{F}$, $|\overline{F}| < m$ et soit $\overline{F}$, $|\overline{F}| = m$ alors pour toute sous-expression propre H de G_i on peut écrire que :

$$\mathscr{E}' \vdash (\sigma)\, \nu\, (\lambda x . H), \qquad |\lambda x . H| < m, \qquad \|\nu\| < k+1.$$

Par hypothèse de récurrence si $\nu \not\equiv \omega$ $(\lambda x . H) X \triangleright H'$ f.n.g. qui vérifie (1) et $\mathscr{E} \vdash \nu H$, $\|\nu\| < k+1$. On en déduit que G_i, $\forall i = 1, p$ se réduit en une expression G'_i qui vérifie (1) et dont toute sous-expression propre a un type de longueur inférieure à $k+1$. Deux cas se présentent suivant que :

– $z \not\equiv x$, $FX \triangleright \lambda y_1 \ldots \lambda y_k . z\, G'_1 \ldots G'_p$ f.n.g. qui vérifie (1);

– $z \equiv x$, $FX \triangleright \lambda y_1 \ldots \lambda y_k\, X\, G'_1 \ldots G'_p$.

Dans ce cas le type de X est σ, $\|\sigma\| = 1$, $\sigma = (\nu_i) \ldots (\nu_p)\, \psi \not\equiv \omega$, σ est donc le type 0 ou 1 ainsi que ψ et ν_i, $\forall i = 1, p$. Par un argument semblable à celui développé dans le lemme 8, FX se réduit à une forme normale.

2e *étape :* Montrons la proposition par récurrence sur $\|\sigma\|$.

Supposons que le théorème est vrai pour toutes les expressions F et X telles que $\|\sigma\| \leqq l$ et montrons le pour $l+1$ par récurrence sur $|\overline{F}|$:

– $|\overline{F}|=1$ les conditions d'application du lemme 6 sont vérifiées;

– supposons la proposition vraie pour tout $\overline{F}$, $|\overline{F}|<m$ et soit $\overline{F}$, $|\overline{F}|=m$ alors pour toute sous-expression propre H de G_i on peut écrire que $\mathscr{E}' \vdash (\sigma)\nu(\lambda x.H)$, $|\lambda x.H|<m$, $\|\sigma\| \leqq l+1$ et par hypothèse de récurrence si $\nu \not\equiv \omega$, $(\lambda x.H)X \triangleright H'$ f. n. g. qui vérifie (1). On en déduit que G_i, $\forall i=1, p$ se réduit en une expression G'_i qui vérifie (1). Deux cas se présentent suivant que :

– $z \not\equiv x$, $FX \triangleright \lambda y_1 \ldots \lambda y_k . z\ G'_1 \ldots G'_p$ f. n. g. qui vérifie (1);

– $z \equiv x$, $FX \triangleright \lambda y_1 \ldots \lambda y_k . X\ G'_1 \ldots G'_p$.

X est de type $\sigma=(\nu_1)\ldots(\nu_p)\psi$, X et G'_1 vérifient (1). XG'_1 a pour type $(\nu_2)\ldots(\nu_p)\psi \not\equiv \omega$, $\|\nu_1\|<\|\sigma\| \leqq l+1$ donc $\|\nu_1\| \leqq l$ et par hypothèse de récurrence $XG'_1 \triangleright X_1$ f. n. g. qui vérifie (1). Le même argument appliqué à X_1 et G'_2 permet de conclure que $X_1\, G'_2 \triangleright X_2$ de type $(\nu_3)\ldots(\nu_p)\psi \not\equiv \omega$ et vérifiant (1).

Ainsi de suite jusqu'à $X_{p-1}\, G'_p \triangleright X_p$ de type $\psi \not\equiv \omega$ vérifiant (1) donc en forme normale gauche.

Théorème 2 : *Toute λ-expression ayant un type différent de ω dans une base quelconque a une forme normale gauche de même type qui vérifie la condition* (1).

Démonstration : Par récurrence sur la taille de l'expression. Il suffit de montrer qu'elle a une forme normale gauche vérifiant (1) :

– $|F|=1$, $F \equiv x$ qui est en forme normale;

– supposons la propriété vraie $\forall F$, $|F|<n$ et soit F, $|F|=n$:

si $F \equiv \lambda x.\overline{F}$ alors $|\overline{F}|<n$ et $\overline{F}$ est en f. n. g. ainsi que F et vérifie (1).

si $F \equiv XY$.

Par hypothèse de récurrence toute sous-expression de X ou de Y ayant un type différent de ω dans une base étendue à une f. n. g., X et Y se réduisent donc à deux expressions X' et Y' qui vérifient les hypothèses de la proposition 6. F possède une f. n. g. qui vérifie (1).

7. PROPRIÉTÉ DES EXPRESSIONS POSSÉDANT LE TYPE 0

Nous montrons que la propriété d'avoir le type 0 pour une λ-expression est une condition suffisante (mais non nécessaire) pour qu'elle possède une forme normale.

Pour cela nous allons nous intéresser aux expressions N ayant un type dans une base $\mathscr{B}$ et vérifiant la condition suivante :

(2) Quelle que soit Z sous-expression de N telle qu'il existe une base $\mathscr{B}' \supset \mathscr{B}$ et $\psi \sqsupseteq 0$ tels que $\mathscr{B}' \vdash \psi Z$ alors Z est en forme normale.

PROPOSITION 7 : *Si* $\mathscr{B} \vdash \tau(NX)$, $\tau \sqsupseteq 0$ *et si* N *et* X *vérifient les conditions* (1) *et* (2) *alors* NX *a une forme normale.*

Démonstration : D'après la proposition 1 si $\mathscr{B} \vdash \tau(NX)$ alors il existe σ tel que $\mathscr{B} \vdash (\sigma)\tau N$ et $\mathscr{B} \vdash \sigma X$. La proposition se démontre par double récurrence sur $\|(\sigma)\tau\|$ et $|N|$.

1^{re} *étape :* $\|(\sigma)\tau\| = 1$.

Deux possibilités :

– (σ) $\tau = 1$, N est de type 1 et X de type 0 ils sont donc tous deux en forme normale d'après (2) et NX a une forme normale d'après la proposition 5;

– (σ) $\tau = 0$, N est de type 0 et X de type 1 NX a une forme normale pour les mêmes raisons.

2^e *étape :* On suppose la propriété vraie pour toutes les expressions NX avec N ayant un type ρ, $\|\rho\| < n$ et soit N de type $\|(\sigma)\tau\| = n$. On sait que $\tau \sqsupseteq 0$ donc que $(\sigma)\tau \not\equiv \omega$ c'est-à-dire que N est en forme normale gauche d'après (1). Deux cas se présentent :

(A) $N \equiv z N_1 \ldots N_m$, $NX \equiv z N_1 \ldots N_m X$. z étant une variable libre elle a un type $(\sigma_1) \ldots (\sigma_{m+1}) \sigma_{m+2} \sqsubseteq 1$. On en déduit que $\sigma_i \sqsupseteq 0$, $\forall i = 1, m+1$ et que N_i sous-expression de N, et X ayant des types positifs sont en forme normale d'après (2). NX est en forme normale.

(B) $N \equiv \lambda x . \overline{N}$ nous raisonnons par récurrence sur la taille de $\overline{N}$:

1^{re} *étape :* $|\overline{N}| = 1$, $\overline{N} \equiv z$:

– $z \not\equiv x$, $NX \triangleright z$ forme normale;

– $z \not\equiv x$, $NX \triangleright X$ a le type $\tau \sqsupseteq 0$ et est en forme normale d'après la condition (2).

2^e *étape :* Supposons la proposition vraie $\forall \overline{N}$, $|\overline{N}| < m$ et soit $\overline{N}$, $|\overline{N}| = m$ $\overline{N} \equiv \lambda y_1 \ldots \lambda y_p . (z N_1 \ldots N_k)$. Le type de N étant $(\sigma)\tau$ on en déduit par la règle R_a que $\mathscr{B}, \sigma x \vdash \tau \overline{N}$. D'autre part on peut écrire que $\tau = (\varphi_1) \ldots (\varphi_p) \psi$ et comme $\tau \sqsupseteq 0$ on déduit $\psi \sqsupseteq 0$, $\varphi_i \sqsubseteq 1$, $\forall i$, $i = 1, p$. $\mathscr{B}' = \mathscr{B}, \varphi_1 y_1, \ldots, \varphi_p y_p$ est donc une base et $\mathscr{B}', \sigma x \vdash \psi (z N_1 \ldots N_k)$ d'où z a pour type $(v_1) \ldots (v_k) \psi$ avec $\mathscr{B}'$, $\sigma x \vdash v_i N_i$, $\forall i$, $i = 1, k$. D'après la règle R_a il existe une déduction de $\mathscr{B}' \vdash (\sigma)$ $v_i (\lambda x . N_i)$, $\forall i$, $i = 1, k$.

Deux cas se présentent suivant que :

(a) $z \not\equiv x$, z variable libre ou, z liée a y_i a nécessairement un type inclus dans 1.

On en déduit que $\nu_i \sqsupseteq 0, \forall i, i=1, k$ que $(\lambda x.N_i)X$ a le type $\nu_i \sqsupseteq 0$ dans une base $\mathscr{B}'$ et que $|\lambda x.N_i|<m$. D'autre part $(\lambda x.N_i)$ ayant le type $(\sigma)\,\nu_i$ il existe une déduction de $\mathscr{B}' \vdash (\sigma)0(\lambda x.N_i)$ et $\|(\sigma)0\| \leqq \|(\sigma)\tau\|$. D'après l'hypothèse de récurrence $(\lambda x.N_i)X$ se réduit à une forme normale N_i' et NX a la forme normale $\lambda y_1 \ldots \lambda y_p.z N_1' \ldots N_k'$.

(*b*) $z \equiv x$ alors $NX \triangleright \lambda y_1 \ldots \lambda y_p.X\,N_1' \ldots N_k'$ avec $N_i' \equiv [X/x]N_i$ de type $\nu_i, \forall i, i=1, k$. D'autre part le type σ est égal au type de z, $\sigma=(\nu_1)\ldots(\nu_k)\psi$ avec $\psi \sqsupseteq 0$ et $\sigma \not\equiv \omega$.

Considérons alors n'importe qu'elle sous-expression propre K de N_i telle qu'il existe une base $\mathscr{B}'' \supset \mathscr{B}$, $\mathscr{B}''$, $\sigma x \vdash \tau_k K$ et $\tau_k \sqsupseteq 0$. On peut alors écrire que $\mathscr{B}'' \vdash (\sigma)\tau_k(\lambda x.K)$ et que $\mathscr{B}'' \vdash (\sigma)0(\lambda x.K)$. Enfin K sous-expression de N_i et X vérifient les conditions (1) et (2) et sont tels que $|\lambda x.K|<m$ et $\|(\sigma)0\|<\|(\sigma)\tau\|$ ce qui permet d'appliquer l'hypothèse de récurrence et d'affirmer que $(\lambda x.K)X$ a une forme normale. Le même raisonnement appliqué à toutes les sous-expressions propres K de N_i ayant un type différent de ω suffit à prouver que $(\lambda x.K)X$ possède une forme normale gauche. On en déduit que N_i' se réduit à une expression N_i'' vérifiant les conditions (1) et (2).

Il suffit maintenant de montrer que $((XN_1'')N_2'' \ldots N_k'')$ de type $\psi \sqsupseteq 0$ possède une forme normale. Pour cela nous montrerons sucessivement que XN_1'', $XN_1''N_2''$, ..., $XN_1'' \ldots N_k''$ peuvent se réduire à des expressions qui vérifient les conditions (1) et (2). La dernière de ces expressions ayant un type positif sera donc en forme normale.

XN_1'' a le type $(\nu_2)\ldots(\nu_k)\psi$ et X le type $\sigma \not\equiv \omega$ dans la base $\mathscr{B}'$.

X est donc en forme normale gauche [condition (1)] et deux cas se présentent :

– $X=u\overline{X}$, $XN_1'' \equiv u\overline{X}\,N_1''$, $\overline{X}$ et N_1'' vérifient les conditions (1) et (2) et il en est de même pour XN_1''. Posons $X_1=XN_1''$ et remarquons que $XN_1'' \ldots N_k''$ vérifiera également (1) et (2);

– $X=\lambda y.\overline{X}$. On en déduit d'après la proposition 1 que y a le type ν_1. Soit alors K une sous-expression propre de $\overline{X}$ telle qu'il existe une base $\mathscr{B}_1 \supset \mathscr{B}'\,\mathscr{B}_1$, $\nu_1 y \vdash \tau_k K$ avec $\tau_k \sqsupseteq 0$. On peut alors écrire que $\mathscr{B}_1 \vdash (\nu_1)\tau_k(\lambda y.K)$ et que $\mathscr{B}_1 \vdash (\nu_1)0(\lambda y.K)$. K sous-expression de X vérifie les conditions (1) et (2), $\|(\nu_1)0\|<n$ puisque

$$\|\nu_1(0)\| \leqq \|(\tau_1)\ldots(\nu_k)\psi\| = \|\sigma\| < \|(\sigma)\tau\|,$$

$(\lambda y.K)\,N_1''$ possède donc une forme normale par hypothèse de récurrence.

Un argument similaire prouve que toute sous-expression propre K de $\overline{X}$ ayant un type différent de ω est telle que $(\lambda y.K)\,N_1''$ possède une forme normale gauche.

$[N_1''/y]\overline{X}$ peut donc se réduire à une expression X_1 qui vérifie les conditions (1) et (2).

On peut recommencer ce raisonnement pour $X_1 N_2''$ et déterminer X_2 vérifiant les conditions (1) et (2) puis déterminer de même $X_3 \ldots X_k$. X_k de type $\psi \sqsupseteq 0$ est donc en forme normale.

THÉORÈME 3 : *Si F est une expression quelconque et s'il existe une base $\mathscr{B}$ telle que $\mathscr{B} \vdash \tau F$, $\tau \sqsupseteq 0$ alors F a une forme normale F' et $\mathscr{B} \vdash \tau F'$.*

Il suffit de montrer la première partie par récurrence sur $|F|$.

Démonstration : Si $|F|=1$ alors $F \equiv x$ en forme normale. Supposons la propriété vraie, $\forall F$, $|F|<n$, et soit F, $|F|=n$.

Deux cas se présentent :

— $F \equiv \lambda x.\overline{F}$ alors d'après la proposition 1 il existe $(\sigma_1)\,\sigma_2 = \tau$ tel que $\mathscr{B}$, $\sigma_1 x \vdash \sigma_2 \overline{F}$. Or $(\sigma_1)\sigma_2 \sqsupseteq 0$, $\sigma_1 \sqsubseteq 1$ et $\sigma_2 \sqsupseteq 0$. Il existe donc une base $\mathscr{B}' \vdash \sigma_2 F$, $\mathscr{B}' = \mathscr{B} \cup \{\sigma_1 x\}$, $\sigma_2 \sqsupseteq 0$ et $|\overline{F}|<n$. Par hypothèse de récurrence $\overline{F}$ a une forme normale;

— $F \equiv XY$. Pour toute sous-expression K de X ou de Y ayant un type $\tau_k \sqsupseteq 0$ (resp. $\tau_k \not\equiv \omega$) dans une base $\mathscr{B}' \supset \mathscr{B}$, K est tel que $|K|<n$ et a une forme normale (resp. une forme normale gauche). X et Y vérifient les conditions (1) et (2) et d'après la proposition 7 XY a une forme normale.

8. ENSEMBLE DES EXPRESSIONS TYPABLES

Dans les paragraphes précédents il a été montré que toute expression ayant le type 0 a une forme normale et que toute expression ayant un type distinct de ω a une forme normale gauche. Nous montrerons ici que les expressions de $\mathscr{N}_\omega$ ne possèdent que le type ω. Les réciproques de ces trois propriétés sont fausses et nous donnerons des contre-exemples.

L'équivalence entre $\mathscr{N}_\omega$ et les formes normales de type 1 a été prouvée et il est aisé de constater que les expressions F ayant le type $\underbrace{(\omega)\ldots(\omega)}_{n \text{ fois}}1$ ont la propriété que $FX_1 \ldots X_n \in \mathscr{N}_\omega$ pour $X_1 \ldots X_n$ quelconques. D'autre part les expressions possédant le type 1 ont également le type 0 par la règle R_i. Elles ont donc une forme normale qui appartient à $\mathscr{N}_\omega$.

Par la suite nous utiliserons les abréviations suivantes :

$$I \equiv \lambda x.x, \qquad K \equiv \lambda x \lambda y.x, \qquad \Delta \equiv \lambda x.xx.$$

LEMME 9 : *$\forall \tau \in T$ il existe une λ-expression F ayant ce type :*

– $\tau \equiv \omega$, $F \equiv \Delta\Delta$;

– $\tau \not\equiv \omega$, *alors* $\tau = (\sigma_1) \ldots (\sigma_n)\, \sigma_{n+1}$ *où* σ_{n+1} *est un type élémentaire* 0 *ou* 1 *et* $(\sigma_1) \ldots (\sigma_n)\, \sigma_{n+1} \sqsubseteq \underbrace{(\omega) \ldots (\omega)}_{n \text{ fois}} 1$.

La λ-expression $\lambda x_1 \ldots \lambda x_n . z$ a le type $(\omega) \ldots (\omega) 1$ et par la règle R_i le type τ.

PROPOSITION 8 : *Si une expression F appartient à $\mathcal{N}_{-\omega}$ alors elle n'a que le type ω.*

En effet supposons qu'elle ait un type $\tau \not\equiv \omega$ alors $\tau = (\sigma_1) \ldots (\sigma_n)\, \sigma_{n+1}$ où σ_{n+1} est un type élémentaire 0 ou 1.

Donc il existe n expressions $X_1 \ldots X_n$ de type $\sigma_1 \ldots \sigma_n$ telle que $FX_1 \ldots X_n$ est de type $\sigma_{n+1} \sqsupseteq 0$ et qui possède une forme normale ce qui est contraire à l'hypothèse.

Nous donnons maintenant quelques exemples qui illustrent les possibilités d'affectation de types aux λ-expressions et qui montrent que la théorie que nous avons présentée ne satisfait pas les conditions que nous nous étions fixées à l'avance.

Exemple 10 : L'exemple 7 $(\lambda x . x (\Delta\Delta)) (\lambda y . z)$ a montré que cette expression a le type 1 donc appartient à $\mathcal{N}_\omega$ alors que dans [6] et [7] cette expression n'a pas de type. L'introduction du type ω a permis de rendre compte de ce que l'on pourrait appeler en logique combinatoire « l'effet cachant » de certains combinateurs comme K.

Exemple 11 : $(\lambda x . (xKz (\Delta\Delta)) (x (KI) (\Delta\Delta) z)) I$.

Cette expression a pour forme normale zz et n'a pas de type ce qui fournit un contre-exemple aux réciproques des théorèmes 2 et 3.

En fait l'analyse de cette expression prouve que l'expression équivalente $((\lambda x . (xKz (\Delta\Delta))) I)\ ((\lambda x . (x (KI)\ (\Delta\Delta) z)) I)$ possède elle un type et que lors de l'affectation les deux occurences de la variable x ont des types distincts compatibles avec l'ensemble des types de I mais incompatibles (incomparables) entre eux. Dans la première expression les deux occurences de x étant liées par le même λ un seul type peut leur être affecté et c'est ω. Dans la deuxième expression chaque x a un type particulier. Des résultats semblables sont obtenus si on considère l'opérateur de point fixe $Y = \lambda f . ((\lambda x . f (xx))\ (\lambda x . f (xx)))$. Le type le plus général qui peut être affecté à Y est $((\omega)\tau)\tau'$, $\forall \tau$ et τ', tels que $\tau' \sqsubseteq \tau$. L'argument de Y est une fonction de type $(\omega)\tau$ c'est-à-dire une fonction constante par rapport à son premier argument.

Exemple 12 : $\lambda x \lambda y . y$ a pour type $(\omega)\,(0)\,0$; Y a pour type $((\omega)\,(0)\,0)\,(0)\,0$ et d'après R_c: $Y(\lambda x \lambda y . y)$ à pour type $(0)\,0$. De fait $Y(\lambda x \lambda y . y)$ se réduit en $\lambda y . y$ forme normale.

Exemple 13 : Nous noterons par $\bar{n}$, $\overline{\text{pred}}$, $\overline{\text{zéro}}$ les λ-expressions classiques représentant respectivement l'entier reconstruit, la fonction prédecesseur, le prédicat d'égalité à $\bar{0}$ à valeur dans $\{\lambda xy.x,\ \lambda xy.y\}$ et utilisées dans le langage CUCH [2].

L'expression $F = \lambda f \lambda x . (((\overline{\text{zero}}\ x)\,\bar{1})\,(f\,(\overline{\text{pred}}\ x)))$ correspond à la fonctionnelle récursive $\tau[f] \equiv$ si $x = 0$ alors 1 sinon $f(x-1)$.$(YF)\,\bar{n}$ a la forme normale $\bar{1}$ quel que soit n. Or seule $(YF)\,\bar{0}$ possède un type différent de ω.

En n réductions $(YF)\ \bar{n}$, (a) peut se mettre sous la forme $(F(F(\ldots(F(YF))\ldots)))\,\bar{n}$, (b) où F apparaît $n+1$ fois. On montre que (b) a la forme normale $\bar{1}$ obtenue par une suite de réductions distinctes de celle du radical YF. On montre que (b) a un type $\sqsupseteq 0$ et l'analyse de la déduction de ce type fait apparaître que les différentes occurrences de F dans (b) ont des types $\tau_1, \ldots, \tau_n$ distincts et non ordonnés par la relation $\sqsubseteq$. Il n'est donc pas possible d'affecter dans (a) un type τ à F qui soit plus grand que tous les τ_i ce qui explique que (a) n'a que le type ω.

Nous pouvons en conclure qu'il ne suffit pas de remplacer l'hypothèse de stratification « toute expression a un type et un seul » par l'hypothèse « toute expression qui a un type, a les types qui lui sont contenus » pour pouvoir affecter un type à toute expression ayant une forme normale. Cette constatation nous a amené à étendre la théorie des types en considérant pour chaque variable ou sous-expression un ensemble fini de types d'une manière analogue à celle définie dans [7]. Cette extension nous permet de généraliser les résultats de [7] sur le λ-I-calcul au λ-calcul général et fera l'objet de la deuxième partie de l'article qui sera publiée dans cette même revue.

BIBLIOGRAPHIE

1. H. P. Barendregt, *Some Extensional Term Models for Combinatory Logics and λ-calculi*, Ph. D. Thesis, Utrecht University, 1971.
2. C. Böhm, *The CUCH as a Formal and Description Language*, *in* Formal Language, Description Languages for Computer Programming, T. B. Steele, Jr, éd., 1966, p. 179-197, North Holland, Amsterdam.
3. C. Böhm et M. Dezani-Ciancaglini, *Lambda-Terms as Total or Partial Function on Normal Forms, in Lambda Calculus and Computer Science Theory*, C. Böhm, éd., Lecture Notes in Computer Science n° 37, 1975, p. 96-121, Springer-Verlag.

4. C. BÖHM et M. DEZANI-CIANCAGLINI, *Termination Test Inside λ-Calculus,* Automata Languages and Programming, (ICALP'77) A. SALOMAS, éd., Lecture Notes in Computer Science, n° 52, 1977, p. 95-110, Springer-Verlag.
5. H. B. CURRY, J. R. HINDLEY et J. P. SELDIN, *Combinatory Logic*, Amsterdam, North Holland, vol. II 1972.
6. M. COPPO et M. DEZANI-CIANCAGLINI, *A proposal for a New Type Assigment for λ-terms*, Rapport Interne, Université de Turin, 1976.
7. M. COPPO et M. DEZANI-CIANCAGLINI, *A Generalized Type Theory for λ-calculus*, Rapport Interne, Université de Turin, 1977.
8. M. COPPO et M. DEZANI-CIANCAGLINI, *A New Type Assigment for λ-terms, in* Archiv für Math. Logik und Grundlagenforschung, 19, 1978, p. 1-17.
9. M. COPPO, M. DEZANI-CIANCAGLINI et P. SALLE, *Functional Characterisation of Some Semantic Equalities Inside λ-Calculus*, Automata Languages and Programming, (ICALP'79), E. MAURER, éd. Lecture Notes in Computer Science n° 71, 1979, p. 133-146, Springer-Verlag.
10. J. P. LANDIN, *A Correspondence Between Algol*-60 *and Church's λ-notation*, C.A.C.M., vol. 8, 1965, p. 89-101 et 158-165.
11. J. H. MORRIS, *Lambda Calculus Models of Programming Languages*, Ph. D. Thesis, M.I.T., 1968.
12. L. NOLIN, *Les modèles informatiques des λ-Calculs*, *in* λ-Calculus and Computer Science Theory, C. BÖHM, éd., Lecture Notes in Computer Science, n° 37, Springer-Verlag, 1975, p. 166-176.
13. B. ROBINET, *Contribution à l'étude des réalités informatiques*, Thèse de Doctorat, Paris, 1974.
14. B. ROBINET et F. NOZICK, *Sémantique des structures de contrôle*, R.A.I.R.O. Informatique théorique, vol. 11, n° 1, 1977, p. 63-74.
15. B. ROBINET, *Un modèle fonctionnel des structures de contrôle*, R.A.I.R.O. Informatique théorique, vol. 11, n° 3, 1977, p. 213-236.
16. L. E. SANCHIS, *Types of Combinatory Logic*, Notre-Dame Journal of Formal Logic, (5), 1964, p. 161-180.
17. P. SALLE, *Types et étiquettes dans le λ-calcul* (à paraître).
18. P. SALLE, *La notion de types en λ-calcul*, Groupe Programmation et Languages A.F.C.E.T., Bulletin n° 3, 1978 p. 61-77.
19. P. SALLE, *Une extension de la théorie des types en λ-calcul*, *in* Automata, Languages and Programming (ICALP'78), Lecture Notes in Computer Science, n° 62, 1978, p. 398-410, Springer-Verlag.
20. P. SALLE et J. L. DURIEUX, *L'échappement comme sémantique des structures de contrôle*, Actes du Congrès A.F.C.E.T. TTI, Gif/Yvette, novembre 1978, p. 77-87.
21. C. P. WADSWORTH, *The Relation Between Lambda Expressions and Their Denotations in Scott's Models for the λ-Calculus*, Séminaire I.R.I.A., 1974.