UNE EXTENSION DE LA THEORIE DES TYPES EN λ-CALCUL

Patrick SALLE

Laboratoire Langages et Systèmes Informatiques
Equipe de Recherche Associée au CNRS

UNIVERSITE PAUL SABATIER
118, Route de Narbonne

31077 - TOULOUSE - CEDEX

Introduction

Nous proposons tout d'abord une théorie qui permet l'affectation d'un type à des expressions sans forme normale (14). Nous montrons dans cette théorie qu'une expression admettant un type possède suivant la valeur de ce type une forme normale ou une forme normale gauche. Ensuite nous étendons cette théorie en affectant à chaque expression un ensemble fini de types. Cette extension rend possible l'affectation d'un type significatif à toutes les expressions solvables (BARENDREGT (1)) c.a.d. ayant une forme normale gauche. Nous montrons que deux expressions α-β-η convertibles ont le même ensemble de types. Nous déduisons de cette propriété l'équivalence pour une expression entre avoir un type et posséder suivant la valeur de ce type une forme normale ou une forme normale gauche. La décidabilité de l'affectation n'est montrée que pour les expressions en forme normale.

L'intérêt d'une théorie des types associée à un système formel (où à un langage de programmation) est de fournir un moyen de vérifier statiquement qu'une expression du système, syntactiquement bien formée, vérifie également certaines propriétés sémantiques. CURRY (5) reconstruit la logique mathématique à partir de la logique combinatoire. La théorie des types qu'il propose lui permet d'éliminer les expressions sans type c'est-à-dire sans signification. Depuis le λ -calcul s'est révélé un bon formalisme pour exprimer la sémantique des langages de programmation (BOHM (2), LANDIN (8), MORRIS (9), NOLIN (10), ROBINET (11)). Tout programme peut être traduit de manière automatique en une λ expression. 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 les 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 par :

(I) un ensemble de types de base fini ou dénombrable
et les deux règles de construction

(II) chaque type de base est un type

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

Suivant les théories un choix particulier est effectuë 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 (9), SANCHIS (12) 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 (9) 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 $\subseteq$ qui structurent l'ensemble des types et qui sont justifiées 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

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

a') chaque variable libre à un type unique. (Toute occurence de cette variable a donc tous les types inclus dans le type de la variable)

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. Deux expressions du λ-calcul 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 nous plaçons dans le λ-calcul général et nous introduisons un type de base supplémentaire ω de caractère universel.

Après avoir étudié la structure de l'ensemble engendré par les types de base 0, 1, ω , nous définissons les règles d'affectation et nous donnons les résultats principaux sur les propriétés des expressions typées (Théorêmes 6, 7 et 8). Après avoir analysé les limitations de cette théorie sur des exemples nous présentons une extension dont les résultats principaux (Théorèmes 9 et 12) sont appliqués au combinateur Y .

Tous les résultats sont obtenus sur le λ-calcul pur où les variables libres sont considérées comme des constantes sans signification sémantique et auxquelles peuvent être affectés tous les types inclus dans le type 1. Pour obtenir une théorie appliquée nous pensons introduire des δ-règles et par voie de conséquence des types de base supplémentaires.

1 DEFINITIONS ET NOTATIONS

Définition 1 : $\mathcal{N}_\omega$ est l'ensemble des formes normales telles que si $N \in \mathcal{N}_\omega$, $\forall$ n entier, $\forall$ $X_1, \ldots\ldots$ Xn formes normales $(\ldots((NX_1)X_2)\ldots Xn)$ 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, $\forall X_1, \ldots\ldots$ Xn quelconques,

$(\ldots((NX_1)\ X_2)\ldots\ldots\ldots Xn)$ n'a pas de forme normale

C'est l'ensemble des termes insolvables (Barendregt (1) , Wadsworth (12))

Définition 3 : une forme normale gauche est une expression de la forme $\lambda x_1 \ldots\ldots \lambda x_n$ $z M_1 \ldots . M_p$ avec $n \geq o$, $p \geq o$ et M_i une λ expression quelconque $\forall_i$. (z peut être identique à l'un des x_i)

Notation : l'expression F possède le type τ sera noté par τF

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

2 L'ENSEMBLE DES TYPES T

Il est construit à partir de trois types de base $0,1,\omega$ dont nous donnons les motivations sémantiques

2.1 Les types de base

. ω est le type universel que possède toute λ expression (th 1)

. 0 toute forme normale a le type 0. D'autre part si 0F alors F a une forme normale (th 2 et 8)

. 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 $\mathcal{N}_\omega$ (th 4 et 5)

Axiomes d'équivalence

Ao : $0 = (1)\ 0$ 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 corrrespond à la définition de $\mathcal{N}_\omega$

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

l'emploi du type 1 se justifie

- par sa complémentarité avec le type 0 et son rôle dans la définition d'une base (paragraphe 3)
- par le fait que dans un 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 (I), (II) et (III) à partir des types de base $0,1,\omega$

Définition 4 : 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 Ao, A_1 et A_ω .

Exemple 1 ((0)ω) (0) 1 et (ω) ((0)(0) 1) sont des types et sont équivalents à (ω) 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 minimum (nb. minimum d'occurences de 0, 1, ω)

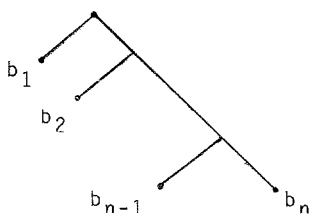
Définition 5 : La longueur d'un type τ , notée $\|\tau\|$ est la longueur du représentant de longueur minimum de la classe d'équivalence de τ

Exemple 2 : $\|((0)\ 0)\ 1\| = 3$, $\|((0)\ 1)\ 0\| = 1$ et $\|(\tau_1)\ldots(\tau_n)\omega\| = 1$

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

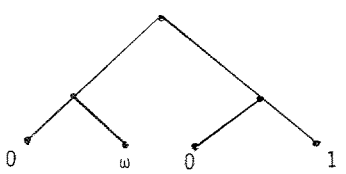
$$\tau = (\tau_1)(\tau_2)\ldots(\tau_{n-1})\tau_n$$

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



où b_i désigne le sous arbre associé à τ_i dont les feuilles sont libellées par les types de base.

Exemple 3 : Le type ((0) ω) (o) 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 6 : 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

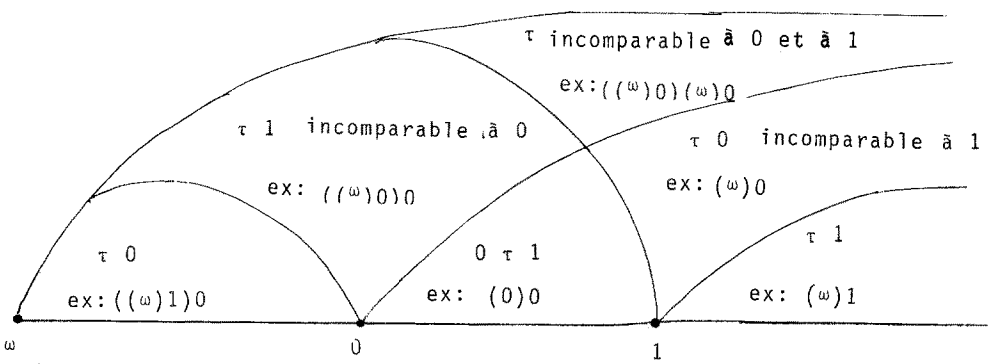
(P6) Les relations $=, \sqsubset, \sqsubseteq, \neq$ sont toujours décidables

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 .

Définition 7 : une base $\mathcal{B}$ est un ensemble d'affectations de la forme σx où 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 8 : une base étendue $\mathcal{E}$ 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.

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

Le système de déduction

Axiome Ap : si $\mathcal{B}$ est une base et si $\sigma x \in \mathcal{B}$ alors $\mathcal{B} \vdash \sigma x$

Règle Rc : si $\mathcal{B} \vdash (\sigma)\tau F$ et $\mathcal{B} \vdash \sigma G$ alors $\mathcal{B} \vdash \tau(FG)$

Règle Ra : si $\mathcal{B}, \sigma x \vdash \tau F$ et si x n'apparaît pas dans $\mathcal{B}$ alors $\mathcal{B} \vdash (\sigma)\tau(\lambda x.F)$

Règle Rc : si $\mathcal{B} \vdash \sigma F$ et si $\sigma = \tau$ alors $\mathcal{B} \vdash \tau F$

Règle Ri : si $\mathcal{B} \vdash \sigma F$ et si $\tau \sqsubset \sigma$ alors $\mathcal{B} \vdash \tau F$

Les règles Ap, Rc et Ra sont classiques en théorie des types et les règles Rc et Ri 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 $\mathcal{E}$ sont les mêmes que par rapport à une base $\mathcal{B}$. L'introduction des bases étendues est rendue nécessaire par la remarque suivante. Lors de l'application de la règle Ra où $\mathcal{B}, \sigma x \vdash \tau F$ σ peut être quelconque. Le type de F est donc un type défini par rapport à une base étendue $\mathcal{E} = \mathcal{B} \cup \sigma x$ alors que celui de $\lambda x.F$ est défini par rapport à la base $\mathcal{B}$. Plus généralement dans une expression F typée par rapport à une base $\mathcal{B}$ les variables libres de F ont un type défini dans $\mathcal{B}$ 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 $C(\ )$ 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 $\mathcal{E} = \mathcal{B} \cup \sigma_1 x_1, \ldots, \sigma_n x_n$ mais ce type n'a de signification pour H que dans le contexte $C(\ )$.

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

$$\left.\begin{array}{l} \text{Rc} \quad \dfrac{(\sigma)\tau\, x \vdash (\sigma)\tau\, x}{} \\ \text{Ra} \quad \dfrac{(\sigma)\tau\, x \vdash \tau(x\ x)}{\vdash ((\sigma)\tau)\tau \quad (\lambda x.\ x\ x)} \end{array}\right\} \text{Si } (\sigma)\tau \sqsubset \sigma$$

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 $((o)o)\ o$ qui contiennent le type O. Δ est en forme normale.

Exemple 6 : On déduit d'après Rc 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.

$$\text{Rc}\ \frac{(\omega)\tau\ \ y \vdash (\omega)\tau\ \ y\ ,\ \ \omega\ (\Delta\Delta)}{}$$

$$\text{Ra}\ \frac{(\omega)\tau\ \ y \vdash \tau\ (y\ (\Delta\Delta))}{\vdash ((\omega)\tau)\tau\ \ (\lambda\ y.\ (y\ (\Delta\Delta)))}$$

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 O.

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

$$\text{Ra}\ \frac{\sigma z \vdash \sigma z}{}$$

$$\text{Ra}\ \frac{\vdash (\sigma)\sigma\ (\lambda z.z)}{}$$

$$\text{Rc}\ \frac{\vdash (\omega)(\sigma)\sigma\ (\lambda x \lambda z\ .z), \vdash ((\omega)(\sigma)\sigma)(\sigma)\sigma A}{\vdash\ (\sigma)\sigma\ \ \ A(\ \lambda x \lambda z.z)}$$

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

4 Résultats principaux

Nous montrons d'abord par recurrence sur la taille des expressions les théorêmes suivants.

Théorême 1 : Toute expression a le type ω
Il justifie le caractère universel de ω

Théorême 2 : Toute forme normale a le type O si on affecte le type 1 à ses variables libres

Théorême 3 : Si N est en forme normale alors $\mathcal{B} \vdash_\tau N$ est décidable .

Grâce à ces résultats de base nous avons étudié les propriétés des expressions ayant un type supérieur ou égal a 1 qui se résument par

Théorême 4 : Si $N \in \mathcal{K}_\omega$ alors $\exists\, \mathcal{B}$ telle que $\mathcal{B} \vdash 1\ N$

Théorême 5 : Si $\mathcal{B} \vdash \tau\ N$, $\tau \sqsupseteq 1$ et N en forme normale alors $N \in \mathcal{K}_\omega$

La preuve de ce théorême se fait par recurrence sur la structure de l'expression en considérant l'ensemble des variables remplaçables de N (voir définition dans(3)). On montre également que

Théorême 6 : $\forall$ M , N formes normales si $\exists \mathcal{B}$ telle que $\mathcal{B} \vdash 1\ N$ alors MN et N M ont des formes normales.

La preuve est faite grâce à la définition et aux propriétés de $\mathcal{K}_\omega$

Nous étudions ensuite les propriétés des expressions ayant un type distinct de ω

Théorême 7 : Toute expression ayant un type distinct de ω a une forme normale gauche.

La preuve de ce théorême est très longue et se fait par inductions successives sur la taille des expressions et sur la taille des types de toutes les sous expressions. On considère d'abord les combinaisons de deux expressions en forme normale gauche dont toutes les sous expressions ayant un type distinct de ω sont également en forme normale gauche. On généralise ensuite le résultat à des expressions quelconques. A partir du théorême 7 on peut montrer que

Théorême 8 : Toute expression ayant un type $\tau \sqsupseteq 0$ a une forme normale.

La preuve de ce théorême se fait également par double induction sur les tailles des expressions et de leurs types. On considère dans un premier temps la combinaison de deux expressions telles que toutes leurs sous expressions ayant un type $\sqsupseteq 0$ est en forme normale, puis on généralise le résultat à toutes les expressions. Etant donné la taille des démonstrations il n'est pas possible d'indiquer les schémas des preuves.

Remarques : Les réciproques des Théorêmes 7 et 8 sont fausses.

On montre que tout type correspond à au moins une expression.

Nous analysons les limitations de cette théorie en étudiant le type de l'opérateur de point fixe $Y = \lambda f . ((\lambda x. f\ (x\ x))\ (\lambda x. f\ (x\ x)))$

Le type le plus général qui peut être affecté à Y est $((\omega)\tau)\tau'$ $\forall$ τ et τ', tels que $\tau' \sqsubseteq \tau$. L'argument de Y est une fonction de type $(\omega)\ \tau$ c.a.d. une fonction constante par rapport à son premier argument.

Exemple 9 : $\lambda x\ \lambda y. y$ a pour type $(\omega)\ (o)\ o$

Y a pour type $((\omega)\ (o)\ o)\ (o)\ o$ et d'après Rc

Y $(\lambda x\ \lambda y. y)$ à pour type $(o)\ o$. De fait Y $(\lambda x \lambda y . y)$ se réduit en $\lambda y . y$ forme normale.

Exemple 10 : Nous noterons par $\overline{n}$, $\overline{pred}$, $\overline{zero}$ les λ expressions classiques représentant l'entier reconstruit, la fonction predecesseur, le prédicat

d'égalité à $\overline{0}$ à valeur dans $\{\lambda x \lambda y . x , \lambda x \lambda y . y\}$ et utilisées dans le langage CUCH (2)

L'expression F = $\lambda f \lambda x . (((\overline{Zero}\ x) \overline{1}) (f (\overline{pred}\ x)))$ correspond à la fonctionnelle recursive

$T[f] \equiv$ si $x = o$ alors 1 sinon $f(x - 1)$.

$(Y F) \overline{n}$ a la forme normale $\overline{1}$ quelque soit n . Or seule $(Y F) \overline{o}$ possède un type différent de ω

En n réductions $(Y F) \overline{n}$, (a) peut se mettre sous la forme $(F (F (\ldots . (F (Y F)) \ldots .))) \overline{n}$, (b) ou F apparait $n + 1$ fois. On montre que (b) a la forme normale $\overline{1}$ obtenue par une suite de réductions distinctes de celle du radical Y F. On montre que (b) a un type $\sqsupseteq 0$ et l'analyse de la déduction de ce type fait apparaître que les différentes occurences 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 . Cette constation 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).

5 Extension de la théorie des types

Nous présentons l'ensemble des types, sa structure, les règles d'affectation aux expressions puis les résultats principaux et leur application à un exemple.

5-1 Ensemble des types

Il est défini à partir des types de base et des opérations de composition et de construction de séquences.

Définition 9 : T' est le plus petit ensemble défini par

- 0 , 1, ω appartiennent à T'
- si $\sigma_1, \ldots, \sigma_n, \tau \in T'$ alors $[\sigma_1, \ldots, \sigma_n]\tau \in T'$

$[\sigma_1, \ldots, \sigma_n]$ est appelée une séquence et possède les propriétés d'un ensemble mathématique id est l'indépendance par rapport à l'ordre d'écriture de ses éléments et au nombre de représentants d'un même élément.

Exemple 11 : $[\omega] [0 , 1 , 0] 0$ et $[\omega] [1 , 0] 0$ sont deux écritures du même type

Nous introduisons les trois axiomes d'équivalence

A'_o : $[1]\ 0 = 0$

A'_1 : $[0]\ 1 = 1$

A'_ω ; $[\bar{\tau}]\ \omega = \omega$ où $\bar{\tau}$ est une abréviation pour $\tau_1, \ldots\ldots, \tau_n$

La justification sémantique de ces axiomes est la même que celle de A_o , A_1 , A_ω . De même les propriétés (P1) , (P2) , la définition de la longueur d'un type $\| \quad \|$ et la propriété (P4) restent valables. (P3) devient (P'3) $\forall \tau \in T', \forall n\ , \exists \bar{\tau}_1, \ldots\ldots\ldots, \bar{\tau}_{n-1}$ séquences et $\tau_n \in T'$ tels que $\tau = [\bar{\tau}_1] \ldots\ldots [\bar{\tau}_{n-1}]\ \tau_n$

Nous définissons les ordres partiels $\sqsubseteq$ et $\subseteq$ entre types et séquences par

- $\omega \sqsubseteq 0 \sqsubseteq 1$

- $[\sigma_1, \ldots\ldots, \sigma_n] \subseteq [\tau_1, \ldots\quad, \tau_m]$ si et seulement si
  $\forall i (1 \le i \le n), \exists j\ (1 \le j \le m)$ tel que $\sigma_i \sqsubseteq \tau_j$

- $[\tau_1, \ldots\quad, \tau_m]\tau_{m+1} \sqsubseteq [\sigma_1, \ldots, \sigma_n]\sigma_{n+1}$ si et seulement si
  $\tau_{m+1} \sqsubseteq \sigma_{n+1}$ et $[\sigma_1, \ldots\quad, \sigma_n] \subseteq [\tau_1, \ldots\quad, \tau_m]$

On montre que les propriétés (P5) à (P8) sur la decidabilité des relations et la structure de treillis de l'ensemble des types restent vraies .

5-2 Règles d'affectation

Une base $\mathcal{B}$ est un ensemble d'affectations du type $\sigma\ x$ ou x est une variable et σ est un type $\sqsubseteq 1$. Plusieurs types peuvent être affectés à la même variable. Si $\bar{\sigma} = \sigma_1, \ldots\ldots, \sigma_n$ alors $\bar{\sigma}\ x$ est une abréviation pour $\sigma_1\ x\ ,\ \sigma_2\ x\ , \ldots\ \sigma_n\ x$. Le système d'affectation se déduit du système décrit au paragraphe 3 de la manière suivante.

- l'axiome Ap et les règles Re et Ri sont inchangées

- les règles Rc et Ra sont remplacées par

Règle R'c : Si $\mathcal{B} \vdash [\bar{\sigma}]\tau\ X$ où $\bar{\sigma} = \sigma_1, \ldots\ldots, \sigma_n$ et si $\forall i\ (1 \le i \le n)\ \mathcal{B} \vdash \sigma i\ Y$ alors $\mathcal{B} \vdash \tau\ (X\ Y)$

Règle R'a : Si $\mathcal{B}$, $\bar{\sigma}\ x \vdash \tau\ X$ et si x n'apparait pas dans $\mathcal{B}$ alors $\mathcal{B} \vdash [\bar{\sigma}]\tau(\lambda x.X)$

Nous introduisons également la notion de base étendue $\mathcal{E}$ et les remarques du paragraphe 3 restent valables quand à leur utilisation.

Exemple 12 : L'expression [λx . (((x K) a (Δ Δ)) (x (K I) (Δ Δ) a))] (λ y . y) qui n'a pas de type dans la théorie non étendue, à un type positif et une forme normale. Ceci est du au fait que les deux occurences de x ont deux types non comparables et non compatibles simultanément avec un type de λ y . y .

5-3 Résultats principaux

L'introduction des séquences permet tout d'abord de montrer les théorêmes suivants :

Théorême 9 : Si X et X' sont deux expressions α - β - η convertibles alors

$$\mathcal{B} \vdash \sigma X \quad \Rightarrow \mathcal{B} \vdash \sigma X'$$

D'autre part à partir des théorêmes 7 et 8 et par récurrence sur le nombre de séquences du type, la longueur du type et la taille des expressions, on montre que

Théorême 10 : $\forall$ X si $\mathcal{B} \vdash \tau X$, $\tau \neq \omega$ alors X a une forme normale gauche .

Théorême 11 : $\forall$ X si $\mathcal{B} \vdash \tau X$, $\tau \exists 0$ alors X a une forme normale .

De ces trois théorêmes on déduit le résultat principal

Théorême 12 : $\forall$ X , X a une forme normale (forme normale gauche) <=>

$\exists \mathcal{B}$ et $\tau \exists 0$ ($\tau \neq \omega$) tels que $\mathcal{B} \vdash \tau X$

La théorie étendue permet de donner un type significatif à toute expression qui n'est pas un terme insolvable (1), id est à la représentation de tout programme qui calcule effectivement quelque chose. Si on applique cette théorie à Y l'ensemble des types que l'on peut affecter à Y est d'après le théorême 12 l'ensemble des types que l'on peut affecter aux expressions de la forme $\lambda f.\ f\ (\ f\ (\ .\ .\ .\ .$ . . (($\lambda x.\ f\ (\ x\ x\ )\ )\ (\lambda x.\ f\ (\ x\ x\ )\ )$)...)) obtenues par réductions successives de Y . C'est l'ensemble des types de la forme $[[\omega]\tau_1, [\tau_1']\ \tau_2, \ .\ .\ .$ . . $[\tau_n']\tau_{n+1}\ ]\tau_{n+1}$ avec $\tau_i \exists \tau_i'$, $\forall i=1,n$

On en déduit une condition nécessaire et suffisante sur le type de F pour que YFX ait une forme normale :

- F doit avoir les types $[\omega]\tau_1, \ldots, [\tau_n']\tau_{n+1}$ $\tau_i \exists \tau_i'$, $\forall i=1,n$
- si σX alors $\tau_{n+1} = [\sigma]\tau$ et $\tau \exists 0$

B I B L I O G R A P H I E

1 - H.P. BARENDREGT "Some extensional term models for combinatory logics and λ-calculi", Ph. D. Thesis, UTRECHT Univ., the Netherlands, 1971.

2 - C. BOHM "The CUCH as a formal and description language" In Formal language, description languages for computer Programming, edited by T.B. Steel Jr. ; pp 179-197, North Holland, Amsterdam 1966.

3 - C. BOHM, M. DEZANI-CIANCAGLINI "Lambda-terms as total or partial Function on Normal Forms, Lambda Calculus and Computer Science Theory". Lectures Notes in Computer Science, 37, (1975), Springer Verlag, 96-121.

4 - C. BOHM, M. DEZANI-CIANCAGLINI "Termination Test inside λ-calculus", to appear in : Automata languages and Programming, ed. A. Salomaa, Lecture Notes in Computer Science, Springer-Verlag.

5 - H.B. CURRY, JR. HINDLEY, J.P. SELDIN "Combinatory Logic", Vol II, Amsterdam, North Holland, (1972)

6 - M. COPPO, M. DEZANI-CIANCAGLINI " A new type Assignment for λ-terms, Rapport Interne, Université de Turin, (1976).

7 - M. COPPO, M. DEZANI-CIANCAGLINI "A generalized type theory for λ-calculus, Rapport Interne, Université de Turin, (1977)

8 - P. J. LANDIN "A correspondence between ALGOL-60 and CHURCH'S λ-notation", CACM, vol. 8, February and March 1965, pp 89-101, 158-165.

9 - J.H. MORRIS "Lambda Calculus Models of Programming Languages", Ph.D., M.I.T. 1968

10 - L. NOLIN "Les modèles informatiques des λ-Calculs" , λ-Calculus and Computer Science Theory Proceedings of the Symposium held in Rome C. BOHM Ed, Springer Verlag, (1975), pp 166-176

11 - B. ROBINET "Contribution à l'étude des réalités informatiques" thèse Doctorat, n° I.P. 74-9, Paris, (1974).

12 - L.E. SANCHIS "Types of Combinatory Logic" Notre Dame Journal of Formal logic (5) 1964 pp. 161-180

13 - P. SALLE " La notion de type en λ-calcul" Groupe Programmation et Languages AFCET, Bulletin n°3, (1978).

14 - P. SALLE "Une théorie des types généralisée en λ-calcul" (à paraître)

15 - C.P. WADSWORTH "The relation between Lambda expressions and their denotations in Scott's Models for the Calculus", Seminaire IRIA 1974.