Systèmes de types purs et séquents classique.

Romain Kervac

Laboratoire de l'Informatique du Parallélisme

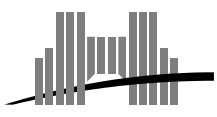
École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON-UCBL n° 5668

Pure Type Systems and Classical Sequents

Romain Kervarc Oct 2006

Research Report N° 2006-36

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : lip@ens-lyon.fr



Pure Type Systems and Classical Sequents

Romain Kervarc

Oct 2006

Abstract

The $\overline{\lambda}\mu\tilde{\mu}$-calculus was designed by P.-L. Curien and H. Herbelin. One of its interest is that its terms can be interpreted as derivations in the classical sequent calculus. One of the its lacks is the fact that it has only simple types. Our purpose here is to extend the calculus to higher-order types, and especially those of the calculus of constructions, a calculus designed by T. Coquand and G. Huet in order to provide a very general typed language for proof assistants based on λ-calculus. In order to do this, we place ourselves in the framework of pure type systems, which are a very general formalism allowing to represent many interesting type systems, among which those of Barendregt's λ-cube which is a hierarchical presentation of the calculus of constructions. We show that our systems satisfy some fundamental properties, such as subject reduction, and strong normalisation on the λ-cube.

Keywords: λ-calculus, $\bar{\lambda}\mu\tilde{\mu}$-calculus, classical logic, sequent calculus, pure type systems, higher order types, calculus of constructions, λ-cube, subject reduction, strong normalisation.

Résumé

Le $\overline{\lambda}\mu\tilde{\mu}$-calcul a été introduit par P.-L. Curien et H. Herbelin. L'un de ses intérêts est que ses termes peuvent être interprétés comme des dérivations dans le calcul des séquents classiques. L'un de ses défauts est qu'il n'a que des types simples. Nous nous proposons ici d'étendre ce calcul à des types d'ordre supérieur, et particulièrement à ceux du calcul des constructions, un calcul introduit par T. Coquand et G. Huet pour fournir un langage typé très général pour les assistants à la démonstration basés sur le λ-calcul. Pour faire cela, nous nous plaçons dans le cadre des systèmes de types purs, qui sont un formalisme très général permettant de représenter de nombreux systèmes de types intéressants, parmi lesquels ceux du λ-cube de Barendregt, qui est une présentation hiérarchique du calcul des constructions. Nous montrons que nos systèmes satisfont des proprétés fondamentales comme la réduction du sujet et la normalisation forte sur le λ-cube.

Mots-clés: λ-calcul, $\bar{\lambda}\mu\tilde{\mu}$-calcul, logique classique, calcul des séquents, systèmes de types purs, types d'ordre supérieur, calcul des constructions, λ-cube, réduction du sujet, normalisation forte.

Contents

Introduction

S. Berardi et J. Terlouw ont, indépendamment l'un de l'autre, fourni dans leurs travaux en 1989 des méthodes générales permettant d'engendrer de manière systématique des systèmes de types à la Church pour le λ-calcul. Ces méthodes ont abouti aux *systèmes de types purs*, qui sont un formalisme simple et élégant permettant de décrire de nombreux systèmes de types, parmi lesquels en particulier ceux dits du λ-cube, cette décomposition hiérarchique du calcul des constructions que l'on vient de présenter. Comme on l'a mentionné, cette décomposition a été introduite par H. Barendregt, lequel a donné ultérieurement dans [2] une présentation formelle et systématique des systèmes de types purs, ainsi que de leurs principales propriétés.

Syntaxe et sémantique

De manière formelle, les systèmes de types purs sont définis comme suit :

Définition 0.1 : *système de types pur ; sorte, axiome, règle*
Par *système de types pur*, on entend un triplet $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ vérifiant que $\mathcal{A} \subseteq \mathcal{S}^2$ et $\mathcal{R} \subseteq \mathcal{S}^3$.
Les éléments de $\mathcal{S}$, $\mathcal{A}$ et $\mathcal{R}$ sont respectivement appelés *sortes*, *axiomes* et *règles* du système de types purs $\mathfrak{T}$.

Naturellement, à cette définition formelle, qui constitue en quelque sorte une spécification du système de types, vient s'ajouter un calcul sous-jacent, basé sur le λ-calcul typé à la Church et défini comme suit :

Définition 0.2 : *$\mathfrak{T}$-expressions*
Soient $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ un système de types purs, et $\mathcal{U}$ un ensemble infini de variables.
On définit alors l'ensemble $\mathcal{E}(\mathfrak{T})$ des *$\mathfrak{T}$-expressions* par la grammaire algébrique suivante :

$$\begin{array}{rcll} \mathrm{E} & ::= & \sigma & \text{(sorte)} \\ & | & x & \text{(variable)} \\ & | & \mathrm{E\,E} & \text{(application)} \\ & | & \lambda x{:}\mathrm{E.E} & \text{(abstraction)} \\ & | & \Pi x{:}\mathrm{E.E} & \text{(quantification)} \end{array}$$

où les symboles σ et x décrivent respectivement les ensembles des sortes et des variables ($\sigma \in \mathcal{S}$; $x \in \mathcal{U}$).

Sur ces termes, on définit de façon usuelle les notions de variables libres et liées :

Définition 0.3 : *variables libres/liées pour les systèmes de types purs implicites*
Soit $\mathfrak{T}$ un système de types pur. Soit M une $\mathfrak{T}$-expression.
On définit l'ensemble $fv(\mathrm{M})$ des *variables libres* de M inductivement comme suit :

- $fv(\sigma) = \emptyset$;
- $fv(x) = \{x\}$;
- $fv(\Pi x{:}\mathrm{L.M}) = (fv(\mathrm{M})\backslash\{x\}) \cup fv(\mathrm{L})$;
- $fv(\lambda x{:}\mathrm{L.M}) = (fv(\mathrm{M})\backslash\{x\}) \cup fv(\mathrm{L})$;
- $fv(\mathrm{M\,N}) = fv(\mathrm{M}) \cup fv(\mathrm{N})$.

De façon analogue, on définit l'ensemble $bv(\mathrm{M})$ des *variables liées* de M comme suit :

- $bv(\sigma) = \emptyset$;
- $bv(x) = \emptyset$;
- $bv(\Pi x{:}\mathrm{L}.\mathrm{M}) = \{x\} \cup bv(\mathrm{L}) \cup bv(\mathrm{M})$;
- $bv(\lambda x{:}\mathrm{L}.\mathrm{M}) = \{x\} \cup bv(\mathrm{L}) \cup bv(\mathrm{M})$;
- $bv(\mathrm{M}\ \mathrm{N}) = bv(\mathrm{M}) \cup bv(\mathrm{N})$.

On définit également la classique α-conversion :

Définition 0.4 : *α-conversion*
Soit $\mathfrak{T}$ un système de types purs. On appelle α-conversion la relation $\equiv_\alpha$ définie sur $\mathcal{E}(\mathfrak{T})$ par :

- si $\mathrm{M} = x \in \mathcal{U}$, $\mathrm{M} \equiv_\alpha \mathrm{N}$ si $\mathrm{N} = x$;
- si $\mathrm{M} = \sigma \in \mathcal{S}$, $\mathrm{M} \equiv_\alpha \mathrm{N}$ si $\mathrm{N} = \sigma$;
- si $\mathrm{M} = \lambda x{:}\mathrm{A}.\mathrm{P}$, $\mathrm{M} \equiv_\alpha \mathrm{N}$ si $\mathrm{N} = \lambda y{:}\mathrm{B}.\mathrm{R}$, avec $\mathrm{A} \equiv_\alpha \mathrm{B}$ et $\mathrm{P}[x{:=}z] \equiv_\alpha \mathrm{R}[y{:=}z]$ pour tout z sauf un nombre fini ;
- si $\mathrm{M} = \Pi x{:}\mathrm{A}.\mathrm{P}$, $\mathrm{M} \equiv_\alpha \mathrm{N}$ si $\mathrm{N} = \lambda y{:}\mathrm{B}.\mathrm{R}$, avec $\mathrm{A} \equiv_\alpha \mathrm{B}$ et $\mathrm{P}[x{:=}z] \equiv_\alpha \mathrm{R}[y{:=}z]$ pour tout z sauf un nombre fini ;
- si $\mathrm{M} = \mathrm{PQ}$, $\mathrm{M} \equiv_\alpha \mathrm{N}$ si $\mathrm{N} = \mathrm{RS}$ avec $\mathrm{P} \equiv_\alpha \mathrm{R}$ et $\mathrm{Q} \equiv_\alpha \mathrm{S}$.

Il est aisé de vérifier que $\equiv_\alpha$ est une relation de congruence sur les $\mathfrak{T}$-expressions.

L'α-conversion étant une congruence, on peut considérer les expressions à α-conversion près. On raisonne alors non plus sur les expressions, mais sur leurs classes d'équivalence modulo α-conversion, que l'on appelle les *termes* du calcul :

Définition 0.5 : *$\lambda\mathfrak{T}$-calcul*
On considère l'ensemble quotient $\Lambda\mathfrak{T} = \mathcal{E}(\mathfrak{T})/\equiv_\alpha$, dont les éléments sont appelés *termes du $\lambda\mathfrak{T}$-calcul*. L'α-conversion $\equiv_\alpha$ étant une congruence, les opérations de $\mathcal{E}(\mathfrak{T})$ (*i.e.* l'abstraction, l'application, la quantification) s'étendent canoniquement aux termes de $\Lambda\mathfrak{T}$.

Dans la suite, on adoptera pour le choix des représentants des termes du calcul la *convention de Barendregt*, qui stipule qu'une variable liée – dont on peut toujours librement changer le nom par α-conversion – doit avoir un nom distinct de toute autre variable. (En particulier, cela interdit de faire apparaître une variable à la fois libre et liée dans un terme, et oblige de donner à deux variables liées distinctes des noms distincts.)

On dote ensuite le calcul sous-jacent à un système de type pur ainsi défini d'une sémantique opérationnelle par l'adjonction de la relation de β-réduction du λ-calcul typé à la Church :

Définition 0.6 : *β-réduction, β-conversion*
Soit $\mathfrak{T}$ un système de types purs. La β-réduction $\underset{\beta}{\longrightarrow}$ est la relation de réduction sur les $\mathfrak{T}$-expressions induite par la règle :

$$(\beta) \qquad (\lambda x{:}\mathrm{A}.\mathrm{B})\,\mathrm{C} \quad \longrightarrow \quad \mathrm{B}[x{:=}\mathrm{C}]$$

où $[x{:=}C]$ est une substitution implicite – *i.e.* $B[x{:=}C]$ désigne l'expression B où chaque occurrence libre de x est remplacée par une sous-expression C.
La β-conversion $\equiv_\beta$ est la clôture réflexive, symétrique et transitive de $\xrightarrow[\beta]{}$.

Les notions *supra* constituent l'aspect calculatoire des systèmes de types purs. Sur cela vient se greffer un aspect logique, par le biais de jugements de typage, qui peuvent être interprétés *via* la correspondance de Curry-Howard comme des assertions logiques.

Le typage dans les systèmes de types purs s'effectue par le moyen suivant :

$$\frac{(\sigma, \tau) \in \mathcal{A}}{\vdash \sigma : \tau}\ \text{(axiome)}$$

$$\frac{\Gamma \vdash A : \rho \qquad \Gamma, x : A \vdash B : \sigma \qquad (\rho, \sigma, \tau) \in \mathcal{R}}{\Gamma \vdash \Pi x{:}A.B : \tau}\ \text{(règle)}$$

$$\frac{\Gamma \vdash A : \sigma \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}\ \text{(hypothèse)}$$

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : \sigma \qquad x \notin \mathrm{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}\ \text{(affaiblissement)}$$

$$\frac{\Gamma \vdash (\Pi x{:}A.B) : \sigma \qquad \Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x{:}A.M : (\Pi x{:}A.B)}\ \text{(}\Pi\text{ - introduction)}$$

$$\frac{\Gamma \vdash M : (\Pi x{:}A.B) \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x{:=}N]}\ \text{(}\Pi\text{ - élimination)}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash B : \sigma \qquad A \equiv_\beta B}{\Gamma \vdash M : B}\ \text{(conversion)}$$

Table 1: Figures d'inférence de type valides pour les systèmes de types purs (implicites)

Définition 0.7 : *assertion, contexte, jugement (valide) de typage*
Soit $\mathfrak{T}$ un système de types purs.
On appelle *assertion de typage* un couple de $\mathfrak{T}$-expressions noté M : N. Dans un tel couple, le premier élément est appelé *sujet* de l'assertion et le second, *prédicat* de l'assertion.
On appelle *contexte de typage* une suite finie[1] d'assertions de types dont les sujets sont des variables distinctes. L'ensemble de ces variables est appelé le *domaine* du contexte, et leur suite ordonnée, son *support*. Le contexte vide est noté [ ], et la concaténation de contextes de domaines disjoints est notée par une virgule.
On appelle *jugement de typage* une expression de la forme $\Gamma \vdash_{\mathfrak{T}} M : N$, où Γ est un contexte de typage et M : N une assertion de typage. Un jugement est dit *valide* s'il dérive de l'application des figures[2] d'inférence présentées dans la table 1 *supra*. L'indice $\mathfrak{T}$ peut être omis s'il n'y a pas d'ambiguïté.

[1] et non un ensemble : le fait que termes et types ne soient pas des catégories syntaxiques distinctes engendre des dépendances entre variables, ce qui impose que les contextes de typage soient ordonnés

[2] S'inspirant de la dénomination de Gentzen *Schlußfigur*, on parle de figures plutôt que de règles, ce dernier terme étant réservé aux règles d'un système de types pur, *i.e.* aux éléments de $\mathcal{R}$ dans le système $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$.

Définition 0.8 : *contexte bien formé*
Un contexte $\Gamma = [x_1 : A_1, \ldots, x_n : A_n]$ est dit *bien formé* si pour tout k de $[\![1, n]\!]$, il existe une sorte σ_k telle que $[x_1 : A_1, \ldots, x_{k-1} : A_{k-1}] \vdash A_k : \sigma_k$.

Propriétés

On va maintenant rappeler diverses propriétés générales des systèmes de types purs, desquelles on ne donnera pas de démonstrations, car celles-ci peuvent être trouvées dans [2], auquel pourra se référer le lecteur désireux de les consulter.

Lemme 0.1 : *Initialisation pour les systèmes de types purs implicites*
Soient Γ un contexte bien formé et $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ un système de types pur.

(*i*) Si $(\sigma : \tau) \in \mathcal{A}$, alors $\Gamma \vdash \sigma : \tau$.

(*ii*) Si $(u : V \in \Gamma$, alors $\Gamma \vdash u : V$.

Lemme 0.2 : *Engendrement pour les systèmes de types purs implicites*
Soient Ξ un contexte et M, T deux $\lambda\mathfrak{T}$-termes tels que $\Xi \vdash M : T$. Alors :

(i) si $M = \sigma \in \mathcal{S}$ alors
$\exists\, \tau \in \mathcal{S}\ (T \equiv_\beta \tau\ \wedge\ (\sigma : \tau) \in \mathcal{A})$;
(ii) si $M = x \in \mathcal{U}$ alors
$\exists\, \tau \in \mathcal{S}\ \exists U \in \Lambda\mathfrak{T}\ (\Xi \vdash U : \tau\ \wedge\ (x{:}U) \in \Xi\ \wedge\ T \equiv_\beta U)$;
(iii) si $M = \Pi x{:}A.B$ alors
$\exists\, (\rho,\sigma,\tau) \in \mathcal{R}\ (\Xi \vdash A : \rho\ \wedge\ \Xi, x : A \vdash B : \sigma\ \wedge\ T \equiv_\beta \tau)$;
(iv) si $M = \lambda x{:}A.B$ alors
$\exists\, \tau \in \mathcal{S}\ \exists C \in \Lambda\mathfrak{T}\ (\Xi \vdash (\Pi x{:}A.C) : \tau \wedge \Xi, x : A \vdash B : C \wedge T \equiv_\beta \Pi x{:}A.C)$;
(v) si $M = A\ B$ alors
$\exists C, D \in \Lambda\mathfrak{T}\ (\Xi \vdash A : (\Pi x{:}C.D)\ \wedge\ \Xi \vdash B : C\ \wedge\ T \equiv_\beta D\langle x{:=}B\rangle)$.

Lemme 0.3 : *Substitution pour les systèmes de types purs implicites*
Soit $\mathfrak{T}$ un système de types purs. Soient $\Gamma, x : A, \Delta \vdash_{\mathfrak{T}} M : B$ un jugement de typage valide et N un $\mathfrak{T}$-terme tels que $\Gamma \vdash_{\mathfrak{T}} N : A$ soit un jugement de typage valide.
Alors le jugement de typage $\Gamma, \Delta[x{:=}N] \vdash_{\mathfrak{T}} M[x{:=}N] : B[x{:=}N]$ est valide.

Théorème 0.4 : *Correction pour les systèmes de types purs implicites*
Soit $\mathfrak{T}$ un système de types purs. Soient Γ un contexte de typage et A, B deux $\mathfrak{T}$-termes tels que le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide.
Alors B est une sorte ou il existe une sorte σ telle que $\Gamma \vdash_{\mathfrak{T}} B : \sigma$ soit valide.

Théorème 0.5 : *Réduction du sujet pour les systèmes de types purs implicites*
Soit $\mathfrak{T}$ un système de types purs.
Soient Γ un contexte de typage et A, A′, B trois $\mathfrak{T}$-termes tels que le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide et $A \xrightarrow[\beta]{} A'$.
Alors le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A' : B$ est valide.

Construction du λ-cube en systèmes de types purs

Dans ce formalisme, le λ-cube peut se construire de façon systématique et élégante. Pour ce faire, on considère une classe particulière de systèmes de types purs, dits élémentaires :

Définition 0.9 : *système de types pur élémentaire (complet)*
Étant donné un ensemble non vide de sortes $\mathcal{S}$, on appelle *règle élémentaire* tout triplet de la forme (σ, τ, τ), ce que l'on dénote en abrégé par $[\sigma, \tau]$. L'ensemble des règles élémentaires formées avec les éléments de $\mathcal{S}$ se note $\mathcal{R}^e_{\mathcal{S}}$.
Un système de types pur $\mathfrak{T} = \mathcal{S}, \mathcal{A}, \mathcal{R})$ est dit *élémentaire* si $\mathcal{R} \subseteq \mathcal{R}^e_{\mathcal{S}}$ et *élémentaire complet* si $\mathcal{R} = \mathcal{R}^e_{\mathcal{S}}$.

Le système de types purs correspondant au λ-calcul simplement typé est le système élémentaire suivant : $\lambda{\to} = (\{*, \Box\}, \{* : \Box\}, \{[*, *]\})$.

Les systèmes de types du λ-cube sont alors définis simplement comme les systèmes élémentaires prolongeant $\lambda{\to}$, *i.e.* les systèmes qui ont mêmes sortes et axiome que $\lambda{\to}$ et dont les règles sont élémentaires et contiennent $[*, *]$.

Suivant que l'on ajoute l'une des trois autres règles élémentaires, on parcourt alors une arête dans le cube : vers le haut, on ajoute $[\Box, *]$; vers la droite, $[*, \Box]$; et vers le fond, $[\Box, \Box]$.

De plus, on remarque que le sommet du λ-cube, le calcul des constructions, correspond au système élémentaire complet prolongeant $\lambda{\to}$, dont la spécification formelle est $\lambda C = (\{*, \Box\}, \{* : \Box\}, \mathcal{R}^e_{\{*,\Box\}})$.

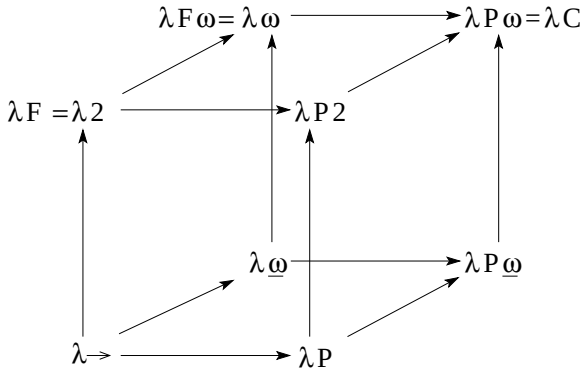


Figure 1: Le λ-cube de Barendregt

On voit ici à quel point le formalisme des systèmes de types purs est adapté à la construction du λ-cube, en particulier lorsque l'on s'intéresse d'un peu plus près à la spécification des systèmes du λ-cube.

Les systèmes comportent deux sortes, $*$ et $\Box$, qui caractérisent respectivement les types d'expressions à valeur de termes et les types d'expressions à valeur de type, ce que confirme l'axiome $* : \Box$, qui dit simplement qu'$*$, sorte typant les types d'expression à valeur de termes, est elle-même subséquemment un type d'expressions à valeurs de types, donc typable par $\Box$.

Outre les résultats présentés précédemment, qui étaient satisfaits en toute généralité pour tout système de types pur. On va maintenant donner des résultats plus particuliers, qui seraient faux en général, mais sont vrais dans le cas particulier des systèmes du λ-cube.

Définition 0.10 : *normalisation forte*
Un système de types pur $\mathfrak{T}$ est dit *fortement normalisant* s'il vérifie la propriété suivante : soient Γ un contexte de typage et A, B deux $\mathfrak{T}$-termes tels que le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide. Alors A et B n'admettent pas de chaîne infinie de β-réduction.

On peut remarquer que, du fait du théorème de correction des types, cette définition est équivalente à une définition où l'on exige simplement la normalisation forte de sujet du jugement, celle du prédicat provenant du fait qu'il est une sorte (donc un terme β-irréductibles), soit lui-même le sujet d'un jugement de correction.

Théorème 0.6 : *Normalisation forte dans les s.t.p. implicites du λ-cube*
Les systèmes de types purs du λ-cube ($\lambda\!\!\rightarrow$, $\lambda\underline{\omega}$, $\lambda 2$, λP, $\lambda\omega$, $\lambda P2$, $\lambda P\underline{\omega}$, $\lambda P\omega$) sont fortement normalisants.

I. Le $\bar{\lambda}\mu\tilde{\mu}$-calcul

1. Historique

La problématique générale dans laquelle se place est la suivante : la correspondance de Curry-Howard(-De Bruijn), que l'on a déjà présentée au paragraphe 2.1 du chapitre I, a été à son origine développée dans le cadre de la logique intuitionniste. Toutefois, l'observation que certaines tautologies de la logique classique évoquent le typage de certains opérateurs de contrôle conduisit T. Griffin en 1990 dans [5] à entamer l'extension de cette correspondance à la logique classique. L'une des pistes les plus prometteuses dans cette optique fut le $\lambda\mu$-calcul de M. Parigot (cf. en particulier [12]). Le système de types associé à ce calcul est basé sur un formalisme de déduction naturelle pour la logique classique.

Une approche alternative, qui est celle que l'on va appliquer ici, est celle du $\bar{\lambda}\mu\tilde{\mu}$-calcul, introduit dans [3] en 2000 par P.-L. Curien et H. Herbelin, dont les termes représentent des dérivations dans un calcul de séquent, tandis que la réduction sur ces termes représentent l'élimination des coupures. L'un des avantages de cette approche par le calcul des séquents est sa symétrie intrinsèque, qui peut être mise en parallèle avec la symétrie inhérente de la logique, et spécialement de la logique classique.
Dans [3], le $\bar{\lambda}\mu\tilde{\mu}$-calcul est introduit avec un système de types simples, et ses versions en appel par valeur et en appel par nom sont encodées en λ-calcul simplement typé *via* une traduction CPS – ce qui montre en particulier la forte normalisation de ce système de types.

Le $\bar{\lambda}\mu\tilde{\mu}$-calcul a par ailleurs été enrichi en calcul à substitutions explicites. On pourra en particulier se reporter à l'article [13] d'E. Polonovski, où celui-ci étudie la forte normalisation d'un $\bar{\lambda}\mu\tilde{\mu}$-calcul à substitutions explicites pour la réduction sans restriction. L'une des importantes difficultés posées par ce problème est que cette réduction n'est pas confluente, du fait d'une paire critique. La méthode exposée dans [13] fait usage d'une notion de « candidats de symétries », trouvant son origine dans l'article [1] de F. Barbanera et S. Berardi.

Par ailleurs, dans l'important article [4] de D. Dougherty, S. Ghilezan et P. Lescanne, on trouve une étude des propriétés computationnelles du $\bar{\lambda}\mu\tilde{\mu}$-calcul non typé, et en particulier une caractérisation les termes fortement normalisants (en appel par valeur ou par nom) au moyen d'un système de types à intersection et union, sous forme d'un calcul de séquents analogue à celui des types simples définis dans [3].

2. Syntaxe et réduction

Les termes du $\bar{\lambda}\mu\tilde{\mu}$-calcul sont définis comme suit :

Définition 1.11 : *$\bar{\lambda}\mu\tilde{\mu}$-calcul ; appelants, appelés, capsules*
Soient $\mathcal{X}$ et $\mathcal{V}$ deux ensembles infinis dénombrables de variables appelées respectivement *variables d'appelant* et *variables d'appelé*. On définit alors trois ensembles d'expressions au moyen de la grammaire algébrique suivante :

$$\begin{array}{lcll} \mathrm{R} & ::= & x & \textit{(variable d'appelant)} \\ & | & \lambda x.\mathrm{R} & \textit{(λ-abstraction)} \\ & | & \mu\alpha.\mathrm{C} & \textit{(μ-abstraction)} \\ \mathrm{E} & ::= & \alpha & \textit{(variable d'appelé)} \\ & | & \mathrm{R}\bullet\mathrm{E} & \textit{(empilement)} \\ & | & \tilde{\mu}x.\mathrm{C} & \textit{($\tilde{\mu}$-abstraction)} \\ \mathrm{C} & ::= & \langle \mathrm{R} \parallel \mathrm{E} \rangle & \textit{(capsule)} \end{array}$$

$$\sigma \in \mathcal{S} \ ; \ x \in \mathcal{X} \ ; \ \alpha \in \mathcal{V}$$

dont les trois principaux symboles décrivent chacun une espèce d'expression :

- R décrit les $\mathfrak{T}$-appelants ;
- E décrit les $\mathfrak{T}$-appelés ;
- C décrit les $\mathfrak{T}$-capsules.

On observe qu'il existe deux espèces différentes de variables : les variables dites respectivement *d'appelant*, qui décrivent l'ensemble $\mathcal{X}$, et *d'appelé*, qui décrivent l'ensemble $\mathcal{V}$. On observera, dans toute cette partie, la convention suivante : les variables d'appelant sont dénotées par des lettres latines minuscules, tandis que les variables d'appelé sont dénotées par des lettres grecques minuscules.

On peut naturellement définir canoniquement les notions de variables libres et liées de la manière suivante :

Définition 1.12 : *variables libres/liées pour $\bar{\lambda}\mu\tilde{\mu}$*
On définit les variables d'appelant (respectivement d'appelé) libres ou liées d'une expression comme appartenant à une partie de $\mathcal{X}$ (respectivement $\mathcal{V}$) définie par induction sur l'expression.
Les ensembles des variables d'appelant libres, des variables d'appelant liées, des variables d'appelé libres, des variables d'appelé liées de M sont respectivement dénotées par $fv_{\mathcal{X}}(\mathrm{M})$, $bv_{\mathcal{X}}(\mathrm{M})$, $fv_{\mathcal{V}}(\mathrm{M})$, $bv_{\mathcal{V}}(\mathrm{M})$.
Les variables d'appelant libres sont définies comme suit :

- $fv_{\mathcal{X}}(x) = \{x\}$;
- $fv_{\mathcal{X}}(\lambda x.\mathrm{R}) = fv_{\mathcal{X}}(\mathrm{R}) \setminus \{x\}$;
- $fv_{\mathcal{X}}(\mu\alpha.\mathrm{C}) = fv_{\mathcal{X}}(\mathrm{C})$;
- $fv_{\mathcal{X}}(\alpha) = \emptyset$;

- $fv_{\mathcal{X}}(\mathrm{R}\bullet\mathrm{E}) = fv_{\mathcal{X}}(\mathrm{R}) \cup fv_{\mathcal{X}}(\mathrm{E})$;
- $fv_{\mathcal{X}}(\tilde{\mu}x.\mathrm{C}) = fv_{\mathcal{X}}(\mathrm{C}) \setminus \{x\}$;
- $fv_{\mathcal{X}}(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = fv_{\mathcal{X}}(\mathrm{R}) \cup fv_{\mathcal{X}}(\mathrm{E})$;

et les variables d'appelant liées comme suit :

- $bv_{\mathcal{X}}(x) = \emptyset$;
- $bv_{\mathcal{X}}(\lambda x.\mathrm{R}) = bv_{\mathcal{X}}(\mathrm{R}) \cup \{x\}$;
- $bv_{\mathcal{X}}(\mu\alpha.\mathrm{C}) = bv_{\mathcal{X}}(\mathrm{C})$;
- $bv_{\mathcal{X}}(\alpha) = \emptyset$;
- $bv_{\mathcal{X}}(\mathrm{R}\bullet\mathrm{E}) = bv_{\mathcal{X}}(\mathrm{R}) \cup bv_{\mathcal{X}}(\mathrm{E})$;
- $bv_{\mathcal{X}}(\tilde{\mu}x.\mathrm{C}) = bv_{\mathcal{X}}(\mathrm{C}) \cup \{x\}$;
- $bv_{\mathcal{X}}(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = bv_{\mathcal{X}}(\mathrm{R}) \cup bv_{\mathcal{X}}(\mathrm{E})$.

Les variables d'appelant libres sont définies comme suit :

- $fv_{\mathcal{V}}(x) = \emptyset$;
- $fv_{\mathcal{V}}(\lambda x.\mathrm{R}) = fv_{\mathcal{V}}(\mathrm{R})$;
- $fv_{\mathcal{V}}(\mu\alpha.\mathrm{C}) = fv_{\mathcal{V}}(\mathrm{C}) \setminus \{\alpha\}$;
- $fv_{\mathcal{V}}(\alpha) = \{\alpha\}$;
- $fv_{\mathcal{V}}(\mathrm{R}\bullet\mathrm{E}) = fv_{\mathcal{V}}(\mathrm{R}) \cup fv_{\mathcal{V}}(\mathrm{E})$;
- $fv_{\mathcal{V}}(\tilde{\mu}x.\mathrm{C}) = fv_{\mathcal{V}}(\mathrm{C})$;
- $fv_{\mathcal{V}}(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = fv_{\mathcal{V}}(\mathrm{R}) \cup fv_{\mathcal{V}}(\mathrm{E})$;

et les variables d'appelé liées comme suit :

- $bv_{\mathcal{V}}(x) = \emptyset$;
- $bv_{\mathcal{V}}(\lambda x.\mathrm{R}) = bv_{\mathcal{V}}(\mathrm{R})$;
- $bv_{\mathcal{V}}(\mu\alpha.\mathrm{C}) = bv_{\mathcal{V}}(\mathrm{C}) \cup \{\alpha\}$;
- $bv_{\mathcal{V}}(\alpha) = \emptyset$;
- $bv_{\mathcal{V}}(\mathrm{R}\bullet\mathrm{E}) = bv_{\mathcal{V}}(\mathrm{R}) \cup bv_{\mathcal{V}}(\mathrm{E})$;
- $bv_{\mathcal{V}}(\tilde{\mu}x.\mathrm{C}) = bv_{\mathcal{V}}(\mathrm{C})$;
- $bv_{\mathcal{V}}(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = bv_{\mathcal{V}}(\mathrm{R}) \cup bv_{\mathcal{V}}(\mathrm{E})$.

Naturellement, à cette notion de variable libre ou liée est associée une notion d'α-conversion, qui se définit de façon usuelle. De nouveau, on peut considérer comme termes du calcul les éléments du quotient par l'α-conversion, auxquels s'étendent canoniquement toutes les opérations de $\bar{\lambda}\mu\tilde{\mu}$. On applique comme précédemment la convention de Barendregt pour le choix des représentants.

Parmi tous les termes exposés ci-dessus, les expressions essentielles de $\bar{\lambda}\mu\tilde{\mu}$, *i.e.* celles sur lesquelles se font les calculs, sont les capsules. Leur sémantique opérationnelle est donnée par la définition suivante :

Définition 1.13 : *réduction pour* $\bar{\lambda}\mu\tilde{\mu}$
La réduction, notée $\rightarrow$, est définie comme la clôture contextuelle de l'union des trois axiomes de réduction suivants :

$$\begin{array}{lrcl} (\lambda) & \langle \lambda x.\mathrm{M} \parallel \mathrm{N}\bullet\mathrm{E}\rangle & \rightarrow & \langle \mathrm{M}[\mathrm{N}/x] \parallel \mathrm{E}\rangle \\ (\mu) & \langle \mu\alpha.\mathrm{C} \parallel \mathrm{E}\rangle & \rightarrow & \mathrm{C}[\mathrm{E}/\alpha] \\ (\tilde{\mu}) & \langle \mathrm{R} \parallel \tilde{\mu}x.\mathrm{C}\rangle & \rightarrow & \mathrm{C}[\mathrm{R}/x] \end{array}$$

Ces trois règles correspondent aux trois différentes actions possibles au sein d'une capsule. Intuitivement, les deux éléments d'une capsule ont chacune leur fonction propre : l'appelant peut être vu comme un processus allant chercher des données dans l'appelé, que l'on peut voir comme une pile d'éléments – c'est là ce que l'on voit avec la règle (λ). Par ailleurs, chacun de l'appelant et de l'appelé peut demander à l'autre de prendre la place de l'une de ses variables d'espèce correspondante – c'est là la signification des règles (μ) et ($\tilde{\mu}$).

On constate que la notion de réduction ainsi obtenue n'est pas confluente. En effet, si l'on applique les deux dernières règles à la capsule $\langle \mu\xi.\mathrm{C}_1 \parallel \tilde{\mu}x.\mathrm{C}_2\rangle$, on obtient la paire critique $(\mathrm{C}[\tilde{\mu}x.\mathrm{C}_2/\xi], \mathrm{C}[\mu\xi.\mathrm{C}_1/x])$, qui n'est pas joignable, comme on peut le voir sur le simple contre-exemple suivant : on considère la capsule $\mathrm{C} = \langle \mu\xi.\langle a \parallel \alpha\rangle \parallel \tilde{\mu}x.\langle b \parallel \beta\rangle\rangle$, où a, b (respectivement α, β) sont des variables d'appelant (respectivement d'appelés) distinctes. On observe d'une part que $\mathrm{C} \xrightarrow[(\mu)]{} \langle a \parallel \alpha\rangle$ et d'autre part que $\mathrm{C} \xrightarrow[(\tilde{\mu})]{} \langle b \parallel \beta\rangle$. Ces deux capsules étant de toute évidence irréductibles et distinctes, donc injoignables, il est impossible de clore le diagramme.

Pour traiter ce problème de confluence, on peut décider de donner à l'une des règles (μ) ou ($\tilde{\mu}$) la priorité sur l'autre, ce qui revient en fait à choisir une stratégie d'évaluation des capsules. Si l'on donne la priorité à (μ), on obtient une stratégie d'appel par valeur (*call by value*), tandis que si l'on donne la priorité à ($\tilde{\mu}$), on obtient une stratégie d'appel par nom (*call by name*).

3. Séquents et typage

Les jugements de type pour le $\bar{\lambda}\mu\tilde{\mu}$-calcul nécessitent une présentation particulière, car ils sont assez différents des jugements usuels. Il s'agit en fait de *séquents à formules actives*, que l'on va définir ici de manière plus ou moins formelle.

Soient un ensemble Φ de formules définies sur une signature donnée. On appelle *séquent* un couple de parties de Φ noté $\Gamma \vdash \Delta$. Un séquent s'interprète comme suit : la conjonction des formules de Γ implique la disjonction des formules de Δ. Les séquents se dérivent au moyens

d'axiomes, d'introductions gauches et droites pour chaque connecteur de la signature des formules, et de coupures. On donne ici comme exemple dans la table 2 les figures de dérivation du calcul des séquents sur les variables propositionnels et l'implication $\rightarrow$:

$$\frac{}{\Gamma, \alpha \vdash \alpha, \Delta}\ (\text{axiome})$$

$$\frac{\Gamma \vdash \alpha, \Delta \quad \Gamma, \beta \vdash \Delta}{\Gamma, \alpha \rightarrow \beta \vdash \Delta}\ (\rightarrow\text{-G}) \qquad \frac{\Gamma, \alpha \vdash \beta, \Delta}{\Gamma \vdash \alpha \rightarrow \beta, \Delta}\ (\rightarrow\text{-D})$$

$$\frac{\Gamma \vdash \alpha, \Delta \quad \Gamma, \alpha \vdash \Delta}{\Gamma \vdash \Delta}\ (\text{coupure})$$

Table 2: Calcul des séquents sur la signature $\{\rightarrow\}$

On remarque dans ces figures de dérivation qu'une formule est en général distinguée : la formule où a lieu l'introduction du connecteur. Si l'on considère maintenant le calcul des séquents dans une perspective de traitement automatique, il semble important de repérer cette formule, d'où l'idée d'introduire une notion de *formule active*, que l'on va définir ici.

Étant donné l'ensemble Φ de formules, on construit l'ensemble $\hat{\Phi} = \Phi \times \{0, 1\}$, dont les éléments sont de deux espèces : les formules inactives, de la forme $(\varphi, 0)$ et les formules actives, de la forme $(\varphi, 1)$. Plutôt que d'écrire ces couples, on adopte la convention d'écriture suivante :

Convention :
Les formules inactives sont écrites normalement, tandis que les formules actives sont indiquées par l'emploi d'un fond grisé.

On appelle *séquent à formule active* un séquent dont les formules appartiennent à $\hat{\Phi}$, et dont au plus une est active. Il y en a donc de trois types : sans formule active, avec une formule active à gauche, avec une formule active à droite.

L'éventuelle formule active joue un rôle particulier dans la dérivation : dans toute figure d'introduction, la formule où le connecteur a été introduit est active et dérive d'une formule active dans les prémisses. En outre, la coupure fait disparaître les formules actives, et de nouvelles figures apparaissent pour faire apparaître une formule active dans un séquent n'en ayant pas. Pour poursuivre l'exemple précédent, on donne dans la table 2 les figures de dérivation du calcul des séquents à formules actives sur les variables propositionnels et l'implication $\rightarrow$:

Les jugements de typage du $\bar{\lambda}\mu\tilde{\mu}$-calcul sont en fait sous la forme de tels séquents à formule active.

Définition 1.14 : *assertion, contexte, jugement (valide) de typage dans* $\bar{\lambda}\mu\tilde{\mu}$
Soit $\mathfrak{T}$ un système de types purs. On appelle *assertion de typage* un couple de $\mathfrak{T}$-expressions noté M : R, où M peut être un appelant ou un appelé, et R est un appelant. Dans un tel couple, le premier élément est appelé *sujet* de l'assertion et le second, *prédicat* de l'assertion. Une assertion peut être dite *active*.
On appelle *contexte de typage* un ensemble d'assertions de typage non actives dont les sujets sont des variables distinctes de même nature. Si ces variables sont des variables d'appelant,

$$\frac{}{\Gamma,\ \alpha\ \vdash \alpha, \Delta}\ \text{(ax-G)} \qquad \frac{}{\Gamma, \alpha \vdash\ \alpha\ , \Delta}\ \text{(ax-D)}$$

$$\frac{\Gamma \vdash\ \alpha\ , \Delta \quad \Gamma,\ \beta\ \vdash \Delta}{\Gamma,\ \alpha{\to}\beta\ \vdash \Delta}\ (\to\text{-G}) \qquad \frac{\Gamma, \alpha \vdash\ \beta\ , \Delta}{\Gamma \vdash\ \alpha{\to}\beta\ , \Delta}\ (\to\text{-D})$$

$$\frac{\Gamma, \alpha \vdash \Delta}{\Gamma,\ \alpha\ \vdash \Delta}\ \text{(ax-G)} \qquad \frac{\Gamma \vdash \alpha, \Delta}{\Gamma \vdash\ \alpha\ , \Delta}\ \text{(ax-D)}$$

$$\frac{\Gamma \vdash\ \alpha\ , \Delta \quad \Gamma,\ \alpha\ \vdash \Delta}{\Gamma \vdash \Delta}\ \text{(coupure)}$$

Table 3: Calcul des séquents à formule active sur la signature $\{\to\}$

on parle de *contexte gauche*. Si ce sont des variables d'appelé, on parle de *contexte droit*. Le contexte vide est noté $\emptyset$, et la concaténation de contextes d'un même côté dont les supports sont disjoints est notée par une virgule.
On appelle *jugement de typage* une expression de l'une des trois formes suivante, Γ et Δ étant respectivement un contexte gauche de typage et un contexte droit de typage :

- $\Gamma;\ \mathrm{E} : \mathrm{A}\ \vdash \Delta$ où E est un appelé ;
- $\Gamma \vdash\ \mathrm{R} : \mathrm{A}\ ; \Delta$ où R est un appelant ;
- $\mathrm{C} : (\Gamma \vdash \Delta)$ où C est une capsule.

Un jugement est dit *valide* s'il dérive de l'application des figures[3] d'inférence présentées dans la table 4. L'indice $\mathfrak{T}$ peut être omis s'il n'y a pas d'ambiguïté.

$$\frac{}{\Gamma, x : \mathrm{A} \vdash\ x : \mathrm{A}\ , \Delta}\ \text{(ax-D)} \qquad \frac{}{\Gamma,\ \alpha : \mathrm{A}\ \vdash \alpha : \mathrm{A}, \Delta}\ \text{(ax-G)}$$

$$\frac{\Gamma, x : \mathrm{A} \vdash\ \mathrm{R} : \mathrm{B}\ , \Delta}{\Gamma \vdash\ \lambda x.\mathrm{R} : \mathrm{A}{\to}\mathrm{B}\ , \Delta}\ (\to\text{-D}) \qquad \frac{\Gamma \vdash\ \mathrm{R} : \mathrm{A}\ , \Delta \quad \Gamma,\ \mathrm{E} : \mathrm{B}\ \vdash \Delta}{\Gamma,\ \mathrm{R}{\bullet}\mathrm{E} : \mathrm{A}{\to}\mathrm{B}\ \vdash \Delta}\ (\to\text{-G})$$

$$\frac{\Gamma \vdash\ \mathrm{R} : \mathrm{A}\ , \Delta \quad \Gamma,\ \mathrm{E} : \mathrm{A}\ \vdash \Delta}{\langle \mathrm{R} \parallel \mathrm{E} \rangle : (\Gamma \vdash \Delta)}\ \text{(coupure)}$$

Table 4: Figures d'inférence de type valides pour les types simples pour $\bar{\lambda}\mu\tilde{\mu}$

II. Systèmes de types purs pour le $\bar{\lambda}\mu\tilde{\mu}$-calcul

Dans cette partie, l'on va présenter dans un premier paragraphe une syntaxe du $\bar{\lambda}\mu\tilde{\mu}$-calcul étendue pour les systèmes de types purs, puis, dans un deuxième paragraphe, le typage

[3] On conserve encore ici cette dénomination inspirée de G. Gentzen.

pour ce système, et enfin, dans un troisième paragraphe, on exposera une notion de bonne formation et de correction des types.

1. Syntaxe et réduction

On adopte ici la présentation du $\bar{\lambda}\mu\tilde{\mu}$-calcul donnée dans [4], qui a été présentée dans la section précédente. Naturellement, il va être nécessaire d'étendre la syntaxe pour l'adapter aux nécessités des systèmes de types purs, qui imposent que termes et types ne soient pas syntaxiquement séparables.

La définition formelle des systèmes de types purs va, une nouvelle fois, rester inchangée : par système de types pur explicite, on entend toujours une spécification du système sous la forme d'un triplet $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ de sortes, axiomes et règles, indiquant l'expressivité du système de type en précisant les espèces permises de dépendances, indépendamment du calcul sous-jacent et du système d'inférence de types qui viendront s'adjoindre à cette spécification.

Le calcul susmentionné est défini comme suit :

Définition 2.15 : *$\mathfrak{T}$-appelants, $\mathfrak{T}$-appelés, $\mathfrak{T}$-capsules*
Soit $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$. Soient $\mathcal{X}$ et $\mathcal{V}$ deux ensembles infinis dénombrables de variables appelées respectivement *variables d'appelant* et *variables d'appelé*. On définit alors trois ensembles d'expressions au moyen de la grammaire algébrique suivante :

$$
\begin{array}{lcll}
\mathrm{R} & ::= & x & \textit{(variable d'appelant)} \\
 & | & \lambda x{:}\mathrm{R}.\mathrm{R} & \textit{(λ-abstraction)} \\
 & | & \mu\alpha.\mathrm{C} & \textit{(μ-abstraction)} \\
 & | & \sigma & \textit{(sorte)} \\
 & | & \Pi x{:}\mathrm{R}.\mathrm{R} & \textit{(Π-quantification)} \\
\mathrm{E} & ::= & \alpha & \textit{(variable d'appelé)} \\
 & | & \mathrm{R}\bullet\mathrm{E} & \textit{(empilement)} \\
 & | & \tilde{\mu}x.\mathrm{C} & \textit{($\tilde{\mu}$-abstraction)} \\
\mathrm{C} & ::= & \langle \mathrm{R} \,||\, \mathrm{E} \rangle & \textit{(capsule)}
\end{array}
$$

$$\sigma \in \mathcal{S} \ ; \ x \in \mathcal{X} \ ; \ \alpha \in \mathcal{V}$$

dont les trois principaux symboles décrivent chacun une espèce d'expression :

- R décrit les $\mathfrak{T}$-appelants ;
- E décrit les $\mathfrak{T}$-appelés ;
- C décrit les $\mathfrak{T}$-capsules.

On observe la même convention que précédemment pour les variables, à savoir que les variables d'appelant sont dénotées par des lettres latines minuscules, et les variables d'appelé, par des lettres grecques minuscules. Pour éviter la confusion entre ces dernières et les sortes, qui sont également dénotées par des lettres grecques minuscules, on utilisera de préférence les deux premiers tiers de l'alphabet pour les variables et le dernier tiers pour les sortes.

On remarque divers changements par rapport au $\bar{\lambda}\mu\tilde{\mu}$-calcul usuel. D'une part, on est passé d'un calcul à la Curry à un calcul à la Church : les λ-abstractions sont maintenant

annotées. En outre, on remarque l'adjonction au calcul de deux constructions de type appelant : les sortes et la Π-quantification.

Le choix a été fait, ici, de traiter comme des appelants les constructions de types. Il aurait également été possible d'envisager de les traiter comme appelés. Le choix a été fait ici de les considérer comme des appelants car, dans les systèmes de types purs, il n'y a pas de distinction entre les variables sur lesquelles on peut effectuer une λ-abstraction et celles sur lesquelles on peut effectuer une −quantification, ce que l'on a souhaiter préserver ici. Ce choix sera évoqué au paragraphe suivant, avec le typage.

On introduit de façon canonique la notion de variable libre et liée comme suit :

Définition 2.16 : *variables libres/liées dans les systèmes de types purs pour* $\bar{\lambda}\mu\tilde{\mu}$
On définit les variables d'appelant (respectivement d'appelé) libres ou liées d'une expression comme appartenant à une partie de $\mathcal{X}$ (respectivement $\mathcal{V}$) définie par induction sur l'expression.
Les ensembles des variables d'appelant libres, des variables d'appelant liées, des variables d'appelé libres, des variables d'appelé liées de M sont respectivement dénotées par $fv_{\mathcal{X}}$ (M), $bv_{\mathcal{X}}$ (M), $fv_{\mathcal{V}}$ (M), $bv_{\mathcal{V}}$ (M).
Les variables d'appelant libres sont définies comme suit :

- $fv_{\mathcal{X}}(x) = \{x\}$;
- $fv_{\mathcal{X}}(\lambda x{:}\mathrm{A}.\mathrm{R}) = fv_{\mathcal{X}}(\mathrm{A}) \cup (fv_{\mathcal{X}}(\mathrm{R}) \setminus \{x\})$;
- $fv_{\mathcal{X}}(\mu\alpha.\mathrm{C}) = fv_{\mathcal{X}}(\mathrm{C})$;
- $fv_{\mathcal{X}}(\sigma) = \emptyset$;
- $fv_{\mathcal{X}}(\Pi x{:}\mathrm{A}.\mathrm{R}) = fv_{\mathcal{X}}(\mathrm{A}) \cup (fv_{\mathcal{X}}(\mathrm{R}) \setminus \{x\})$;
- $fv_{\mathcal{X}}(\alpha) = \emptyset$;
- $fv_{\mathcal{X}}(\mathrm{R}\bullet\mathrm{E}) = fv_{\mathcal{X}}(\mathrm{R}) \cup fv_{\mathcal{X}}(\mathrm{E})$;
- $fv_{\mathcal{X}}(\tilde{\mu}x.\mathrm{C}) = fv_{\mathcal{X}}(\mathrm{C}) \setminus \{x\}$;
- $fv_{\mathcal{X}}(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = fv_{\mathcal{X}}(\mathrm{R}) \cup fv_{\mathcal{X}}(\mathrm{E})$;

et les variables d'appelant liées comme suit :

- $bv_{\mathcal{X}}(x) = \emptyset$;
- $bv_{\mathcal{X}}(\lambda x{:}\mathrm{A}.\mathrm{R}) = bv_{\mathcal{X}}(\mathrm{A}) \cup bv_{\mathcal{X}}(\mathrm{R}) \cup \{x\}$;
- $bv_{\mathcal{X}}(\mu\alpha.\mathrm{C}) = bv_{\mathcal{X}}(\mathrm{C})$;
- $bv_{\mathcal{X}}(\sigma) = \emptyset$;
- $bv_{\mathcal{X}}(\Pi x{:}\mathrm{A}.\mathrm{R}) = bv_{\mathcal{X}}(\mathrm{A}) \cup bv_{\mathcal{X}}(\mathrm{R}) \cup \{x\}$;
- $bv_{\mathcal{X}}(\alpha) = \emptyset$;
- $bv_{\mathcal{X}}(\mathrm{R}\bullet\mathrm{E}) = bv_{\mathcal{X}}(\mathrm{R}) \cup bv_{\mathcal{X}}(\mathrm{E})$;

- $bv_{\mathcal{X}}\,(\tilde{\mu}x.\mathrm{C}) = bv_{\mathcal{X}}\,(\mathrm{C}) \cup \{x\}$;
- $bv_{\mathcal{X}}\,(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = bv_{\mathcal{X}}\,(\mathrm{R}) \cup bv_{\mathcal{X}}\,(\mathrm{E})$.

Les variables d'appelant libres sont définies comme suit :

- $fv_{\mathcal{V}}\,(x) = \emptyset$;
- $fv_{\mathcal{V}}\,(\lambda x{:}\mathrm{A}.\mathrm{R}) = fv_{\mathcal{V}}\,(\mathrm{A}) \cup fv_{\mathcal{V}}\,(\mathrm{R})$;
- $fv_{\mathcal{V}}\,(\mu\alpha.\mathrm{C}) = fv_{\mathcal{V}}\,(\mathrm{C}) \setminus \{\alpha\}$;
- $fv_{\mathcal{V}}\,(\sigma) = \emptyset$;
- $fv_{\mathcal{V}}\,(\Pi x{:}\mathrm{A}.\mathrm{R}) = fv_{\mathcal{V}}\,(\mathrm{A}) \cup fv_{\mathcal{V}}\,(\mathrm{R})$;
- $fv_{\mathcal{V}}\,(\alpha) = \{\alpha\}$;
- $fv_{\mathcal{V}}\,(\mathrm{R}\bullet\mathrm{E}) = fv_{\mathcal{V}}\,(\mathrm{R}) \cup fv_{\mathcal{V}}\,(\mathrm{E})$;
- $fv_{\mathcal{V}}\,(\tilde{\mu}x.\mathrm{C}) = fv_{\mathcal{V}}\,(\mathrm{C})$;
- $fv_{\mathcal{V}}\,(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = fv_{\mathcal{V}}\,(\mathrm{R}) \cup fv_{\mathcal{V}}\,(\mathrm{E})$;

et les variables d'appelé liées comme suit :

- $bv_{\mathcal{V}}\,(x) = \emptyset$;
- $fv_{\mathcal{V}}\,(\lambda x{:}\mathrm{A}.\mathrm{R}) = bv_{\mathcal{V}}\,(\mathrm{A}) \cup bv_{\mathcal{V}}\,(\mathrm{R})$;
- $bv_{\mathcal{V}}\,(\mu\alpha.\mathrm{C}) = bv_{\mathcal{V}}\,(\mathrm{C}) \cup \{\alpha\}$;
- $fv_{\mathcal{V}}\,(\sigma) = \emptyset$;
- $fv_{\mathcal{V}}\,(\Pi x{:}\mathrm{A}.\mathrm{B}) = bv_{\mathcal{V}}\,(\mathrm{A}) \cup bv_{\mathcal{V}}\,(\mathrm{B})$;
- $bv_{\mathcal{V}}\,(\alpha) = \emptyset$;
- $bv_{\mathcal{V}}\,(\mathrm{R}\bullet\mathrm{E}) = bv_{\mathcal{V}}\,(\mathrm{R}) \cup bv_{\mathcal{V}}\,(\mathrm{E})$;
- $bv_{\mathcal{V}}\,(\tilde{\mu}x.\mathrm{C}) = bv_{\mathcal{V}}\,(\mathrm{C})$;
- $bv_{\mathcal{V}}\,(\langle \mathrm{R} \parallel \mathrm{E} \rangle) = bv_{\mathcal{V}}\,(\mathrm{R}) \cup bv_{\mathcal{V}}\,(\mathrm{E})$.

Tout comme précédemment, cette définition produit une notion d'α-conversion, qui permet d'identifier les termes aux classes d'équivalences pour l'α-conversion des expressions, et l'on peut de nouveau adopter la convention de Barendregt pour le choix des représentants.

La réduction demeure inchangée par rapport au $\bar{\lambda}\mu\tilde{\mu}$-calcul usuel, hormis bien évidemment le fait que l'on est passé d'un calcul à la Curry à un calcul à la Church. Elle est donc définie comme suit :

Définition 2.17 : *réduction (de tête) dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*
La réduction, notée $\to$, est définie comme la clôture contextuelle de l'union des trois axiomes de réduction suivants :

$$\begin{array}{lrcl} (\lambda) & \langle \lambda x{:}\mathrm{A}.\mathrm{M} \parallel \mathrm{N}\bullet\mathrm{E}\rangle & \to & \langle \mathrm{M}[\mathrm{N}/x] \parallel \mathrm{E}\rangle \\ (\mu) & \langle \mu\alpha.\mathrm{C} \parallel \mathrm{E}\rangle & \to & \mathrm{C}[\mathrm{E}/\alpha] \\ (\tilde{\mu}) & \langle \mathrm{R} \parallel \tilde{\mu}x.\mathrm{C}\rangle & \to & \mathrm{C}[\mathrm{R}/x] \end{array}$$

Une réduction est dite *de tête* si elle est un axiome, *i.e.* la relation de réduction de tête $\xrightarrow[\mathrm{H}]{}$ est définie comme l'union des trois axiomes (λ), (μ), $(\tilde{\mu})$.

Tout comme dans le $\bar{\lambda}\mu\tilde{\mu}$-calcul, il convient d'observer que cette réduction a une paire critique engendrée par les deux dernières règles, et que celle-ci n'est pas joignable. La réduction $\to$ n'est donc pas confluente. On pourra se reporter à la partie précédente à ce propos (paragraphe 1.2).

Toutefois, il est possible d'intervertir deux pas de réduction quand le rédex impliqué dans le premier est inclus dans celui impliqué dans le second, *i.e.* :

Lemme 2.7 : *Interversion de rédex*
Soient C et D deux capsules et $\mathcal{G}[\,]$ un contexte tels que $\mathrm{C} = \mathcal{G}[\mathrm{D}]$. On suppose que $\mathrm{C} \longrightarrow \mathcal{G}[\mathrm{D}'] \xrightarrow[\mathrm{H}]{} \mathrm{C}'$ avec $\mathrm{D} \xrightarrow[\mathrm{H}]{} \mathrm{D}'$.
Alors il existe $\mathcal{G}'[\,]$ tel $\mathrm{C}' \xrightarrow[\mathrm{H}]{} \mathcal{G}'[\mathrm{D}] \xrightarrow{*} \mathcal{G}'[\mathrm{D}'] = \mathrm{C}'$.
Démonstration.
Il y a différents cas possibles :

a. $\mathrm{C} = \langle \lambda x{:}\mathcal{H}[\mathrm{D}].\mathrm{M} \parallel \mathrm{N}\bullet\mathrm{E}\rangle$.
Alors $\mathrm{C}' = \langle \mathrm{M}[\mathrm{N}/x] \parallel \mathrm{E}\rangle$, *et l'on a bien* $\mathrm{C} \xrightarrow[\mathrm{H}]{} \langle \mathrm{M}[\mathrm{N}/x] \parallel \mathrm{E}\rangle = \mathrm{C}'$.

b. $\mathrm{C} = \langle \lambda x{:}\mathrm{A}.\mathcal{H}[\mathrm{D}] \parallel \mathrm{N}\bullet\mathrm{E}\rangle$.
Alors $\mathrm{C}' = \langle \mathcal{H}[\mathrm{D}'][\mathrm{N}/x] \parallel \mathrm{E}\rangle$, *et l'on a bien* $\mathrm{C} \xrightarrow[\mathrm{H}]{} \langle \mathcal{H}[\mathrm{D}][\mathrm{N}/x] \parallel \mathrm{E}\rangle \longrightarrow \mathrm{C}'$.

c. $\mathrm{C} = \langle \lambda x{:}\mathrm{A}.\mathrm{M} \parallel \mathcal{H}[\mathrm{D}]\bullet\mathrm{E}\rangle$.
Alors $\mathrm{C}' = \langle \mathrm{M}[\mathcal{H}[\mathrm{D}']/x] \parallel \mathrm{E}\rangle$, *et l'on a bien* $\mathrm{C} \xrightarrow[\mathrm{H}]{} \langle \mathrm{M}[\mathcal{H}[\mathrm{D}]/x] \parallel \mathrm{E}\rangle \xrightarrow{*} \mathrm{C}'$.

d. $\mathrm{C} = \langle \lambda x{:}\mathrm{A}.\mathrm{M} \parallel \mathrm{N}\bullet\mathcal{H}[\mathrm{D}]\rangle$.
Alors $\mathrm{C}' = \langle \mathrm{M}[\mathrm{N}/x] \parallel \mathcal{H}[\mathrm{D}']\rangle$, *et l'on a bien* $\mathrm{C} \xrightarrow[\mathrm{H}]{} \langle \mathrm{M}[\mathrm{N}/x] \parallel \mathcal{H}[\mathrm{D}]\rangle \longrightarrow \mathrm{C}'$.

e. $\mathrm{C} = \langle \mu\alpha.\mathcal{H}[\mathrm{D}] \parallel \mathrm{E}\rangle$.
Alors $\mathrm{C}' = \mathcal{H}[\mathrm{D}'][\mathrm{E}/\alpha]$, *et l'on a bien* $\mathrm{C} \xrightarrow[\mathrm{H}]{} \mathcal{H}[\mathrm{D}][\mathrm{E}/\alpha] \longrightarrow \mathrm{C}'$.

f. $\mathrm{C} = \langle \mu\alpha.\mathrm{Z} \parallel \mathcal{H}[\mathrm{D}]\rangle$.
Alors $\mathrm{C}' = \mathrm{Z}[\mathcal{H}[\mathrm{D}']/\alpha]$, *et l'on a bien* $\mathrm{C} \xrightarrow[\mathrm{H}]{} \mathrm{Z}[\mathcal{H}[\mathrm{D}]/\alpha] \xrightarrow{*} \mathrm{C}'$.

g. $C = \langle \mathcal{H}[D] \parallel \tilde{\mu}x.Z \rangle$.
Alors $C' = Z[\mathcal{H}[D']/x]$, *et l'on a bien* $C \xrightarrow[H]{} Z[\mathcal{H}[D]/x] \xrightarrow{*} C'$.

h. $C = \langle R \parallel \tilde{\mu}x.\mathcal{H}[D] \rangle$.
Alors $C' = \mathcal{H}[D'][R/x]$, *et l'on a bien* $C \xrightarrow[H]{} \mathcal{H}[D][R/x] \longrightarrow C'$.

Dans tous les cas, l'on peut bien effectuer la réduction de tête en premier. Q.E.D.

On en déduit la possibilité de réorganiser une réduction infinie selon une stratégie allant des capsules les plus intérieures aux capsules les plus extérieures :

Corollaire 2.8 : *Réduction par l'extérieur*
Soit $(C_n)_{n\in\mathbb{N}}$ une chaîne de réduction infinie.

(*i*) Si $(C_n)_{n\in\mathbb{N}}$ ne comprend aucun pas de réduction de tête, alors $C_0 = \mathcal{G}[D_0]$ avec D_0 admettant une réduction infinie comprenant un nombre non nul de pas de réduction de tête.

(*ii*) Si $(C_n)_{n\in\mathbb{N}}$ comprend une infinité de pas de réduction de tête, alors C_0 admet une réduction de tête infinie.

(*iii*) Si $(C_n)_{n\in\mathbb{N}}$ comprend un nombre fini non nul de pas de réduction de tête et C_0 n'admet pas de réduction de tête infinie, alors il existe $k > 0$ tel que $C_0 \xrightarrow[H]{+} C_k = \mathcal{G}[D_0]$ avec D_0 admettant une réduction infinie comprenant un nombre non nul de pas de réduction de tête.

Démonstration.

(i) C'est une simple conséquence du lemme des tiroirs.

(ii) On a $C_0 \xrightarrow[k]{} C_k \xrightarrow[H]{} C_{k+1}$, *où le pas ainsi isolé est le premier pas de réduction de tête. On a alors, en appliquant le lemme précédent k fois,* $C_0 \xrightarrow[H]{} C'_1 \xrightarrow{*} C_{k+1} \longrightarrow \cdots$. *On poursuit ainsi la construction en appliquant le même processus à* C'_1 *pour obtenir une réduction de tête de longueur non bornée, et l'on peut conclure.*

(iii) Soit j tel qu'à partir de C_j *aucun pas de réduction ne soit de tête. De même que précédemment, l'on obtient* $C_0 \xrightarrow[H]{+} C'_k \xrightarrow{*} C_j \longrightarrow \cdots$, *Une alternative se présente alors :*

- *si* $C'_k \xrightarrow{*} C_j$ *ne contient pas de pas de réduction de tête, on peut alors conclure en appliquant le cas (i) à* C'_k ;
- *sinon, on répète la construction précédente, ce qui augmente la taille de la réduction de tête issue de* C_0.

Comme C_0 *n'admet pas de réduction de tête infinie, nécessairement la première branche de l'alternative finit par être réalisée.* Q.E.D.

Si l'on se trouve dans le troisième cas, l'on peut ensuite réappliquer la transformation indiquée dans ce corollaire à la capsule incluse, et ainsi de suite. On obtient dans tous les cas une chaîne de réduction dite *imbriquée* :

Définition 2.18 : *réduction imbriquée dans les systèmes de types purs pour* $\bar{\lambda}\mu\tilde{\mu}$
Soit $C_0 \longrightarrow \cdots \longrightarrow C_n$ une chaîne de réduction ; pour tout i, il existe un contexte $\mathcal{G}_i[\,]$ tel que

$C_i = \mathcal{G}_i[D_i]$ et $C_{i+1} = \mathcal{G}_i[D'_i]$ avec $D_i \xrightarrow[H]{} D'_i$.
La chaîne de réduction est dite *imbriquée* si elle vérifie la propriété suivante : pour tout i, il existe un contexte $\mathcal{H}_i[\,]$ tel que $\mathcal{G}_{i+1}[\,] = \mathcal{G}_i[\mathcal{H}_i[\,]]$.

On voit donc que la stratégie par l'extérieur est perpétuelle. De telles réductions par l'extérieur seront intéressantes par la suite pour ce qui concerne la normalisation forte.

2. Typage

Les jugements de type des systèmes de types purs pour le $\bar{\lambda}\mu\tilde{\mu}$-calcul sont analogues à ceux des types simples : il s'agit de séquents à formule active, qui peuvent donc être de trois formes, selon que leur formule active est en partie gauche, en partie droite, ou absente.

Définition 2.19 : *assertion, contexte, jugement de typage des systèmes de types purs pour* $\bar{\lambda}\mu\tilde{\mu}$
Soit $\mathfrak{T}$ un système de types purs. On appelle *assertion de typage* un couple de $\mathfrak{T}$-expressions noté M : R, où M peut être un appelant ou un appelé, et R est un appelant. Dans un tel couple, le premier élément est appelé *sujet* de l'assertion et le second, *prédicat* de l'assertion. Une assertion peut être dite *active*.
On appelle *contexte gauche de typage* une suite finie – non un ensemble – d'assertions de typage non actives dont les sujets sont des variables d'appelant distinctes. Le contexte gauche vide est noté [ ]. On appelle *contexte droit de typage* un ensemble d'assertions de typage non actives dont les sujets sont des variables d'appelés distinctes. Le contexte droit vide est noté $\emptyset$. Pour ces deux types de contexte, la concaténation de contextes de supports disjoints est notée par une virgule.
On appelle *jugement de typage* une expression de l'une des trois formes suivante, Γ et Δ étant respectivement un contexte gauche de typage et un contexte droit de typage :

- $\Gamma, E : A \vdash \Delta$ où E est un appelé ;
- $\Gamma \vdash R : A, \Delta$ où R est un appelant ;
- C : $(\Gamma \vdash \Delta)$ où C est une capsule.

Un jugement est dit *valide* s'il dérive de l'application des figuresd'inférence présentées dans la table 5. L'indice $\mathfrak{T}$ peut être omis s'il n'y a pas d'ambiguïté.

On remarque dans les figures de la table *infra* une relation $\equiv$. Il ne s'agit pas ici de la clôture réflexive, symétrique et transitive de $\longrightarrow$ (ce qui n'aurait pas de sens car cette relation n'est pas confluente), mais de celle de $\xrightarrow[(\lambda)]{}$.

En observant la figure (Π-D), on peut voir pourquoi le choix a été fait de traiter les types comme des appelants et non comme des appelés : en effet, on souhaite préserver le fait qu'une λ-abstraction est associée à une Π-quantification sur la même variable. Lorsque l'on considère la figure (Π-G), on peut remarquer qu'il pourrait être possible d'introduire des types duaux en ajoutant des constructeurs de types aux appelés, car une fois la variable liée, elle peut être identifiée par α-conversion.

$$\frac{(\sigma,\tau)\in\mathcal{A}}{[]\vdash \sigma:\tau\,;\emptyset}\ \text{(axiome)}$$

$$\frac{\Gamma\vdash A:\rho\,;\Delta \quad \Gamma,x:A\vdash B:\sigma \quad (\rho,\sigma,\tau)\in\mathcal{A}}{\Gamma\vdash (\Pi x{:}A.B):\tau\,;\Delta}\ \text{(règle)}$$

$$\frac{\Gamma\vdash A:\sigma\,;\Delta}{\Gamma,x:A\vdash x:A\,;\Delta}\ \text{(hyp-D)} \qquad \frac{\Gamma\vdash A:\sigma\,;\Delta \quad A\notin\mathcal{S}}{\Gamma;\,\alpha:A\vdash\Delta,\alpha:A}\ \text{(hyp-G)}$$

$$\frac{\Gamma\vdash R:B\,;\Delta \quad \Gamma\vdash A:\sigma\,;\Delta}{\Gamma,x:A\vdash R:B\,;\Delta}\ \text{(aff-g-D)} \qquad \frac{\Gamma;\,E:B\vdash\Delta \quad \Gamma\vdash A:\sigma\,;\Delta \quad A\notin\mathcal{S}}{\Gamma,x:A;\,E:B\vdash\Delta}\ \text{(aff-g-G)}$$

$$\frac{\Gamma\vdash R:B\,;\Delta \quad \Gamma\vdash A:\sigma\,;\Delta}{\Gamma\vdash R:B\,;\Delta,\alpha:A}\ \text{(aff-d-D)} \qquad \frac{\Gamma;\,E:B\vdash\Delta \quad \Gamma\vdash A:\sigma\,;\Delta \quad A\notin\mathcal{S}}{\Gamma;\,E:B\vdash\Delta,\alpha:A}\ \text{(aff-d-G)}$$

$$\frac{\Gamma\vdash (\Pi x{:}A.B){:}\sigma\,;\Delta \quad \Gamma,x:A\vdash R:B\,;\Delta}{\Gamma\vdash (\lambda x{:}A.R):(\Pi x{:}A.B)\,;\Delta}\ \text{(Π-D)} \qquad \frac{\Gamma\vdash (\Pi x{:}A.B){:}\sigma\,;\Delta \quad \Gamma;\,E{:}B[R/x]\vdash\Delta \quad \Gamma\vdash R{:}A\,;\Delta}{\Gamma;\,R\bullet E:\Pi x{:}A.B\vdash\Delta}\ \text{(Π-G)}$$

$$\frac{\Gamma\vdash R:B\,;\Delta \quad \Gamma\vdash A:\sigma\,;\Delta \quad A\equiv B}{\Gamma\vdash R:A\,;\Delta}\ \text{(≡-D)} \qquad \frac{\Gamma;\,E:B\vdash\Delta \quad \Gamma\vdash A:\sigma\,;\Delta \quad A\equiv B}{\Gamma;\,E:A\vdash\Delta}\ \text{(≡-G)}$$

$$\frac{C:(\Gamma\vdash\Delta',\alpha:A)}{\Gamma\vdash \mu\alpha.C:A\,;\Delta'}\ (\mu) \qquad \frac{C:(\Gamma',x:A\vdash\Delta)}{\Gamma';\,\tilde{\mu}x.C:A\vdash\Delta}\ (\tilde{\mu})$$

$$\frac{\Gamma\vdash R:A\,;\Delta \quad \Gamma;\,E:A\vdash\Delta}{\langle R\parallel E\rangle:(\Gamma\vdash\Delta)}\ \text{(coupure)}$$

Table 5: Figures d'inférence de type pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$

3. Correction des types

Comme il est habituel avec les systèmes de types purs, la notion de contextes bien formés et de correction des types revêt une importance particulière. Pour ce faire, on introduit la définition suivante :

Définition 2.20 : *couple de contextes bien formé*
Un couple de contextes (Γ, Δ) où les contextes sont respectivement de la forme $\Gamma = [x_1 : \mathrm{T}_1, \ldots, x_n : \mathrm{T}_n]$ et $\Delta = \{\alpha_1 : \mathrm{A}_1, \ldots, \alpha_k : \mathrm{A}_k\}$ est dit *bien formé* si les conditions suivantes sont toutes deux remplies :

i. pour tout i de $[\![1, n]\!]$, il existe une sorte σ_i telle que :
$[x_1 : \mathrm{T}_1, \ldots, x_{i-1} : \mathrm{T}_{i-1}] \vdash \mathrm{T}_i : \sigma_i, \Delta$;

ii. pour tout j de $[\![1, k]\!]$, il existe une sorte τ_j telle que :
$\Gamma \vdash \mathrm{A}_j : \tau_j, \Delta \setminus \{\alpha_j : \mathrm{A}_j\}$.

Naturellement, on peut montrer que les coupless de contextes intervenant dans un jugement validement dérivé sont bien formées :

Lemme 2.9 : *Bonne formation de couple*
Soit $\Gamma; \mathrm{E} : \mathrm{A} \vdash \Delta, \Gamma \vdash \mathrm{R} : \mathrm{A}; \Delta$ ou $\mathrm{C} : (\Gamma \vdash \Delta)$ un jugement valide.
Alors (Γ, Δ) est bien formé.

Démonstration.
Par induction sur le jugement en discriminant selon la dernière figure :

a. *(axiome) : ce cas est trivial.*

b. *(règle), (Π-G), (Π-D), (≡-G), (≡-D), (μ), (μ̃), (coupure) : ces cas sont évidents par hypothèse d'induction, car toutes ces figures ont une prémisse ayant le même couple de contextes que la conclusion.*

c. *(hyp-D)* : $\dfrac{\Gamma \vdash \mathrm{A} : \sigma ; \Delta}{\Gamma, x : \mathrm{A} \vdash x : \mathrm{A} ; \Delta}$
Par hypothèse d'induction, (Γ, Δ) *est bien formée, et pour montrer que* $((\Gamma, x : \mathrm{A}), \Delta)$ *est bien formée, il suffit donc de montrer qu'il existe* σ *telle que* $\Gamma \vdash A : \sigma; \Delta$*, et cela est assuré par la prémisse.*

d. *(hyp-G)* : $\dfrac{\Gamma \vdash \mathrm{A} : \sigma ; \Delta \quad \mathrm{A} \notin \mathcal{S}}{\Gamma; \alpha : \mathrm{A} \vdash \Delta, \alpha : \mathrm{A}}$
Par hypothèse d'induction, (Γ, Δ) *est bien formée. Pour montrer que* $(\Gamma, (\Delta, \alpha : \mathrm{A}))$ *est bien formée, il suffit de montrer deux choses : qu'il existe* σ *telle que* $\Gamma \vdash A : \sigma; \Delta$*, ce qui est assuré par la prémisse, et que, si* $\Gamma = [x_1 : \mathrm{T}_1, \ldots, x_n : \mathrm{T}_n]$*, alors pour tout* i *de* $[\![1, n]\!]$*, il existe une sorte* σ_i *telle que* $[x_1 : \mathrm{T}_1, \ldots, x_{i-1} : \mathrm{T}_{i-1}] \vdash \mathrm{T}_i : \sigma_i, \Delta, \alpha : \mathrm{A}$*. La bonne formation de* (Γ, Δ) *donne que* $[x_1 : \mathrm{T}_1, \ldots, x_{i-1} : \mathrm{T}_{i-1}] \vdash \mathrm{T}_i : \sigma_i, \Delta$*, et, avec la prémisse* $\Gamma \vdash A : \sigma; \Delta$*, on peut obtenir ce que l'on souhaite par une figure d'(aff-d-D).*

e. *(aff-d-D), (aff-g-D) : ces deux cas sont analogues à (hyp-D).*

f. *(aff-d-G), (aff-g-G) : ces deux cas sont analogues à (hyp-G).*

Q.E.D.

À l'aide des quatre figures d'affaiblissement, on peut alors déduire le corollaire suivant :

Corollaire 2.10 : *Correction des prédicats de contexte*
Soit (Γ, Δ) un couple bien formé. Alors pour toute assertion $x : A$ dans Γ ou pour toute assertion $\alpha : A$ dans Δ, il existe une sorte $\sigma \in \mathcal{S}$ telle que $\Gamma \vdash A : \sigma ; \Delta$.

Avec tout ceci, on peut maintenant démontrer le théorème de correction des types, qui s'énonce de manière très analogue à celui des systèmes de types purs présentés au chapitre I :

Théorème 2.11 : *Correction des types dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*
Soit $\Gamma \vdash \mathrm{R} : \mathrm{A} ; \Delta$ ou $\Gamma ; \mathrm{E} : \mathrm{A} \vdash \Delta$ un jugement de type valide. Alors soit $A \in \mathcal{S}$, soit il existe une sorte $\sigma \in \mathcal{S}$ telle que $\Gamma \vdash \mathrm{A} : \sigma ; \Delta$.

Démonstration.
Par induction sur la dérivation du jugement.
Tous les cas hormis (μ) *et* $(\tilde{\mu})$ *sont évidents car, pour toutes ces figures, soit* A *est une sorte, soit la propriété* $\Gamma \vdash \mathrm{A} : \sigma ; \Delta$ *est garantie par une prémisse.*
Restent les cas de (μ) *et* $(\tilde{\mu})$. *Pour ceux-ci, la propriété dérive directement du corollaire (2.10).* Q.E.D.

Il est intéressant de remarquer que, comme on l'avait déjà noté au chapitre précédent, on voit surgir à nouveaux deux notions de correction des types : l'une pour les prédicats de contexte, et l'une pour les prédicats d'assertion. Ces notions sont les mêmes que celles des systèmes de types purs usuels (cf. chapitre I).

III. Propriétés remarquables

Dans cette partie, on donne tout d'abord dans un premier paragraphe quelques lemmes de dérivation, puis, dans les deux paragraphes suivants, deux résultats importants : la réduction du sujet et la correction des types.

1. Lemmes d'inférence de types

Les lemmes suivants sont utiles pour reconstruire des dérivations de types.

Lemme 3.12 : *Initialisation pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*
Soit (Γ, Δ) un couple bien formé. Alors :

(*i*) si $(\sigma, \tau) \in \mathcal{A}$, alors $\Gamma \vdash \sigma : \tau ; \Delta$;

(*ii*) si $(x : \mathrm{T}) \in \Gamma$, alors $\Gamma \vdash x : \mathrm{T} ; \Delta$;

(*iii*) si $(\alpha : \mathrm{A}) \in \Delta$, alors $\Gamma ; \alpha : \mathrm{A} \vdash \Delta$.

Démonstration.
Par induction sur la taille n du couple, i.e. *le nombre total d'assertions dans ses deux composantes.*

- $n = 0$. *Dans ce cas, le point (i) est une application d'une figure d'(axiome) et les points (ii) et (iii) sont triviaux.*
- $n > 0$. *On a alors :*

  (i) *On a, par hypothèse d'induction,* $\Gamma' \vdash \sigma : \tau \,;\Delta'$ *avec, selon les cas,* $\Gamma = \Gamma'$ *et* $\Delta = \Delta', \alpha : \mathrm{A}$ *ou bien* $\Gamma = \Gamma', x : \mathrm{A}$ *et* $\Delta = \Delta'$. *En outre, par définition de la bonne formation, on a* $\Gamma' \vdash \mathrm{A} : \sigma \,;\Delta$ *ce qui permet d'appliquer la figure d'affaiblissement adéquate pour obtenir* $\Gamma \vdash \sigma : \tau \,;\Delta$.

  (ii) *Soit on se trouve dans un cas analogue à celui de la propriété (i), soit on a* $\Gamma = \Gamma', x : \mathrm{T}$. *Dans ce dernier cas, par définition de la bonne formation, on a* $\Gamma' \vdash \mathrm{T} : \sigma \,;\Delta$, *ce qui permet d'obtenir, par une figure d'(hyp-R), que* $\Gamma, x : \mathrm{T} \vdash x : \mathrm{T} \,;\Delta$.

  (iii) *Cette démonstration est symétrique de la précédente.*

Q.E.D.

Lemme 3.13 : *Engendrement pour les systèmes de types purs pour* $\bar{\lambda}\mu\tilde{\mu}$
Soit un jugement de typage valide.

(*i*) Si le jugement est de la forme $\Gamma \vdash \mathrm{R} : \mathrm{T} \,;\Delta$, alors R doit avoir l'une des formes suivantes :

  a. $\mathrm{R} = \sigma \in \mathcal{S}$. Dans ce cas, il existe $\tau \in \mathcal{S}$ telle que $\mathrm{T} \equiv \tau$ et $(\sigma, \tau) \in \mathcal{A}$.
  b. $\mathrm{R} = x \in \mathcal{X}$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et $(x : \mathrm{U}) \in \Gamma$ telles que $\Gamma \vdash \mathrm{U} : \tau \,;\Delta$ and $\mathrm{T} \equiv \mathrm{U}$.
  c. $\mathrm{R} = \Pi x{:}\mathrm{A}.\mathrm{B}$. Dans ce cas, il existe $(\rho, \sigma, \tau) \in \mathcal{R}$ tel que $\Gamma \vdash \mathrm{A} : \rho \,;\Delta$, $\Gamma, x : \mathrm{A} \vdash \mathrm{B} : \sigma \,;\Delta$, et $\mathrm{T} \equiv \tau$.
  d. $\mathrm{R} = \lambda x{:}\mathrm{A}.\mathrm{N}$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et un terme B tels que $\Gamma \vdash (\Pi x{:}\mathrm{A}.\mathrm{B}) : \tau \,;\Delta$ and $\Gamma, x : \mathrm{A} \vdash \mathrm{N} : \mathrm{B} \,;\Delta$ and $\mathrm{T} \equiv \Pi x : \mathrm{A}.\mathrm{B}$
  e. $\mathrm{R} = \mu\alpha.\mathrm{C}$. Dans ce cas, il existe un terme U tel que $\mathrm{C} : (\Gamma \vdash \Delta, \alpha : \mathrm{U})$ et $\mathrm{T} \equiv \mathrm{U}$.

(*ii*) Si le jugement est de la forme $\Gamma; E : T \vdash \Delta$, alors E doit avoir l'une des formes suivantes :

  a. $\mathrm{E} = \alpha \in \mathcal{V}$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et $(\alpha : \mathrm{U}) \in \Delta$ tels que $\Gamma; \mathrm{U} : \tau \vdash \Delta$ et $\mathrm{T} \equiv \mathrm{U}$.
  b. $\mathrm{E} = \mathrm{R}\bullet\mathrm{F}$. Dans ce cas, il existe $\tau \in \mathcal{S}$, des termes A et B et une variable $x \in \mathcal{X}$ telle que $\Gamma \vdash (\Pi x{:}\mathrm{A}.\mathrm{B}) : \tau \,;\Delta$, $\Gamma \vdash \mathrm{R} : \mathrm{A} \,;\Delta$, $\Gamma; \mathrm{F} : \mathrm{B}[\mathrm{R}/x] \vdash \Delta$ et $\mathrm{T} \equiv \Pi x{:}\mathrm{A}.\mathrm{B}$
  c. $\mathrm{E} = \tilde{\mu}x.\mathrm{C}$. Dans ce cas, il existe un terme U tel que $\mathrm{C} : (\Gamma, x : \mathrm{U} \vdash \Delta)$ et $\mathrm{T} \equiv \mathrm{U}$.

(*iii*) Si le jugement est de la forme $\mathrm{C} : (\Gamma \vdash \Delta)$ avec $\mathrm{C} = \langle \mathrm{R} \,||\, \mathrm{E} \rangle$, alors il existe $\tau \in \mathcal{S}$ et un terme A tel que $\Gamma \vdash \mathrm{R} : \mathrm{A} \,;\Delta$ et $\Gamma; \mathrm{E} : \mathrm{A} \vdash \Delta$.

Démonstration.
Bien que son énoncé semble plus complexe en raison du plus grand nombre de cas possibles, ce lemme se montre exactement comme son analogue pour les systèmes de types purs implicites présenté au chapitre I.
Q.E.D.

Lemme 3.14 : *Dépendances dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*
Les énoncés suivants sont vrais :

(*i*) Soit $\Gamma \vdash X : T ; \Delta$ ou $\Gamma; X : T \vdash \Delta$ un jugement valide. Alors :

  a. toute variable libre de X est dans le domaine de Γ ou Δ selon qu'elle est d'appelant ou d'appelé ;
  b. toute variable libre de T est dans le domaine de Γ.

(*ii*) Soit (Γ, Δ) un couple bien formé. Alors :

  a. si $\Gamma = \Gamma', x : T, \Gamma''$, alors toute variable libre de T est dans le domaine de Γ' ;
  b. si $\Delta = \Delta, \alpha : A$, alors toute variable libre de A est dans le domaine de Γ.

Démonstration.

(i) Cette propriété se démontre facilement par induction sur la dérivation (pour le point b., la condition $A \notin \mathcal{S}$ dans les figures (hyp-G), (aff-d-G) et (aff-g-G) assure que seules les variables d'appelant sont des variables de types).

(ii) C'est un corollaire du point (i) et du théorème de correction des types (2.11).

Q.E.D.

2. Réduction du sujet

Le but ici est de montrer que les systèmes de types purs définis ici pour le $\bar{\lambda}\mu\tilde{\mu}$-calcul vérifient la réduction du sujet. Pour ce faire, on commence par montrer un lemme de substitution, dont il n'est plus nécessaire d'aborder l'utilité, qui a été discutée dans les chapitres précédents.

Lemme 3.15 : *Substitution dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*
Soit un jugement de type valide.

(*i*) Si ce jugement est de la forme $\Gamma \vdash R : T ; \Delta$, alors :

  a. si $\Gamma = \Gamma', x : U, \Gamma''$ et $\Gamma' \vdash S : U ; \Delta$,
  alors $\Gamma', \Gamma''[S/x] \vdash R[S/x] : T[S/x] ; \Delta[S/x]$;
  b. si $\Delta = \Delta', \alpha : A$ et $\Gamma; F : A \vdash \Delta'$,
  alors $\Gamma \vdash R[F/\alpha] : T ; \Delta'$.

(*ii*) Si ce jugement est de la forme $\Gamma; E : T \vdash \Delta$, alors :

  a. si $\Gamma = \Gamma', x : U, \Gamma''$ et $\Gamma' \vdash S : U ; \Delta$,
  alors $\Gamma', \Gamma''[S/x]; E[S/x] : T[S/x] \vdash \Delta[S/x]$;

b. si $\Delta = \Delta', \alpha : A$ and $\Gamma; \boxed{F : A} \vdash \Delta'$,
alors $\Gamma; \boxed{E[F/\alpha] : T} \vdash \Delta'$.

(*iii*) Si le jugement est de la forme $C : (\Gamma \vdash \Delta)$ avec $C = \langle R \parallel E \rangle$, alors :

a. si $\Gamma = \Gamma', x : U, \Gamma''$ et $\Gamma' \vdash \boxed{S : U}; \Delta$,
alors $C[S/x] : (\Gamma', \Gamma''[S/x] \vdash \Delta[S/x])$;

b. si $\Delta = \Delta', \alpha : A$ and $\Gamma; \boxed{F : A} \vdash \Delta'$,
alors $C[F/\alpha] : (\Gamma \vdash \Delta')$.

Démonstration.
Par induction directe sur la dérivation du jugement. Q.E.D.

Théorème 3.16 : *Réduction du sujet dans les systèmes de types purs pour* $\bar{\lambda}\mu\tilde{\mu}$
Si $C : (\Gamma \vdash \Delta)$ et $C \longrightarrow C'$ alors $C' : (\Gamma \vdash \Delta)$.

Démonstration.
Soit C *égale à* $\langle R \parallel E \rangle$. *Il y a trois cas, selon la règle de réduction qui a été utilisée :*

(λ) *Dans ce cas,* $R = \lambda x{:}A.M$, $E = N \bullet F$, *et* $C' = \langle M[N/x] \parallel F \rangle$. *Par le lemme d'engendrement (3.13), il existe un type* T *tel que* $\Gamma \vdash \boxed{R : T}; \Delta$ *et* $\Gamma; \boxed{E : T} \vdash \Delta$. *En utilisant deux fois le lemme d'engendrement (3.13) et la figure de conversion, l'on obtient qu'il existe* B *tel que* $T \equiv \Pi x{:}A.B$ *et* $\Gamma, x : A \vdash \boxed{M : B}; \Delta$ *et* $\Gamma; \boxed{F : B[N/x]} \vdash \Delta$ *et* $\Gamma \vdash \boxed{N : A}; \Delta$. *En utilisant le lemme de substitution Using the substitution lemma (3.15) avec le premier et le dernier de ces trois jugements, on obtient que* $\Gamma \vdash \boxed{M[N/x] : B}; \Delta[N/x]$. *Comme le couple* (Γ, Δ) *est bien formé et que* x *n'apparaît pas dans* Γ, *par le lemme (3.14), on obtient que* $\Gamma \vdash \boxed{M[N/x] : B}; \Delta$, *et donc, par une coupure,* $C' : (\Gamma \vdash \Delta)$.

(μ) *Dans ce cas,* $R = \mu\alpha.C''$ *et* $C' = C''[E/\alpha]$. *Par le lemme d'engendrement (3.13), il existe un type* V *tel que* $\Gamma \vdash \boxed{R : V}; \Delta$ *et* $\Gamma; \boxed{E : V} \vdash \Delta$. *En utilisant de nouveau le lemme d'engendrement (3.13), on obtient que* $C'' : (\Gamma \vdash \Delta, \alpha : V)$, *et donc, par le lemme de substitution (3.15), que* $C' : (\Gamma \vdash \Delta)$.

($\tilde{\mu}$) *Dans ce cas,* $R = \mu\alpha.C''$ *and* $C' = C''[E/\alpha]$. *Cette démonstration est analogue à la précédente.* Q.E.D.

3. Normalisation forte pour le λ-cube

Dans ce paragraphe, on va montrer que les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$ dont la spécification correspond à un système du λ-cube sont fortement normalisants. On ne considérera donc que des systèmes du λ-cube, *i.e.* avec deux sortes $*$ et $\Box$, un seul axiome $* : \Box$, et des règles prises parmi les règles élémentaires et comprenant au moins $(*, *, *)$.

Soit $\mathfrak{T}$ un système de types pur du λ-cube.
On définit trois traductions $[\![\cdot]\!]$, $[\![\cdot]\!]^\eta$, $[\![\cdot]\!]$ sur les termes typables (respectivement, capsules, appelants, appelés), et vérifiant :

- si $R : A$, alors $[\![R]\!]^\eta : [\![A]\!]^\eta$;

- si $E : A$, alors $[\![E]\!] : (\Pi r{:}[\![A]\!]^\eta.U_E) = T_E$ avec U_E un type donné pour un certain η ;
- si $C = \langle R \parallel E \rangle$, avec $R : A$ et $E : A$, alors $[\![C]\!] : [\![A]\!]^\eta$ pour un certain η ;

ceci s'entendant dans un contexte adapté.

La traduction est définie ainsi (lorsqu'elles sont employées, p et θ désignent des variables inédites) :

$[\![\langle R \parallel E \rangle]\!] = p(\lambda\eta{:}T_E.([\![E]\!]\ [\![R]\!]^\eta)\ [\![E]\!])$

$[\![\sigma]\!]^\eta = \sigma$;

$[\![\Pi x{:}A.B]\!]^\eta = \Pi x{:}[\![A]\!]^\eta.[\![B]\!]^\eta$;

$[\![x]\!]^\eta = x$;

$[\![\lambda x{:}A.R]\!]^\eta = \lambda x{:}[\![A]\!]^\eta.[\![R]\!]^\eta$;

$[\![\mu\alpha.C]\!]^\eta = [\![C]\!][\eta/\alpha]$;

$[\![\alpha]\!] = \alpha$;

$[\![R\bullet E]\!] = \lambda r{:}[\![A]\!]^\theta.(p(\lambda\eta{:}T_E.([\![E]\!]\ (r[\![R]\!]^\eta))\ [\![E]\!]))$ où $R\bullet E : A$;

$[\![\tilde{\mu}x.C]\!] = \lambda x{:}[\![A]\!]^\theta.[\![C]\!]$ où $\tilde{\mu}x.C : A$.

On vérifie facilement que cette traduction est bien définie et produit bien des termes typables comme stipulé *supra*. En effet, on peut vérifier pour chaque construction que si les capsules, appelants et appelés sous-termes de cette construction ont des traductions de type indiqué, alors la traduction de la construction a bien le type indiqué. Pour les appelants, ceci est simple. Pour les appelés, il faut s'assurer que la traduction d'un appelant de type A est une fonction prenant comme argument un terme de type $[\![A]\!]^\theta$, ce qui est bien le cas. Pour une capsule $\langle R \parallel E \rangle$, on a donc, comme $R : A$ et $E : A$ – et donc $[\![R]\!]^\eta : [\![A]\!]^\eta$ et $[\![E]\!] : \Pi r{:}[\![A]\!]^\eta.U_E$ –, que $[\![E]\!]\ [\![R]\!]^\eta : U_E[[\![R]\!]^\eta/r]$, et donc $\lambda\eta{:}T_E.([\![E]\!]\ [\![R]\!]^\eta)\ [\![E]\!] : U_E[[\![R]\!]^\eta/r][[\![E]\!]/\eta]$, si bien que, en choisissant la variable p du type $\Pi c{:}U_E[[\![R]\!]^\eta/r][[\![E]\!]/\eta].[\![A]\!]^\eta$, où c est une variable inédite, on obtient bien que la capsule est du type souhaité.
Par ailleurs, on peut préciser ici les types des traductions d'appelant :

- si $R\bullet E : \Pi x{:}A.B$ (*i.e.* $R{:}A$ et $E{:}B[R/x]$), alors $T_{R\bullet E} = \Pi r{:}[\![\Pi x{:}A.B]\!]^\eta.B[r/x]$ avec la variable d'appelant libre de $[\![R\bullet E]\!]$ du type adéquat comme indiqué *supra* pour les capsules ;
- si $\tilde{\mu}x.C : A$ et $[\![C]\!] : Y$, alors $T_{\tilde{\mu}x.C} = \Pi x{:}[\![A]\!]^\eta.Y$.

En outre, comme les variables sont traduites, tant pour les termes que pour les types, par des variables correspondantes, elle est compositionnelle vis-à-vis de la substitution.
Intuitivement, la variable η introduite dans la traduction des appelants permet d'indiquer l'endroit où celle-ci pourrait accueillir un appelé, et est destinée à être liée par le lieur présent dans la traduction de la cellule.

Lemme 3.17 : *Simulation*
Soient C une capsule typable et C′ telle que $C \xrightarrow[H]{} C'$ (C′ est typable par réduction du sujet). Alors il existe un contexte $\mathcal{K}[\ ]$ tel que $[\![C]\!] \xrightarrow[\beta]{+} \mathcal{K}[[\![C']\!]]$.

Démonstration.
Il y a trois cas possibles :

a. $C = \langle \lambda x{:}A.M \parallel N \bullet E \rangle \longrightarrow \langle M[N/x] \parallel E \rangle = C'$. *On a alors :*
$$\begin{aligned}
[\![C]\!] &= \mathcal{K}[[\![N \bullet E]\!]\ [\![\lambda x{:}A.M]\!]^{\eta}] \\
&= \mathcal{K}[\lambda r{:}[\![Y]\!]^{\theta}.(p(\lambda \eta{:}T.([\![E]\!]\ (r[\![N]\!]^{\eta}))\ [\![E]\!]))(\lambda x{:}[\![A]\!]^{\eta}.[\![M]\!]^{\eta})] \\
&\xrightarrow[\beta]{} \mathcal{K}[p(\lambda \eta{:}T.([\![E]\!]\ (\lambda x{:}[\![A]\!]^{\eta}.[\![M]\!]^{\eta}[\![N]\!]^{\eta}))\ [\![E]\!])] \\
&\xrightarrow[\beta]{} \mathcal{K}[p(\lambda \eta{:}T.([\![E]\!]\ ([\![M]\!]^{\eta}[[\![N]\!]^{\eta}/x]))\ [\![E]\!])] \\
&= \mathcal{K}[[\![C']\!]].
\end{aligned}$$

b. $C = \langle \mu\alpha.Z \parallel E \rangle \longrightarrow Z[E/\alpha] = C'$. *On a alors :*
$$\begin{aligned}
[\![C]\!] &= p(\lambda \eta{:}T.([\![E]\!]\ [\![\mu\alpha.Z]\!]^{\eta})\ [\![E]\!]) \\
&\xrightarrow[\beta]{} \mathcal{K}[[\![\mu\alpha.Z]\!]^{\eta}[[\![E]\!]/\eta]] \\
&= \mathcal{K}[[\![Z]\!][\eta/\alpha][[\![E]\!]/\eta]] \\
&= \mathcal{K}[[\![Z]\!][[\![E]\!]/\alpha]].
\end{aligned}$$

c. $C = \langle R \parallel \tilde{\mu}x.Z \rangle \longrightarrow Z[R/x] = C'$. *On a alors :*
$$\begin{aligned}
[\![C]\!] &= \mathcal{K}[[\![\tilde{\mu}x.E]\!]\ [\![R]\!]^{\eta}] \\
&= \mathcal{K}[(\lambda x{:}[\![A]\!]^{\theta}.[\![Z]\!])[\![R]\!]^{\eta}] \\
&\xrightarrow[\beta]{} \mathcal{K}[[\![Z]\!][[\![R]\!]^{\eta}/x]] \\
&= \mathcal{K}[[\![C']\!]].
\end{aligned}$$

Dans tous les cas, on a bien au moins un pas de β-réduction pour parvenir à un contexte contenant la traduction de C'. Q.E.D.

Corollaire 3.18 : *Normalisation forte dans le λ-cube pour* $\bar{\lambda}\mu\tilde{\mu}$
Tous les systèmes de types purs pour le $\bar{\lambda}\mu\tilde{\mu}$-calcul du λ-cube sont fortement normalisants.
I.e. : Soit C une $\mathfrak{T}$-capsule $\mathfrak{T}$-typable. Alors C n'admet pas de réduction infinie.

Démonstration.
Soit une suite infinie de capsules $(Z_i)_{i\in\mathbb{N}}$ *telle que pour tout i,* $Z_i \longrightarrow Z_{i+1}$ *et* Z_0 *soit typable. En appliquant à la chaîne de réduction infinie* $(C_i)_{i\in\mathbb{N}}$ *la transformation indiquée dans le corollaire 2.8, l'on obtient une réduction infinie imbriquée* $C_0 \xrightarrow[H]{} \mathcal{G}_0[C_1 \longrightarrow \mathcal{G}_0[\mathcal{G}_1[C_2]] \longrightarrow \cdots$ *avec* $C_i \xrightarrow[H]{} \mathcal{G}_i[C_{i+1}]$.
On a alors, par le lemme de simulation précédent, $[\![C_i]\!] \xrightarrow[\beta]{+} \mathcal{K}_i[[\![C_{i+1}]\!]]$.
On pose alors $T_i = \mathcal{K}_0[\mathcal{K}_1[\cdots\mathcal{K}_{i-1}[C_i]\cdots]]$, *et l'on vérifie que* $(T_i)_{i\in\mathbb{N}}$ *est une chaîne infinie de réduction de termes typables dans le calcul des constructions, ce qui est absurde.* Q.E.D.

Conclusion

L'on a étudié ici une extension des systèmes de types purs à un calcul dont le typage repose sur l'utilisation d'un calcul de séquents à formule active classiques. Les calculs basés sur des types de logique classique sont de plus en plus étudiés, et l'enrichissement des théories de typage sous-jacentes à ces calculs semble être une approche intéressante.

Une piste intéressant à étudier pourrait consister à combiner cette approche avec une extension à des substitutions explicites (cf. [10, 9]) : en effet, il existe des versions de $\bar{\lambda}\mu\tilde{\mu}$ à substitutions explicites (cf. [13]), et il pourrait être intéressant de leur étendre les systèmes de types purs en mettant en œuvre les techniques employées ici.

Bibliographie

[1] Barbanera (Franco) & Berardi (Stefano), « A symmetric lambda calculus for calssical program extraction ». *Information and Computation*, 125(2) : p. 103–117, 1996.

[2] Barendregt (Hendrik Pieter), « Lambda calculi with types ». *In* S. Abramsky, D. Gabbay, & T. Maibaum, éd., *Handbook of Logic in Computer Science*, volume 2 (*Background: Computational Structures*), p. 117–309. Oxford University Press, Oxford, U.K., 1992.

[3] Curien (Pierre-Louis) & Herbelin (Hugo), « The duality of computation ». *In Proc. of International Conference on Functional Programming*, Montréal, Canada, 2000. Association for Computing Machinery Press.

[4] Dougherty (Daniel J.), Ghilezan (Silvia) & Lescanne (Pierre), « Characterizing strong normalization in a language with control operators ». *In* E. Moggi & D. S. Warren, éd., *Proc. of 6th International Conference on Principles and Practice of Declarative Programming*, Verona, Italia, 2004. Association for Computing Machinery Press.

[5] Griffin (Timothy), « A formulae-as-types notion of control ». *In Proc. of 17th ACM Symposium on Principles of Programming Languages*, San Francisco, California, U.S.A., 1990, p. 47–58. Association for Computing Machinery Press.

[6] Herbelin (Hugo), *Séquents qu'on calcule*. Thèse de doctorat, Université Paris VII Denis Diderot, janvier 1995.

[7] Herbelin (Hugo), « Explicit substitutions and reducibility ». *Journal of Logic and Computation*, 11(3) : p. 29–449, 2001.

[8] Herbelin (Hugo), *C'est maintenant qu'on calcule*. Mémoire d'habilitation à diriger les recherches, Université Paris XI Sud (Orsay), décembre 2005.

[9] Kervarc (Romain), « Explicit substitution for the λ-cube ». Communication at the 2nd workshop on Coq and Rewriting, Palaiseau, France, 2004.

[10] Kervarc (Romain) & Lescanne (Pierre), « Pure type systems, cut, and explicit substitution ». *In* D. Kesner, F. Van Raamsdonk, & J. B. Wells, éd., *Proc. of the 2nd International Workshop on Higher Order Rewriting (part of the 15th International Conference on Rewriting Techniques and Applications)*, Aachen, Nordrhein-Westfallen, Deutschland, 2004, p. 72–77.

[11] Miquel (Alexandre), *Le calcul des constructions implicite : syntaxe et sémantique*. Thèse de doctorat, Université Paris VII Denis Diderot, décembre 2001.

[12] Parigot (Michel), « $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction ». *In* A. Voronkov, éd., *Proc. of International Conference on Logic Programming and Automated Reasoning*, Санкт Петербург, Россия, 1992. *Lecture Notes in Computer Science*, volume 624, p. 190–201. Springer Verlag.

[13] Polonovski (Emmanuel), « Strong normalization of $\lambda\mu\tilde{\mu}$-calculus with explicit substitution ». *In Proc. of Foundations of Software Science and Computation Structures*, Barcelona, España, 2004. *Lecture Notes in Computer Science*, volume 2987, p. 423–437. Springer Verlag.

[14] Urban (Christian), *Classical Logic and Computation*. Ph.D. Thesis, University of Cambridge, octobre 2000.