DEA SEMANTIQUE, PREUVES et LANGAGES

Cours de Lambda-calcul

Année 2002-2003

Thérèse HARDIN-ACCART

Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie
8, Rue du Capitaine Scott, 75015 Paris
`Therese.Hardin@lip6.fr`
`http:\\www-spi.lip6.fr/~hardin`

Les notes qui suivent servent de base au cours du module de tronc commun "Lambda-calcul". En les rédigeant, j'ai d'abord cherché à donner une présentation intuitive des notions abordées, quitte à sacrfier parfois à une exposition complètement rogoureuse, qui pourra être trouvée dans les ouvrages donnés en référence. Ces notes ne constituent qu'une version provisoire. Toute remarque, suggestion est la bienvenue.

1 Un tour guidé des λ-calculs

La notion de fonction peut être abordée de plusieurs points de vue. On peut, dans un premier point de vue dit *extensionnel*, définir une fonction comme un graphe (ensemble de paires (argument, valeur)). Cette présentation, généralement attribuée à Dirichlet, est intéressante pour résoudre certains problèmes (résolution d'équations différentielles, recherche d'une meilleure approximation) mais elle perd son caractère intuitif dès lors que les valeurs rendues sont elles-mêmes des fonctions. Par exemple, cette difficulté se révèle lorsque l'on veut définir la dérivée partielle d'une fonction. L'interprétation des fonctions en tant que relations procède de ce point de vue extensionnel et conduit aux notions de domaine et codomaine de la théorie des ensembles, par exemple.

Le point de vue *intentionnel* considère qu'une fonction définit une règle de correspondance. Lorsqu'on applique cette règle de correspondance à un argument, on obtient (si possible) une valeur. La notion primitive ici est celle de l'*application* (notée souvent par simple juxtaposition). Les règles de correspondance peuvent être exprimées de plusieurs manières. On peut par exemple introduire un symbole I et la règle $Ix \rightarrow x$ (i.e. I appliqué à x s'évalue en x) pour exprimer la fonction Identité. Dès lors, on peut évaluer II par exemple. Notez que cette définition de l'identité n'a pas utilisé la notion de variable : le x qui sert à exprimer la règle désigne un argument quelconque et non la variable de la fonction. La Logique Combinatoire, $\mathcal{LC}$, est une théorie de la fonctionnalité qui permet d'exprimer intentionnellement des fonctions sans utiliser de noms de variable, dans l'esprit de l'exemple ci-dessus. Les théories des substitutions explicites rentrent aussi dans ce cadre.

Le point de vue intentionnel semble donc plus général que le point de vue extensionnel : un graphe peut toujours être interprété comme une correspondance. La réciproque est fausse : aussi bien en analyse (théorème des fonctions implicites par exemple) qu'en $\mathcal{LC}$, on peut établir des propriétés de fonctions sans connaître leur graphe. Cette discussion sur les liens entre les deux points de vue se complique lorqu'on la prolonge à l'égalité entre fonctions. Comment formuler que deux fonctions sont intentionnellement (extensionnellement) égales? L'égalité intentionnelle induit-elle l'égalité extensionnelle et réciproquement? Des réponses seront apportées dans la suite du cours.

Le Lambda-calcul est une théorie de la fonctionnalité, introduite à la fin des années 30 par le logicien Alonzo Church, qui procède d'un point de vue *intentionnel*. Une fonction y est définie comme l'abstraction d'un nom dans une expression, abstraction qui définit la correspondance. Par exemple, l'abstraction du nom x dans l'expression $x+1$ construit la fonction $x \longrightarrow x+1$, appartenant à l'ensemble $N \longrightarrow N$, qui peut être comprise comme un processus de calcul du successeur. Cette fonction n'a, a priori, nullement besoin d'être nommée. Pour l'utiliser, il suffit de pouvoir l'appliquer à un nombre donné : $(x \longrightarrow x+1)3 = 4$. Les fonctions du λ-calcul sont donc des fonctions anonymes, construites par abstraction (fonctionnelle) d'une variable ($\lambda x.$) dans un terme. Ce sont donc des fonctions d'une (seule) variable. Elles peuvent être appliquées à n'importe quel terme du λ-calcul, y compris à elles-mêmes (la fonction identité $\lambda x.x$ peut être appliquée à elle-même). Ce λ-calcul, dit pur ou non-typé, propose donc une notion de fonction plus générale que celle de la théorie classique des ensembles. En effet, celle-ci interdit d'appliquer une fonction à elle-même (l'identité, fonction de A dans A, ne peut pas être appliquée à elle-même, puisqu'elle appartient à l'ensemble A^A, disjoint de A). La notion fondamentale qui sous-tend le λ-calcul pur n'est pas celle d'ensemble — donc de domaine et de codomaine — mais celle de correspondance.

Pour retrouver une notion de fonction plus proche de celle de la théorie classique des ensembles, il faut restreindre les règles de formation des termes, de façon à interdire l'application d'une fonction à elle-même. Cela se fait à l'aide de la notion de typage. Il existe plusieurs systèmes de typage, dont la complexité augmente avec le nombre de termes typables.

Le langage des types peut être défini par un ensemble de types de base ($\{Nat\}$ par exemple) et le constructeur $\longrightarrow$. Le lambda-calcul ainsi obtenu est dit *simplement typé*. Une variable a un type unique (La fonction identité ($\lambda x : nat.x$) y est de type $Nat \longrightarrow Nat$). Elle ne peut être appliquée qu'à une expression de type Nat. Les fonctions retrouvent ainsi un domaine et un codomaine et ne peuvent être appliquées qu'à des valeurs de leur domaine. Les restrictions dûes à ce typage peuvent cependant sembler trop contraignantes. On ne peut pas, avec ce système de types, faire apparaître le fait que la définition de l'identité est indépendante du type de la variable abstraite — la fonction identité se contente de rendre son argument — et qu'il suffit de s'assurer que l'application de l'identité est faite de façon cohérente. Pour rendre compte de ce point, on enrichit le langage des types par adjonction de variables de type (souvent représentées par des lettres grecques) et de quantificateurs sur ces variables. On obtient ainsi le *λ-calcul typé polymorphe*. l'identité y est définie comme la fonction $\lambda x : \sigma.x$, qui a pour type $\forall\sigma.(\sigma \longrightarrow \sigma)$, où σ est une variable de type. Elle peut être appliquée à une valeur de n'importe quel type, disons t, et elle rend un résultat de type t. Le noyau fonctionnel des langages ML est un λ-calcul typé avec variables de type (les `'a`). Les quantificateurs universels sur ces variables de type sont autorisés, mais seulement en position prénexe (au début de l'écriture du type). Cette restriction permet de décider si un terme est typable et donc d'offrir un algorithme de synthèse de type au programmeur ML.

Les variables de type étant introduites, il devient possible d'enrichir le langage des types comme on l'a fait pour le langage des termes. On pourra par exemple parler du type $\exists\sigma.\sigma \longrightarrow \sigma$. Plus généralement, le langage des types peut être enrichi avec des produits, des sommes, etc. de manière à représenter des types de données vérifiant certaines propriétés. Les λ-calculs ainsi obtenus sont dits *avec typage d'ordre supérieur*. Ils sont utilisés en informatique en particulier comme langages de description de vérificateurs de théorèmes. Par exemple, le langage COQ, développé à l'INRIA, repose sur un λ-calcul typé d'ordre supérieur appelé *langage des Constructions* dont le système de types est très riche. On peut y définir le type des listes de longueur n, le système synthétise alors le principe de récurrence sur ces listes. On peut y spécifiier un algorithme de tri d'une liste, prouver que cette spécification construit une liste triée ayant les mêmes eléments que la liste argument puis extraire automatiquement de la preuve le programme implémentant le tri examiné.

Le λ-calcul pur, malgré (ou peut-être à cause) de son laxisme au niveau de l'application, est une théorie intéressante de plusieurs points de vue. Tout d'abord, du point de vue de la calculabilité, les termes du λ-calcul représentent exactement toutes les fonctions récursives partielles (résultat de Kleene). Du point de vue de la programmation, le λ-calcul est un outil essentiel puisqu'il s'agit d'un système formel qui décrit la construction et l'évaluation des fonctions. Il apparaît ainsi comme le prototype des langages de programmation dits fonctionnels (ou de la partie fonctionnelle des autres langages). Dans ces langages, un programme est d'abord une fonction. Il est donc très naturellement modélisé comme une expression du λ-calcul. L'expression $(\lambda x.a)$ représente la fonction de paramètre formel x, de corps a, le terme $(\lambda x.a)b$ correspond à l'appel de cette fonction sur le paramètre effectif b. Le résultat de cet appel est une valeur, celle du corps de la fonction a, dans lequel on a remplacé toutes les occurrences du paramètre formel x par la valeur effectivement fournie b. Cette évaluation est exprimée dans le λ-calcul par la règle de β-réduction :

$$(\lambda x.a)b \to_\beta a\{b/x\}$$

$a\{b/x\}$ est le résultat de la substitution de x par b dans a.

L'étude du λ-calcul renseigne donc les informaticiens sur leurs programmes. Par exemple, plusieurs stratégies de calcul sont possibles pour obtenir le résultat d'un programme. On peut utiliser une règle simple —évaluer d'abord l'argument b — avec le risque que ce calcul s'avère inutile si le paramètre formel x ne figure pas dans a. On peut aussi ne calculer la valeur des paramètres effectifs que si cela devient nécessaire. Cette possibilité est intéressante car elle permet de manipuler des structures de données potentiellement infinies comme des flots. Pour comparer ces différents choix, prouver leur correction, il suffit d'étudier les dérivations (suite de réductions) construites avec la règle de simplification β. Le résultat fondamental dans ce domaine est le théorème de confluence, dit encore propriété de Church-Rosser, pour la β-réduction. Il signifie que l'ordre dans lequel les différentes étapes de β-réduction sont menées, donc l'ordre dans lequel les évaluations sont effectuées, n'a pas d'importance : on parvient toujours à la même valeur. D'autres théorèmes affinent ce résultat : comparaison des différentes dérivations d'un terme, définition des approximations syntaxiques d'un λ-terme, etc.

Revenons un instant sur les λ-calculs typés. La majorité de ces langages possèdent la propriété de *terminaison forte* : toutes les suites de β-réductions sont finies et terminent toutes sur le même terme, appelé *forme normale* du λ-terme. Cette propriété, très intéressante par ailleurs, fait que les termes des λ-calculs typés ne peuvent pas représenter tous les programmes, tous ceux en particulier qui ne s'arrêtent pas et que l'on souhaite pourtant étudier, ne serait-ce que pour les rejeter. En revanche, le λ-calcul pur permet, au moins en théorie, de représenter tous les programmes et donc de les étudier.

On a abordé le λ-calcul par quelques aspects syntaxiques et montré son intérêt. Encore faut-il s'assurer que tout cette construction n'est pas que syntaxe et qu'il est possible de définir des modèles (des réalisations effectives) du λ-calcul. La question de la consistance — existence d'un modèle dans lequel il existe au moins deux termes différents n'ayant pas la même image — est résolue ainsi. Il n'existe pas d'ensemble X, contenant son propre ensemble de fonctions $X \longrightarrow X$. Mais, si l'on ajoute de la structure à X, par exemple une topologie, et si on restreint $X \longrightarrow X$ à ne contenir que les fonctions continues pour cette topologie, alors on peut obtenir l'inclusion de $X \longrightarrow X$ dans X. Cette théorie des modèles, appelée sémantique dénotationnelle, est dûe à Scott (1969). Elle est importante car elle permet, non seulement de donner une signification à tout terme du λ-calcul, mais aussi de modéliser la construction d'une fonction récursive par celle d'un point-fixe et de prendre en compte des types de données.

Pour terminer ce tour guidé des λ-calculs (qui est aussi un tour guidé du tronc commun du DEA), remarquons qu'il peut sembler surprenant qu'un langage de la logique, créé avant que l'informatique ne voie le jour, puisse se révéler si utile à la programmation. Mais programmer un tri par exemple n'est jamais que construire une preuve, de manière effective, que la liste résultat est triée. Les questions soulevées par ces méthodes effectives de construction rejoignent donc celles des logiciens des années 30, qui essayaient de bâtir une théorie constructive des mathématiques. Cela explique cette convergence.

2 La syntaxe du λ-calcul

Définition 2.1 Soit V un ensemble dénombrable de variables. L'ensemble des termes du λ-calcul, noté $\mathbf{\Lambda_V}$, est le plus petit ensemble vérifiant :

1. $V \subseteq \mathbf{\Lambda_V}$
2. Si $a \in \mathbf{\Lambda_V}$, alors $\lambda\, x\, .a \ \in \mathbf{\Lambda_V}$
   $\lambda\, x$ est dit lieur de la variable x dans le terme a. $\lambda\, x\, .a$ représente la fonction de la variable x (i.e. de paramètre formel x), ayant pour corps l'expression a.
3. Si a et $b \in \mathbf{\Lambda_V}$, alors $a\, b \ \in \mathbf{\Lambda_V}$
   $(a\, b)$ est l'application (notée par juxtaposition) du terme a au terme b. Le terme $(a\, b)$ peut être vu comme la valeur rendue par la fonction a, appliquée à l'argument (i.e. au paramètre effectif) b.

Exercice 2.1 *Pourquoi ce "plus petit ensemble" existe-t-il? Quelle méthode de raisonnement déduit-on de son existence?*

<u>Notation</u> L'opérateur d'application associe à gauche: $a\, b\, c\, d \equiv (((a\, b)\, c)\, d)$. D'autre part, on regroupe des abstractions successives sous un seul λ : $\lambda x.\lambda y.a \equiv \lambda x\, y.a$.

Appliquer la fonction $\lambda x.a$ au paramètre effectif b consiste à remplacer dans le corps a les occurrences du paramètre formel x par b. Ce calcul remplace donc le terme $(\lambda x.a)b$ par le terme $a\{b/x\}$ (a dans lequel b remplace x). Il peut être compris comme une étape de réécriture par une règle qui est notée $\xrightarrow{\beta}$.

Avant de définir précisément cette règle $\xrightarrow{\beta}$, il faut expliquer ce qu'est ce remplacement. Tout d'abord, dans l'expression $\lambda x.yx$, les variables x et y ont des rôles très différents. x est une variable muette, qui pourrait être changée en z par exemple à condition de changer aussi λx en λz. Quand on appliquera cette fonction $\lambda x.yx$ à une valeur v, on remplacera x par v. y est réellement une variable, c'est-à-dire un symbole dont la valeur est inconnue/quelconque/ indifférente/elle-même ou, version modèle, un symbole qui recevra une valeur par une interprétation dans un domaine donné. Il faut donc bien faire la distinction entre variables libres (comme y dans $\lambda x.yx$) et variables liées (comme x dans $\lambda x.yx$). Bien sûr, une variable libre dans un sous-terme peut devenir liée dans le terme : par exemple, dans le terme $\lambda y.(\lambda x.yx)$, y est devenue liée. Toute valeur de y ne pourra dès lors être obtenue que par application de $\lambda y.(\lambda x.yx)$ à un argument.

Le remplacement ne concerne donc que les occurrences de la variable liée par le lieur de la fonction. L'argument, qui peut être, rappelons-le, un terme quelconque, peut lui-aussi contenir des variables libres. Examinons par exemple le terme $(\lambda\, y.(\lambda\, t.\, t\, y)\, y)(t\, x)$. L'argument $(t\, x)$ contient deux variables libres t et x. Si on se contente d'effectuer le remplacement textuel de la variable y par le terme $(t\, x)$ dans $((\lambda\, t.\, t\, y)\, y)$, on obtient le terme $(\lambda\, t\, .t(t\, x))(t\, x)$. Or, dans le sous-terme $(\lambda\, t.\, t\, y)$ la variable t est une variable muette, $(\lambda\, t.\, t\, y)$ peut être remplacé par $(\lambda\, f.\, f\, y)$: il s'agit simplement d'un changement du nom du paramètre formel. Le résultat correct de la β-réduction est donc :

$$((\lambda\, f.\, f\, y)y)\{t\, x/y\} = (\lambda\, f.\, f\, (t\, x))(t\, x)$$

Le phénomène ainsi mis en évidence s'appelle la *capture de variables* (ici, de la variable t par le lieur λt) et il faut bien sûr l'éviter. Cela nous oblige à définir précisèment les notions de variable libre, variable liée et celle de remplacement, appelé en général *substitution*.

Définition 2.2 Soit $a \in \mathbf{\Lambda_V}$. $V(a)$ désigne l'ensemble des variables de a. $FV(a)$, ensemble des variables libres de a et $BV(a)$, ensemble des variables liées de a, sont définis par récurrence sur la structure de a ainsi :

1. $FV(x) = \{x\}$ et $BV(x) = \emptyset$.

2. $FV(a\,b) = FV(a) \cup FV(b)$ et $BV(a\,b) = BV(a) \cup BV(b)$
3. $FV(\lambda\, x.a) = FV(a) \setminus \{x\}$ et $BV(\lambda\, x.a) = BV(a) \cup \{x\}$

Exercice 2.2 *Les ensembles $FV(a)$ et $BV(a)$ sont-ils nécessairement disjoints? Donner des exemples pour étayer votre réponse.*

Définition 2.3 Soient a et $b \in \mathbf{\Lambda_V}$. La substitution du terme b à toutes les occurrences libres de la variable x dans le terme a est notée $a\{b/x\}$ et est définie ainsi :

1. $x\{b/x\} = b$ et $y\{b/x\} = y$ si $y \not\equiv x$
2. $(a_1\, a_2)\{b/x\} = (a_1\{b/x\})\ (a_2\{b/x\})$
3. $(\lambda\, x.a)\{b/x\} = \lambda\, x.a$
4. $(\lambda\, y.a)\{b/x\} = \lambda\, y.\, (a\ \{b/x\})$
   si $y \not\equiv x$ et ($y \notin FV(b)$ ou $x \notin FV(a)$)
5. $(\lambda\, y.a)\{b/x\} = \lambda\, z.\, (a\ \{z/y\})\ \{b/x\}$
   si $y \not\equiv x$ et ($y \in FV(b)$ et $x \in FV(a)$) ; $z \notin V(ab)$.

Un λ-terme a est dit *fermé* si $FV(a) = \emptyset$.

La définition de la substitution donnée ici permet d'éviter la capture de variables. La variable z introduite par les clauses 4. et 5. est parfois dite *variable fraîche*. Ce changement du nom d'un paramètre formel est axiomatisé par la règle de α-conversion.

Définition 2.4 La règle de α-conversion est définie par:

$$\lambda\, t.a \xrightarrow{\alpha} \lambda\, f.a\{f/t\}$$

A condition de travailler modulo α-conversion, on peut remplacer les clauses 3–4–5 par la clause suivante (cf [**?**, p. 27]) :

$$(\lambda\, y.a)\{b/x\} = \lambda\, y.\, (a\ \{b/x\})$$

que nous adopterons. Ce choix simplifie notablement les démonstrations : on fait l'hypothèse qu'à chaque étape de la preuve, on peut choisir le "bon" représentant canonique et on ne mentionne pas ce choix. Notons toutefois que cette solution n'est pas directement utilisable dans une implémentation et qu'il faut donc gérer la α-conversion dès que l'on souhaite construire un interpréteur de λ-termes. Nous reviendrons sur cette α-conversion au cours de l'étude des $\lambda\sigma$-calculs.

Pour visualiser l'opération de substitution, il est commode de représenter les λ-termes sous forme d'arbres. Les feuilles d'un tel arbre sont des variables et les noeuds sont soit des noeuds application soit des noeuds abstraction. Un sous-terme — donc un sous-arbre — est défini par le chemin qui va de la racine de l'arbre à son propre sommet. La notion de chemin est formalisée par celle d'occurrence.

Définition 2.5 *L'ensemble des occurrences d'un terme a est un ensemble de mots sur l'alphabet $\{0,\ 1,\ 2\}$. Le sous-terme à l'occurrence u de a, noté $a_{|u}$, et l'ensemble des occurrences de a, noté $O(a)$, sont définis comme suit, par récurrence sur la structure de a :*

1. *$O(x) = \{\epsilon\}$ (ϵ le mot vide) et $a_{|\epsilon} = a$.*

2. $O(a\,b) = \{\epsilon\} \cup 1O(a) \cup 2O(b)$. $(ab)_{|1u} = a_{|u}$ *et* $(ab)_{|2u} = b_{|u}$.
3. $O(\lambda x\, a) = \{\epsilon\} \cup 0O(a)$. $(\lambda x.a)_{|0u} = a_{|u}$

On note $a[u \longleftarrow b]$ *le terme* a *dans lequel* $a_{|u}$ *est remplacé textuellement par* b. *Deux sous-termes sont dits disjoints si leurs occurrences sont disjointes.*

L'ensemble $O(a)$ *est muni d'un ordre partiel, défini par l'ordre préfixe.*

On peut maintenant localiser la notion de liaison.

Proposition 2.6 *L'occurrence* u *d'une variable* x *dans un terme* a *est liée si et seulement si* $u = vw$ *et* $a_{|v} = \lambda x.b$. *Le lieur de cette occurrence* u *est l'occurrence* v *maximale réalisant cette condition.*

La démonstration se fait par récurrence sur la structure du terme. Le lecteur est vivement encouragé à la faire. •

Définition 2.7 *Un contexte* $C[\,]$ *est un terme contenant un "trou". Plus précisément, un contexte est un terme défini de manière analogue à un* λ*-terme à partir de* V *et de la constante* Ω, *encore notée* $[\,]$. *Le contexte* $C[M]$ *est construit à partir des contextes* C *et* M *en remplaçant textuellement l'occurrence de* Ω *par* M. *Le contexte peut contenir plusieurs occurrences de* Ω. *Ce remplacement textuel peut donc conduire à des captures de variable.*

Lemme 2.8 (Lemme de substitution) *Si* a, b, $c \in \mathbf{\Lambda_V}$, *si* $x \neq y$, *si* $x \notin FV(c)$, *alors:*

$$a\{b/x\}\{c/y\} \equiv a\{c/y\}\{b\{c/y\}/x\}$$

Si, de plus, $y \notin FV(b)$, *alors:*

$$a\{b/x\}\{c/y\} \equiv a\{c/y\}\{b/x\}$$

<u>Preuve</u>

La preuve se fait par récurrence sur la structure de a. •

Ayant défini la substitution, nous pouvons définir la β-réduction.

2.1 définition de la β-réduction

Définition 2.9 La notion de β-réduction est la relation :

$$(\lambda x.M)N \;\rightarrow\; M\{N/x\}$$

Un terme de la forme $(\lambda x.M)N$ est appelé un β-*radical* (redex en anglais). Le terme $M\{N/x\}$ est appelé son *réduit* ou *contracté*.

La β-réduction est la relation définie par : $a \xrightarrow{\beta} b$ si

$$a_{|u} = (\lambda x.M)N \;\; b = a[u \longleftarrow M\{N/x\}]$$

Une suite de β-réductions est appelée une β-dérivation.

Un terme a se β-réduit donc sur un terme b s'il possède un β-radical à une occurrence u et si b est obtenu en remplaçant dans a ce radical par son réduit. On va donner une définition plus constructive — ou moins opérationnelle — de la β-réduction. La première définition est la plus intuitive mais n'est pas très facile à manipuler dans les preuves.

Définition 2.10 Une relation R sur $\mathbf{\Lambda_V}$ est *compatible* si :

$$(a\ R\ b) \Longrightarrow (ac\ R\ bc),\ (ca\ R\ cb),\ (\lambda x.a\ R\ \lambda x.b)$$

Une *congruence* est une relation d'équivalence compatible.
Une *théorie* est définie par la donnée d'une famille de termes et d'une congruence sur ces termes.

Définition 2.11 Une *notion de réduction* est une relation binaire sur $\mathbf{\Lambda_V}$.
Une relation de réduction — ou *réduction* — sur $\mathbf{\Lambda_V}$ est la fermeture compatible d'une notion de réduction.
Si R est une réduction, alors R^* désigne la fermeture réfléxive et transitive de cette relation.
$=_R$ désigne la congruence engendrée par R. On la représente souvent par un zig-zag. Pourquoi?

Lemme 2.12 *Une relation R est compatible si et seulement si elle vérifie la propriété suivante:*

$$(a\ R\ b) \Longrightarrow \forall c \in \Lambda, \forall u \in O(c), (c[u \longleftarrow a]\ R\ c[u \longleftarrow b])$$

<u>Preuve</u>
Si la propriété est vérifiée, en prenant $c = \lambda x.x$ et $u = 0$ puis $c = xc_2$ et $u = 1$ puis $c = c_1 x$ et $u = 2$, on montre que R est compatible. La réciproque se montre par récurrence sur la longueur de u. Si $u = 0v$, alors c est de la forme $\lambda x.c_0$. Par hypothèse de récurrence, $c_0[v \longleftarrow a]\ R\ c_0[v \longleftarrow b]$. R étant compatible, $\lambda x.c_0[v \longleftarrow a]\ R\ \lambda x.c_0[v \longleftarrow b]$. Les deux autres cas se traitent de la même façon.
•

Définition 2.13 La β-réduction est la relation de réduction obtenue par fermeture de la notion de réduction définie par:

$$(a, b) \in \beta \text{ ssi } a = (\lambda x.M)N \text{ et } b = M\{N/x\}$$

Lemme 2.14 *Les deux définitions de la β-réduction sont équivalentes.*

<u>Preuve</u>
Simple manipulation des définitions et lemmes précédents. Notez que la substitution fait partie des deux définitions alors que le remplacement textuel devient implicite dans la seconde version. Il y est internalisé par la propriété de compatibilité. •

Exercice 2.3 *Donner les dérivations des termes suivants.*
II où $I = (\lambda x.x)$
$\Delta\Delta$ où $\Delta = (\lambda x.x\,x)$
WW où $W = (\lambda x.x\,x\,x)$.

Exercice 2.4 *Soit R une relation compatible. Montrer que, si $a\,R\,b$, alors pour tout contexte C, on a $C[a]\,R\,C[b]$.*

Définition 2.15 *Un terme $a \in \mathbf{\Lambda_V}$ est en forme normale si aucun de ses sous-termes n'est un β-radical.*

Exercice 2.5 *Donner des exemples de λ-termes en forme normale.*
Donner des exemples de λ-termes qui ne sont pas en forme normale mais en possèdent une.
Donner des exemples de λ-termes qui ne possèdent pas de forme normale.
Est-ce que si, pour tout b tel que $a \xrightarrow{*} b$, on a $a \equiv b$, alors a est en forme normale?

2.2 Définition de la β-interconvertibilité

Nous allons présenter le λ-calcul comme une théorie équationnelle.

Définition 2.16 *La relation de β-interconvertibilité, notée $=_\beta$, est la fermeture réfléxive, symétrique et transitive de la β-réduction. On la note aussi $\mathbf{\Lambda} \vdash a = b$.*

Donner la définition de la relation d'interconvertibilité par une famille de règles d'inférence.

Définition 2.17 *Une théorie $\mathcal{T}$ équationnelle est consistante s'il existe deux termes clos a et b tes que $(a = b)$ ne soit pas prouvable dans $\mathcal{T}$.*

Proposition 2.18 *Le λ-calcul, muni de la relation de β-interconvertibilité, est une théorie consistante.*

Preuve
Soient a et b deux termes clos en forme normale, différents, et tels que $a =_\beta b$. Le théorème de Church-Rosser, vu plus tard, affirme qu'il existe un terme c tel que $a \stackrel{\beta^*}{\to} c$ et $b \stackrel{\beta^*}{\to} c$. Comme a est en forme normale, on $a \equiv c$. De même, $b \equiv c$. Donc, $a \equiv b$. $\bullet$

Exercice 2.6 *Démontrer que la β-interconvertibilité est la fermeture par congruence de la notion de β-réduction. Donner une autre version de cette définition à l'aide de règles d'inférence.*

La définition axiomatique de $=_\beta$ est la suivante ·

$$\frac{}{(\lambda x.a)b =_\beta a\{b/x\}}\text{Ax} \qquad \frac{}{a =_\beta a}\text{Refl}$$

$$\frac{a =_\beta b}{b =_\beta a}\text{Sym} \qquad \frac{a =_\beta b \qquad b =_\beta c}{a =_\beta c}\text{Trans}$$

$$\frac{a =_\beta b \qquad c =_\beta d}{a\ c =_\beta b\ d}\text{App} \qquad \frac{a =_\beta b}{\lambda x.a =_\beta \lambda x.b}\text{Lam}$$

Proposition 2.19 *Si $a_1 \stackrel{\beta}{\to} b_1$, si $a_2 \stackrel{\beta}{\to} b_2$, peut-on affirmer que $a_1\{a_2/x\} \stackrel{\beta}{\to} b_1\{b_2/x\}$?*
Montrer que, si $a_1 =_\beta a_2$, si $b_1 =_\beta b_2$, alors $a_1\{b_1/x\} =_\beta a_2\{b_2/x\}$. On peut traiter d'abord les deux cas particuliers $a_1 \equiv a_2$ et $b_1 \equiv b_2$.

On démontre d'abord deux lemmes.

Lemme 2.20 *Si $a =_\beta c$, alors $a\{b/x\} =_\beta c\{b/x\}$*

Preuve
Soit $P(a)$ la propriété suivante :

$$P \equiv \forall c.(\forall b.(a =_\beta c) \implies a\{b/x\} =_\beta c\{b/x\})$$

On démontrer que $\forall a.P(a)$, par récurrence sur l'arbre de preuve de $a =_\beta c$

Si la preuve de $a =_\beta c$ se termine par $a \equiv (\lambda y.M)N =_\beta M\{N/y\} \equiv c$, il faut montrer que $((\lambda y.M)N)\{b/x\} =_\beta (M\{N/y\})\{b/x\}$. Or:

$$((\lambda y.M)N)\{b/x\} =_\beta ((\lambda y.M)\{b/x\})(N\{b/x\}) =_\beta (M\{b/x\})\{N\{b/x\}/y\}$$

On conclut avec le lemme de substitution. $\bullet$

Lemme 2.21 *Si $b =_\beta c$, alors $a\{b/x\} =_\beta a\{c/x\}$*

Preuve
Soit $P(a) \equiv \forall b, c.(b =_\beta c) \implies a\{b/x\} =_\beta a\{c/x\}$. On démontre $\forall a.P(a)$ par récurrence structurelle sur a. •

L'égalité cherchée s'obtient par application des deux lemmes:

$$a_1\{b_1/x\} =_\beta a_1\{b_2/x\} =_\beta a_2\{b_2/x\}$$

3 Confluence du λ-calcul

Proposition 3.1 *La β-réduction est faiblement confluente.*

Preuve
Soit un terme a contenant deux radicaux $R_1 = (\lambda x.M)N$ à l'occurrence u et $R_2 = (\lambda y.P)Q$ à l'occurrence v. Posons $a \overset{R_1}{\to} b$ et $a \overset{R_2}{\to} c$. Alors, $b = a[u \longleftarrow M\{N/x\}]$ et $c = a[v \longleftarrow P\{Q/y\}]$. Si les occurrences u et v sont disjointes, alors $b_{|v} = R_2$ et $c_{|u} = R_1$. Il suffit de réduire b en v et c en u pour obtenir le même terme. Si u (par exemple) est un préfixe de v, on a deux cas suivant que R_2 est dans M ou dans N. Si $v = u1w$, alors R_2 est dans M, il est donc transformé en $(\lambda y.P\{N/x\})Q\{N/x\}$. On réduit ce radical dans b, on réduit R_1 dans c et on constate que l'on aboutit au même terme. Si $v = u2w$, alors R_2 figure dans N, il est donc dupliqué en autant de copies qu'il y a d'ocurrences de x dans M. En réduisant chacune de ces copies d'une part dans b et en réduisant R_1 dans c, on obtient le même terme. •

Comme la β-réduction ne vérifie pas la propriété du diamant (cf. preuve de confluence locale), comme elle n'est pas noethérienne (cf. Ω), il faut une méthode originale pour montrer la confluence de la β-réduction. Il en existe plusieurs. Nous allons d'abord utiliser celle de Taït et Martin-Löf. Elle est fondée sur l'idée suivante: trouver une relation, notée B, qui soit contenue dans β^* et qui vérifie la propriété du diamant. Intuitivement, la relation B réduit un certain nombre de radicaux en parcourant le terme depuis les feuilles.

Définition 3.2 *la relation B est définie sur $\mathbf{\Lambda_V}$ par :*

1. $a \overset{B}{\to} a$

2. $\dfrac{a \overset{B}{\to} b}{\lambda x.a \overset{B}{\to} \lambda x.b}$

3. $\dfrac{a_1 \overset{B}{\to} a_2 \quad b_1 \overset{B}{\to} b_2}{a_1 b_1 \overset{B}{\to} a_2 b_2}$

4. $\dfrac{a_1 \overset{B}{\to} a_2 \quad b_1 \overset{B}{\to} b_2}{(\lambda x.a_1)b_1 \overset{B}{\to} a_2\{b_2/x\}}$

Proposition 3.3 *Si $a_1 \overset{B}{\to} a_2$, si $b_1 \overset{B}{\to} b_2$, alors $a_1\{b_1/x\} \overset{B}{\to} a_2\{b_2/x\}$.*

Preuve
La propriété $P(a_1, a_2)$ est définie par :

$$P(a_1, a_2) \equiv (\forall b_1, b_2 . b_1 \stackrel{B}{\to} b_2 \implies a_1\{b_1/x\} \stackrel{B}{\to} a_2\{b_2/x\})$$

Il faut donc démontrer: $\forall a_1, a_2.((a_1 \stackrel{B}{\to} a_2) \implies P(a_1, a_2))$

La preuve se fait par récurrence sur la structure de l'arbre de preuve de $a_1 \stackrel{B}{\to} a_2$, quel que soit l'arbre de preuve de $b_1 \stackrel{B}{\to} b_2$, comme en témoigne l'énoncé de P.

Le seul cas non immédiat est le suivant:

$$\frac{M_1 \stackrel{B}{\to} M_2 \quad N_1 \stackrel{B}{\to} N_2}{a_1 \equiv (\lambda y.M_1)N_1 \stackrel{B}{\to} M_2\{N_2/y\} \equiv a_2}$$

pour lequel il faut montrer:

$((\lambda y.M_1)N_1)\{b_1/x\} \stackrel{B}{\to} (M_2\{N_2/y\})\{b_2/x\}$

En utilisant deux fois l'hypothèse de récurrence, on peut construire l'arbre suivant:

$$\frac{M_1)\{b_1/x\} \stackrel{B}{\to} M_2\{b_2/x\} \quad N_1\{b_1/x\} \stackrel{B}{\to} N_2\{b_2/x\}}{(\lambda y.M_1\{b_1/x\})(N_1\{b_1/x\}) \stackrel{B}{\to} (M_2\{b_2/x\})\{N_2\{b_2/x\}/y\}}$$

et on conclut avec le lemme de substitution.

•

Proposition 3.4 *La relation $\stackrel{B}{\to}$ est fortement confluente.*

<u>Preuve</u>

Soit $a \stackrel{B}{\to} b$ et $a \stackrel{B}{\to} c$. Montrons qu'il existe d tel que $b \stackrel{B}{\to} d$ et $c \stackrel{B}{\to} d$. La preuve se fait par récurrence sur la construction de l'arbre de preuve de $a \stackrel{B}{\to} b$, quel que soit l'arbre de preuve de $a \stackrel{B}{\to} c$.

1. $a \equiv b$, alors prendre $d = c$.

2. $a \equiv \lambda x.a_1 \stackrel{B}{\to} \lambda x.b_1 \equiv b$

   il faut montrer que, si $\lambda x.a_1 \stackrel{B}{\to} c$, alors $c \equiv \lambda x.c_1$ et $a_1 \stackrel{B}{\to} c_1$.

3. $a \equiv a_1 a_2 \stackrel{B}{\to} b_1 b_2 \equiv b$. La discussion se fait sur la conclusion de la preuve de $a_1 a_2 \stackrel{B}{\to} c$. Deux cas sont possibles:

   - $c \equiv c_1 c_2$ et $a_i \stackrel{B}{\to} c_i$
   - $a_1 \equiv \lambda x.a_{10}$ et $c \equiv c_{10}\{c_2/x\}$ et $a_{10} \stackrel{B}{\to} c_{10}$ et $a_2 \stackrel{B}{\to} c_2$.

     Alors, il faut d'abord montrer que $b_1 \equiv \lambda x.b_{10}$ et $a_{10} \stackrel{B}{\to} b_{10}$ et $a_2 \stackrel{B}{\to} b_2$.

     Il existe donc d_1 tel que $b_{10} \stackrel{B}{\to} d_1$ et $c_{10} \stackrel{B}{\to} d_1$ De même, il existe d_2 tel que $b_2 \stackrel{B}{\to} d_2$ et $c_2 \stackrel{B}{\to} d_2$ Il suffit alors de prendre $d \equiv d_1\{d_2/x\}$.

4. $a \equiv (\lambda x.a_1)a_2 \stackrel{B}{\to} b_1\{b_2/x\} \equiv b$.

   La discussion se fait encore sur la conclusion de la preuve de $(\lambda x.a_1)a_2 \stackrel{B}{\to} c$. Deux cas sont possibles

- $c \equiv (\lambda x.c_1)c_2$. Cas dèjà étudié.
- $c \equiv c_1\{c_2/x\}$. On utilise le lemme **??**.

•

Proposition 3.5

$$\beta \subseteq B \subseteq \beta^*$$

Preuve
Prendre la définition axiomatique de la réduction β pour montrer que $\beta \subseteq B$ et celle de B pour montrer que ; $B \subseteq \beta^*$. •

Théorème 3.6 (Taït- Martin-Löf) *la β-réduction est confluente.*

Théorème 3.7 (Church-Rosser)

1. *Si $a =_\beta b$ alors il existe c tel que $a \overset{\beta^*}{\to} c$ et $b \overset{\beta^*}{\to} c$.*
2. *Si $a =_\beta b$ et si b est en β-forme normale, alors $a \overset{\beta^*}{\to} b$.*
3. *Tout terme $a \in \mathbf{\Lambda_V}$ a au plus une forme β-normale.*
4. *Si deux termes distincts a et b sont en β-forme normale, alors $a \neq_\beta b$. La théorie du λ-calcul est consistante.*

Preuve
Le premier point se montre par récurrence sur la preuve de $a =_\beta b$. Cet énoncé est valable pour toute relation confluente. Les autres points sont des conséquences évidentes du point 1. •

Exercice 3.1 *Décrire les formes normales des λ-termes.*

Montrer que si $a \overset{\beta^*}{\to} \lambda x_1 x_2 \cdots x_n \,.\, x b_1 \cdots b_p$ et si $a \overset{\beta^*}{\to} \lambda y_1 y_2 \cdots y_m \,.\, y c_1 \cdots c_q$, alors $n = m$, $p = q$ et $x = y$.

4 η-réduction

Le λ-calcul peut être enrichi avec une règle supplémentaire de calcul, appelée la η-réduction.

Définition 4.1 *La relation de η-réduction, notée η, est la relation définie par : $a \overset{\eta}{\to} b$ si*

$$a_{|u} = \lambda x.Mx \;\; b = a[u \longleftarrow M] \texttt{ si } x \notin FV(M)$$

La η-réduction traduit donc le fait que la construction du terme $(a\,x)$ puis l'abstraction de x dans $(a\,x)$ sont inutiles, si $x \notin FV(a)$. En effet, dans ce cas, $(\lambda x.(ax))b \overset{\beta}{\to} ab$: la fonction $\lambda x.ax$ n'est donc pas distingable de la fonction a : appliquées toutes deux à un même terme, elles donnent le même résultat.

Cette règle permet donc de supprimer des abstractions inutiles comme celle qui suit dans la fonction CAML :

```
let f =
 let g = function y -> corps_g
 in
 function x ->   g  x;;
```

qui s'écrirait aussi bien:

```
let f =
  function y -> corps_g
```

Cette règle est aussi utilisée dans l'autre sens, essentiellement dans les λ-calculs typés, car elle facilite le processus d'unification de deux termes. On parle alors de η-extension.

La théorie $\beta\eta$ est la théorie engendrée par $\beta \cup \eta$. L'extensionnalié peut aussi être exprimée de la manière suivante. Soit `ext` la règle équationnelle donnée par :

Soit $x \in V$ telle que $x \notin FV(ab)$, si $a\,x = b\,x$, alors $a = b$.

Soit $\lambda + \mathbf{ext}$ la théorie obtenue en étendant la relation $=_\beta$ par cette équation.

Proposition 4.2 *Les théories $\lambda + \mathbf{ext}$ et $\beta\eta$ sont équivalentes.*

<u>Preuve</u>

Montrons d'abord que η est une conséquence de $\lambda + \mathbf{ext}$. Appliquons $\lambda x.a\,x$ et a à une variable $y \notin FV(a)$. On a $(\lambda x.a\,x)y \xrightarrow{\beta} ay$ puisque $x \notin FV(a)$. Donc $(\lambda x.a\,x) =_{\lambda+\mathtt{ext}} a$.

Réciproquement, montrons que la règle `ext` est une conséquence de la théorie $\beta\eta$. Soient a et b tels que $ax =_{\beta\eta} bx$. Alors, grâce à la règle ξ, $\lambda x.a\,x =_{\beta\eta} \lambda x.b\,x$. Donc, $a =_{\beta\eta} b$. •

On va montrer que la réduction $\beta\eta$ est confluente, en utilisant le lemme de Hindley-Rosen.

Proposition 4.3 *1. η vérifie la propriété du diamant.*

2. β^* et η^* commutent.

3. $\beta\eta$ est confluente.

<u>Preuve</u>

1. La preuve se fait par récurrence sur la preuve de $a \xrightarrow{\eta} b$, quelle que soit la preuve de $a \xrightarrow{\eta} c$.

2. On démontre d'abord que si deux relations R_1 et R_2 vérifiant le diagramme suivant, alors R_1^* et R_2^* commutent.

$$\begin{tikzcd} M \arrow[r, "R_1"] \arrow[d, "R_2"'] & N \arrow[d, dashed, "R_2^*"] \\ P \arrow[r, dashed, "R_1 \cup \epsilon"] & Q \end{tikzcd}$$

   Puis on montre que η et β vérifient ce diagramme, par cas sur les positions respectives des radicaux. Le cas des radicaux emboîtés $\lambda x.(\lambda y.a)x$ utilise explicitement la α-conversion. Dans le cadre des λ-calculs typés, ce cas peut poser problème, si les λ ne sont pas décorés du même type.

3. Il reste à utiliser le lemme de Hindley-Rosen. Il affirme que, si deux relations R_1 et R_2 sont confluentes et si elles commutent, alors $R_1 \cup R_2$ est confluente.

•

5 Le pouvoir d'expression du λ-calcul

Nous allons montrer que les fonctions récursives sont définissables dans le λ-calcul. Donnons tout d'abord quelques termes du λ-calcul, permettant de définir les eléments essentiels d'un langage de programmation.

5.1 Programmation en λ-calcul

5.1.1 Les booléens

On pose :

$$\mathtt{true} = \lambda x\, y.x \quad \mathtt{false} = \lambda x\, y.y$$

Alors, $\mathtt{true}\ a\ b \xrightarrow{\beta^*} a$ et $\mathtt{false}\ a\ b \xrightarrow{\beta^*} b$. Donc, les λ-termes `true` et `false` agissent comme des opérateurs de choix, c'est-à-dire qu'ils ne retiennent que l'un de leurs deux arguments. La conditionnelle est définie par les dérivations suivantes:

$$\mathtt{cond}\,\mathtt{true}\,a\,b \xrightarrow{\beta^*} a \quad \mathtt{cond}\,\mathtt{false}\,a\,b \xrightarrow{\beta^*} b$$

On peut tenter de le construire tel que:

$$\mathtt{cond}\,\mathtt{true}\,a\,b \xrightarrow{\beta^*} \mathtt{true}\,a\,b \xrightarrow{\beta^*} a$$

Donc, `cond` doit construire l'application de son argument booléen à ses deux autres arguments. D' où, $\mathtt{cond} = \lambda p\, x\, y.(p\ x\ y)$.

Le terme `true` est souvent appelé `K`. Il vérifie, pour tout a, $\mathtt{K}\, a = \lambda y.a$. Notons que, dans cet énoncé, on fait usage de la convention de renommage : $y \notin FV(a)$. Le terme `false` sera simplement noté `F` dans la suite.

5.1.2 Les couples

Soient a et $b \in \mathbf{\Lambda_V}$. On veut construire un terme `C` et deux termes $\mathtt{C}_1$ et $\mathtt{C}_2$ tels que :

$$\mathtt{C}_1(\mathtt{C}\,a\,b) \xrightarrow{\beta^*} a \qquad \mathtt{C}_2(\mathtt{C}\,a\,b) \xrightarrow{\beta^*} b$$

Il suffit d'obtenir:

$$\mathtt{C}_1(\mathtt{C}\,a\,b) \xrightarrow{\beta^*} \mathtt{K}ab \qquad \mathtt{C}_2(\mathtt{C}\,a\,b) \xrightarrow{\beta^*} \mathtt{F}ab$$

Or, $\mathtt{K}ab =_\beta (\lambda c.cab)\mathtt{K}$. D' où les termes :

$$\mathtt{C} \equiv \lambda x\, y\, z\, .\, z\, x\, y \qquad \mathtt{C}_1 \equiv \lambda\, x.x\ (\lambda\, x\, y.x) \qquad \mathtt{C}_2 \equiv \lambda\, x.x\ (\lambda\, x\, y.y)$$

Vérifer que ces termes permettent bien de représenter les paires et les projections et que l'on a l'égalité suivante :

$$\mathtt{C}(\mathtt{C}_1(\mathtt{C}\,M\,N))(\mathtt{C}_2(\mathtt{C}\,M\,N)) \xrightarrow{\beta}{}^{*} (\mathtt{C}\,M\,N)$$

5.1.3 Les entiers et l'arithmétique entière

Il existe plusieurs familles de termes qui peuvent représenter les entiers ainsi que les opérations sur les entiers. En voici une. On peut la comprendre comme : un entier n est représenté par l'imbrication de $(n-1)$ paires.

On pose donc : $\overline{0} = \lambda x.x$ et $\overline{n+1} = \mathtt{C}\ \mathtt{F}\ \overline{n}$. La fonction sucesseur est représentée par le terme $\mathtt{S}$ défini par $\mathtt{S} = \lambda x.\mathtt{C}\ \mathtt{F}\ x$. Vérifier que $\mathtt{S}\ \overline{n} = \overline{n+1}$.

La fonction prédecesseur est représentée par le terme $\mathtt{P}$ défini par $\mathtt{P} = \lambda x.x\mathtt{F}$. Vérifier que $\mathtt{P}\ \overline{n+1} = \overline{n}$ et que $\mathtt{P}\ \overline{0} = \mathtt{F}$.

Le test d'égalité à zéro est représenté par le terme $\mathtt{Z}$ défini par : $\mathtt{Z} = \lambda xx\mathtt{K}$. Vérifier que $\mathtt{Z}\ \overline{n+1} = \mathtt{F}$ et que $\mathtt{Z}\ \overline{0} = \mathtt{K}$. les opérations seront définies un peu plus tard.

5.1.4 Les combinateurs de point-fixe

Proposition 5.1 *Quel que soit le λ-terme F, il existe un terme X tel que $F\,X =_\beta X$. X est dit point-fixe de F.*

Prendre $W \equiv \lambda x.F(x\,x)$ et $X \equiv W\,W$. •

Définition 5.2 *Un combinateur de point fixe est un terme, disons fix, tel que*

$$\forall a \in \Lambda,\ a\,(fix\,a) = fix\,a$$

Les combinateurs les plus connus sont le combinateur de Curry Y et le combinateur de Turing Θ.

$$Y \equiv \lambda F.WW \quad \Theta \equiv VV \quad \text{où} \quad V = \lambda xy.y(xxy)$$

Proposition 5.3 *Pour tout $a \in \Lambda$, $Ya =_\beta a(Ya)$ mais $a(Ya)$ n'est pas un réduit de Ya.
Pour tout $a \in \Lambda$, $\Theta a \longrightarrow^{\beta^*} a(\Theta a)$*

La preuve est laissée en exercice.

Exercice 5.1 *Soit le terme* $\mathtt{recfact} = \lambda fx.if\ n = 0\ then\ 1\ else\ f(n-1)$. *Soit le terme* $\mathtt{Fact} = \Theta\ \mathtt{recfact}$. *Montrer que* $\mathtt{Fact}\ 3 = 6$.

5.2 Complétude combinatoire

Dans le cas où $\{x_1, \cdots, x_n\} \not\subseteq FV(b_1) \cup \cdots \cup FV(b_n)$, la substitution de b_i à x_i est indépendante de l'ordre des substitutions. On introduit donc une méta-notation de substitution simultanée $\{b_1/x_1 \cdots b_n/x_n\}$ pour traduire ce fait. Chaque fois que cette notation sera utilisée, on supposera que la condition précédente sur les variables est vérifiée.

Proposition 5.4 *Soit a tel que $FV(a) \subseteq \{x_1, \cdots, x_n\}$. Il existe un terme b vérifiant $b\,x_1 \cdots x_n = a$. De plus, pour tous termes $c_1, \cdots, c_n$, on a $b\,c_1 \cdots c_n = a\{c_1/x_1 \cdots c_n/x_n\}$.*

Preuve
Il suffit de poser $b = \lambda x_1. \cdots x_n.a$. •

Proposition 5.5 *Soit a un terme tel que $FV(a) \subseteq \{f, x_1, \cdots, x_n\}$. Il existe un terme F tel que $Fc_1 \cdots c_n = a\{F/f, c_1/x_1, \cdots, c_n/x_n\}$.*

Preuve

Il ne suffit plus de faire l'abstraction des variables comme dans le cas précédent : F doit apparaître dans sa définition. On va donc utiliser un combinateur de point fixe. Soit $G = \lambda f.x_1 \cdots x_n.a$ et $F = \Theta G$. Alors $Fc_1 \cdots c_n \stackrel{\beta^*}{\to} a\{F/f, c_1/x_1, \cdots, c_n/x_n\}$. •

Exercice 5.2 *Trouver un λ-terme F tel que $F a\, b =_\beta FbaF$.*

5.3 La définissabilité

On va montrer dans un premier temps que les fonctions récursives totales sont représentables dans le λ-calcul. Ce résultat sera ensuite étendu aux fonctions récursives partielles.

Définition 5.6 *Une fonction ϕ définie sur N^p à valeurs dans N est dite λ-définissable s'il existe un λ-terme F tel que:*

$$\forall n_1, \cdots, n_p \in N, F\, \overline{n}_1 \cdots \overline{n}_p = \overline{\phi(n_1, \cdots, n_p)}$$

Proposition 5.7 *Les projections ${P^i}_p(n_0, \cdots, n_p) = n_i$ $(0 \leq i \leq p)$, la fonction constante égale à 0, la fonction successeur sont λ-définissables.*

Preuve

Les termes définissant ces fonctions ont été vus précédemment. •

Proposition 5.8 *La classe des fonctions λ-définissables est close par composition.*

Preuve

Soient ϕ et ψ_1, ..., ψ_n des fonctions λ-définissables respectivement par G et H_i. La fonction θ définie par :

$$\theta(p_1, \cdots, p_k) = \phi(\psi_1(p_1, \cdots, p_k), \cdots, \psi_n(p_1, \cdots, p_k))$$

est λ-définie par le terme

$$\lambda x_1. \cdots x_k.G(H_1 x_1 \cdots x_k) \cdots (H_n x_1 \cdots x_k)$$

•

Proposition 5.9 *La classe des fonctions λ-définissables est close par récursion primitive.*

Preuve

Soient ϕ la fonction définie par récursion primitive comme suit:

$$\phi(0, n_1, \cdots, n_k) = \theta(n_1, \cdots, n_k)$$

$$\phi(p+1, n_1, \cdots, n_k) = \psi(\phi(p, (n_1, \cdots, n_k)), p, n_1, \cdots, n_k)$$

Supposons θ et ψ λ-définies respectivement par G et H. On veut donc définir un terme F tel que :

$F\, p\, n_1 \cdots n_k =$
`if` $(\mathtt{Z}\, p)$
  `then` $G\, n_1 \cdots n_k$
  `else` $H\ (F\ (\mathtt{P\,p}) n_1 \cdots n_k)\ (\mathtt{P\,p})\ n_1 \cdots n_k$

Ce terme est donc défini par une équation de point-fixe. L'exercice sur les points-fixes permet d'affirmer que ce terme existe. Il doit être construit à partir de `cond`. •

Reste à montrer que la classe des fonctions λ-définissables est close par minimalisation. Il faut donc définir un λ-terme calculant la plus petite valeur "vérifiant une propriété". Pour tout λ-terme P, on pose :

$$H_P = \Theta\,(\lambda hz.\,\texttt{if}\ \ (Pz)\ \ \texttt{then}\ \ z\ \ \texttt{else}\ \ h\,(\texttt{S}\,z))$$

$$\mu(P) = H_P\overline{0}$$

Proposition 5.10 *Soit P un λ-terme tel que, pour tout $\overline{n}$, $P\overline{n} = \texttt{K}$ ou $P\overline{n} = \texttt{F}$. Alors:*

1.

$$H_P\,z \overset{\beta^*}{\to} \texttt{if}\ (Pz)\ \texttt{then}\ z\ \texttt{else}\ H_P(\texttt{S}z)$$

2. Si $\exists n.P(\overline{n}) = \texttt{K}$, on pose $m = \mu n.\{\overline{n}|P(\overline{n}) = \texttt{K}\}$. Alors $\mu(P) = \overline{m}$.

Preuve
Le point 1. s'obtient par un simple calcul (à faire).

Pour le point 2. , on montre que, si $P(\overline{n}) = \texttt{K}$, alors $H_P(\overline{n}) = \overline{n}$. On montre ensuite que si $P(\overline{n}) = \texttt{F}$, alors $H_P(\overline{n}) = H_P(\overline{(n+1)})$. D'où le résultat. •

Proposition 5.11 *La classe des fonctions λ-définissables est close par minimalisation.*

Preuve
Soit la fonction définie par:

$$\phi(n_1, \cdots, n_p) = \mu m.(\psi(n_1, \cdots, n_p, m) = 0)$$

où ψ est une fonction λ-définie par G, telle que $\forall n_1, \cdots, n_p, \exists m, \psi((n_1, \cdots, n_p, m) = 0$. C'est ici qu'intervient l'hypothèse de totalité des fonctions. Il suffit de définir F par:

$$Fx_1 \cdots x_p = \mu(\lambda y.\texttt{Z}\ (Gx_1 \cdots x_p y))$$

•

Théorème 5.12 (Kleene) *Les fonctions λ-définissables sont exactement les fonctions récursives.*

Preuve
Rappelons que toutes les fonctions considérées sont des fonctions totales. L'ensemble des fonctions récursives est par définition le plus petit ensemble contenant les projections, le test à zéro, la fonction successeur, clos par composition, récursion primitive et minimalisation. Il est donc contenu dans l'ensemble des fonctions λ-définissables. Réciproquement, si ϕ est λ-définissable par F, alors son graphe est récursivement énumérable: il suffit d'énumérer les n-uplets de λ-termes $(\overline{n}_1, \cdots, \overline{n}_p, \texttt{m})$ vérifiant $(F\ \overline{n}_1 \cdots \overline{n}_p = \texttt{m})$. La fonction ϕ est donc récursive. •

Le système d'entiers choisi pour traiter de la définissabilité, appelé système d'entiers *standard*, peut être remplacé par n'importe quelle suite (d_i) de λ-termes à condtion que cette suite vérifie les propriétés suivantes. Les d_i sont des termes clos ayant tous une forme normale et il existe des termes `Succ`, `Pred` et `Zero` tels que:

1. `Succ` $d_n = d_{n+1}$.
2. `Zero` $d_0 = \mathcal{K}$ et `Zero` $d_{n+1} = \texttt{F}$.

3. $\texttt{Pred}\ d_{i+1} = d_i$.

Par exemple, les entiers de Church constituent un tel système. Ils sont définis par:

1. $d_0 = \lambda fx.x$ et $d_n = \lambda fx.\, f^n\, x$.
2. $\texttt{Succ} = \lambda nfx.f(nfx)$.
3. $\texttt{Zero}= \lambda nyx.n(\mathcal{K}x)y$
4. $\texttt{add} = \lambda mnfz.(mf)(nfz)$.
5. $\texttt{mult} = \lambda mnp.m(n\,p)$
6. $\texttt{exp} = \lambda mn.nm$

Les entiers de Church sont souvent appelés *itérateurs* de Church. Pourquoi?

La définition directe du prédecesseur est plus compliquée. On peut tout simplement se ramener aux entiers définis précédemment en construisant un terme f tel que $f\overline{n} =_\beta d_n$ et un terme f^{-1} tel que $f^{-1}(d_n) =_\beta \overline{n}$.

Pour f, on peut choisir le terme solution de l'équation suivante:

$$f = \lambda x.\texttt{if}\ \texttt{Z}\, x\, then\, d_0\, else\, \texttt{Succ}(f\texttt{P}x)$$

Pour f^{-1}, on peut choisir le terme $\lambda x.x\texttt{S}\overline{0}$.

De même, on peut utiliser ces inter-traductions pour définir les opérations sur les entiers standard.

Proposition 5.13 *Théorème du double point fixe Pour tous termes f et g, il existe des termes a et b tels que $a = fab$ et $b = gab$.*

6 Stratégies de calcul

Certains λ-termes n'ont pas de forme normale (par exemple Ω), d'autres ont une forme normale mais possèdent aussi des dérivations infinies (par exemple $(\lambda xy.y)\Omega$), enfin, certains termes ont une forme normale *forte* : toutes leurs dérivations conduisent à cette forme normale. Il y a donc deux problèmes à résoudre.

- Si un terme possède une forme normale, existe-t-il une stratégie de dérivation qui y amène à coup sûr?
- Parmi toutes les dérivations menant à la forme normale, y en a-t-il de meilleures? (Que veut d'ailleurs dire "meilleure"?).

Exemple 6.1 Soit $\Delta = \lambda x.x\,x$ et $\texttt{I} = \lambda y.y$

Examinons les deux dérivations suivantes:

$$\Delta(\texttt{I}x) \stackrel{\beta}{\to} \Delta x \stackrel{\beta}{\to} xx$$

$$\Delta(\texttt{I}x) \stackrel{\beta}{\to} (\texttt{I}x)(\texttt{I}x) \stackrel{\beta}{\to} x(\texttt{I}x) \stackrel{\beta}{\to} xx$$

La première dérivation qui réduit d'abord les radicaux les plus internes (les plus profonds dans l'arbre) est donc plus courte que la dérivation réduisant d'abord le plus externe. Considérons ensuite les dérivations suivantes:

$$(\lambda x.x\mathtt{I})\,\lambda y.(\Delta(yz)) \stackrel{20}{\to} (\lambda x.x\mathtt{I})\,(\lambda y.(yz)(yz)) \stackrel{\epsilon}{\to} (\lambda y.(yz)(yz))\mathtt{I} \stackrel{\epsilon}{\to} (\mathtt{I}z)(\mathtt{I}z)$$

.

$$(\lambda x.x\mathtt{I})\,\lambda y.(\Delta(yz)) \stackrel{\epsilon}{\to} \lambda y.(\Delta(yz))\mathtt{I} \stackrel{\epsilon}{\to} \Delta(\mathtt{I}z) \stackrel{2}{\to} \Delta z \to zz$$

.

La dérivation réduisant toujours le radical le plus interne est cette fois plus longue que la seconde dérivation. Trouver une stratégie optimale n'a donc rien d'évident. Ce problème a d'ailleurs été résolu très récemment (travaux de Lamping, Khatail puis Abadi, Lévy et Gonthier puis Asperti, etc.), ces travaux s'appuyant eux-mêmes sur les résultats fondamentaux de Lévy sur l'optimalité.

6.1 Les résidus

Pour étudier les stratégies de calcul, il faut pouvoir suivre les radicaux d'un terme ou d'un réduit de ce terme au cours d'une dérivation. Soit a un terme contenant une famille de radicaux R_i. Soit b tel que $a \stackrel{\beta}{\to} b$. Un radical de b peut provenir d'un radical R_k de a : il sera dit *résidu* de R_k dans b. Il peut aussi être créé par la réduction (par exemple , $(\lambda x.xa)(\lambda y.y) \stackrel{\beta}{\to} (\lambda y.y)a)$. Tout radical de a est soit réduit par la réduction, soit effacé, soit possède au moins un résidu dans b. Nous allons définir précisément la notion de *résidu d'une occurrence* (sous-entendu d'un radical).

Définition 6.2 Soit une dérivation ρ de a vers b. Soit $u \in Occ(a)$ tel que $a|_u$ soit un radical. On note u/ρ l'ensemble des résidus de u après ρ donc les résidus de u dans b. Cet ensemble est défini par:

1. si $\rho = \emptyset$, alors $u/\rho = \{u\}$.
2. $u/(\rho\sigma) = (u/\rho)/\sigma = \{w | w \in v/\sigma \text{ et } v \in u/\rho\}$.
3. si $|\rho| = 1$, si $a \stackrel{v}{\to} b$ alors:
   (a) si u et v disjoints, alors $u/\rho = \{u\}$.
   (b) si $u < v$, alors $u/\rho = \{u\}$.
   (c) si $u = v$, alors $u/\rho = \emptyset$.
   (d) si $v < u$, alors $a|_v = (\lambda x.M)N$. si $u = v10u_1$, alors $u/\rho = \{vu_1\}$. Si $u = v2u_2$ alors $u/\rho = \{vwu_2 | w\ t.q.\ \text{lieur}(v10w) = v1\}$.

Exemple 6.3 On va souligner le radical considéré et ses résidus. Soit $\Delta = \lambda x.x\,x$ et $I = \lambda y.y$.

$$\Delta\underline{(Iz)} \stackrel{\beta}{\to} \underline{(Iz)(Iz)}$$

$$\underline{(Iz)}(\Delta(Ix)) \stackrel{\beta}{\to} \underline{(Iz)}((Ix)(Ix))$$

$$\underline{I(\Delta(Ix))} \stackrel{\beta}{\to} \underline{I((Ix)(Ix))}$$

$$\underline{\Delta(Ix)} \stackrel{\beta}{\to} (Ix)(Ix)$$

$$\underline{\Delta\Delta} \stackrel{\beta}{\to} \Delta\Delta$$

Remarque 6.1 1. Une occurrence résidu d'un radical est encore un radical. D'où la notation R/ρ, désignant les résidus (de l'occurrence) du radical R après ρ.

2. Si $S \in R/\rho$, alors S peut être différent de R :
$\lambda x.\underline{(\lambda y.xy)a)})b \xrightarrow{\beta} \underline{(\lambda y.by)a\{b/x\}}$.

3. Soient $\rho : a \xrightarrow{\beta^*} b$ et $\sigma : a \xrightarrow{\beta^*} b$. Alors u/ρ et u/σ peuvent être différents. Soient les dérivations ρ et σ suivantes:

$$\mathtt{I}(\mathtt{I}x) \xrightarrow{\epsilon} \mathtt{I}x \text{ et } \mathtt{I}(\mathtt{I}x) \xrightarrow{2} \mathtt{I}x$$

Dans cet exemple, $\epsilon/\rho = \emptyset$ et $\epsilon/\sigma = \{\epsilon\}$. Les résidus dépendent donc des dérivations effectuées et non pas seulement des termes de départ et d'arrivée.

4. Les résidus de deux radicaux disjoints peuvent ne pas être disjoints. Soit le terme $a = (\lambda x.R_1)R_2$ où $R_1 = (\lambda u.x)v$. Soit la réduction $a \xrightarrow{\epsilon} (\lambda u.R_2)v$. Le résidu de R_1 contient le résidu de R_2.

5. Un même radical peut avoir des résidus emboîtés.

$$\Delta(\lambda x.\underline{\mathtt{I}}x) \xrightarrow{\beta} (\lambda x.\underline{\mathtt{I}}x)(\lambda x.\underline{\mathtt{I}}x) \xrightarrow{\beta} \underline{\mathtt{I}}(\lambda x.\underline{\mathtt{I}}x)$$

6. Un radical de b est dit créé par la réduction $a \xrightarrow{u} b$ s'il n'est pas résidu d'un radical de a. Il n'y a que trois cas possibles de créations:
   (a) $a|_v = (\lambda x.(\lambda y.M))NP$ et $u = v1$. Le radical créé est $(\lambda y.M\{N/x\})P$.
   (b) $a|_v = (\lambda x.x)(\lambda y.M)N$ et $u = v1$. Le radical créé est $(\lambda y.M)N$.
   (c) $a|_u = (\lambda x.(C[\,xM\,]))(\lambda y.N)$. Le radical créé est $(\lambda y.N)M$.

6.2 Le théorème des Développements Finis

Le théorème des Développements Finis affirme que, si l'on se contente de réduire les radicaux présents dans un terme et leurs résidus, alors, quelle que soit la stratégie suivie, on obtient le même terme, en un nombre fini d'étapes.

Pour démontrer le théorême des Développements Finis, mais aussi pour prouver un certain nombre d'autres résultats, on a besoin de marquer certains radicaux. On va donc introduire cette notion à travers un exercice. Ce marquage de radicaux permet de se dispenser de manipuler explicitement les ensembles résidus.

Exercice 6.1 *La plupart des questions de cet exercice sont très simples. La solution consiste le plus souvent en une reformulation de preuves vues en cours.*

Soit V un ensemble de variables. Soit M un ensemble de marques. L'ensemble Λ_m des λ-termes (avec radicaux) marqués est le plus petit ensemble tel que :

1. $V \in \Lambda_m$
2. $\lambda x.a \in \Lambda_m$ et $a\, b \in \Lambda_m$ si a et $b \in \Lambda_m$
3. $(\lambda_m x.a)b \in \Lambda_m$ si a et $b \in \Lambda_m$ et $m \in M$. Ce terme est dit marqué avec m.

1. Donner des exemples de termes marqués. Définir la substitution dans Λ_m. On conservera la notation de $\mathbf{\Lambda_V}$.

2. La notion de réduction sur Λ_m est la réunion de β et β_m définies par:

$$(\lambda x.a)b \stackrel{\beta}{\rightarrow} a\{b/x\} \qquad (\lambda_m x.a)b \stackrel{\beta_m}{\rightarrow} a\{b/x\}$$

   Définir la réduction associée notée $m\beta$ et la relation d'interconvertibilité.

3. Définir la fonction B_m sur Λ_m qui réduit en une seule étape tous les radicaux marqués d'un terme. On pourra s'inspirer de la définition de B. Pourquoi B_m est-elle bien une fonction et pas seulement une relation?

4. Soit la fonction $|\ | : \Lambda_m \rightarrow \mathbf{\Lambda_V}$ qui enlève les marques. Démontrer que si $a \stackrel{m\beta^*}{\rightarrow} b$, alors $|a| \stackrel{\beta^*}{\rightarrow} |b|$ et réciproquement.

5. Soient a et $b \in \Lambda_m$. Montrer que:

$$B_m(a\{b/x\}) \equiv (B_m(a))\{B_m(b)/x\}$$

6. Démontrer la propriété suivante:

$$\begin{tikzcd} a \arrow[r, "m\beta^*"] \arrow[d] & b \arrow[d] \\ B_m(a) \arrow[r, dashed, "\beta^*"] & B_m(b) \end{tikzcd}$$

7. Démontrer que $|a| \stackrel{\beta^*}{\rightarrow} B_m a$.

8. **Strip Lemma** Démontrer que:

$$\begin{tikzcd} a \arrow[r, "\beta^*"] \arrow[d, "\beta"'] & b \arrow[d, "\beta^*"'] \\ c \arrow[r, dashed, "\beta^*"] & B_m(b) \end{tikzcd}$$

9. En déduire la confluence de la β-réduction.

<u>Preuve</u>
Les points 1., 2. et 3. sont des reformulations des notions analogues pour la β-réduction. Le point 4. repose sur la remarque que le marquage conserve les radicaux et que la relation $m\beta$ contient la relation β.

Le point 5. se montre par récurrence structurelle sur a, en montrant au passage un lemme de substitution pour β_m.

Le point 6. se montre par récurrence sur la longueur de la $m\beta$ dérivation.

Le point 7. se montre par récurrence structurelle sur a.

Le point 8. se montre en marquant le radical réduit dans a pour obtenir c et en utilisant tous les résultats précédents.

•

Définition 6.4 Soit $\mathcal{F}$ un ensemble d'occurrences de radicaux d'un terme a. On note $\mathcal{F}/\rho$ l'ensemble $\{v | v \in u/\rho$ et $u \in \mathcal{F}\}$. L'ensemble des dérivations relatives à $\mathcal{F}$ est le plus petit sous-ensemble, noté $D(\mathcal{F})$, des dérivations de a tel que:

1. $\emptyset \in D(\mathcal{F})$,
2. si $\rho \in D(\mathcal{F})$ et $a \xrightarrow{\rho} b$, si $\sigma = b \xrightarrow{v} c$ et $v \in \mathcal{F}/\rho$, alors $\rho\sigma \in D(\mathcal{F})$.

Un développement de $\mathcal{F}$ est une dérivation ρ relative à $\mathcal{F}$ telle que $\mathcal{F}/\rho$ soit vide (parfois appelé développement complet).

Le théorème suivant affirme qu'il n'est pas possible de réduire indéfiniment des radicaux de $\mathcal{F}$. L'emboîtement possible de radicaux posait le problème. De plus, quelle que soit la stratégie choisie pour les réaliser, les développements de $\mathcal{F}$ confluent sur un même terme.

Théorème 6.5 *Soit $\mathcal{F}$ un ensemble de radicaux de a. Il n'y a pas de dérivation infinie relative à $\mathcal{F}$.*

Preuve

Il existe plusieurs preuves de ce théorême, toutes assez techniques.

Pour parler des développements de $\mathcal{F}$, il faut se donner le moyen de reconnaître les radicaux de $\mathcal{F}$. On va donc les marquer. On obtient donc un terme de Λ_m. Montrer qu'il n'y a pas de dérivation infinie relative à $\mathcal{F}$ revient à montrer que tout développement de $\mathcal{F}$ termine. On va définir une fonction de poids sur Λ_m comme suit et montrer que le poids d'un terme diminue strictement par β_m-réduction.

Définition d'une fonction de poids A toute occurrence d'une variable dans un terme, on associe un entier strictement positif, on note x^n l'association de x à l'entier n et l'entier n est appelé le poids de l'occurrence de la variable x.

Le poids d'un terme a, noté $||a||$, est la somme des poids de ses variables.

Poids décroissant Un poids est dit décroissant si, quel que soit le radical marqué $(\lambda_m x.a)b$, le poids de toute occurrence de x dans a est strictement supérieur au poids de b.

Tout terme peut être muni d'un poids décroissant On numérote les occurrences des variables de a de la droite vers la gauche, en commençant à 0. Par exemple, le terme $(\lambda x.xxx)(\lambda y.yy)$ devient $(\lambda_F x.x_4x_3x_2)(\lambda y.y_1y_0)$. Le poids de l'occurrence de la variable de numéro i est alors 2^i.

Lemme Soit $a \in \lambda_m$ et $a \xrightarrow{\beta_m} b$. Alors, $||a|| > ||b||$. De plus, le poids induit par le poids de a sur b est décroissant.

Quand on réduit un radical marqué $R = (\lambda_m x.(..x_{n_1}...x_{n_p})N = (\lambda_m x.M)N$, on remplace chacune des occurrences de x par N, de poids strictement plus petit que chacune de ces occurrences. Si $x \notin R$, alors N est effacé, ce qui fait aussi diminuer strictement le poids.

Tout radical marqué de b, disons $(\lambda_m.x^j.P')Q'$, est résidu d'un radical marqué de a, $(\lambda_m.x^j.P)Q$. Si $R \subseteq P$ ou si $(\lambda_m.x^j.P)Q \subseteq N$, le résultat est évident.

Si $R \subseteq Q$, comme $||R|| > ||M\{N/x\}||$, on a $||Q|| > ||Q'||$. Le poids de b reste donc décroissant.

Si $(\lambda_m.x^j.P)Q \subseteq M$, alors $P' = P\{N/x\}$ et $Q' = Q\{N/x\}$ mais $||Q|| > ||Q'||$.

Conclusion si $a \overset{\beta_m^*}{\rightarrow} b$, alors $||a|| > ||b||$. La relation β_m a donc la propriété de terminaison forte.

•

Théorème 6.6 *Soit $\mathcal{F}$ un ensemble de radicaux de a. Alors:*

1. *Si $a \overset{\rho}{\rightarrow} b$, si $a \overset{\sigma}{\rightarrow} c$ et si ρ et σ sont deux développements complets de $\mathcal{F}$, alors $b \equiv c$.*
2. *De plus, pour tout radical R de a, on a: $R/\rho = R/\sigma$*

<u>Preuve</u>

1. Il suffit de montrer que la relation $a \overset{R}{\rightarrow} b$ si $R \in D(\mathcal{F})$ est confluente. Puisque cette relation est noethérienne, il suffit de montrer qu'elle est faiblement (ou localement) confluente. Il suffit pour cela de reprendre la preuve de la confluence faible de β, en marquant les radicaux, donc de démontrer l'existence du diagramme suivant.

$$\begin{tikzcd} a \arrow[r, "R_1"] \arrow[d, "R_2"'] & b \arrow[d, dashed, "R_2/R_1"] \\ c \arrow[r, dashed, "R_1/R_2"] & d \end{tikzcd}$$

Il faut aussi montrer que R_2/R_1 et R_1/R_2 sont bien des dérivations relatives à $\mathcal{F}$. Il suffit pour cela de constater que le résidu d'un radical marqué est encore un radical marqué.

2. Pour montrer le deuxième point, il suffit d'ajouter le radical considéré à la famille $\mathcal{F}$ sans le réduire. Les deux dérivations ρ et σ se terminent sur un même terme b. Tout radical marqué de b est un résidu à la fois de ρ et σ puisqu'il a subsisté par les deux dérivations (qui ne peuvent ni ajouter ni enlever des marques).

•

Grâce au théorême des Développements Finis, on peut donc parler du développement complet d'une famille de radicaux, noté $a \overset{\mathcal{F}}{\rightarrow} b$, sans ambigüité. Le lemme des mouvements parallèles (parallel moves lemma) est une reformulation du théorème des développements finis.

Proposition 6.7 *Soit un terme a contenant deux familles F_1 et F_2 de radicaux. On a le diagramme suivant:*

$$\begin{tikzcd} a \arrow[r, "F_1"] \arrow[d, "F_2"'] & b \arrow[d, dashed, "F_2/F_1"] \\ c \arrow[r, dashed, "F_1/F_2"] & d \end{tikzcd}$$

<u>Preuve</u>
Il suffit de considérer la famille de radicaux $\mathcal{F} = F_1 \cup F_2$ et d'appliquer le théorème des développements finis. •

On va maintenant redonner une preuve de confluence de β.

Théorème 6.8 *La β-réduction est confluente.*

<u>Preuve</u>
Soit D la relation définie par: $(a\ D\ b)$ s'il existe une famille $\mathcal{F}$ de radicaux de a tel que b soit le résultat d'un développement de $\mathcal{F}$. On montre que β^* est la fermeture transitive de D. Le théorème des développements finis nous assure alors que D vérifie la propriété du diamant. •

La preuve de Taït-Martin-Löf est un cas particulier de cette preuve, dans laquelle la famille réduite par la relation B est choisie de manière à pouvoir utiliser une récurrence structurelle (au lieu d'avoir à montrer la terminaison).

On donne en exercice une autre preuve du théorème des Développements Finis, qui rappelle la preuve de Taït pour la normalisation forte.

Définition 6.9 L'ensemble FD est le plus petit ensemble de Λ_m tel que:

1. $V \subseteq FD$.
2. si $a \in FD$, alors $\lambda x.a \in FD$.
3. si a et $b \in FD$ alors $ab \in FD$.
4. Si $a\{b/x\} \in FD$ et $b \in FD$, alors $(\lambda_m x.a)b \in FD$.

Proposition 6.10 *1. Si a et $b \in FD$ alors $a\{b/x\} \in FD$.*

2. $\Lambda_m \subseteq FD$.

3. toute β_m-dérivation est finie.

<u>Preuve</u>

1. Par récurrence sur la preuve de $a \in FD$.
2. Par récurrence sur la structure du terme $a \in \Lambda_m$.
3. Par récurrence sur la preuve de $a \in FD$, en remarquant que si $a = a_1 a_2$ et si a_1, $a_2 \in FD$, alors $a_1 \notin \lambda x.c$.

•

7 Standardisation

Définition 7.1 Un radical R d'occurrence u dans un terme a est dit *avant* un radical S d'occurrence v dans *l'ordre standard* si:

1. soit $u = w1u'$ et $v = w2v'$. Donc, $a|_w$ est une application dont le fils gauche contient R et le fils droit contient S.

2. $v = uv'$. Donc, R contient S.

Notez que dans les deux cas, le λ de R est à gauche du λ de S dans l'écriture linéaire de a.

Définition 7.2 Une dérivation $\rho\ a = a_0 \xrightarrow{u_1} a_1 \to \cdots a_{i-1} \xrightarrow{u_i} a_i \cdots \xrightarrow{u_n} a_n$ est dite *standard* si, pour tout $1 \leq i < j \leq n$, s'il existe $v_i \in a_{i-1}$ tel que $u_j \in v_i/\rho_{i,j-1}$, alors u_i est avant v_i dans l'ordre standard ($\rho_{i,j-1}$ désigne la partie de ρ entre a_i et a_{j-1}). Autrement dit si le radical réduit à la j-ième étape est un résidu d'un radical v_i présent dans a_{i-1}, alors le radical u_i réduit à la i-ème étape est avant v_i.

Une dérivation est dite *normale* si elle réduit à chaque étape le radical le plus petit dans l'ordre standard.

Une dérivation standard contracte donc les radicaux de l'extérieur vers l'intérieur et ("pour un même niveau") de la gauche vers la droite.

Exercice 7.1 Donner un exemple de dérivation standard non normale, de dérivation normale. Donner un exemple de deux dérivations de mêmes extrémités dont l'une est standard et l'autre non.

Démontrer que si ρ et σ sont deux dérivations, si la dérivation $\rho;\sigma$ (ρ suivie de σ) est standard, alors ρ et σ sont standard. Montrer que la réciproque n'est pas vraie.

Définition 7.3 Un terme a est dit en *forme normale de tête* s'il est de la forme $a = \lambda x_1 \cdots x_n.ya_1 \cdots a_p$, avec $n, p \geq 0$. y est appelée la *variable de tête*. Si a est de la forme $a = \lambda x_1 \cdots x_n.(\lambda y.a_0)a_1 \cdots a_p$, le radical $(\lambda y.a_0)a_1$ est appelé *radical de tête*. Tout autre radical de a est dit *interne* à a. Une dérivation est dite *interne* si elle ne réduit que des radicaux internes. Elle est dite *de tête* si elle ne réduit que des radicaux de tête.

Lemme 7.4 S'il existe, le radical de tête est le plus petit dans l'ordre standard et il est aussi le radical le plus à gauche (mais le radical le plus à gauche n'est pas toujours un radical de tête — donner un exemple). Toute dérivation de tête est donc une dérivation normale. Si D_1 et D_2 sont deux dérivations de tête d'un terme a, alors $D_1 \subseteq D_2$ (ou l'inverse). Toute dérivation de tête est donc prolongeable en une dérivation de tête unique, appelée *dérivation normale de tête*.

Exercice 7.2 Donner un terme a contenant un radical interne R et une dérivation de a vers b tel que un résidu de R soit le radical de tête de b.

Lemme 7.5 *Soit $\sigma = a \xrightarrow{u} b$ une réduction interne de a. Alors,*

1. *Si a a un radical de tête à l'occurrence h, alors b a aussi un radical de tête qui est $h/u = \{h\}$.*

2. *Si b a un radical de tête à l'occurrence h, alors a a aussi un radical de tête à la même occurrence.*

3. *Les résidus des radicaux internes à a sont des radicaux internes de b.*

4. *On a le diagramme suivant:*

$$\begin{tikzcd} a \arrow[r, "u"] \arrow[d, "h"'] & b \arrow[d, dashed, "h"] \\ c \arrow[r, dashed, "u/h"] & d \end{tikzcd}$$

La dérivation u/h peut contenir encore des radicaux de tête, résidus de radicaux internes par la réduction h.

5. *Soit $a = \lambda x_1 \ldots x_p.a_1 \ldots a_k$. Soit $a \stackrel{D(i)}{\rightarrow} b$, où $D(i)$ est une dérivation interne. Alors, b est de la forme $\lambda x_1 \ldots x_p.b_1 \ldots b_k$.*

Lemme 7.6 *Soit $\mathcal{F}$ une famille de radicaux d'un terme a. Tout développement de $\mathcal{F}$ peut être factorisé en un dérivation de tête $D(h)$ suivie d'une dérivation interne, qui est le développement de $\mathcal{F}/D(h)$.*

<u>Preuve</u>
Notons d'abord que, si $\mathcal{F}$ ne contient pas le radical de tête, alors $D(h)$ est la dérivation vide. En effet, les réductions de radicaux internes ne créent pas de radicaux de tête et les résidus des radicaux internes par des dérivations internes sont encore des radiacaux internes.

On fait la preuve par récurrence sur le sup des longueurs des développements de $\mathcal{F}$. Notons h l'éventuel radical de tête de a et $a \stackrel{D(\mathcal{F})}{\rightarrow} b$ le développement de $\mathcal{F}$ considéré.

Considérons le cas où $\mathcal{F}$ ne contient qu'un radical. Si ce radical est interne, soit $D(h)$ est vide et $\mathcal{F}/D(h) = \mathcal{F}$. Si ce radical est le radical de tête, alors $D(h)$ est de longueur 1 et $\mathcal{F}/D(h)$ est vide.

Dans le cas général, on réduit le radical de tête présent dans $\mathcal{F}$ et on finit de développer $\mathcal{F}$. On obtient le diagramme suivant, où $\mathcal{F}/h$ contient encore un éventuel radical de tête.

$$\begin{tikzcd} a \arrow[r, "D(\mathcal{F})"] \arrow[d, "h"'] & b \arrow[d, equal] \\ c \arrow[r, dashed, "\mathcal{F}/h"] & b \end{tikzcd}$$

Le sup des longueurs des développements de $\mathcal{F}/h$ est, d'après le théorème des développements finis, strictement plus petit que celui de $\mathcal{F}$. On peut donc appliquer l'hypothèse de récurrence : il existe une dérivation de tête $D'(h)$ et une dérivation interne $D(i)$ telles que:

$$\begin{tikzcd} c \arrow[r, "D(\mathcal{F}/h)"] \arrow[d, "D'(h)"'] & b \arrow[d, equal] \\ f \arrow[r, dashed, "D(i)"] & b \end{tikzcd}$$

Il reste à accoler les diagrammes, pour obtenir le le résultat, où $D(h)$ est une dérivation de tête et $\mathcal{F}/D(h)$ est une dérivation interne.

$$\begin{tikzcd} a \arrow[r, "D(\mathcal{F})"] \arrow[d, "D(h)"'] & b \arrow[d, equal] \\ f \arrow[r, dashed, "\mathcal{F}/D(h)"] & b \end{tikzcd}$$

•

Lemme 7.7 *Si* $\stackrel{i}{\to}$ *désigne une réduction interne et* $\stackrel{D(h)}{\to}$ *une dérivation de tête, on a le diagramme suivant, où $D'(h)$ est une dérivation de tête et $D(i)$ une dérivation interne.*

$$\begin{tikzcd} a \arrow[r, "i"] \arrow[d, dashed, "D'(h)"'] & b \arrow[d, "D(h)"] \\ c \arrow[r, dashed, "D(i)"] & d \end{tikzcd}$$

<u>Preuve</u>
Soit R le radical interne réduit par $a \stackrel{i}{\to} b$. On va faire une preuve par récurrence sur la longueur de $b \stackrel{D(h)}{\to} d$. On étudie d'abord le cas où cette dérivation est de longueur 1.

Si b a un radical de tête, alors a a aussi un radical de tête H, qui a un seul résidu dans b, donc réduit par $b \stackrel{h}{\to} d$. Soit $\mathcal{F} = \{H; R\}$.

D'après le lemme précédent, la dérivation $a \stackrel{i}{\to} b \stackrel{h}{\to} d$ est un développement de $\mathcal{F}$, qui peut donc être factorisé en une dérivation de tête de a vers c, $D'(h)$, suivie d'une dérivation interne de c vers d, $D(i)$. Mais cette dernière dérivation n'est pas en général de longueur 1. Il faut donc modifier l'hypothèse de récurrence en remplaçant l'hypothèse $a \stackrel{i}{\to} b$ par $a \stackrel{D(i)}{\to} b$ où $D(i)$ désigne un développement d'une famille $\mathcal{F}_1$ de radicaux internes à a. On a donc à montrer:

$$\begin{tikzcd} a \arrow[r, "D(i)"] \arrow[d, dashed, "D'(h)"'] & b \arrow[d, "h"] \\ c \arrow[r, dashed, "D(i)"] & d \end{tikzcd}$$

On pose donc $\mathcal{F} = \{H\} \cup \mathcal{F}_1$ et on factorise ce développement de $\mathcal{F}$. Il existe donc c tel que $a \stackrel{D'(h)}{\to} c$, où $D'(h)$ est une dérivation de tête et $c \stackrel{D'(i)}{\to} d$ un développement de radicaux internes à a. D'où le résultat pour une dérivation de $b \stackrel{h}{\to} d$ de longueur 1.

Le cas général se montre alors sans difficulté, en accolant des instances du diagramme précédent le long de la dérivation $D(h)$. •

Lemme 7.8 *Soit $a \stackrel{\beta*}{\to} b$. Il existe c tel que $a \stackrel{D(h)}{\to} c$ et $c \stackrel{D(i)}{\to} b$, où $D(h)$ et $D(i)$ sont des dérivations respectivement de tête et interne.*

<u>Preuve</u>
La dérivation $a \stackrel{\beta*}{\to} b$ est une suite de dérivations de tête et de dérivations internes. Appliquer le lemme de commutation précédent autant que nécessaire, en faisant permuter toute étape de dérivation de tête avec l' étape de réduction interne la précédant immédiatement dans la dérivation de a vers b considérée. •

Théorème 7.9 *Si $a \stackrel{\beta*}{\to} b$, il existe une dérivation standard de a vers b.*

<u>Preuve</u>
Soit c donné par le lemme précédent : $a \overset{D(h)}{\rightarrow} c$ et $D(h)$ est une dérivation de tête. On fait la preuve sur la taille de b.

Si $b = x$, alors $c = x$ et $a \overset{D(h)}{\rightarrow} c$, qui est une dérivation de tête, est une dérivation standard.

b et c sont de la "même forme" : si $b = \lambda x_1 \cdots \lambda x_n.\, b_1 \cdots b_p$, alors c est de la forme $c = \lambda x_1 \cdots \lambda x_n.\, c_1 \cdots c_p$ et $c_i \overset{\beta^*}{\rightarrow} b_i$. Par hypothèse de récurrence, il existe une dérivation standard D_i de c_i sur b_i. La dérivation $a \overset{D(h)}{\rightarrow} c \overset{D_1}{\rightarrow} \overset{D_2}{\rightarrow} \cdots \overset{D_p}{\rightarrow} b$ est clairement une dérivation standard. •

Corollaire 7.10 *Un terme a a une forme normale de tête si et seulement si sa dérivation normale de tête termine.*

Soit un terme a, dont la dérivation normale de tête mène au terme $\lambda x_1 \cdots x_n.b_1 \cdots b_p$. Alors, pour toute dérivation de a s'arrêtant sur une forme normale de tête $\lambda x_1 \cdots x_n.c_1 \cdots c_p$, on a $b_i \overset{\beta^*}{\rightarrow} c_i$.

<u>Preuve</u>
Soit D une dérivation de a sur une de ses formes normales de tête $\lambda x_1 \cdots x_n.c_1 \cdots c_p$, qui peut donc être supposée standard. Cette dérivation peut contenir des étapes internes et est donc de la forme suivante.

$$D \equiv a \overset{h}{\rightarrow} b \overset{i}{\rightarrow} \cdots \overset{i}{\rightarrow} \lambda x_1 \cdots x_n.c_1 \cdots c_p$$

Alors, b est lui-même une forme normale de tête $\lambda x_1 \cdots x_n.b_1 \cdots b_p$, sinon, comme D est standard, son radical de tête aurait un résidu dans c. Donc, $a \overset{h}{\rightarrow} b$ est la dérivation normale de tête de a et elle est finie. De plus, $b_i \overset{\beta^*}{\rightarrow} c_i$. •

Corollaire 7.11 *Si un terme a a une forme normale, alors sa dérivation normale termine sur sa forme normale.*

<u>Preuve</u>
On peut la faire par récurrence sur la structure du terme comme suit. Soit $\lambda x_1 \cdots x_n.c_1 \cdots c_p$ la forme normale du terme a. La dérivation normale DN débute par la dérivation normale de tête DH qui conduit à $\lambda x_1 \cdots x_n.b_1 \cdots b_p$. c_i est la forme normale de b_i donc la dérivation normale DN_i de b_i se termine sur c_i. Mais DN contient $DH; DN_1; \cdots; DN_p$. Donc, DN s'arrête, sur la forme normale de a. •

7.1 Dérivations équivalentes

Gràce au théorème des Développements Finis, on peut changer d'unité de calcul dans le λ-calcul en considérant comme étape élémentaire, non plus une β-réduction, mais un développement d'une famille de radicaux. On appellera une telle étape une *réduction parallèle* et une *dérivation paralléle* sera une suite de réductions parallèles. Notons qu'une étape de β-réduction est un cas particulier de réduction parallèle.

Deux dérivations sont coïnitiales si elles dérivent le même terme et cofinales si elles construisent le même réduit. La longueur d'une dérivation parallèle est son nombre de réductions parallèles. On note $R_1 + R_2$ la dérivation $R_1; R_2/R_1$ où R_i est une réduction parallèle. Le lemme de mouvements parallèles se reformule comme suit:

Lemme 7.12 (Parallel Moves) *Si R_1 et R_2 sont deux réductions parallèles coïnitiales, alors $R_1 + R_2$ et $R_2 + R_1$ sont cofinales.*

Preuve
Exercice de vocabulaire. •

Lemme 7.13 (Cube) *Si R_1, R_2, R_3 sont trois réductions parallèles coïnitiales, alors:*

1. *$R_1 + R_2$ et $R_2 + R_1$ sont cofinales.*
2. *$R_1 + R_3$ et $R_3 + R_1$ sont cofinales.*
3. *$R_2 + R_3$ et $R_3 + R_2$ sont cofinales.*
4. *$(R_2 + R_1) + R_3$, $(R_1 + R_2) + R_3$ sont cofinales ainsi que toutes celles qui s'en déduisent par permutation circulaire des indices.*

Preuve
Les trois premiers points définissent trois faces du cube. Le quatrième point dit qu'on peut fermer le cube. Faire le dessin et identifier chaque arête en tant que réduction parallèle d'une famille de radicaux à déterminer. • Le théorème de standardisation dit que toute dérivation peut être remplacée par une dérivation standard : ce qui caractérise une dérivation semble donc être plus la famille de radicaux qu'elle réduit que l'ordre même de ces réductions. Cette idée va être mise en place par la relation d'équivalence sur les dérivations, appelée *équivalence par permutation.*

Définition 7.14 *On appelle longueur d' une dérivation le nombre d'étapes de réductions parallèles qu'elle contient*

L'équivalence par permutation est la plus petite relation $=_P$ sur les dérivations telle que:

1. *$D; \emptyset =_P \emptyset; D =_P D$. On élimine les étapes vides.*
2. *$D_1 + D_2 =_P D_2 + D_1$ si D_i de longueur 1.*
3. *$D_1; D_2; D_3 =_P D_1; D_4; D_3$ si $D_2 =_P D_4$: $=_P$ est donc une congruence pour la composition de dérivations.*
4. *$=_P$ est transitive.*

Deux dérivations équivalentes par permutation sont donc coïnitiales et cofinales. La réciproque n'est pas vraie (cf. $\Delta\Delta$).

Proposition 7.15 *Le résidu d'une dérivation D_1 après une dérivation D_2, noté D_1/D_2 et le résidu de D_2 après D_1 peuvent être définis par le diagramme suivant:*

$$\begin{tikzcd} a \arrow[r, "D_1"] \arrow[d, "D_2"'] & b \arrow[d, dashed, "D_2/D_1"'] \\ c \arrow[r, dashed, "D_1/D_2"] & b \end{tikzcd}$$

On a $(D_1/D_2)/D_3 = D_1/(D_2; D_3)$.

Preuve
Le diagramme est construit en accolant des diagrammes fournis par une application du lemme des mouvements parallèles. •

Lemme 7.16 *1. Si on pose $D_1 + D_2 = D_1; (D_2/D_1)$, alors $D_1 + D_2 =_P D_2 + D_1$.*

2. $D_1 =_P D_2$ ssi $D_1/D_2 = \emptyset$ et $D_2/D_1 = \emptyset$. L'équivalence par permutation est donc décidable.

3. Si $D_1; D_2 =_P D_1; D_3$, alors $D_2 =_P D_3$.

4. $D_1 =_P D_2$ ssi, pour toute dérivation D_3, on a $D_3/D_1 =_P D_3/D_2$.

5. Si $D_1 =_P D_2$, alors pour toute dérivation D_3, on a $D_1/D_3 =_P D_2/D_3$.

<u>Preuve</u>

1. Il suffit d'utiliser la définition de l'équivalence par permutation dans le diagramme définissant les résidus des deux dérivations.

2. Si $D_1 =_P D_2$, on examine la dernière clause utilisée pour montrer que D_1 et D_2 sont équivalentes.
   Si $D_2 = D_1; \emptyset$ alors $D_2/D_1 = D_1/D_1; \emptyset/D_1 = \emptyset$. De même pour D_1/D_2.
   Si $D_1 = R_a + R_b$ et $D_2 = R_b + R_a$, faire le calcul en dessinant le cube.
   Si $D_1; D_2; D_3 =_P D_1; D_4; D_3$ parce que $D_2 =_P D_4$, utiliser l'hypothèse de récurrence.
   Le dernier cas se traite aussi par récurrence.

3. Par récurrence sur la preuve de $D_1; D_2 =_P D_1; D_3$.

4. Si $D_1 =_P D_2$ alors $D_3/D_1 = D_3/(D_1 + D_2)$ puisque $D_2/D_1 = \emptyset$. Donc, $D_3/D_1 = D_3/(D_2 + D_1) = D_3/D_2$.
   Dans l'autre sens, on déduit de l'hypothèse que $D_1/D_2 = D_2/D_2 = \emptyset$. De même, $D_2/D_1 = \emptyset$. Donc, $D_1 =_P D_2$.

•

Définition 7.17 *Soient deux dérivations D_1 et D_2 coïnitiales. D_1 est* contenue *dans D_2, noté $D_1 \subseteq D_2$ si il existe D_3 tel que $D_1; D_3 =_P D_2$.*

Proposition 7.18 *1. $D_1 \subseteq D_2$ ssi $D_1/D_2 = \emptyset$.*

2. $=_P$ est l'équivalence associée à $\subseteq$.

Théorème 7.19 *Si D_1 et D_2 sont deux dérivations coïnitiales, alors il existe une plus petite dérivation D telle que $D_1 \subseteq D$ et $D_2 \subseteq D$. De plus, $D = D_1 + D_2$.*

Pour toute dérivation D, il existe une seule dérivation standard D_s telle que $D =_P D_s$.

<u>Preuve</u>
Il est clair que $D_i \subseteq D_1 + D_2$. Soit D_3 un majorant commun de D_1 et D_2. On montre que $D_1 + D_2/D_3 = \emptyset$.

On montre ensuite que deux dérivations standard équivalentes par permutation sont identiques. En effet, soit R_i le premier radical réduit par D_i. Supposons R_1 plus petit que R_2 dans l'ordre standard. Alors, R_1/D_2 n'est pas vide : D_2 n'a réduit que des radicaux supérieurs à R_2 donc supérieurs strictement à R_1. Ceci contredit $D_1 =_P D_2$. Donc, $R_1 = R_2$. De proche en proche, on montre que les deux dérivations sont identiques. •

Ce théorème fournit une nouvelle preuve du théorème de Church-Rosser et une preuve du théorème de standardisation.

8 Arbres de Bohm

8.1 Résolubilité

Une sémantique S du λ-calcul est, par définition, une congruence sur Λ contenant la relation de β-interconvertibilité (éventuellement de η-interconvertibilité). Donc, si $a =_\beta b$ alors $a \; S \; b$ et, si $c \; S \; d$, alors pour tout contexte et toute occurrence u de ce contexte, on a $C[u \leftarrow c] \; S \; C[u \leftarrow d]$. Autrement dit, dans une telle sémantique, deux programmes β-interconvertibles sont équivalents et deux procédures sont équivalentes si, lorsqu'on les appelle dans un même contexte, on obtient des programmes équivalents. $=_\beta$ est clairement un exemple de sémantique et c'est même la plus petite sémantique. Mais, on peut souhaiter une sémantique plus directement liée au résultat de l'exécution d'un programme, i.e. à la forme normale du λ-terme si elle existe. Deux λ-termes ayant des formes normales différentes ne devraient pas être S-équivalents. Cela n'est d'ailleurs pas possible:

Théorème 8.1 (Théorème de Böhm (1968)) *Soient a et b deux λ-termes en forme normale non η-interconvertibles. Il existe alors deux variables x et y, un contexte C et une occurrence u dans ce contexte tel que $C[u \leftarrow a] \xrightarrow{\beta^*} x$ et $C[u \leftarrow b] \xrightarrow{\beta^*} y$.*

La preuve de ce théorème est donnée dans [**?**, **?**]. Elle repose sur la transformation de Böhm, qui s'appuie elle-même sur la notion d'arbre de Böhm, présentée plus loin. Donnons un exemple de construction d'un contexte séparant deux termes $a = \lambda x_1.x_1 z$ et $b = \lambda x_1.x_2 z$. Ces deux termes ne sont manifestement pas η-interconvertibles. Pour les séparer, il faut d'abord éliminer le λ de tête donc construire un contexe (Ωx_3) qui conduit aux termes $(x_3 z)$ et $(x_2 z)$. Reste à éliminer $x_3 z$ et $x_2 z$ et à faire apparaître x et y. Pour cela, on va remplacer x_3 (resp. x_2) par une fonction constante, dont l'application à z aura pour résultat x. Le contexte C cherché est donc: $\lambda x_2 x_3.(\Omega x_3)(\lambda u.x)(\lambda v.y)$.

De ce théorème, on déduit le résultat suivant:

Proposition 8.2 *Toute sémantique S telle qu'il existe deux termes a et b vérifiant les hypothèses du théorème de Böhm et $a \; S \; b$ est* incohérente, *c'est-à-dire identifie tous les λ-termes.*

<u>Preuve</u>
En effet, si $a \; S \; b$, par application du théorème de Böhm, on montre que $x \; S \; y$. Donc, pour tous termes c et d, en considérant le contexte $(\lambda xy.\Omega)cd$, on obtient $(\lambda xy.x)cd \; S \; (\lambda xy.y)cd$ et par suite $c \; S \; d$. •

Peut-on rendre équivalents tous les λ-termes n'ayant pas de forme normale? La réponse est non.

Proposition 8.3 *Toute sémantique S vérifiant $a \; S \; b$ dès que a et b n'ont pas de forme normale est incohérente.*

<u>Preuve</u>
Posons $\Delta = \lambda x.x\,x$, $K = \lambda xy.x$, $P = \lambda xyz.xy(\Delta\Delta)$ et $Q = \lambda xyz.xz(\Delta\Delta)$. P et Q n'ayant clairement pas de forme normale sont donc S-équivalents. Soit C le contexte ΩKab. $C[\Omega \leftarrow P] \xrightarrow{\beta^*} a$ et $C[\Omega \leftarrow Q] \xrightarrow{\beta^*} b$. Donc, pour tous termes a et b, $a \; S \; b$. •

Pour obtenir une sémantique raisonnable, il faut donc distinguer entre les diverses possibilités de non-terminaison. Pour cela, on va regarder le comportement de ces λ-termes dans un contexte : seront identifiées à l'indéfini les "procédures" qui, quel que soit le contexte les utilisant réellement, empêchent la terminaison du "programme" construit en les plaçant dans ce contexte.

Définition 8.4 Un λ-terme a est dit *totalement indéfini* si pour tout contexte C, pour toute occurrence u de Ω dans ce contexte, pour tout terme b, on a:
Si $C[\Omega \leftarrow a]$ a une forme normale, alors $C[\Omega \leftarrow b]$ a aussi une forme normale.

Un terme a est donc totalement indéfini si, dès qu'il existe un terme b tel que $C[\Omega \leftarrow b]$ n'ait pas de forme normale, alors $C[\Omega \leftarrow a]$ n'en a pas non plus. Par exemple, $\Delta\Delta$ est totalement indéfini mais P et Q ne le sont pas.

La sémantique que nous allons construire va identifier les termes totalement indéfinis. La première chose à faire est d'en donner une description plus maniable. On caractérisera les termes non totalement indéfinis par le fait qu'ils ont une forme normale de tête. Rappelons qu'une forme normale de tête (en abrége fnt) est un terme de la forme $\lambda x_1 \cdots x_n . x a_1 \cdots a_p$. Nous allons en étudier quelques propriétés. On a tout d'abord:

Proposition 8.5 *Une forme normale de tête n'est pas totalement indéfinie.*

Preuve
Soit la fnt $\lambda x_1 \cdots x_n . x a_1 \cdots a_p$. Si $x = x_i$, choisir le contexte $\Omega b_1 \cdots b_n$. Alors, $(\lambda x_1 \cdots x_n . x_i a_1 \cdots a_p) b_1 \cdots b_n \xrightarrow{*} b_i a_1 \cdots a_p$. Prendre alors $b_i = \lambda y_1 \cdots y_p . y$ avec y une variable libre. D'autre part, $(\Delta\Delta) b_1 \cdots b_n$ n'a pas de forme normale.

Si, pour tout i, $x \neq x_i$, prendre le contexte $\lambda x . \Omega b_1 \cdots b_n$. On réalise ainsi la capture de x et on se retrouve dans un cas voisin du précédent. •

Définition 8.6 Deux formes normales de tête $\lambda x_1 \cdots x_n . x a_1 \cdots a_p$ et $\lambda x_1 \cdots x_m . y a_1 \cdots a_q$ sont dites *semblables* si $x = y$, $n = m$ et $p = q$.

Proposition 8.7 *Un terme a possède une forme normale de tête si et seulement si sa dérivation de tête termine, sur un terme noté $h(a)$. Parmi toutes les formes normales de tête d'un terme a, s'il en possède, $h(a)$ est une forme minimale au sens suivant: si $a \xrightarrow{\beta^*} b$ et si b est en fnt, alors $h(a) \xrightarrow{\beta^*} b$.*

Preuve
Si la dérivation de tête termine, elle s'arrête sur une fnt $h(a)$. Soit b une fnt de a. Par le théorème de standardisation, il existe une dérivation standard de a vers b. Ou a est déjà en fnt et $h(a) \equiv a$, ou la dérivation standard commence par réduire le radical de tête, sinon, b ne serait pas une fnt. Cette dérivation standard de a se poursuit donc par la réduction du radical de tête (éventuellement) créé par la première étape et se prolonge répétitivement ainsi jusqu'à l'obtention d'une forme normale de tête minimale, qui est $h(a)$, car toute dérivation standard conduisant à une fnt aura à effectuer exactement les mêmes étapes. De plus, la dérivation de a sur $h(a)$ est la dérivation de tête de a, qui termine donc. •

Proposition 8.8 *1. a a une fnt ssi $\lambda x.a$ en a une.*

2. Si $a\{b/x\}$ a une fnt, alors a en a une.

3. Si ab a une fnt, alors a en a une.

Preuve
Le point 1. s'obtient en remarquant que tout réduit de $\lambda x.a$ est de la forme $\lambda x.b$.

Pour le point 2., on montre d'abord que si $a \stackrel{h}{\to} b$, alors $a\{c/x\} \stackrel{h}{\to} b\{c/x\}$. a contient un radical de tête donc est de la forme $\lambda x_1 \cdots x_n.((\lambda y.a_1)a_2) \cdots a_n$. Donc, $b = \lambda x_1 \cdots x_n.(a_1\{a_2/y\}) \cdots a_n$. On peut donc calculer $a\{c/x\}$ et $b\{c/x\}$ et en déduire le résultat cherché.
Supposons alors que a n'ait pas de fnt, alors sa dérivation de tête est infinie. On en déduit donc une dérivation de tête infinie pour $a\{b/x\}$.

Pour montrer le point 3, considérons la dérivation de tête de a. Soit elle contient un réduit de la forme $\lambda x.a_1$. Alors la dérivation de tête de ab contient une étape aboutissant à $(\lambda x.a_1)b$. Ce radical est nécesairement réduit à l'étape suivante. Donc, $a_1\{b/x\}$ a une fnt, donc a_1 et $\lambda x.a_1$ en ont aussi une. Soit la dérivation de tête de a ne comporte aucun réduit de la forme $\lambda x.a_1$. Alors, la dérivation de tête de ab ne contient que des réduits de la forme $a_i b$. Comme elle est finie, celle de a est aussi finie.

•

Définition 8.9

1. Un terme a est dit *résoluble* (ou solvable) si pour tout $b \in \mathbf{\Lambda_V}$, il existe des variables $x_1, \cdots, x_k$, des termes $u_1, \cdots, u_k$, $v_1, \cdots, u_j$ tels que $a\{u_1/x_1, \cdots, u_k/x_k\}\, v_1 \cdots v_j =_\beta b$.
2. Un terme a est *résoluble* si il existe des variables $x_1, \cdots, x_k$, des termes $u_1, \cdots, u_k$, $v_1, \cdots, u_j$ tels que $a\{u_1/x_1, \cdots, u_k/x_k\}\, v_1 \cdots v_j =_\beta I$.
3. Un terme a est *résoluble* si il existe des variables $x_1, \cdots, x_k$, des termes $u_1, \cdots, u_k$, $v_1, \cdots, u_j$ tels que $a\{u_1/x_1, \cdots, u_k/x_k\}\, v_1 \cdots v_j =_\beta x$, où $x \notin FV(a)$.
4. Un terme clos a est *résoluble* si on peut trouver un nombre fini de termes b_i tels que $a\, b_1 \cdots b_n =_\beta I$. Si a n'est pas clos, on demande que $\lambda x_1 \cdots x_n.a$, où les x_i sont les variables libres de a, soit résoluble.

Montrer que toutes ces définitions sont équivalentes.

Proposition 8.10 *a est résoluble si et seulement si a a une forme normale de tête.*

<u>Preuve</u>
Si a est résoluble, alors il existe des termes b_i tels que $a\, b_1 \cdots b_n = I$. Le terme $a\, b_1 \cdots b_n$ a une fnt donc a aussi. Si a a une fnt de la forme $\lambda x_1 \cdots x_n.x_i a_1 \cdots a_p$, le terme
$(\lambda x_1 \cdots x_n.x_i a_1 \cdots a_p) b_1 \cdots b_{i-1}((\lambda y_1 \cdots y_p.I) b_{i+1} \cdots b_n$
se réduit sur I. Sinon, il suffit de clore a pour se ramener au cas précédent. •

Nous avons vu comment représenter les fonctions récursives totales. Pour représenter les fonctions récursives partielles, il faut un moyen de représenter l'indéfini. Soit ϕ une fonction partielle récursive. On peut poser la définition suivante: ϕ est λ-définissable s'il existe un λ-terme F tel que $F\overline{n}_1 \cdots \overline{n}_p = \overline{m}$ si $\phi(n_1, \cdots, n_p) = m$ et $F\overline{n}_1 \cdots \overline{n}_p$ n'a pas de forme normale si $\phi(n_1, \cdots, n_p)$ est indéfini. Mais cette définition conduit à plusieurs difficultés, vu ce qui a été dit plus haut. La définition correcte consiste à traduire "non défini" par "n'est pas résoluble".

Définition 8.11 *Soit ϕ une fonction partielle récursive définie sur N^n à valeurs dans N. ϕ est λ-définissable s'il existe un terme F du λ-calcul tel que:*
si $\phi(n_1, \cdots, n_p) = m$, alors $F\overline{n}_1 \cdots \overline{n}_p = \overline{m}$
si $\phi(n_1, \cdots, n_p)$ est indéfini, alors $F\overline{n}_1 \cdots \overline{n}_p$ n'est pas résoluble.

Théorème 8.12 (Kleene) *Une fonction numérique partielle est récursive partielle si et seulement si elle est λ-définissable.*

La preuve de ce théorême suit le même plan que celle pour les fonctions totales. Mais, elle est techniquement plus compliquée. En particulier, le terme représentant la composition de deux fonctions ϕ et ψ ne peut pas toujours être défini par la composition des représentations de ces deux fonctions. Par exemple, la fonction f définie par $f(x) = 0$ est λ-définie par $F \equiv KZero$, où $Zero$ est un codage de 0. La fonction g partout non définie peut être représentée par $G \equiv K\Omega$. Cependant, la fonction $\lambda x.F(Gx) =_\beta F$ et n'est donc pas totalement indéfinie.

8.2 Arbres de Böhm

Intuitivement, connaître la forme normale d'un terme, c'est posséder l'information maximale sur ce terme. Disposer d'une forme normale de tête d'un terme fournit aussi une information, moins précise, sur ce terme. On va développer cette idée en introduisant la notion d'approximation d'un λ-terme. Pour cela, on introduit une nouvelle constante $\perp$ pour représenter l'indéfini. Un radical est une expression à calculer, donc sa valeur est $\perp$; si on applique une fonction indéfinie à un terme, le résultat est lui aussi indéfini, etc. D'où la définition suivante:

Définition 8.13 L'*approximation directe* d'un λ-terme a, notée $\perp(a)$, est obtenue en remplaçant tous les radicaux de a par $\perp$ puis en appliquant les règles ($\perp a \to \perp$) et ($\lambda x.\perp \to \perp$) jusqu'à obtenir une forme normale pour ces règles (et pour β). On notera $\perp(\Lambda)$ l'ensemble des approximations directes des λ-termes.

Remarque 8.1 Les approximations directes peuvent être caractérisées comme suit. Si un terme a n'est pas en fnt, alors $\perp(a) = \perp$ et si $a = \lambda x_1 \cdots x_n.xa_1 \cdots a_p$, alors $\perp(a) = \lambda x_1 \cdots x_n.x\perp(a_1) \cdots \perp(a_p)$. Ce fait justifie l'appellation " approximation directe". De plus, deux fnt semblables ont des approximations directes semblables.

Les approximations directes peuvent être représentées par des arbres d'arité variable : le préfixe de tête $\lambda x_1 \cdots x_n.x$ constitue la racine, les fils sont les représentations de $\perp(a_i)$.

Exercice 8.1 *Construire la représentation de Δ, de Ω, de Y et de ses dérivés.*

On peut maintenant introduire la notion : "un terme a est moins défini qu'un terme b" en se plaçant dans l'ensemble $\Lambda_\perp$ des λ-expressions pouvant contenir $\perp$.

Définition 8.14 Soient a et $b \in \Lambda_\perp$. La relation a est un *préfixe* de b, notée $a \preceq b$, est définie par :

1. $\forall a, \perp \preceq a$.
2. $x \preceq x$.
3. si $a \preceq b$, alors $\lambda x.a \preceq \lambda x.b$.
4. Si $a_i \preceq b_i$ pour $i = 1, 2$, alors $a_1a_2 \preceq b_1b_2$.

Cet ordre induit un ordre sur $\perp(\Lambda)$: si $\perp(a) \preceq \perp(b)$, alors a sera dit *moins défini* que b.

Proposition 8.15 *L'ordre $\preceq$ est un ordre bien fondé, possédant un élément minimum $\perp$. Si a et $b \in \perp(\Lambda_\perp)$, alors a et b ont une borne inférieure et, s'ils ont un majorant commun, alors ils ont une borne supérieure.*

Preuve
Notez que deux variables différentes ne sont pas comparables. Donc, si $x \neq y$, $a = \lambda x_1 \cdots x_n.xa_1 \cdots a_p$ et $b = \lambda x_1 \cdots x_n.yb_1 \cdots b_p$, ne sont pas comparables. Alors, $inf(a, b) = \perp$.

On pose, si $a = \lambda x_1 \cdots x_n.xa_1 \cdots a_p$ et $b = \lambda x_1 \cdots x_n.xb_1 \cdots b_p$, alors $inf(a, b) = \lambda x_1 \cdots x_n.xinf(a_1, b_1) \cdots inf(a$
On montre que cette définition a un sens par récurrence sur la taille des termes.

De même, on pose, si $a = \lambda x_1 \cdots x_n.xa_1 \cdots a_p$ et $b = \lambda x_1 \cdots x_n.xb_1 \cdots b_p$, alors $sup(a, b) = \lambda x_1 \cdots x_n.xsup(a_1, b_1) \cdots sup(a_n, b_n)$ et $sup(a, \perp) = a$ sinon. On montre de même que cette définition est correcte, en notant, que si a et b sont deux fnt et ont un majorant commun, alors ils sont semblables. • Noter que x et y ne sont pas comparables donc $\lambda \overrightarrow{x}_i.x\overrightarrow{a}_i$ et $\lambda \overrightarrow{x}_i.y\overrightarrow{a}_i$ ne sont pas comparables.

Proposition 8.16 *Si $a \xrightarrow{*} b$, alors a est moins défini que b.*

Preuve
Ou a n'est pas en fnt et $\perp(a) = \perp$ ou a donc b sont en fnt et on conclut par récurrence sur la taille des termes. •

Définition 8.17 *L'ensemble des approximations d'un terme a est l'ensemble $A(a)$ défini par $A(a) = \{\perp(b) | a \xrightarrow{*} b\}$.*

Proposition 8.18 *$\perp(a)$ est l'élément minimum de $A(a)$.*
Si c et $d \in A(a)$, alors $inf(c, d)$ et $sup(c, d) \in A(a)$.

Preuve
la preuve se fait par récurrence sur la taille des termes.
Par définition de $A(a)$, il existe a_1 et a_2 tels que $c = \perp(a_1)$, $d = \perp(a_2)$ et $a \xrightarrow{*} a_i$. Si a_1 par exemple n'est pas en fnt, alors $c = \perp$ et $inf(c, d) = \perp$. De plus, a n'est pas en fnt. Donc, $\perp = \perp(a) \in A(a)$. Si a_1 et a_2 sont en fnt, alors en standardisant les dérivations, on sait que $a \xrightarrow{*} h(a) \xrightarrow{*} a_i$, où $h(a)$ représente la forme normale de tête principale de a, i.e. celle qui est obtenue par LA dérivation normale de tête. Donc, si $h(a) = \lambda x_1 \cdots x_n.xc_1 \cdots c_p$, alors $a_i = \lambda x_1 \cdots x_n.xd_1^i \cdots d_p^i$ et $c_k \xrightarrow{*} d_k^i$. Donc, $\perp(d_k^1)$ et $\perp(d_k^2)$ ont un inf d_k. On en déduit que $\lambda x_1 \cdots x_n.xd_1 \cdots d_p$ est l'inf de c et de d.

Pour la borne supérieure, on fait le même raisonnement en constatant que c et d ont au moins un majorant commun grâce au théorème de Church-Rosser. •

Remarquez que $A(a)$ n'a pas toujours un élément maximal. S'il n'a pas d'élément maximal, alors a n'a pas de forme normale. La réciproque est fausse : si a n'a pas de fnt, alors $A(a) = \perp$.

Les ensembles $A(a)$ ont donc une structure de treillis et sont finis ou infinis. On va d'abord les *clore par le bas*:

Définition 8.19 L'ensemble des approximations généralisées d'un terme a, noté $BT(a)$, est l'ensemble défini par $\{c | c \preceq b \text{ et } b \in A(a)\}$. Cet ensemble est encore appelé *arbre de Böhm du terme a.*

Soit S une partie dirigée de $\perp(\Lambda)$. L'idéal (dirigé) engendré par S est l'ensemble $\{c | c \preceq b \text{ et } b \in S\}$.

On note $BT = \{I | I \subseteq \perp(\Lambda) \text{ et } I \text{ idéal}\}$. On munit BT de l'ordre d'inclusion.

Si le terme a n'a pas de forme normale mais possède une forme normale de tête, son arbre de Böhm est représenté par un arbre infini, qui est la "limite" des arbres représentant les approximations de a. Plus précisément, l'ensemble $BT(a)$ est un idéal($\perp \in BT(a)$ et si $a, b \in BT(a)$, alors il existe $c \in BT(a)$ tel que $a \preceq c$ et $b \preceq c$) et l'ensemble des arbres de Böhm est le complété par idéaux de $\perp(\Lambda)$.

Proposition 8.20 BT vérifie les propriétés suivantes:

1. BT possède un élément minimal $\{\perp\}$.

2. Si I et $J \in BT$, alors $I \cap J = inf(I, J)$.

3. Si I et $J \in BT$ ont un majorant commun, alors :

$$I \sqcup J = \{sup(a, b) | a \in I \text{ et } b \in J\} = sup(I, J)$$

4. BT est complet : toute partie dirigée S de BT admet un sup, qui est $\cup S$, la réunion des éléments de S.

5. $\perp(\Lambda)$ est isomorphe à l'ensemble de ses idéaux principaux $\{a\}$, c'est-à-dire, ceux qui sont engendrés par un seul terme.

Définition 8.21 On note $\preceq$ le préordre défini sur $\mathbf{\Lambda_V}$ par $a \preceq b$ ssi $BT(a) \subseteq BT(b)$. On note $=_{BT}$ l'équivalence associée.

Remarque 8.2 1. $\forall a \in \mathbf{\Lambda_V}, \Omega \preceq a$.

2. Si a possède une f.n.t., si $a \preceq b$, alors b a aussi une f.n.t, semblable à celles de a.

3. Si $a \to^{\beta *} b$, alors $a =_{BT} b$. La réciproque est fausse.

Théorème 8.22 *(Théorème de Continuité) La relation $a \preceq b$ définie par $a \preceq b$ ssi $BT(a) \subseteq BT(b)$ est une relation monotone (si $a \overset{*}{\to} b$, alors $a \equiv b$), stable par passage au contexte, continue par rapport aux approximations ($\perp \preceq a$ et $a =_{BT} sup(c | c \in (BT(a)))$. Cette relation définit donc une sémantique du λ-calcul, qui est la plus petite sémantique continue par approximations.*

Nous avons donc, par des considérations essentiellement syntaxiques, construit un modèle du λ-calcul. Cette construction peut être étendue en considérant la relation de $\beta\eta$-interconvertibilité, de plusieurs manières, qui permettent de retrouver les modèles classiques du λ-calcul.

9 Logique combinatoire

*Ces notes de cours sont établies à partir des ouvrages de Barendregt[**?**], Curien[**?**], Hindley[**?**]*

La Logique Combinatoire a d'abord été introduite par Schöchnfinkel vers 1920 puis redécouverte par Curry. Il s'agit d'une théorie équationnelle du premier ordre, qui sert à décrire la fonctionnalité : on définit un ensemble d'opérateurs primitifs et les moyens de les composer. Il s'agit donc d'une théorie de la fonctionnalité où la notion de variable liée n'existe pas. Elle met l'accent sur le procédé opérationnel du calcul : on combine des éléments atomiques (S, K et I) pour décrire une fonction.

Définition 9.1 *La Logique Combinatoire, notée CL, est définie par la donnée d'un ensemble de variables V, d'un opérateur d'arité 2, noté par juxtaposition, et par trois constantes S, K et I. La théorie est engendrée par deux équations :*

$$K\,x\,y \to x \quad S\,x\,y\,z \to (x\,z)(y\,z) \quad I\,x \to x$$

On peut aussi ne prendre que les constantes S et K et définir I par $I = S\,K\,K$.

Appelons *remplacement* la substitution du premier ordre et notons $a[b/x]$ le remplacement de b aux occurrences de x dans a.

On démontre sans difficulté, en utilisant simplement les résultats généraux sur la réécriture du premier ordre, les lemmes suivants:

Lemme 9.2 *Si $M \overset{\star}{\to} N$, alors :*

1. $M[a_1/x_1, \ldots, a_n/x_n] \overset{\star}{\to} N[a_1/x_1, \ldots, a_n/x_n]$.
2. $P[M/x] \overset{\star}{\to} P[N/x]$

Théorème 9.3 *CL vérifie la propriété de Church-Rosser.*

En effet, ce système est orthogonal (i.e. linéaire gauche, sans paire critique) et donc confluent. On rappelle que la confluence d'un système orthogonal se démontre en prouvant que la réduction d'une famille de radicaux disjoints est fortement confluente.

9.1 Algorithme d'abstraction

Pour établir le fait que CL est bien une théorie de la fonctionnalité, on va définir un algorithme d'abstraction d'une variable x dans une terme M. Le résultat de cet algorthime est un terme de CL, noté $[x]M$.

Définition 9.4 *Soit $x \in V$. L'abstraction de x dans $M \in CL$ est définie par :*

1. $[x]x = I$
2. $[x]M = K\,M$ *si* $x \notin Var(M)$
3. $[x](M\,N) = S([x]M)([x]N)$

Théorème 9.5

$$([x]M)\,N \overset{\star}{\to} M[N/x]$$

Grâce aux propriétés de la réécriture du premier ordre, il suffit de montrer que :

$$([x]M)x \overset{\star}{\to} M$$

Il suffit de le montrer pour une étape de réduction, ce qui se fait par cas sur la construction de $[x]M$.

On en déduit la version suivante du théorème de complétude combinatoire:

Théorème 9.6 *Si $\forall i,j,\ x_i \notin Var(N_j)$, alors:*

$$([x_1, x_2, \ldots, x_n]M)\,N_1\,N_2\,\ldots\,N_n \overset{\star}{\to} M[N_1/x_1, \ldots, N_n/x_n]$$

Théorème 9.7 *Si $x \notin Var(N)$,*

$$([x]M)[N/y] =_{CL} [x](M[N/y])$$

La preuve se fait par cas, sur la construction de $[x]M$.

9.2 intertraduction entre $\mathbf{\Lambda_V}$ et CL

Définition 9.8 *On définit une traduction* $(\_)_{CL}$ *de* $\mathbf{\Lambda_V}$ *vers* CL *par:*

1. $(x)_{CL} = x$
2. $(MN)_{CL} = (M)_{CL}(N)_{CL}$
3. $(\lambda x.M)_{CL} = [x](M)_{CL}$

et une traduction $(\_)_\lambda$ *de* CL *vers* $\mathbf{\Lambda_V}$ *par:*

1. $K_\lambda = \lambda xy.x$
2. $S_\lambda = \lambda xyz.(x\,z)(y\,z)$
3. $(M\,N)_\lambda = M_\lambda\,N_\lambda$

Le but est de montrer que ces deux traductions définissent un isomorphisme entre les théories $\mathbf{\Lambda}$ et CL. Le résultat sera vrai à condition d'ajouter un certain nombre d'axiomes.

Théorème 9.9 *On peut étendre la théorie* CL *par une famille d'axiomes, les axiomes de Curry (donnés plus loin), tels que :*

1. $M =_\beta N \;\Rightarrow\; (M)_{CL} =_{CL} (N)_{CL}$
2. $A =_{CL} B \;\Rightarrow\; A_\lambda =_\beta B_\lambda$
3. $(A)_{\lambda,CL} =_{CL} A$
4. $(M)_{CL,\lambda} =_\beta M$

Ces résultats s'étendent à l'égalité $=_{\beta\eta}$ *en ajoutant un axiome d'extensionalité.*

Le point 1. s'obtient facilement si $M = (\lambda x.P)Q$ et $N = P\{Q/x\}$. Pour obtenir le résultat pour tout M, il suffit de montrer que cette propriété reste vraie en passant au contexte. Cela nécessite de disposer de l'étape de déduction suivante:

$$A = B \Rightarrow [x]A =_{CL} [x]B \quad (\xi)$$

Cette étape ne peut pas être une conséquence des axiomes de CL déjà introduits. En effet:

$$[x]S\,x\,y\,z = S([x]S\,x\,y)([x]z) = S(S([x]S\,x)([x]y))([x]z) = S(S(S(K\,S)I)(K\,y))(K\,z)$$

terme qui est en forme normale, à comparer avec $[x](x\,z)(y\,z)$.

Supposons savoir obtenir cette étape ξ. Pour le point 3., il suffit de montrer le point 3. pour S et pour K. Or,

$$(K)_{\lambda,CL} = (\lambda xy.x)_{CL} = [x](\lambda y.x)_{CL} = [x][y](x)_{CL}$$

Or, $K\,x\,y =_{CL} x$, si ξ est obtenue, on en déduit que

$$[xy].x =_{CL} [xy].K\,x\,y$$

De la même manière, on montre que:

$$(S)_{\lambda,CL} = S\,x\,y\,z$$

Posons $1_n = [ux_1 \dots x_n].u\,x_1 \dots\, x_n$ et introduisons les axiomes :

$$1_2 K = K \quad (K\lambda);\;\; 1_3 S = S \quad (S\lambda)$$

Comme

$$1_2 K = ([u]([x\,y].u\,x\,y))\,K = [x\,y]\,K\,x\,y$$

On en déduit que $(K)_{\lambda,CL} = K$. De même, $(S)_{\lambda,CL} = S$. De plus,

$$K\,M = ([u\,x\,y].u\,x\,y)K\,M = [y]K\,M\,y = ([y\,t]ty)(K\,M)$$

$$S\,M\,N = ([u\,x\,y\,z].u\,x\,y\,z)S\,M\,N = [z]S\,M\,N\,z = ([z\,t]tz)(S\,M\,N)$$

Or, tout terme obtenu par l'algorithme d'abstraction est de la forme $K\,M$ ou $S\,M\,N$ (car $I = S\,K\,K$). On en déduit donc que :

$$1_1([x]A) = [x]A$$

Théorème 9.10 *Si on ajoute à CL les axiomes $K\lambda$, $S\lambda$ et l'axiome*

$$M\,x = N\,x \Rightarrow 1_1\,M = 1_1\,N$$

alors, la théorie obtenue permet d'obtenir le théorême **??** *pour la théorie β.*

Cette réponse n'est pas entièrement satisfaisante car la théorie ci-dessus n'est pas une théorie équationnelle. Pour obtenir la règle ξ comme conséquence équationnelle, il faut introduire trois axiomes supplémentaires :

$$[x\,y]K\,(x\,y) = [x\,y]S\,(K\,x)(K\,y) \quad (abs)$$

$$[x\,y]S(S(K\,K)x)y = [x\,y]x \quad (Kx)$$

$$[x\,y\,z]S(S(S(K\,S)x)y)z = [x\,y\,z]S(S\,x\,z)(S\,y\,z) \quad (Sx)$$

Pour obtenir le théorême **??** pour la théorie $\beta\eta$, il faut encore ajouter un autre axiome:

$$[x]S\,(K\,x)I = [x].x$$

Nous n'avons pas expliqué la genèse de ces axiomes dont la formulation est assez compliquée. Nous n'avons pas non plus fait la preuve du théorème **??**. Ces notes ne donnent donc qu'un très petit aperçu de la Logique Combinatoire. La conclusion de ce survol peut être la suivante : on peut arriver à coder le λ-calcul sans utiliser de noms de variable avec les combinateurs S et K, mais, pour retrouver la puissance de calcul de $\mathbf{\Lambda}$, il faut ajouter des axiomes dont la formulation n'est pas simple. En fait, la théorie CL avec ses axiomes S et K correspond bien à la β-réduction faible, c'est-à-dire sans la règle ξ.

La lecture des ouvrages cités en référence est vivement conseillée.

10 Les λ-calculs avec couples

On a montré comment coder les notions de couple et de projection dans le λ-calcul.

Soient M et $N \in \mathbf{\Lambda_V}$. Soient les termes :

$$C \equiv \lambda f\, s\, d\, .\, d\, f\, s \qquad C_1 \equiv \lambda\, c.c\, (\lambda\, x\, y.x) \qquad C_2 \equiv \lambda\, c.c\, (\lambda\, x\, y.y)$$

On a alors :

$$C_1(C\, M\, N) \xrightarrow{\beta^\star} M \qquad C_2(C\, M\, N) \xrightarrow{\beta^\star} N$$

projection.

Ayant choisi ces termes C, *Fst* et *Snd*, a-t-on réellement ajouté un *opérateur* de construction de couples au λ-calcul? Autrement dit, cette construction est-elle canonique? Est-il possible de prouver l'unicité, modulo β-réduction, des couples bâtis à l'aide d'un terme C fixé?

La réponse est négative : quels que soient les termes utilisés pour représenter le couple et les projections, il existe des termes M tels que :

$$C(\mathit{Fst}\ M)\ (\mathit{Snd}\ M) \not\xrightarrow{\beta^\star} M$$

LA β-réduction ne permet donc pas d'affirmer l'unicité de la définition d'un couple. De plus, Barendregt [**?**] a montré que s'il existait un triplet de termes du λ-calcul (C, C_1, C_2) tel que l'égalité (SP) :

$$C(C_1 M)\ (C_2 M) =_{\beta\eta} M$$

soit vraie pour tout terme M, alors le λ-calcul serait une théorie incohérente. On obtient le même résultat pour la Logique Combinatoire. La démonstration repose sur le fait suivant :
Si on a l'égalité (SP) , alors :

$$C(C_1 x)\ (C_2 x) =_{\beta\eta} x$$

donc :

$$C(C_1 x)\ (C_2 x) \ \longrightarrow^{\beta\eta^\star}\ x$$

Soit $\Omega \equiv (\lambda x.x\, x)(\lambda x.x\, x)$. Alors,

$$C(C_1 \Omega)\ (C_2 \Omega) \ \longrightarrow^{\beta\eta^\star}\ \Omega$$

On montre alors, en les marquant, que nécessairement l'une des occurrences de Ω dans le membre gauche disparaît dans cette dérivation. On remplace ensuite cette occurrence par une nouvelle variable z obtenant ainsi :

$$C(C_1 z)\ (C_2 \Omega) \ \longrightarrow^{\beta\eta^\star}\ \Omega$$

D'où, en composant avec C_1 à gauche puis en réduisant :

$$C_1 z \ =_{\beta\eta}\ C_1 \Omega$$

Remplaçons alors z par $C\, t\, s$ puis par $C\, s\, t$. On obtient :

$$\forall s, t, \quad s =_{\beta\eta} C_1 \Omega =_{\beta\eta} t$$

Par contre, la théorie obtenue en ajoutant explicitement trois constantes et les règles de projection ainsi que la règle (SP) reste cohérente. Nous allons donc l'étudier.

10.0.1 le λc-calcul applicatif

Définition 10.1 Le λc- calcul applicatif, noté $\mathbf{\Lambda_{c,a}}$ est obtenu en ajoutant au λ-calcul pur les constantes D, F, S et les règles :

$$\begin{array}{lrcl} \text{(Fst)} & F\,x\,y & \longrightarrow & x \\ \text{(Snd)} & S\,x\,y & \longrightarrow & y \\ \text{(SP)} & D\,(\,Fx)(\,Sx) & \longrightarrow & x \end{array}$$

définissant ainsi, avec la règle β (respectivement $\beta\eta$), la théorie (βSP) (resp. ($\beta\eta SP$)). Si on enlève la règle (SP), la théorie obtenue s'appelle βP.

Théorème 10.1 $\mathbf{\Lambda_{c,a}}$ *, muni de la théorie βP, vérifie la propriété de Church-Rosser.*
$\mathbf{\Lambda_{c,a}}$ *muni de la théorie βSP, est localement confluent mais ne vérifie pas la propriété de Church-Rosser.*

Démonstration
La démonstration du premier résultat ne présente pas de difficultés. Le second résultat a été obtenu par J. W. Klop [**?**] en 1980 . Mann avait posé le problème en 1972. R. Hindley , J. Staples avaient fourni des versions simplifiées de cette conjecture en 1974-1975 . Elle fut également soulevée par G. Huet et J.J. Levy à propos de l'opération de branchement dans les Schémas de Programmes Récursifs. J.W. Klop a montré qu'aucun de ces systèmes n'est confluent. ■

Donnons d'abord la construction du contre-exemple de J. W. Klop. Celui-ci montre qu'il existe deux dérivés d'un même terme qui ne peuvent être égaux.

Notations
Soit $P = \lambda\, x\lambda\, y.y\,(\,(xx)\,y)$. Soit $Y_T = P\,P$ le point fixe de Turing.

Lemme 10.2 *Soit $M \in \mathbf{\Lambda_{c,a}}$. Alors :*

$$Y_T\,M \xrightarrow{\beta^\star} M\,(Y_T\,M)$$

Proposition 10.3 Le contre-exemple de Klop
Soit E une variable libre. Posons :

$$V \equiv \lambda\, x\lambda\, y.E(\,D(F\,y)(S\,xy)) \qquad C = Y_T\,V \qquad B = Y_T\,C$$

Alors, le terme B admet deux dérivés $E(C\,B)$ et $C(E(C\,B))$ qui n'ont aucun dérivé commun.

Démonstration
D'après le lemme **??**, on a :

$$C \xrightarrow{\beta^\star} V\,C \xrightarrow{\beta^\star} \lambda y.E(\,D(F\,y)(S\,C\,y))$$

$$B \xrightarrow{\beta^\star} C\,B \xrightarrow{\beta^\star} E(\,D(F\,B)(S\,CB))$$

Donc :

$$B \xrightarrow{\beta^\star} E(\,D(F\,(C\,B))(S\,(C\,B)))$$

En utilisant la règle (SP), on obtient :

$$B \xrightarrow{\beta^\star} C\,B \xrightarrow{\beta^\star} E(C\,B)$$

On peut aussi construire la dérivation suivante :

$$B \xrightarrow{\beta^\star} C\,B \xrightarrow{\beta^\star} C(\,E(\,C\,B))$$

Notons (SPE) une (SP)-réduction ainsi instanciée :

$$E\,(\,D\,(F\,x)(S\,x)) \longrightarrow E\,x$$

J. W. Klop montre alors que, dans toute dérivation de CB, on peut faire commuter les étapes de β-réduction et les étapes (SPE). Toute conversion entre $E\,(C\,B)$ et $C(\,E(CB))$ peut donc être factorisée ainsi :

$$E\,(C\,B) \xrightarrow{\beta^\star} \circ\ (SPE)^\star \circ (\,(SPE)^\star)^{-1} \circ \xleftarrow{\beta^\star} C(\,E(CB))$$

Les deux β-dérivations exhibées ci-dessus peuvent être remplacées par des dérivations standard. Or, on montre que toute dérivation standard de $C(\,E(CB))$ conduit au terme $D\,(E\,(C\,B))\,(C(\,E(CB))$. Donc, toute conversion n'utilisant que des étapes standard contient une conversion utilisant moins de symboles et pouvant aussi être standardisée. D'où le résultat. ■

Un autre Contre-Exemple

Le contre-exemple que nous proposons est bâti sur la constatation suivante : la réduction d'un (SP)-radical peut créer un β-radical, qui, une fois réduit, peut faire disparaître toute l'information contenue dans le (SP)-radical. Il ne nécessite pas le théorème de Standardisation. De plus, sa démonstration consiste à montrer qu'un terme $C\,I$ ne peut avoir pour forme normale le terme $\lambda x.x$, en explorant toutes les dérivations possibles de $C\,I$.

Notations
Soit $U = \lambda x\lambda y.D(F(\lambda z.z(xy))(S(\lambda z.zy))(\lambda z.I)$ où $I = \lambda x.x$.
Soit $C = Y_T\,U$ et $B = Y_T\,C$

Lemme 10.4 *Le terme B admet les termes I et $C\,I$ pour dérivés.*

Démonstration
D'après le lemme **??**, on sait que :

$$C \xrightarrow{\beta^\star} UC \qquad B \xrightarrow{\beta^\star} C\,B$$

Donc, quel que soit le terme M, on a :

$$CM \xrightarrow{\beta^\star} D(F(\lambda z.z\,(C\,M)))(S(\lambda z.z\,M))(\lambda z.I)$$

On en déduit la dérivation D_1 de B :

B
$\xrightarrow{\beta^\star}$
$C\,B$
$\xrightarrow{\beta^\star}$
$D(F(\lambda z.z\,(C\,B)))(S(\lambda z.z\,B))(\lambda z.I)$
$\xrightarrow{\beta^\star}$
$D(F(\lambda z.z\,(C\,B)))(S(\lambda z.z\,(C\,B)))(\lambda z.I)$

$\stackrel{(SP)}{\longrightarrow}$
$(\lambda z.z\ (C\ B))(\lambda z.I)$
$\stackrel{\beta}{\longrightarrow}$
$(\lambda z.I)(C\ B)$
$\stackrel{\beta}{\longrightarrow}$
I
Construisons alors la dérivation D_2 de B :

B
$\stackrel{\beta^\star}{\longrightarrow}$
$C\ (C\ B)$
$\stackrel{D_1}{\longrightarrow}$
$C\ I$ ■

Lemme 10.5 *Soit $M \in \mathbf{\Lambda_{c,a}}$, admettant une forme normale N distincte de I.*
Si βSP (resp. $\beta\eta SP$) vérifie la propriété d'unicité des formes normales, alors $C\ M$ et M ne peuvent avoir de dérivé commun.

Démonstration
Le terme

$$X \equiv D\ (F\ (\lambda z.z(C\ M))\ )\ (S\ (\lambda z.zM)\ )\ (\lambda z.I)$$

est un dérivé de $C\ M$. Soit A un dérivé commun à M et $C\ M$. X peut être réécrit sur :

$$D(F\ (\lambda z.zA\ )\ )(S\ (\lambda z.zA\ )\ )\ (\lambda z.I)$$

puis sur I. D'où le résultat. ■

Lemme 10.6 Si $M \stackrel{(\beta SP)^\star}{\longrightarrow} N$, alors :

$$M\ [y \longleftarrow\ U\ ] \stackrel{(\beta SP)^\star}{\longrightarrow} N\ [y \longleftarrow\ U\ ]$$

Démonstration
Elle est évidente.
On a le même résultat pour $\beta\eta SP$.

Proposition 10.7 *I ne peut pas être un β-dérivé (resp. un $\beta\eta$-dérivé) de $C\ I$.*

Démonstration
Faisons la pour les β-dérivations. Elle est identique pour les $\beta\eta$-dérivations.
Nous allons examiner toutes les dérivations possibles de $C\ I$ vers son éventuelle forme normale et montrer qu'aucune ne peut aboutir à I. On appellera R une telle dérivation. $C\,I$ ne contient qu'un seul radical : celui de Y_T. Sa réduction conduit au terme A_1 :

$$A_1 = (\ (\ \lambda y.y\ (Y_T\ y)\ )\ U\ )\ I$$

R peut se prolonger en dérivant le sous-terme $Y_T\,y$. Soit $Red(Y_T\ y)$ le sous-terme ainsi obtenu. R doit nécessairement réduire le radical le plus externe. R contient donc un terme A_2 :

$$A_2 = U(Red(Y_T\ y)[y \longleftarrow\ U]\ )\ I = U(Red(Y_T U)\ )\ I = U(Red(C))\ I$$

R peut se prolonger en dérivant le sous-terme $Red(C)$. Elle doit nécessairement réduire le radical le plus externe. R contient donc un terme A_3 :

$$A_3 \equiv (\lambda y.(D\ (F\ (\lambda z.z(Red\,(C)y)))\ (S\ (\lambda z.zy))\ )\ (\lambda z.I)\ )\ I$$

A_3 contient un sous-terme $D(F..)(S..)$. Si (βSP) est confluente, alors elle vérifie la propriété d'unicité des formes normales et, par conséquent, d'après le lemme **??**, Cy ne peut se dériver sur y. Donc, ce contexte ne peut pas disparaître avant la réduction du radical le plus externe. R contient donc un terme A_4 :

$$A_4 \equiv D(\ (F\ (\lambda z.zRed(CI))\ )\ (S\ (\lambda z.zI)\ ))\ (\lambda z.I)$$

Appelons longueur d'une dérivation le nombre de (SP)-étapes qu'elle contient. Soit $Rmin$ une dérivation de CI sur I de longueur minimale. $Rmin$ doit faire disparaître le contexte $DF(..)S(..)(\lambda z.I)$ Elle contient donc une dérivation de CI sur I et n'est donc pas de longueur minimale. Il n'existe donc pas de dérivation de CI sur I. D'où la contradiction. ■

10.0.2 Le λc-calcul fonctionnel

Définition 10.8 Le λc- calcul fonctionnel, noté $\mathbf{\Lambda_{c,f}}$ est obtenu en ajoutant au λ-calcul pur un opérateur binaire noté $<,>$, deux opérateurs unaires notés fst et snd ainsi que les règles :

$$\begin{array}{lrcl} \text{(Fst)} & fst\,(<\ x,\ y\ >) & \longrightarrow & x \\ \text{(Snd)} & snd\,(<\ x,\ y\ >) & \longrightarrow & y \\ \text{(SP)} & <\ fst\,(x)\ ,\ snd\,(x)\ > & \longrightarrow & x \end{array}$$

définissant ainsi, avec la règle β, une théorie encore notée (βSP). Si on enlève la règle (SP), la théorie obtenue s'appelle encore βP.

Théorème 10.2 $\mathbf{\Lambda_{c,f}}$*, muni de la théorie* βP*, vérifie la propriété de Church-Rosser.*
$\mathbf{\Lambda_{c,f}}$*, muni de la théorie* (βSP)*, ne vérifie pas la propriété de Church-Rosser.*

Démonstration
Elle est obtenue à partir de la précédente en remplaçant le terme U du cas applicatif par le terme W :

$$W \equiv \lambda x\ \lambda y(<\ fst\,(\lambda z.\ z(xy)\,)\ ,\ snd\,(\ \lambda z.zy\ )\ >\ (\lambda z.\ I)\ )$$

■

Remarque 10.1 Pour le λ-calcul, quelle que soit la méthode d'extension choisie, la propriété de Church-Rosser n'est pas conservée. Il n'en est pas de même dans le cas de la Logique Combinatoire. Etendons la Logique Combinatoire, CL, avec trois constantes D, F, S et les règles (Fst), (Snd) et (SP). Soit CLa la théorie ainsi obtenue. J. W. Klop a montré que CLa ne vérifie pas la propriété de Church-Rosser.

Etendons alors CL en ajoutant les opérateurs $<,>$, $Fst\,()$ et $Snd\,()$. Cette fois, la théorie obtenue CLf vérifie la propriété de Church-Rosser. Ce résultat est une conséquence du théorème de Y. Toyama sur les sommes disjointes de systèmes confluents [?]. En fait, on peut considérer que $\mathbf{\Lambda_{c,a}}$ comme CLa contiennent la curryfication de l'opération de couple : Dx est en quelque sorte le curryfié de $<x,->$. CLa contient donc des termes fondamentalement différents de ceux de CLf. Ces différences peuvent donc expliquer les propriétés différentes de ces théories.

11 La notation de de Bruijn

Pour éviter le problème d'α-conversion tout en conservant la structure des termes du λ-calcul, le plus simple consiste à supprimer les noms des variables liées. Pour effectuer correctement la substitution, il suffit de connaitre les occurrences du terme liées par l'abstracteur disparaissant dans la β-réduction. Dans $\mathbf{\Lambda_V}$ ce lien est indiqué par l'identité nom de la variable à l'occurrence examinée, nom de la variable de l'abstracteur. Ce lien peut tout aussi bien être indiqué par la *hauteur de liaison* de l'occurrence : le nombre de λ à franchir pour remonter de l'occurrence examinée jusqu'à l'abstracteur qui la lie. Si on décide de supprimer également le nom des variables libres $\{x_0, \ldots, x_n\}$ d'un terme a, il suffit de considérer a comme sous-terme de $\lambda x_0 \ldots x_n.a$. Cette idée a été développée par N.G. de Bruijn dans le cadre du projet Automath à l'Université d'Eindohven. [**?**].

Définition 11.1 $\mathbf{\Lambda}$, l'ensemble des termes du λ-calcul, décrit avec la notation de de Bruijn, est le plus petit ensemble de termes vérifiant :

1. Si $n \geq 1$, alors $n \in \mathbf{\Lambda}$
2. Si a et $b \in \mathbf{\Lambda}$, alors $a\, b \in \mathbf{\Lambda}$
3. Si $a \in \mathbf{\Lambda}$, alors $\lambda(a) \in \mathbf{\Lambda}$

Définition 11.2 Soit $a \in \mathbf{\Lambda_V}$ tel que $FV(a) \subseteq \{x_1, \ldots x_n\}$. La traduction de a, notée $a_{DB_{\{x_1 \ldots x_n\}}}$, dans $\mathbf{\Lambda}$, est définie par :

1. $x_{DB_{\{x_1, \ldots x_n\}}} = i$ si i est le plus petit indice tel que $x = x_i$
2. $(\lambda\, x.a)_{DB_{\{x_1, \ldots x_n\}}} = \lambda\, .a_{DB_{\{x\} \cup \{x_1, \ldots x_n\}}}$
3. $(a\, b)_{DB_{\{x_1, \ldots x_n\}}} = a_{DB_{\{x_1, \ldots x_n\}}}\, b_{DB_{\{x_1, \ldots x_n\}}}$

L'ensemble des variables libres d'un terme est donc représenté par une liste, que nous appellerons *référentiel*. Pour obtenir la liste des variables libres d'un sous-terme “sous” un abstracteur, il suffit d'ajouter, en tête de la liste, la variable associée à l'abstracteur comme indiqué dans le point 2 de la définition précédente.

Exemple

$$((\lambda\, x.xy)y)_{DB_{\{y\}}} = (\lambda\, x.xy)_{DB_{\{y\}}}(y_{DB_{\{y\}}})$$

$$(\lambda\, x.xy)_{DB_{\{y\}}} = \lambda.(xy)_{DB_{\{x,y\}}} = \lambda\, .1\, 2 \qquad (y_{DB_{\{y\}}}) = 1$$

Donc :

$$((\lambda\, x.xy)\, y)_{DB_{\{y\}}} = (\lambda.1\, 2)\, 1$$

Il n'y a donc pas de correspondance directe entre numéros et noms de variables : y est représentée à la fois par 2 et 1.

Remplacer $(\lambda\, x.a)b$ par $a\{b/x\}$ nécessite d'identifier parmi les numéros ceux concernés par la β-réduction, autrement dit ceux qui sont égaux à leur hauteur de liaison. On peut calculer de façon récursive ces hauteurs de liaison à l'aide d'une famille de compteurs gérée ainsi : on démarre la traversée du terme a avec un seul compteur initialisé à 1. Au passage d'un noeud application, on crée deux exemplaires du compteur de façon à en obtenir un pour chaque branche. Au passage d'un noeud abstraction, on incrémente le compteur. Lorsqu'on arrive à l'occurrence d'un numéro, s'il est supérieur à la valeur du compteur, on le diminue de 1 puisqu'il existe un λ de moins sur le chemin entre ce numéro et l'abstracteur qui le lie. Sinon, si ce numéro est égal à la valeur du compteur, il doit être remplacé par le terme b, qui va donc se trouver ainsi placé sous $(n-1)$ λ, si n est la valeur du compteur. Il faut alors ajuster les numéros de b de façon à ne pas modifier les liaisons dans b. Si le numéro est inférieur à la valeur du compteur, alors il est lié par un abstracteur interne à a et il n'a donc pas à être modifié.

La substitution, dans le formalisme de De Bruijn, est donc définie comme suit :

Définition 11.3 La substitution du terme b à la hauteur n dans le terme a, notée $\sigma_n(a,b)$ ainsi que l'incrémentation avec n à partir de i, notée $\tau_i^n(a)$, sont définies par récurrence ainsi :

$$\begin{array}{rcl}
\sigma_n(a\,c, b) & = & \sigma_n(a,b)\;\sigma_n(c,b) \\
\sigma_n(\lambda.a, b) & = & \lambda(\sigma_{n+1}(a,b)) \\
\sigma_n(m,b) & = & \left\{\begin{array}{rcl} m-1 & \text{si} & m>n \\ \tau_0^n(b) & \text{si} & m=n \\ m & \text{si} & m<n \end{array}\right. \\
\\
\tau_i^n(m) & = & \left\{\begin{array}{rcl} m+n-1 & \text{si} & m>i \\ m & \text{si} & m\leq i \end{array}\right. \\
\tau_i^n(a\,c) & = & \tau_i^n(a)\;\tau_i^n(c) \\
\tau_i^n(\lambda(a)) & = & \lambda(\tau_{i+1}^n(a))
\end{array}$$

Définition 11.4 Dans $\mathbf{\Lambda}$, la β-réduction est la relation de réécriture définie par la règle :

$$(\lambda\,.a)b \longrightarrow \sigma_1(a,b)$$

References

[1] Barendregt H., *The Lambda-Calculus.* Volume 103, Elsevier Science Publishing Company, Amsterdam, 1984. *Généralement considéré comme la Bible du λ-calcul*

[2] H. P. Barendregt. *Pairing without conventional restraints.* Zeitschr. J. Math And Logik und Grundlagen d. Math, 20:289–306, 1974.

[3] N. de Bruijn. *Lambda-Calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser Theorem.* Indag. Math., 34(5):381–392, 1972.

[4] N. de Bruijn. *Lambda-Calculus notation with namefree formulas involing symbols that represent reference transforming mappings* Indag. Math.,40 :348–356, 1978.

[5] N. de Bruijn. *A namefree lambda calculus with facilities for internal definition of expressions and segments.* T.H.-Report 78-WSK-03, Technological University Eindhoven, Eindhoven - The Netherlands, August 1978.

[6] J. M. Carthy. *Recursive Functions of Symbolic Expressions and their Computation by by machine.* C.A.C.M, 3(4):184–195, 1960.

[7] A. Church. *The Calculi of Lambda-Conversion.* Ann. of Math. Studies, 6, 1941.

[8] Curien P.-L., *Categorical Combinators, Sequential algorithms and Functional Programming.* Second Edition, Birkäuser Boston, 1993.

[9] H. B. Curry. *Combinatory Logic.* Volume 1, North-Holland, 1958.

[10] Hindley R. J., Seldin J., *Introduction to Combinators and λ-calculus* London Mathematical Society Student Texts, Cambridge University Press 1986. *Contient une présentation du λ-calcul et de la logique combinatoire : syntaxe, modèles et typage*

[11] Hindley R. J., Seldin J., *To H. B. Curry : Essays on Combinatory Logic, Lambda-calculus and Formalism* Academic Press, 1980. *Un recueil d'articles devenus, pour la plupart, des classiques du domaine.*

[12] J. W. Klop. *Combinatory Reduction Systems.* Thèse, Mathematisch Centrum Amsterdam, 1982.

[13] Klop J.W. , *Combinatory Reduction Systems*, Math. Center Tracts 129, Amsterdam, 1980. *Thèse de J. W. Klop, assez difficile à trouver. Lecture recommandée comme complément au cours.*

[14] Krivine J.L.,

Lambda-Calcul, types et modèles Masson 1991 *Lecture recommandée comme complément au cours.*

[15] Lalement R. *Logique, réduction, résolution* Masson 1990. *Une présentation parfois succinte de toutes les notions abordées en tronc commun.*

[16] Lévy J.-J., *Réductions correctes et optimales dans le λ-calcul*, Thèse d'Etat, 1978, Univ. Paris 7. *Lecture recommandée comme complément au cours.*

[17] Krivine J.L., *Théorie Axiomatique des Ensembles* Collection SUP, Presses Universitaires de France

[18] Steelund S. *Combinators, λ-terms and proof theory* Reidel Publishing Compagny

[19] Y. Toyama. *On the Church-Rosser Property for the direct sum of Term Rewriting Systems.* 1985. J.A.C.M.