



HAL
open science

Systèmes de types purs et substitutions explicites

Romain Cyril Kervarc

► **To cite this version:**

Romain Cyril Kervarc. Systèmes de types purs et substitutions explicites. Logique en informatique [cs.LO]. École normale supérieure de Lyon, 2007. Français. NNT : . tel-01232508

HAL Id: tel-01232508

<https://theses.hal.science/tel-01232508>

Submitted on 23 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

N° d'ordre : 399

N° attribué par la bibliothèque : 07ENSL0 399

THÈSE

en vue d'obtenir le grade de

Docteur de l'École normale supérieure de Lyon

dans la spécialité informatique

*préparée au Laboratoire de l'Informatique et du Parallélisme
au titre de l'école doctorale de Mathématiques et Informatique fondamentale*

présentée et soutenue publiquement le 15 février 2007

par Romain Cyril KERVARC

SYSTÈMES DE TYPES PURS ET SUBSTITUTIONS EXPLICITES

sous la direction de M. Pierre LESCANNE,

après avis de M^{me} Delia KESNER & M. Ralph MATTHES, rapporteurs,

devant la commission d'examen formée de :

M ^{me} Delia KESNER,	professeur des universités,	rapporteur,
MM. Ralph MATTHES,	chargé de recherches habilité,	rapporteur,
René DAVID,	professeur des universités,	examinateur (président),
Philippe DE GROOTE,	directeur de recherches,	examinateur,
Pierre LESCANNE,	professeur des universités,	directeur de thèse.

SYSTÈMES DE TYPES PURS ET SUBSTITUTIONS EXPLICITES

THÈSE DE DOCTORAT

Romain Cyril KERVARC

Systemes de types purs
et substitutions explicites

Thèse de doctorat

École normale supérieure de Lyon

Ratio et oratio...

M. Tullius Cicero
De Officiis I, xvi-50

REMERCIEMENTS

Il est d'usage de placer en tête d'une thèse quelques lignes destinées à manifester aux diverses personnes de son entourage sa reconnaissance, sous une forme parfois légère, parfois cryptique, mais sincère. Voici donc, dans un semblant d'ordre n'ayant rien à voir avec celui du cœur, quelques dédicaces.

Un environnement de travail plaisant est d'un grand secours au long d'une thèse, et je souhaite remercier à ce propos le directeur du Laboratoire d'Informatique et du Parallélisme, Jean-Michel MULLER, toujours aimable et accessible ; les secrétaires si compétentes, souriantes, efficaces et compréhensives, en particulier Corinne IAFRATE et Sylvie BOYER.

Ensuite viennent les différents cobureaux, également doctorants au LIP, que j'ai eu au cours de ces années de thèse, et que je remercie tous pour leur présence amicale et les discussions variées que nous avons pu avoir ensemble : David TELLER, Étienne LOZES, Damien POUS, Stéphane LE ROUX, Aurélien PARDON.

Nombreux sont les membres du LIP qui mériteraient d'être mentionnés ici, et je les prie de m'excuser de ne pouvoir tous les citer. Je nommerai cependant les membres de l'équipe de recherche PLUME, dont je faisais partie, et qui, par leurs commentaires en groupe de travail et les discussions plus informelles ou amicales que nous avons pu avoir, ont bien gagné mes remerciements : Jean DUPRAT, Tom HIRSCHOWITZ, et surtout Daniel HIRSCHKOFF, à qui j'adresse toute ma gratitude pour son soutien et sa grande gentillesse.

À propos de l'ouverture aux autres disciplines, toujours enrichissante, je souhaite remercier Sacha BOURGEOIS-GIRONDE de m'avoir permis de suivre son séminaire de philosophie analytique et de monter un séminaire commun entre nos deux unités.

Pour conclure cette partie des remerciements concernant la recherche, toute ma gratitude va également aux membres de mon jury de soutenance : René DAVID, qui a bien voulu le présider, Philippe DE GROOTE, qui a accepté d'en faire partie, et tout particulièrement Delia KESNER et Ralph MATTHES qui, ayant accepté d'être rapporteurs, ont rempli cette fonction avec une conscience professionnelle digne d'éloges et un esprit constructif ; enfin, le dernier membre de mon jury, mon directeur de thèse, Pierre LESCANNE, mérite tout naturellement sa part de remerciements pour avoir accepté d'encadrer ma thèse.

Ma thèse ayant été financée par une allocation couplée, j'ai eu le privilège d'enseigner durant trois ans en tant que moniteur normalien à l'École normale supérieure de Lyon, au département de mathématiques et informatique. Mes pensées vont donc tout naturellement à Simone DURAND, l'indispensable secrétaire de département, aux personnes avec qui j'ai travaillé pour les cours qui me sont échus, en particulier Sylvie BOLDO, et à tous les élèves des promotions 2002 à 2004 de l'École à qui j'ai eu l'occasion d'enseigner quelque chose, avec un clin d'œil ému à la promotion 2003, celle à qui j'ai donné mon tout premier cours, et à qui j'ai le plus enseigné. Le climat de ces cours, alliant respect et proximité, humour et

travail, ainsi que leur contenu, pris dans mes disciplines favorites, en ont fait des moments inoubliables pour moi, et je remercie mes élèves pour l'assiduité et l'intelligence dont ils faisaient montre (même le vendredi matin à huit heures), pour avoir publié dans le journal des élèves mon sujet de partiel de réécriture (chose inédite, et à l'honneur de laquelle je suis sensible), et pour d'autres choses encore – bref, pour leur bon esprit.

Par ailleurs, en tant que cette fois-ci qu'étudiant, je souhaite remercier les enseignants de l'ÉNS dont j'ai pu suivre les cours au long de ma scolarité et de ma thèse, avec une mention spéciale pour le département des langues dirigé par M^{me} RANCUREL, dont j'ai suivi avec plaisir les cours de littérature anglaise ; pour les autres langues, mes remerciements vont à Pierre DESHUSSES et Frau RENZ pour celle de Goethe, à la Signora MILITI et au Signore PINNA pour celle de Dante, à Jenny JOFFER pour le suédois, et tout particulièrement à TERADA-sensei et ONO-sensei, dont les cours nous ont fait découvrir tant la langue que la culture, toutes deux passionnantes, du Pays du Soleil levant.

Si je n'ai jamais cédé à la lassitude qui guette parfois le doctorant, c'est sans nul doute avec l'aide de ceux qui me font la grâce de compter au nombre de mes amis. Je commencerai par deux fidèles parmi les fidèles, que je connais depuis le collège et le lycée, Simon CAUBET et Anne LECACHEUX, dont la présence régulière à mes côtés malgré la distance qui nous sépare depuis des années est pour moi inestimable ; et par Vincent LANGLOIS, qui fut mon colocataire durant notre scolarité à l'ÉNS Lyon, et que j'ai toujours grand plaisir à retrouver, ainsi que sa sœur Aline.

La logique est une discipline issue de la philosophie, et j'ai eu l'avantage d'avoir à mes côtés, pendant toute la durée de ma thèse, une fidèle amie pour représenter cette *alma parens* avec autant d'érudition que de pétillance : Aude BANDINI. Parmi mes condisciples à l'ÉNS, certains sont tout comme moi restés à Lyon pour y faire leur doctorat, et notre petite société fut durant ces années un havre de réconfort au milieu des grains que le doctorant ne manque pas d'essuyer, ce qui fait qu'il serait trop long d'énumérer toutes les raisons que j'ai de remercier Jérôme JOUBERT et Nathalie BLANC, Vincent MIRABET, François LUX et Caroline GENRE (avec une mention particulière de cette dernière) ; sans oublier la relève : Johannes HAGMANN.

La distance n'efface pas les liens d'amitié, n'empêche pas même de discuter autour d'une tasse de thé, boisson pour laquelle Jérôme HÉNIN partage ma vénération, ou encore de se retrouver à l'occasion entre anciens lyonnais, comme Olivier LAFON, Alexis PEAUCELLE et Charlotte LABALETTE (sans oublier leur petite Sophie), Jill AVAZERI, Claire DE BROCHE DES COMBES et Anne ARTERO. J'ai également une petite pensée pour la mémoire de Bruno REYSSAT.

Par ailleurs, des presque condisciples, jeunes mais intrépides face aux caïmans, méritent leur part de remerciements pour leur amitié rafraîchissante : David LEFRANC, Étienne DUCHESNE, Sattisvar TANDABANY, Julien SEIN, Céline COLSON, Camille HOCHEDÉZ.

Le passage à Lyon lors de ma soutenance de Luca MARSELLA, que j'avais connu à Munich, fut comme une bouffée d'air bavarois, et me donne ici l'occasion de remercier

mes amis Erasmus de ce qu'ils m'ont apporté : Silvio PIRANDELLO, Katia RUNSER, Nicolas DE SAINT-ROMAIN, Anne-Laure CHARTIER, Carole CHOMEL DE VARAGNES, Kathrin PFLÜGER, Claudia SILVESTRI DEMANDT, Lucrezia GORINI, Maria ERLANDSSON, Jorge GARCÍA ONCINS. À tous, *danke schön, grazie mille, thank you, tack, gracias, dank u, gratias*.

L'ÉNS permet également à ses étudiants de se retrouver dans de nombreux clubs, dans mon cas liés à la musique et au spectacle, permettant de se changer agréablement les idées un soir par semaine, et j'en remercie leurs membres, outre ceux mentionnés plus haut pour d'autres raisons. L'Orchestre rassemble des étudiants de l'ÉNS, de Centrale et de Sup de Co, et je citerai plus particulièrement Antonin FRADIN, Sylvain LATOUR, Anne-Lise RICHARD, mais aussi Cyril PETERSCHMITT, Vincent DUCROS, Sébastien AEBERLI, Gaëlle GUIGNARD, Cécile LAVERGNE, Séverin TREILLE. La Fanfare de l'ÉNS permet également de libérer certains instincts chez le musicien, ce que résume son motto : « plus de bruit ! », et je nommerai spécialement Scooby, Jésus, Simoun, Grand Laurent, Élise, Marge. La Chorale est un plaisir hebdomadaire, partagé avec de nombreuses personnes, dont son chef Jean-François LE MARÉCHAL, Eudes HARTEMANN, Jean-Claude DIDELOT, Marie-Madeleine FIERS, Julie VERMEILLE, Marie DEVAUTOUR. Le Club Théâtre donne l'occasion de monter sur scène devant ses élèves ailleurs qu'en salle de cours, et je remercie spécialement Julie DENÈUD, Maxime FEYEU, Guillaume DUMAZER, Céline BERNI, Martin VALLON, Sarah JASSIONNESSE, Joris MITHALAL, et nos metteurs en scène, Nadine DOURIAUD et Séverine PUEL.

Durant la fin de ma thèse, un précieux soutien me fut apporté par Bruno, Jean-Baptiste, Xavier, Jérémy et le P. Michel CACAUD, et je les assure ici de toute ma gratitude.

Enfin, je conclurai ces remerciements en consacrant un paragraphe à ma famille, dont les membres m'ont toujours entouré et soutenu, et en premier lieu mes parents, à qui je dois tant que je ne peux dire que ce simple mot, *multa paucis* : merci.

J'adresse également mes remerciements à mes cousins germains : Gaëlle, Yann, Pierre, Thomas, François, ainsi qu'à tous mes oncles, tantes, et cousins, et en particulier au Dr Rolf et Christiana SCHMIDT-SEYFANG pour leurs encouragements et à ma cousine Karen, dont la venue à Lyon fut un rayon de soleil pendant la fin de ma rédaction de thèse.

Enfin, je remercie vivement pour leur soutien moral mes deux grand-mères : Rosa, fervente épistolière de qui j'ai reçu de nombreuses lettres d'encouragement, et Micheline, dont le grand courage devant les épreuves me fut un exemple à suivre, et j'adresse avec une pensée émue une dédicace particulière à mes deux grands-pères, qui m'ont quitté avant l'aboutissement de cette thèse.

TABLE DES MATIÈRES

INTRODUCTION	1
CHAP. PREMIER – Généralités	7
1. Λ-calcul	9
1.1. Définition	9
1.2. Explicitation de méta-opérations	15
1.3. Ordre récursif de chemin	21
2. Λ-calcul typé	22
2.1. Typage à la CURRY ou à la CHURCH	23
2.2. Le λ -cube	26
2.3. Types pour les calculs à substitutions explicites	28
3. Systèmes de types purs	29
3.1. Définition et caractéristiques	30
3.2. Propriétés générales	34
3.3. Construction du λ -cube en systèmes de types purs	36
CHAP. II – Systèmes de types purs explicites	41
1. Définition	43
1.1. Historique	43
1.2. Syntaxe et réduction	44
1.3. Typage	59
2. Particularités remarquables	72
2.1. La figure d'inférence d'expansion	72
2.2. Le prédicat « pseudo-sortie »	79
2.3. Ordre sur les dérivations	85

3. Propriétés	88
3.1. Lemmes de dérivation	88
3.2. Réduction du sujet	95
3.3. Conservation de la normalisation forte	123
CHAP. III – Systèmes de types purs et séquents classiques	129
1. Le $\bar{\lambda}\mu\bar{\mu}$-calcul	131
1.1. Historique	131
1.2. Syntaxe et réduction	132
1.3. Séquents et typage	134
2. Systèmes de types purs pour le $\bar{\lambda}\mu\bar{\mu}$-calcul	137
2.1. Syntaxe et réduction	137
2.2. Typage	142
2.3. Correction des types	145
3. Propriétés remarquables	146
3.1. Lemmes d'inférence de types	147
3.2. Réduction du sujet	149
3.3. Normalisation forte	151
CONCLUSION	159
ANNEXE A – Index	165
Index des figures, tables et dérivations	167
Index des définitions	168
Index des énoncés	170
ANNEXE B – Glossaire	173
ANNEXE C – Bibliographie	177

INTRODUCTION

La théorie des types est actuellement considérée comme un outil fondamental en informatique, car elle établit un lien entre la logique d'une part et la programmation d'autre part, ce qui permet de produire des langages de programmation particulièrement intéressants, qui allient la puissance de calcul d'un langage Turing-complet à la rigueur de la logique, laquelle permet *via* les types associés à chaque programme d'assurer que celui-ci vérifie certaines propriétés.

Ainsi, si l'on considère par exemple des langages fonctionnels de la famille ML, on peut garantir grâce au typage certaines propriétés : si l'on a une fonction u de type $\text{int} \rightarrow \text{bool}$, on est assuré que si cette fonction est appliquée à un argument n de type int , l'évaluation de l'expression $u(n)$ s'achèvera en un temps fini, et produira une valeur de type bool . Le grand avantage de cette approche est que cette propriété peut être vérifiée lors du typage, donc avant d'exécuter le programme.

Dans tous ces langages de programmation, il existe une séparation nette entre l'univers des termes (c'est ainsi que l'on nomme les programmes en théorie des types) et l'univers des types : ces derniers forment une catégorie syntaxique autonome, construite à partir de types élémentaires (les entiers, les booléens, les flottants, etc.) et de constructeurs (types fonctionnels, types produits, listes, etc.), et ils se placent au-dessus des termes. Dans ces langages, être un type est une simple propriété syntaxique.

En ce qui concerne les termes, en revanche, bien que ceux-ci soient définis de manière assez analogue aux types au moyen de termes élémentaires (variables, constantes, liste vide, etc.) et de constructeurs (abstraction fonctionnelle, ajout à une liste, application, etc.), ainsi que, dans certains systèmes, de types, la simple correction syntaxique ne suffit pas : il faut également une vérification de type, qui assure que les différents éléments du programme respectent bien les contraintes de types qui leur sont associées, et qui, en outre, produit également le type du programme.

On se trouve alors en général confronté à deux problèmes algorithmiques principaux : la *vérification de type* et l'*inférence de type*, qui peuvent s'énoncer respectivement comme suit :

- étant donné un terme M et un type T , M a-t-il le type T ?
- étant donné un terme M , M a-t-il un type, et, si oui, quel est-il ?

Ces deux problèmes sont en général décidables dans les langages effectivement implantés. Toutefois, il existe divers systèmes de types où cela n'est pas le cas.

On peut naturellement se demander pourquoi considérer de tels systèmes, où l'inférence, voir la vérification, ne sont pas décidables. En fait, cette indécidabilité est souvent le prix à payer pour disposer d'un système de types plus expressif, qui

permettra donc de vérifier des propriétés plus fines : en effet, il est patent que plus la théorie logique sous-jacente à un système de types est puissante, plus la variété des propriétés que l'on pourra vérifier syntaxiquement au moyen du typage est grande – et, malheureusement, plus les chances d'indécidabilité augmentent.

Pour revenir à l'exemple des langages de la famille ML, certaines propriétés ne peuvent pas être vérifiées au moyen de types. Ainsi, si l'on considère un programme manipulant des listes d'entiers, il existera des programmes bien typés qui sur certaines données produiront une erreur : si f est une fonction qui à une liste associe la valeur de son 12^e élément, et si s est une liste de taille $i < 12$, l'expression $f(s)$ sera bien typée, et pourtant son exécution aboutira à une erreur. Cela est dû au fait que lorsque l'on type s , on sait que l'on a affaire à une liste, mais on ne dispose d'aucune information sur sa taille.

Une idée naturelle pour résoudre ce problème est l'adjonction d'une information au type de s : au lieu de se contenter d'indiquer que s est une liste d'entiers, on indique que s est une liste de n entiers, où n est une expression à valeur entière, et l'on obtient ainsi ce que l'on appelle un type *dépendant* d'un terme. Il appert tout de suite que, si l'on impose pas, comme c'est le cas ici, que n soit une constante, la barrière syntaxique entre termes et types s'effondre, et il est nécessaire d'introduire dans les types une vérification de type : être un type n'est plus une simple propriété syntaxique, car il faut vérifier que chaque dépendance a le type adéquat.

Une fois ces types dépendants introduits, l'on se rend compte que la construction fonctionnelle habituelle \rightarrow n'est plus suffisante. En effet, si l'on dispose de tels types, on peut vouloir que la fonction k qui à n associe une liste de n éléments initialisés à 0 soit dotée d'un type indiquant que $k(n)$ est une liste de n entiers, ce qui, à y regarder de plus près, impose que la valeur de l'argument (et non seulement son type) intervienne dans le type du résultat, ce que ne permet pas la flèche. On ajoute donc une construction appelée *produit dépendant*, sous la forme d'un quantificateur que l'on note Π .

Intuitivement, si f est de type $\Pi a:U.V(a)$ (on écrit $V(a)$ pour indiquer que cette expression peut dépendre de a), et si x est de type U , alors $f(x)$ sera de type $V(x)$. La fonction k ci-dessus pourra donc avoir pour type $\Pi n:\text{int}.\text{(int list)}_n$. Naturellement, dans le cas où le produit dépendant est de la forme $\Pi x:A.B$, avec B ne dépendant pas de x , on retrouve le type \rightarrow habituel. D'un point de vue logique, le produit dépendant correspond à une quantification universelle à domaine restreint, *i.e.* une formule de la forme $\forall x \in X (Y)$.

Comme on l'a mentionné plus haut, il convient, dans un système de types dépendants, de vérifier la bonne formation des types, au moyen d'un jugement de typage analogue à celui d'un terme. L'idée peut alors sembler naturelle d'ajouter une constante de type `type` : de même qu'avant un terme `M` pouvait être un entier (« `M` est de type `int` »), une liste de flottants (« `M` est de type `float` »), une fonction des booléens dans eux-mêmes (« `M` est de type `bool`→`bool` »), un terme peut maintenant de surcroît être un type (« `M` est de type `type` »). En outre, il faut donner un type à `type`, et l'on peut alors décréter que `type` est de type `type`. Un tel système fut introduit par P. MARTIN-LÖF en 1971. Malheureusement, il s'est avéré que ce système était incohérent (*i.e.* on pouvait y trouver un programme ayant pour type l'absurde¹), comme l'a montré J.-Y. GIRARD dans [45].

Pour pallier ce problème, qui est lié à la circularité de l'assertion que « `type` est de type `type` », il suffit en fait d'introduire plusieurs constantes de types. En général, on en introduit deux : l'une, `*`, tient le rôle de `type` ci-dessus, (c'est le type des termes types, *i.e.* des termes qui typent des termes) tandis que l'autre, `□`, sert à typer `*` (c'est le type des termes qui typent des termes types), et est elle-même non typable. Par la suite, on peut utiliser ces constantes, que l'on nomme *sortes*, pour déterminer les différentes dépendances que l'on souhaite permettre entre termes et types, et l'on peut en introduire plus de deux. Les différents systèmes de types pouvant être ainsi obtenus sont groupés au sein d'un même formalisme sous le nom générique de systèmes de types purs.

L'élégance et la sobriété desdits formalismes étant grandes, et leur pouvoir d'expressivité élevé, vient alors naturellement l'idée d'étendre ces systèmes à d'autres calculs que le λ -calcul usuel. On peut envisager différents types d'extension possible : une extension à une variante syntaxique du λ -calcul, le but étant alors d'obtenir une théorie des types identique à celle des systèmes de types purs, *i.e.* s'interprétant en logique intuitionniste d'ordre supérieur, pour le calcul considéré ; ou encore une extension à un calcul autre que le λ -calcul, présentant des mécanismes de typage différents, pour obtenir des systèmes de types purs dans un autre cadre logique, par exemple la logique classique.

Cette thèse étudie différentes extensions de cette notion de systèmes de types purs, principalement à des calculs à substitutions explicites, et à des calculs de séquents classiques, en s'intéressant particulièrement à la préservation des propriétés de ces systèmes.

¹en l'occurrence, son avatar $\Pi a:\text{type}.a$, autrement dit la contradiction $\forall \varphi (\varphi)$ issue du mode de raisonnement *ex falso quodlibet sequitur*

Le premier chapitre, essentiellement historique, présente tout d'abord le λ -calcul dans sa généralité, la question de l'explicitation des méta-opérations, et des outils usuels. Il traite ensuite du λ -calcul typé de façon générale et des approches usuelles du λ -cube et des types pour les calculs à substitutions explicites.

Le deuxième chapitre présente une extension des systèmes de types purs à un calcul à substitutions explicites inspiré du λx -calcul. Dans un premier temps, il présente les différentes approches existant dans la littérature pour les systèmes de types purs avec substitution explicite, ainsi que le calcul sur lequel on va se baser (qui est une extension de λx reposant sur un calcul introduit dans [30], mais résolvant le problème posé par ce dernier, qui n'est pas un système de réécriture) et définit les nouveaux systèmes faisant l'objet de cette partie. Les éléments remarquables de ces systèmes sont ensuite discutés, parmi lesquels on peut essentiellement citer trois choses : un mécanisme de reconnaissance de sortes, permettant de clarifier la question de la correction des types rendue plus complexe du fait de l'explicitation de la substitution ; une figure de typage réalisant une expansion contrôlée issue d'une approche du typage des substitutions dans une optique de synthèse de type où la substitution est considérée comme une opération paresseuse, et enfin un ordre sur les dérivations permettant de raisonner par induction non plus seulement sur des sous-arbres réels d'une dérivation, mais également sur des pseudo-sous-arbres reconstruits par l'application d'un lemme (*i.e.* permettant de considérer non la dérivation effective, mais une dérivation plus commode parmi celles possibles). Enfin, on y montre certaines propriétés de ces systèmes, en particulier la réduction du sujet, toujours délicate dans le contexte des substitutions explicites, et d'ailleurs non vérifiée dans divers systèmes, en particulier dans [9].

Le troisième chapitre présente tout d'abord un historique du $\bar{\lambda}\mu\bar{\nu}$ -calcul, dont les jugements de type s'interprètent comme un calcul des séquents classiques, puis étend ce calcul à un formalisme inspiré des systèmes de types purs, dans une approche résolument syntaxique. On y définit des notions de correction des types et de bonne formation des contextes de typage, qui sont rendues délicates par le fait que l'on a deux contextes avec des dépendances imbriquées. On étudie ensuite certaines propriétés usuelles des systèmes de types purs. Enfin, on montre un résultat de normalisation forte pour les types simples, pouvant sans doute être étendu aux types polymorphes et aux types d'ordre supérieur, et l'on présente une technique possible pour obtenir des résultats de normalisation forte par un encodage dans le système de type pur intuitionniste, la non-confluence du calcul initial étant compensée par l'apparition de contextes autour des traductions, la présence de ces contextes étant rendue non gênante au moyen d'une réorganisation particulière des chemins de réduction ayant une propriété de perpétualité.

CHAPITRE PREMIER

Généralités

— SOMMAIRE —

1. Λ-calcul	9
1.1. Définition	9
1.2. Explication de méta-opérations	15
1.3. Ordre récursif de chemin	21
2. Λ-calcul typé	22
2.1. Typage à la CURRY ou à la CHURCH	23
2.2. Le λ -cube	26
2.3. Types pour les calculs à substitutions explicites	28
3. Systèmes de types purs	29
3.1. Définition et caractéristiques	30
3.2. Propriétés générales	34
3.3. Construction du λ -cube en systèmes de types purs	36

1. λ -calcul

Dans cette partie, l'on va effectuer un bref rappel historique des concepts et des problématiques liées au λ -calcul, et l'on discutera les différents aspects des questions syntaxiques ou opérationnelles liées à ce calcul.

1.1. Définition

Le λ -calcul, inventé par A. CHURCH vers 1930 (cf. [18, 17]), procède d'une volonté d'abolir la distinction entre objets, fonctions, foncteurs, etc., et de les considérer tous comme des objets de première classe.

Syntaxe

Les termes de ce calcul² sont définis à partir d'un ensemble infini \mathcal{X} de variables par la grammaire algébrique suivante :

$$\begin{array}{lll} M, M' & ::= & x \quad (\text{le terme est une } \textit{variable}) \\ & | & (\lambda x.M) \quad (\text{le terme est une } \textit{abstraction}) \\ & | & (MM') \quad (\text{le terme est une } \textit{application}) \end{array}$$

Afin d'éviter le fastidieux usage systématique des parenthèses, on choisit d'adopter les conventions d'écriture suivantes :

- l'application est associative à gauche,
e.g. : $A B C$ désigne $((A B) C)$;
- l'application est prioritaire sur l'abstraction,
e.g. : $\lambda x.A B$ désigne $(\lambda x.(A B))$;
- les abstractions peuvent être groupées,
e.g. : $\lambda xyz.C$ désigne $(\lambda x.(\lambda y.(\lambda z.C)))$.

Il existe également une autre convention de parenthésage des λ -termes, introduite par J.-L. KRIVINE (cf. *e.g.* [65]), et présentant principalement l'avantage d'être univoque :

$$M, M' \quad ::= \quad x \quad | \quad \lambda x.M \quad | \quad (M)M'.$$

Toutefois, l'autre convention étant plus usuelle, et plus proche de l'intuition courante, c'est celle qui sera adoptée ici.

²on parle également de λ -calcul à *noms*, par opposition aux calculs à indices que l'on verra au paragraphe suivant

On peut naturellement définir formellement une notion de sous-terme et de sous-terme propre :

- M est un sous-terme de M ;
- M est un sous-terme propre de $\lambda x.M$, $A M$ et $M B$;
- si M est un sous-terme propre de N , alors c est un sous-terme de N .

On adopte en général la notation $M \subseteq N$ (respectivement $M \subset N$) pour indiquer que M est un sous-terme (respectivement un sous-terme propre) de N .

Le λ -calcul est un paradigme des langages fonctionnels, et ses deux opérations élémentaires, l'abstraction et l'application, dont on va détailler dans ce qui vient la syntaxe et la sémantique, s'interprètent en termes fonctionnels.

L'abstraction

L'abstraction par rapport à la variable x permet de passer de l'expression $f(x)$ à la fonction f . C'est l'opération qui permet de considérer les fonctions comme des objets de première classe : on peut ainsi écrire $f = \lambda x.A$, où A est un terme représentant la valeur de $f(x)$ en toute généralité, au lieu de se contenter de définir les fonctions en extension (*i.e.* en les identifiant à leur graphe).

On voit ainsi par exemple que $\lambda x.x$ représente la fonction identité, et il appert que $\lambda y.y$ représente la même fonction. Une telle variable située sous la portée d'une abstraction est dite *liée*, ce qui s'oppose aux variables *libres*. Un terme sans variable libre est dit *clos*.

Les termes obtenus par renommage d'une variable liée sont dits α -convertibles. La relation d' α -conversion est une relation d'équivalence sur l'ensemble des λ -termes, et elle est une congruence pour l'abstraction et l'application. Il est nécessaire de prendre en compte cette conversion, ce que l'on peut faire par différents moyens, dont les deux plus usuels sont :

- a. On considère comme termes du λ -calcul non les termes définis *supra*, mais leurs classes d'équivalence pour l' α -conversion, auxquelles s'étendent canoniquement l'abstraction et l'application, ces opérations étant compatibles avec l' α -conversion.
- b. On considère l' α -conversion comme une opération « administrative », qu'il est parfois nécessaire d'effectuer « à la volée ».

La première approche présente l'avantage d'être plus formelle, tandis que la suivante se perçoit peut-être de façon plus intuitive. Dans les deux cas, l' α -conversion est reléguée au rang de méta-opération, et les problèmes posés par ces dernières seront abordés *infra*.

L'application

L'application, comme son nom l'indique, permet d'appliquer une fonction à son argument. Toutefois, l'on peut remarquer qu'usuellement, en mathématique, il convient de distinguer l'application purement syntaxique d'une fonction à un argument et la valeur de l'image d'un argument par une fonction – *e.g.*, si l'on a $f : n \mapsto n^2$, un calcul est nécessaire pour passer de $f(3)$ à 9. Ce type de calcul sera génériquement appelé une *réduction*.

Cette question latente en mathématique se pose naturellement de façon particulièrement aiguë dans le cadre du λ -calcul, qui, étant un calcul formel, se doit de traiter avec la plus grande rigueur les mécanismes de réduction : il n'est pas question d'identifier deux termes distincts (respectivement, le terme appliquant f à x et celui représentant l'image de x par f), et la relation de réduction permettant de passer l'argument dans la fonction, *i.e.* permettant d'aller du terme où l'on applique f à x – *i.e.* où l'on écrit $f(x)$ – au terme valant $f(x)$ – *i.e.* où l'on a calculé $f(x)$ – doit être décrite, ce qui confèrera au λ -calcul une sémantique opérationnelle de réduction.

L'opération qui effectue ce passage d'argument est la β -réduction, induite par l'axiome suivant :

$$(\lambda x.M)N \xrightarrow{\beta} M\{x \leftarrow N\}$$

où $M\{x \leftarrow N\}$ désigne le terme M où toutes les occurrences libres (cf. *supra*) de x ont été remplacées par un sous-terme N .

Cette substitution est une opération définie au niveau du méta-langage inductivement comme suit :

- $x\{x \leftarrow N\} = N$;
- $y\{x \leftarrow N\} = y$ (avec $y \neq x$) ;
- $(\lambda y.A)\{x \leftarrow N\} = \lambda y.(A\{x \leftarrow N\})$ (avec $y \notin \text{fv}(N)$, $y \neq x$) ;
- $(AB)\{x \leftarrow N\} = (A\{x \leftarrow N\})(B\{x \leftarrow N\})$.

On remarque qu'il manque un cas dans cette définition : le cas $(\lambda x.A)\{x \leftarrow N\}$. Ce cas est problématique, car il pose la question de la capture des variables. On voit facilement ce problème dans un exemple. Si l'on considère *e.g.* $K = \lambda u.\lambda v.u$, l'on voit que ce terme est la version curryfiée de la première projection canonique, *i.e.* qu'il représente une fonction prenant successivement deux arguments, et gardant le premier. Si l'on applique ce terme à une variable, on s'attend à ce que le résultat de cette application soit une fonction constante égale à ladite variable. Mais si l'on considère le terme Kv , on s'aperçoit qu'il se réduit à $(\lambda v.u)\{u \leftarrow v\}$, ce qui est égal à $\lambda v.v$, si bien que la variable v , libre dans Kv , est passée sous la portée d'une

abstraction, se retrouvant liée par lui : elle a été « capturée » par ce λ ; et le résultat de la substitution est l'identité et non la fonction constante attendue. Mais si l'on avait considéré un terme α -convertible à $\lambda u.\lambda v.u$, par exemple $\lambda u.\lambda w.u$, l'on aurait évité ce problème.

Afin d'éviter les captures décrites ici, l'on emploie différents moyens, suivant la façon dont on considère l' α -conversion. Si l'on se place dans le cas a. évoqué *supra*, on peut simplement dire que pour trouver un représentant de la classe du terme issu de la réduction, il faut appliquer celle-ci à un représentant bien choisi de la classe du terme que l'on réduit, et par « bien choisi », on peut par exemple entendre que le représentant doit respecter la convention de Barendregt, qui veut qu'une variable libre n'apparaisse pas liée, et que deux variables liées aient des noms distincts. Si l'on se place dans le cas b., on dit en général que l'on effectue implicitement les α -conversions requises pour éviter la capture en même temps que l'on effectue la β -réduction. Les problèmes liés à cette approche seront évoqués *infra*.

Propriétés de la β -réduction

Lorsque l'on étudie une relation comme la β -réduction, il est naturel de s'intéresser à certaines de ses propriétés, dont les principales se répartissent en deux catégories :

- 1°) les propriétés de confluence permettent de clore des diagrammes :
 - la propriété du losange : $\forall x\forall y\forall z(x\rightarrow y \wedge x\rightarrow z \Rightarrow \exists t(y\rightarrow t \wedge z\rightarrow t))$
(ce qui revient à dire que la relation commute avec elle-même),
 - la confluence : $\forall x\forall y\forall z(x\overset{*}{\rightarrow}y \wedge x\overset{*}{\rightarrow}z \Rightarrow \exists t(y\overset{*}{\rightarrow}t \wedge z\overset{*}{\rightarrow}t))$, ou encore (ces deux énoncés sont équivalents) la propriété de CHURCH-ROSSER : $\forall u\forall v(u\overset{*}{\leftarrow}v \Rightarrow \exists w(u\overset{*}{\rightarrow}w \wedge v\overset{*}{\rightarrow}w))$,
 - la confluence locale : $\forall x\forall y\forall z(x\rightarrow y \wedge x\rightarrow z \Rightarrow \exists t(y\overset{*}{\rightarrow}t \wedge z\overset{*}{\rightarrow}t))$;
- 2°) les propriétés de normalisation permettent d'assurer une terminaison :
 - la normalisation forte assure qu'aucun terme n'admet de suite infinie de réduction,
 - la normalisation faible assure que tout terme admet une suite finie de réduction se terminant sur un terme irréductible.

Chacune des propriétés de chaque catégorie est plus forte que la suivante, et l'on peut montrer le lemme de Newman, qui assure qu'une relation fortement normalisante est localement confluente si, et seulement si, elle est confluente.

La β -réduction est localement confluente et confluente, mais n'a pas la propriété du losange. Si l'on considère le terme $U = (\lambda x.xx)((\lambda y.y)a)$, on voit que

$U \xrightarrow{\beta} V = ((\lambda y.y)a)((\lambda y.y)a)$ et $U \xrightarrow{\beta} W = (\lambda y.xx)a$, mais V ne peut se réduire en un pas qu'en $a((\lambda y.y)a)$ ou $(\lambda y.y)a$, tandis que W ne peut se réduire qu'en un seul terme, qui est aa .

Par ailleurs, la β -réduction n'est pas normalisante, ne fût-ce que faiblement, comme on le voit avec le terme $\Omega = (\lambda x.xx)(\lambda x.xx)$, qui se β -réduit uniquement à lui-même.

La β -conversion

La β -réduction induit une relation d'équivalence, la β -conversion, notée \equiv_{β} , définie comme la clôture réflexive, symétrique et transitive de la β -réduction. Du fait de la confluence de cette dernière, cette conversion est en fait identique à la β -joignabilité \downarrow_{β}^* , définie par : $A \downarrow_{\beta}^* B$ s'il existe C tel que $A \xrightarrow{\beta}^* C$ et $B \xrightarrow{\beta}^* C$.

La β -conversion représente une égalité au sens de l'approche mathématique du λ -calcul : en effet, elle identifie le terme appliquant une fonction à un argument et le terme représentant l'image de cet argument par cette fonction, tout comme, pour reprendre l'exemple précédent où $f : x \mapsto x^2$, on écrit en mathématique que $f(3) = 9$.

Dans cette optique, il arrive que l'on considère le λ -calcul non comme un calcul de processus, mais comme une théorie équationnelle dont la β -conversion constitue l'égalité démontrable.

Extensionnalité

Une autre réduction, dite η -réduction, permet de simplifier les expressions fonctionnelles. Elle est induite par l'axiome suivant :

$$\lambda x.Mx \xrightarrow{\eta} M \quad \text{si } x \text{ n'est pas libre dans } M$$

et l'on peut montrer qu'elle rend la théorie du λ -calcul extensionnelle, *i.e.* que :

$$\text{si } \forall x (Mx \equiv_{\beta\eta} Nx), \quad \text{alors } M \equiv_{\beta\eta} N$$

– ce qui est faux avec la seule β -équivalence : $\lambda u.yu$ et y ne sont pas β -convertibles, étant irréductibles donc injoignables, et ils sont néanmoins extensionnellement égaux, car pour tout x , $(\lambda u.yu)x \equiv_{\beta} yx$.

On peut noter que l'extensionnalité en question n'est que purement syntaxique : le λ -calcul peut être employé pour implanter divers calculs, comme l'arithmétique, mais dans ce cas-là, comme tous les termes ne représentent pas des entiers, il peut arriver que deux termes fonctionnels coïncident sur tous les entiers sans pour autant être η -convertibles.

Les indices de DE BRUIJN

L' α -conversion est une méta-opération, *i.e.* une opération décrite dans une méta-théorie. Cette particularité fait que cette opération échappe au langage, et s'effectue d'une façon non contrôlée. En particulier, cela influe sur les résultats que l'on montre : ainsi, lorsqu'on énonce la confluence (locale) de la β -réduction, l'on ne montre pas vraiment la propriété que l'on a en tête (à gauche sur la figure *infra*) mais plutôt une autre (à droite sur cette même figure).

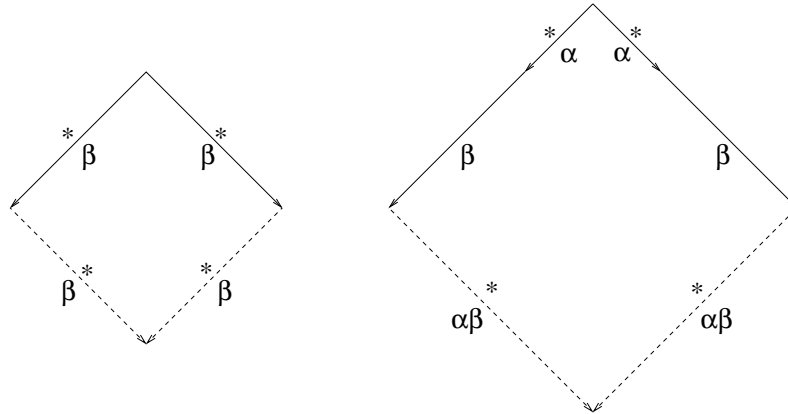


FIG. 1 – Confluence de β avec α -conversion implicite et explicite

En outre, dans le λ -calcul à noms, seule une référence commune à un même nom établit une connexion entre une variable et le λ la liant, ce qui n'est pas satisfaisant dans une optique d'implantation du λ -calcul, où, pour effectuer les opérations de substitution, il est nécessaire de déterminer si une variable est liée, donc de trouver, étant donnée une variable, s'il existe un λ la liant.

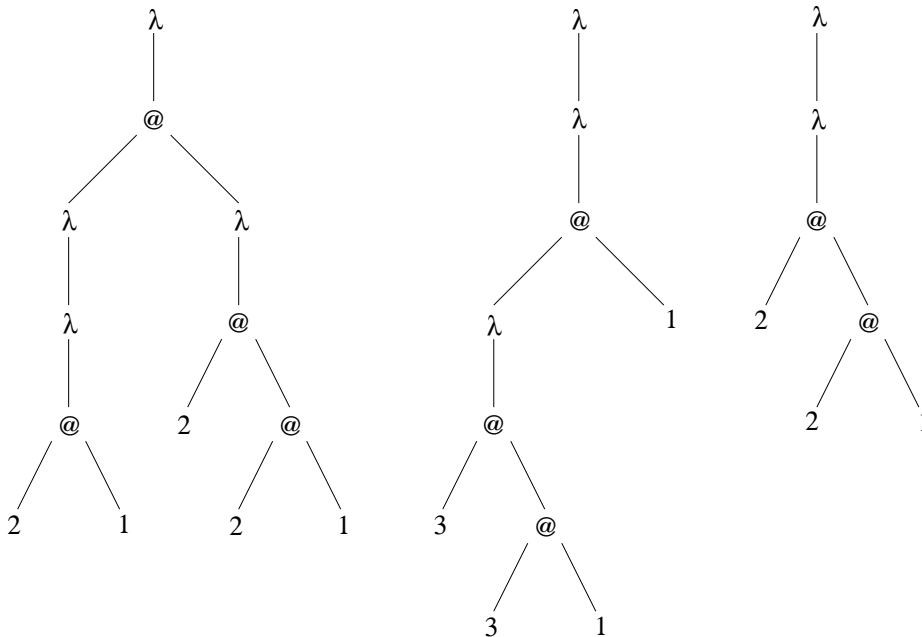
Afin d'éviter ce problème, N. DE BRUIJN a proposé (cf. *e.g.* [27]) une variante du λ -calcul, où les variables ne sont plus indiquées par un nom, mais par un simple *indice* indiquant le rang du λ la liant lorsqu'on remonte dans l'arbre syntaxique du terme. Ainsi, $\lambda x.(\lambda y. \lambda z. yz)x$ se notera $\lambda(\lambda\lambda 21)1$.

On supprime ainsi les difficultés liées à l' α -conversion en supprimant purement et simplement cette dernière, car on introduit une façon canonique de désigner les variables. Toutefois, le prix à payer en est élevé. D'une part, les termes obtenus sont moins lisibles. D'autre part, il est nécessaire de mettre à jour les indices d'un terme en le substituant sous une abstraction.

En effet, si l'on considère par exemple la réduction :

$$\lambda w.(\lambda y.\lambda z.yz)(\lambda x.w(wx)) \xrightarrow{\beta} \lambda w.\lambda z.(\lambda x.w(wx))z \xrightarrow{\beta} \lambda w.\lambda z.w(wz)$$

dont les termes se transposent en calcul à indices de DE BRUIJN respectivement comme suit (on adopte ici une présentation sous forme d'arbres, car elle permet de lire plus aisément les indices : pour savoir par quel λ ils sont liés, il suffit de remonter dans l'arbre en partant des feuilles – une feuille d'indice n étant liée par le n^{e} λ rencontré en remontant vers la racine) :



On voit bien sur la figure *supra* que l'indice représentant la variable w dans le sous-terme $\lambda w.w(wx)$ est au départ 2, puis, suite au passage sous un λ , est mis à jour à 3, puis, suite à l'élimination par une β -réduction du λ d'un redex situé plus haut dans l'arbre, à 2.

En outre, la question de l'indice à donner à des variables libres est problématique dans ce calcul, en particulier quand se pose la question de la mise d'un terme au sein d'un contexte pouvant lier ses variables libres.

1.2. Explicitation de méta-opérations

Comme on l'a vu précédemment avec l' α -conversion, les méta-opérations posent problème car elles échappent au contrôle du langage, tandis qu'il serait souhaitable – par exemple dans une optique d'implantation du λ -calcul, laquelle, est-il besoin de le rappeler, est d'un grand intérêt, ne serait-ce que dans le cadre des assistants automatiques de démonstration – de pouvoir expliciter ces opérations, *i.e.* les introduire dans le calcul.

La substitution

Une méta-opération d'une importance primordiale est la substitution. Il est intéressant de noter que cette question de l'explicitation de la substitution s'est posée très tôt, car on peut déjà lire dans l'introduction du livre [22] de logique combinatoire de 1958 de H. CURRY ET R. FEYS que la substitution est un sujet d'intérêt majeur en logique, et que le λ -calcul ne le traite pas convenablement car les substitutions n'y font pas partie du calcul. Depuis, cette question a été abondamment traitée.

L'une des premières études recensant les différents calculs à substitutions explicites proposés dans la littérature est celle de P. LESCANNE, [70], datant de 1994. D'autres systèmes ont vu le jour depuis, mais on peut trouver dans cette référence suffisamment de différents calculs pour pouvoir se faire une idée de leur variété.

Les calculs à substitutions explicites se divisent en deux familles principales, tout comme le λ -calcul : les calculs à indices de DE BRUIJN, dans la mouvance de [1, 2] ou encore [96], et les calculs à noms, dans celle de [14].

Naturellement, tous ces calculs vérifient des propriétés différentes au détriment d'autres, selon la mouvance dans laquelle ils s'inscrivent. Il est fréquent qu'il faille ainsi renoncer à certaines propriétés pourtant importantes, comme la simulation pas à pas de la β -réduction, la confluence sur les termes avec méta-variables, la normalisation forte du calcul de substitutions et la préservation de la normalisation forte. On peut ici noter les travaux de R. David et B. Guillaume, qui, dans [24, 25], fournissent et étudient un calcul ayant toutes ces propriétés. L'on s'intéressera ici plus particulièrement aux calculs conservant des propriétés liées à la normalisation forte, à propos desquels on pourra se reporter à [39].

Dans tous ces calculs, on part du constat que la β -réduction telle qu'elle est habituellement présentée :

$$(\lambda x.A)B \quad \xrightarrow{\beta} \quad B\{x \leftarrow A\}$$

n'est pas satisfaisante car d'une part, d'un point de vue pratique, la substitution doit bien être implantée d'une façon ou d'une autre, et d'autre part, d'un point de vue algorithmique, la règle *supra* ne peut pas servir de mesure élémentaire de complexité d'une réduction, car la substitution peut être aussi facile que difficile, suivant le nombre d'occurrences libres de x dans B .

Ainsi survient naturellement l'idée de décomposer la β -réduction en une réduction produisant une substitution « paresseuse », devant être ensuite propagée dans le terme au moyen d'un certain nombre de pas de réduction.

Différents calculs existent, dont beaucoup sont à indices de DE BRUIJN – dans ce cas, on explicite également en général la mise à jour des indices. On pourra citer, par exemple, λs , λt et $\lambda \sigma$. Un autre calcul s'inspirant des indices de DE BRUIJN est le calcul λv , que l'on va exposer brièvement ici.

Par ailleurs, il existe des calculs à noms à substitutions explicites. Le plus connu, sur lequel on s'appuiera beaucoup dans la suite, est le calcul λx , et sa variante λx^- , qui fera après λv l'objet d'une présentation détaillée.

Le λv -calcul

Ce calcul a été introduit par P. LESCANNE. Il ne s'agit pas d'un calcul à noms, mais il n'emploie pas exactement les indices de DE BRUIJN usuels. L'un des grands avantages de ce calcul est la limpidité de sa syntaxe et de sa sémantique. En outre, l'indispensable mise à jour des indices y est effectuée avec une grande simplicité. Le calcul présente de surcroît l'intérêt d'être un système de réécriture orthogonal, ce qui, sans rentrer dans des considérations de réécriture³ superflues ici, est une propriété intéressante dans ce cadre.

La syntaxe du calcul est la suivante :

$$\begin{aligned} A, B &::= \underline{n} \mid \lambda A \mid AB \mid A[s] \\ s &::= A/ \mid \uparrow(s) \mid \uparrow \\ n &\in \mathbb{N} \setminus \{0\} \end{aligned}$$

Tout comme dans les calculs à indices de DE BRUIJN, les variables sont représentées par des entiers, mais au contraire desdits calculs, l'indice 1 est lié non par le premier λ se trouvant au-dessus de lui, mais par le premier λ de l'arbre en partant de la racine.

³pour plus d'information sur les notions de réécriture, l'on pourra se reporter *e.g.* à [91, 3]

La sémantique du $\lambda\nu$ -calcul est donnée par :

(Bêta)	$(\lambda A)B$	\longrightarrow	$A[B/]$
(App)	$(AB)[s]$	\longrightarrow	$A[s]B[s]$
(Lambda)	$(\lambda A)[s]$	\longrightarrow	$\lambda A[\uparrow(s)]$
(F-Var)	$\underline{1}[A/]$	\longrightarrow	A
(R-Var)	$\underline{n+1}[A/]$	\longrightarrow	\underline{n}
(F-Lift)	$\underline{1}[\uparrow(s)]$	\longrightarrow	$\underline{1}$
(R-Lift)	$\underline{n+1}[\uparrow(s)]$	\longrightarrow	$\underline{n}[s][\uparrow]$
(Shift)	$\underline{n}[\uparrow]$	\longrightarrow	$\underline{n+1}$

On voit que le mécanisme de mise à jour des indices est intégré au calcul sous la forme de deux opérations simples, respectivement le *lifting* \uparrow et le *shifting* \uparrow , qui permettent de décaler les rangs d'indices de manière que la substitution soit correctement propagée. L'on peut montrer que ce système est confluent, et qu'il simule correctement la β -réduction.

Le $\lambda\mathbf{x}$ - et le $\lambda\mathbf{x}^-$ -calcul

Le $\lambda\mathbf{x}$ -calcul et le $\lambda\mathbf{x}^-$ -calcul s'appuient sur la même syntaxe, qui étend celle du λ -calcul par une nouvelle construction :

M, N	$::=$	x	<i>(variable)</i>
		$\lambda x.N$	<i>(abstraction)</i>
		MN	<i>(application)</i>
		$M\langle x \leftarrow N \rangle$	<i>(substitution)</i>

Intuitivement, cette nouvelle construction, $M\langle x \leftarrow N \rangle$, représente le terme M dans lequel on souhaite remplacer les occurrences de x par N , mais où cette substitution n'a pas encore été effectuée.

La notion de β -réduction est remplacée dans ces deux systèmes par la notion de $\beta\mathbf{x}$ -réduction, qui comprend un premier axiome représentant la contraction d'un β -rédex (dit B), et divers axiomes traitant des substitutions, formant une sous-notion de réduction dite \mathbf{x}^- -réduction :

$(\lambda x.A)B$	\xrightarrow{B}	$B\langle x \leftarrow A \rangle$
$(\lambda x.A)\langle y \leftarrow C \rangle$	\xrightarrow{x}	$\lambda x.(A\langle y \leftarrow C \rangle)$
$(A B)\langle y \leftarrow C \rangle$	\xrightarrow{x}	$(A\langle y \leftarrow C \rangle)(B\langle y \leftarrow C \rangle)$
$y\langle y \leftarrow C \rangle$	\xrightarrow{x}	C
$x\langle y \leftarrow C \rangle$	\xrightarrow{x}	x

(dans tous les axiomes de réduction *supra*, on suppose $x \neq y$).

La notion de réduction dans λx comprend un axiome supplémentaire :

$$M\langle y \leftarrow C \rangle \xrightarrow{x} M \quad \text{si } y \text{ n'est pas libre dans } M.$$

Cet axiome ne peut pas être simulé par les autres axiomes de la x -réduction, car il peut arriver qu'une substitution soit bloquée par une autre : ainsi, si y n'apparaît pas libre dans $A\langle x \leftarrow B \rangle$, la réduction $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{\beta x} A\langle x \leftarrow B \rangle$ est possible dans λx , mais dans λx^- , on ne peut pas propager la substitution $\langle y \leftarrow C \rangle$ à l'intérieur du terme $A\langle x \leftarrow B \rangle$. En effet, ceci requerrait un axiome de réduction dit de *composition* :

$$A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \longrightarrow A\langle y \leftarrow C \rangle\langle x \leftarrow B\langle y \leftarrow C \rangle \rangle,$$

mais en général, on écarte cet axiome, car il pose de nombreux problèmes, que l'on va aborder.

La question de la composition de substitutions

L'un des problèmes principaux soulevé par l'explicitation des substitutions est celui de leur composition. En effet, les substitutions implicites sont compositionnelles, *i.e.* :

$$A\{x \leftarrow B\}\{y \leftarrow C\} = A\{y \leftarrow C\}\{x \leftarrow B\{y \leftarrow C\}\}$$

ce qui est une propriété particulièrement intéressante, car elle permet de choisir l'ordre dans lequel les substitutions sont effectuées.

Dans le cas des substitutions explicites, on n'a naturellement pas égalité entre $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle$ et $A\langle y \leftarrow C \rangle\langle x \leftarrow B\langle y \leftarrow C \rangle \rangle$. Ces deux termes sont certes convertibles, mais lorsque l'on évalue l'un d'eux, on ne peut pas choisir l'ordre dans lequel on effectue les substitutions, et on ne peut pas décider de retarder l'une pour favoriser l'autre – ce qui peut d'ailleurs sembler contraire à la perception intuitive de la substitution explicite comme une opération « paresseuse ».

Les problèmes posés par l'introduction d'un axiome de réduction permettant d'effectuer cette composition sont multiples. D'une part, le calcul obtenu serait fortement divergeant, puisqu'un tel axiome change un rédex en un autre rédex. D'autre part, suivant les calculs, il fait apparaître d'autres difficultés, comme la non-confluence sur les termes ouverts, ou la non-préservation de la normalisation.

C'est pourquoi il faut en général se contenter d'une simple convertibilité entre $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle$ et $A\langle y \leftarrow C \rangle\langle x \leftarrow B\langle y \leftarrow C \rangle \rangle$. On verra par la suite que la composition soulève différents problèmes.

On peut ici mentionner certaines variantes de $\lambda\mathbf{x}$ introduites pour essayer de traiter un peu mieux ce problème de la composition, à savoir $\lambda\mathbf{x}\mathbf{c}$ et $\lambda\mathbf{x}//\mathbf{c}$ (cf. [11]). La syntaxe de $\lambda\mathbf{x}//\mathbf{c}$ est la suivante :

$$\begin{array}{ll} \mathbf{M}, \mathbf{M}_1, \dots, \mathbf{M}_k ::= & x \quad (\text{variable}) \\ & | \lambda x. \mathbf{M} \quad (\text{abstraction}) \\ & | \mathbf{M} \mathbf{M}_1 \quad (\text{application}) \\ & | \mathbf{M} \langle x_1, \dots, x_k \leftarrow \mathbf{M}_1, \dots, \mathbf{M}_k \rangle \quad (\text{substitution parallèle}) \end{array}$$

pour $k > 0$, les x_i étant deux à deux distincts.

Sa sémantique opérationnelle de réduction $\xrightarrow{\beta\mathbf{x}//\mathbf{c}}$ consiste en l'union de $\xrightarrow{\beta\mathbf{x}//}$ et de $\xrightarrow{//\mathbf{c}}$, respectivement définies comme les clôtures contextuelles des axiomes de réduction suivants :

$$\begin{array}{lll} (\lambda x. \mathbf{A}) \mathbf{B} & \xrightarrow{\mathbf{B}} & \mathbf{A} \langle x \leftarrow \mathbf{B} \rangle \\ (\lambda y. \mathbf{A}) \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle & \xrightarrow{\vec{x}//} & \lambda y. (\mathbf{A} \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle) \\ (\mathbf{A} \mathbf{B}) \langle x \leftarrow \mathbf{C} \rangle & \xrightarrow{\vec{x}//} & \mathbf{A} \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle \mathbf{B} \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle \\ x_i \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle & \xrightarrow{\vec{x}//} & \mathbf{M}_i \\ v \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle & \xrightarrow{\vec{x}//} & v \quad \text{si pour tout } i, v \neq x_i \\ \mathbf{A} \langle \vec{x} \leftarrow \tilde{\mathbf{M}} \rangle \langle \vec{y} \leftarrow \tilde{\mathbf{N}} \rangle & \xrightarrow{//\mathbf{c}} & \mathbf{A} \langle \vec{x}, \vec{y} \leftarrow \tilde{\mathbf{M}} \langle \vec{y} \leftarrow \tilde{\mathbf{N}} \rangle, \tilde{\mathbf{N}} \rangle \end{array}$$

Quant à $\lambda\mathbf{x}\mathbf{c}$, il s'agit d'une tentative pour débarasser $\lambda\mathbf{x}//\mathbf{c}$ de son parallélisme. Sa syntaxe est la même que celle de $\lambda\mathbf{x}$, et sa sémantique opérationnelle de réduction $\xrightarrow{\beta\mathbf{x}\mathbf{c}}$ est définie comme l'union de $\xrightarrow{\beta\mathbf{x}}$ et $\xrightarrow{\mathbf{c}}$, cette dernière étant définie comme la clôture contextuelle de l'axiome suivant :

$$\mathbf{A} \langle x \leftarrow \mathbf{X} \rangle \langle y \leftarrow \mathbf{Y} \rangle \xrightarrow{\mathbf{c}} \mathbf{A} \langle x \leftarrow \mathbf{X} \langle y \leftarrow \mathbf{Y} \rangle \rangle \quad \text{si } y \notin \text{fv}(\mathbf{A}).$$

Ces calculs ne préservent malheureusement pas la normalisation forte, puisque l'axiome de réduction $//\mathbf{C}$ (respectivement \mathbf{C}) transforme un rédex en un rédex. C'est pourquoi l'on préférera s'en tenir à $\lambda\mathbf{x}$, qui ne présente pas ce défaut.

Explicitation d'autres opérations

L'une des autres opérations que l'on peut songer à expliciter est l' α -réduction, afin de s'attaquer aux problèmes posés par cette opération, que l'on a présentés au paragraphe précédent. Comme l'a indiqué K.H. ROSE dans sa thèse [83], la substitution et l' α -conversion peuvent être explicitées indépendamment l'une de l'autre.

Si les calculs à indices de DE BRUIJN traitent symptomatiquement le problème de l' α -conversion implicite, ils demeurent imparfaits, du fait d'une part de leur lisibilité moindre (et c'est là un facteur qu'il faut prendre en compte, car même les assistants automatiques de démonstration supposent une interaction avec l'humain) et d'autre part du fait du lourd formalisme qu'il est nécessaire de mettre en œuvre pour la mise à jour des indices – lequel, si l'on veut l'explicitier, nécessite l'introduction d'un système de réécriture souvent complexe, et présente l'inconvénient de devoir effectuer des remises à jour d'indices lors du placement d'un terme au sein d'un contexte. Une solution réside peut-être dans les calculs comme $\lambda\nu$, que l'on a présenté plus haut, qui sont des calculs à indices, donc sans α -conversion implicite, mais restent toutefois simples et lisibles.

D'autres approches, où l'on explicite l' α -conversion dans un calcul à noms, commencent à apparaître ; on pourra entre autres se reporter principalement aux travaux de M. GABBAY ET A. PITTS, et aussi, plus récemment, de R. VESTERGAARD. Un grand intérêt de ces calculs est qu'ils donnent une idée plus précise de ce que l'on montre exactement dans le cadre habituel avec α -conversion implicite, et l'on peut ainsi se rendre compte que l'omniprésence de l' α -conversion est telle qu'il n'est certes pas superflu de se pencher sur le problème de son implantation.

Par ailleurs, on peut également songer à rendre explicite l' η -conversion (que l'on a introduite *supra*). Ce problème a été étudié par D. BRIAUD dans [15], mais est hors de propos ici.

1.3. Ordre récursif de chemin

Les ordres récursifs de chemin (en anglais, *Recursive Path Order*) sont une classe d'ordres bien fondés très utiles pour le raisonnement sur la normalisation forte. Introduits dans [29] par N. DERSCHOWITZ et développés dans [59] par S. KAMIN et J.-J. LÉVY, ils se définissent sur les termes de langages du premier ordre.

Avant de les définir, on rappelle deux classes d'ordres souvent utilisées pour comparer des suites finies d'éléments d'un même ensemble :

Définition 1.1 : *ordre lexicographique, ordre multi-ensemble*

Soit $(E, <)$ un ensemble partiellement strictement ordonné. L'on note E^* l'ensemble des suites finies d'éléments de E , et l'on désigne par ε la suite vide, par une virgule la concaténation de deux suites. En outre, si $\bar{w} \in E^*$, on dénote par $|\bar{w}|$ le nombre de ses éléments, et, pour $e \in E$, l'on dénote par e^p la suite de taille p dont chaque élément est égal à e (si $p = 0$, c'est ε ; sinon, $e^p = e, e^{p-1}$).

L'on définit sur E^* deux ordres stricts partiels, respectivement, l'*ordre lexicographique*, noté $<^{\text{lex}}$ et défini inductivement comme suit :

$$\frac{\bar{w} \neq \varepsilon}{\varepsilon <^{\text{lex}} \bar{w}} \quad \frac{a < b}{a, \bar{u} <^{\text{lex}} b, \bar{v}} \quad \frac{\bar{u} <^{\text{lex}} \bar{v}}{a, \bar{u} <^{\text{lex}} a, \bar{v}}$$

et l'*ordre multi-ensemble*, noté $<^{\text{mult}}$ et défini inductivement comme suit :

$$\frac{a < b \quad p \in \mathbb{N}}{\bar{u}, a^p, \bar{v} <^{\text{mult}} \bar{u}, b, \bar{v}} \quad \frac{\bar{u} <^{\text{mult}} \bar{v} \quad \sigma \in \mathfrak{S}_{|\bar{u}|}, \tau \in \mathfrak{S}_{|\bar{v}|}}{\sigma(\bar{u}) <^{\text{mult}} \tau(\bar{v})} \quad \frac{\bar{u} <^{\text{mult}} \bar{v} \quad \bar{v} <^{\text{mult}} \bar{w}}{\bar{u} <^{\text{mult}} \bar{w}}$$

où \mathfrak{S}_n désigne le groupe symétrique d'ordre n .

Proposition 1.1 : *Bonne fondation des ordres de suites finies*

Soit $(E, <)$ un ensemble partiellement strictement ordonné. Si $<$ est bien fondé, alors $<^{\text{lex}}$ et $<^{\text{mult}}$ sont bien fondés sur E^* .

Cette propriété étant notoire, on n'en donne pas ici de démonstration.

Définition 1.2 : *ordre récursif de chemin*

Soit Σ un ensemble (possiblement infini) de symboles de fonctions (d'arités diverses) nommé *signature*. On suppose que Σ est muni d'un ordre partiel \triangleright nommé *préséance* et d'une application totale $\zeta : \Sigma \rightarrow \{\text{lex}, \text{mult}\}$ nommée *statut*. On note \mathcal{T}_Σ l'algèbre des termes formés sur la signature Σ .

On définit alors les relations binaires $>_{\text{RPO}}$ et \geq_{RPO} inductivement comme suit :

$$\frac{}{s \geq_{\text{RPO}} s} \quad \frac{\exists i \in \llbracket 1, m \rrbracket (s_i \geq_{\text{RPO}} t)}{f(s_1, \dots, s_m) >_{\text{RPO}} t} \quad \frac{f \triangleright g \quad \forall j \in \llbracket 1, n \rrbracket (f(s_1, \dots, s_m) >_{\text{RPO}} t_j)}{f(s_1, \dots, s_m) >_{\text{RPO}} g(t_1, \dots, t_n)}$$

$$\frac{s >_{\text{RPO}} t \quad f = g \quad \forall j \in \llbracket 1, n \rrbracket (f(s_1, \dots, s_m) >_{\text{RPO}} t_j) \quad (s_1, \dots, s_m) >_{\text{RPO}}^{s(f)} (t_1, \dots, t_n)}{s \geq_{\text{RPO}} t \quad f(s_1, \dots, s_m) >_{\text{RPO}} g(t_1, \dots, t_n)}$$

Proposition 1.2 : *Bonne fondation de l'ordre récursif de chemin*

Soient Σ une signature et \triangleright une préséance sur Σ . Si \triangleright est bien fondé, alors $>_{\text{RPO}}$ est bien fondé sur \mathcal{T}_Σ .

Cette propriété étant notoire, on n'en donne pas ici de démonstration.

2. Λ -calcul typé

Le λ -calcul, qui met sur le même plan objets et fonctions, permet d'appliquer des fonctions à elles-mêmes. Ce type d'auto-référence est bien connu pour être

une source de paradoxes – l’un des plus célèbres étant le paradoxe du barbier, découvert en 1901 par B. RUSSELL.

Une méthode classique pour éviter ce genre de paradoxes est l’ajout de *types*, théorie que B. RUSSELL introduisit dans [84] en 1903, et développa dans les *Principia Mathematica* [85], ouvrage qu’il écrivit en 1910 avec A. WHITEHEAD, et considéré depuis, tant par les logiciens que par les philosophes, comme un ouvrage majeur dans le développement de la pensée logique.

Le λ -calcul a été, au cours de son histoire, muni de nombreux systèmes de types, que l’on peut essentiellement diviser en deux grandes familles : les systèmes à la CURRY et les systèmes à la CHURCH, ce qui fera l’objet du premier paragraphe. Le deuxième paragraphe traitera d’un résultat de théorie de la démonstration fondamental dans le cadre de la démonstration automatique, qui établit une correspondance entre typage et démonstration. Le troisième montrera une présentation des systèmes à la CHURCH les plus connus, où ceux-ci sont vus comme membres d’une hiérarchie cubique culminant à un système particulièrement riche : le calcul des constructions (cf. [20]). Le lecteur désireux de plus amples détails pourra consulter la présentation générale effectuée par H. BARENDREGT dans [6] à propos des différents systèmes de types existant pour le λ -calcul.

2.1. Typage à la CURRY ou à la CHURCH

On distingue habituellement deux façons de donner des types aux λ -termes : les systèmes de types à la CURRY et les systèmes de types à la CHURCH. La différence est d’ordre syntaxique : dans les systèmes à la CURRY, on se contente de donner à un terme un type dans un contexte assignant des types aux variables libres, tandis que dans les systèmes à la CHURCH, on modifie la syntaxe du λ -calcul en annotant les variables liées par une abstraction avec le type attendu.

Pour illustrer cette différence, on va présenter le système de types le plus élémentaire du λ -calcul, les *types simples* (cf. [16]), dans leurs deux versions :

$$\begin{array}{l} \sigma, \tau \quad ::= \quad \alpha \quad | \quad \sigma \rightarrow \tau \\ \\ \begin{array}{l} \text{M, N} \quad ::= \\ \quad | \\ \quad | \end{array} \quad \begin{array}{l} \text{[CURRY]} \\ x \\ \lambda x.M \\ MN \end{array} \quad \begin{array}{l} \text{[CHURCH]} \\ x \\ \lambda x:\sigma.M \\ MM' \end{array} \end{array}$$

où x décrit un ensemble \mathcal{X} de variables et α un ensemble Υ de variables de types (ces deux ensembles étant disjoints).

Les jugements de typages sont ensuite construits au moyen de *figures d'inférence*⁴. Ces figures sont dans chaque système au nombre de trois. Deux sont communes aux deux systèmes :

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma} \text{ (variable)} \quad \frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \text{ (application)}$$

tandis que la troisième est spécifique à chacun :

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x. M : \sigma \rightarrow \tau} \text{ (abstraction-CURRY)} \quad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x. \sigma. M : \sigma \rightarrow \tau} \text{ (abstraction-CHURCH)}.$$

Les systèmes à la CURRY présentent l'avantage de pouvoir donner plus de types aux termes représentant des fonctions polymorphes, tandis que les systèmes à la CHURCH ont pour atout que la vérification de type y est décidable.

Cependant, si les types simples, du fait de leur grande sobriété, peuvent se définir indifféremment dans les deux familles de systèmes, ce n'est pas le cas en général, du fait des particularités de certains systèmes.

Systèmes à la CURRY

Le plus usuel des systèmes n'existant *a priori* qu'en version de Curry est le système des *types à intersection*, dont la principale différence avec les types simples consiste en l'adjonction d'un connecteur binaire dans le langage des types :

$$\sigma, \tau \quad ::= \quad \alpha \mid \sigma \rightarrow \tau \mid \sigma \wedge \tau,$$

ledit connecteur s'interprétant comme une conjonction⁵ au moyen des figures de typage suivantes :

$$\frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \wedge \tau} \text{ } (\wedge\text{-I}) \quad \frac{\Gamma \vdash M : \sigma \wedge \tau}{\Gamma \vdash M : \sigma} \text{ } (\wedge\text{-É-1}) \quad \frac{\Gamma \vdash M : \sigma \wedge \tau}{\Gamma \vdash M : \tau} \text{ } (\wedge\text{-É-2})$$

Puisqu'il est possible de donner plusieurs types à un terme – par exemple, on peut dériver $\vdash \lambda x. x : (\alpha \rightarrow \alpha) \wedge (\alpha \rightarrow \alpha)$ – et que c'est là l'intérêt de ce système par rapport aux types simples, il appert qu'il serait dépourvu d'intérêt, sinon de sens, d'annoter les abstractions par le type de la variable abstraite, réduisant par là-même le système à un système dégénéré et proche des types simples (où simplement les variables pourraient avoir plusieurs types).

⁴On n'utilisera pas ici le vocabulaire plus usuel de *règles* d'inférence pour éviter la confusion avec une autre acception de ce terme, qui sera introduit par la suite pour désigner l'un des éléments caractéristiques des systèmes de types purs. Ce nom de figures est inspiré de G. GENTZEN, qui, dans ses travaux, emploie à cet usage le mot *Schlussfiguren*, que l'on traduit ainsi.

⁵du moins, en première approximation – cf. *infra*

Systèmes à la CHURCH

Les systèmes ne pouvant être définis que dans la version de CHURCH sont typiquement des systèmes incluant des types dits *dépendants* – *i.e.* des types pouvant dépendre de termes – ce qui est en particulier le cas du calcul des constructions.

Dans ces systèmes, il est impossible de tenir termes et types séparés : ils appartiennent à un même monde, et l'on ajoute à la syntaxe une construction permettant de construire des types produits (qui sont une généralisation de la flèche et d'une forme de quantification). Ainsi, les termes sont définis par la grammaire algébrique suivante :

$$\begin{array}{lcl}
 A, M, N & ::= & x \\
 & | & \lambda x:A.M \\
 & | & MN \\
 & | & \Pi x:A.M
 \end{array}$$

et le type $\Pi x:A.B$ représente le type des termes qui à un terme N de type A associent un terme de type $B\{x \leftarrow N\}$. Naturellement, si x ne survient pas dans B , ceci équivaut au type fonctionnel $A \rightarrow B$. Par ailleurs, suivant la forme de B , on voit bien qu'une forme de quantification sur x peut se produire.

Ces systèmes seront abordés plus particulièrement au § 2.2, lequel sera consacré au λ -cube.

Correspondance de CURRY-HOWARD(-DE BRUIJN)

La correspondance de CURRY-HOWARD-(DE BRUIJN) est un résultat fondamental de théorie des types, car elle établit le lien entre typage et démonstration : les types sont identifiés à des formules logiques, tandis qu'un terme est considéré comme une démonstration de son type.

Ainsi, si l'on considère les figures de typage données dans ce qui précède, on peut, en omettant les termes, reconnaître des règles de logique.

On peut ainsi établir un parallèle entre le typage d'une variable et l'emploi d'un axiome :

$$\overline{\Gamma, x : \sigma \vdash x : \sigma} \quad \leftrightarrow \quad \overline{\Gamma, \sigma \vdash \sigma}$$

ainsi qu'entre les types flèches et les applications :

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash M : \sigma \rightarrow \tau} \quad \Leftrightarrow \quad \frac{\Gamma, \sigma \vdash \tau}{\Gamma, \sigma \rightarrow \tau}$$

$$\frac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \quad \Leftrightarrow \quad \frac{\Gamma \vdash \sigma \rightarrow \tau \quad \Gamma \vdash \sigma}{\Gamma \vdash \sigma}.$$

Quant au connecteur \wedge des types à intersection, il s'interprète en première approximation comme une conjonction logique. Toutefois, cette interprétation n'est pas totalement exacte, comme l'a montré entre autres J. HINDLEY dans [57].

2.2. Le λ -cube

Le λ -cube, représenté sur la figure suivante :

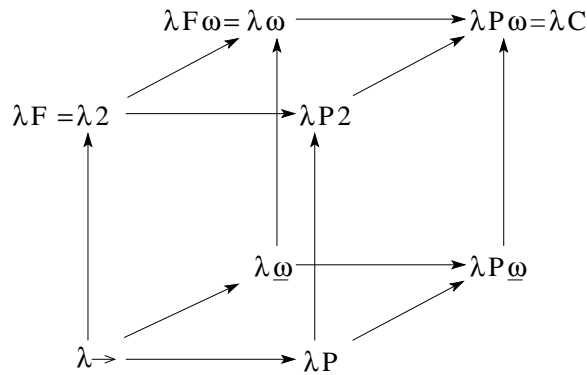


FIG. 2 – Le λ -cube de BARENDREGT

est une approche du calcul des constructions où celui-ci n'est plus vu comme une entité indivisible, mais comme un empilement de différentes sortes de dépendances, organisant différents sous-systèmes remarquables du calcul des constructions dans une hiérarchie tridimensionnelle formant un cube ayant à sa base le λ -calcul simplement typé et à son sommet le calcul des constructions, que l'on va donc ici présenter à travers le λ -cube, et non comme on peut le voir souvent comme un calcul monolithique.

Comme on le disait, à la base de ce cube se trouvent les types simples, qui contiennent l'espèce de dépendance le plus simple, auquel il convient d'ajouter trois autres espèces de dépendances pour parvenir au calcul des constructions.

Selon ce que l'on ajoute, on obtient les sous-systèmes remarquables suivants :

- le système $\lambda 2$, appelé λ -calcul à types du second ordre ou encore λ -calcul à types polymorphes, n'est en fait autre que le système F de J.-Y. GIRARD, et le système $\lambda\omega$ son système $F\omega$, tous deux datant de 1972 (cf. [45, 47]) ;
- le système λP peut être vu à peu près comme l'un des systèmes de la famille des langages AUTOMATH de N. DE BRUIJN, datant de 1980, ou comme le système LF de R. HARPER, F. HONSELL et G. PLOTKIN (cf. [52]) en 1987 ;
- le système $\lambda P2$ a été étudié sous ce nom en 1988 par G. LONGO et E. MOGGI dans [71] ;
- le système $\lambda P\omega$ s'apparente à un système introduit dans [82] en 1991 par G.R. RENARDEL DE LAVALETTE ;
- le système $\lambda\omega$ n'a guère de précédent historique.

et au sommet de tous ces systèmes culmine le calcul complet, combinant les diverses constructions de chacun : le calcul des constructions, étudié en 1988 par T. COQUAND et G. HUET dans [20].

Les systèmes $\lambda\rightarrow$ et $\lambda 2$ peuvent aussi être donnés en leur version à la CURRY, et il existe dans [44] une version à la CURRY par P. GIANNINI et S. RONCHI DELLA ROCCA de $\lambda\omega$, datant de 1988. S'il semble également possible de donner des versions à la CURRY de $\lambda\omega$, en revanche, il paraît nécessaire d'avoir un typage à la CHURCH dès l'instant où l'on a le type de dépendances introduit dans λP et, naturellement, dans les calculs le contenant, *i.e.* $\lambda P2$, $\lambda P\omega$, et le calcul des constructions.

Les quatre espèces de dépendances dont on vient de parler sont les suivantes :

- les termes dépendant de termes ;
- les termes dépendant de types ;
- les types dépendant de types ;
- les types dépendant de termes.

Les termes dépendant de termes sont à la base du système de typage ; c'est l'unique espèce de dépendance existant dans $\lambda\rightarrow$, et on la trouve dans tous les systèmes. Ainsi, lorsque l'on déduit de $F : \alpha\rightarrow\beta$ et $M : \alpha$ que $FM : \beta$, on a construit un terme dépendant d'un terme : le terme FM dépend du terme M .

Les termes dépendant de types se trouvent dans le système F (ou $\lambda 2$) : ainsi, dans ce système, on peut construire des termes de la sorte : $G : \forall\alpha(\tau)$. Ces termes vérifient que lorsqu'on les applique à un type A, le terme obtenu est $GA : \tau\{\alpha\leftarrow A\}$, et l'on peut ainsi construire des termes dépendant de types : le terme GA dépend du type A.

Les types dépendant de types sont une idée assez naturelle survenant en observant qu'il existe une relation de type fonctionnel entre types : ainsi, $\alpha\rightarrow\alpha$ et $\beta\rightarrow\beta$

semblent être l'image par une même fonction de, respectivement, α et β . Pour pouvoir formaliser ce type de dépendance, il suffit d'introduire des fonctions pouvant abstraire non plus des variables de termes mais aussi des variables de types, ce que fait le système $\lambda\omega$. Dans de tels systèmes, les types ne sont plus de simples constructions d'un méta-langage informel, mais bel et bien des constructions du système lui-même, et l'on dispose d'un type spécial (que l'on appellera par la suite une *sorte*), noté $*$, pour les reconnaître. Les fonctions manipulant ces types sont appelées *constructeurs*, et l'on peut leur donner des « types » induits par la grammaire $K ::= * \mid K \rightarrow K$, qui sont identifiés dans le système au moyen d'un nouveau symbole de constante, noté \square . Pour revenir au petit exemple *supra*, la fonction serait $f = \lambda\alpha:*. \alpha \rightarrow \alpha$, et $f\beta$ serait un type dépendant du type β .

Enfin, les types dépendant de termes sont en fait intuitivement assez faciles à saisir, mais sont toutefois difficiles à formaliser. Un exemple simple est le type $A^N \rightarrow B$, où A et B sont des types et N est un terme représentant un entier naturel. On introduit cette possibilité en étendant la grammaire vue *supra* pour les « types » de constructeurs par $K ::= * \mid K \rightarrow K \mid N \rightarrow K$, où N est un terme. Si l'on a ensuite $F : A \rightarrow *$, et $N : A$ un terme de type A , on obtient donc en $FN : *$ un type dépendant du terme N . Cette dépendance est liée à l'idée de produit cartésien pour les types : en effet, si pour tout terme $a : A$ on peut former un type B_a , on peut vouloir créer la fonction $\lambda a: A. B_a$. Cette fonction devrait avoir pour type le produit cartésien de tous les B_a . Cette notion permet d'étendre le cas des types \rightarrow des termes dépendant de termes : en effet, on peut les considérer comme un produit cartésien où B ne dépendrait pas de a . Cette présentation intuitive des types dépendants ayant déjà été abordée dans l'introduction, l'on ne s'y étendra pas davantage.

2.3. Types pour les calculs à substitutions explicites

Une fois le calcul étendu par les substitutions explicites, il est intéressant d'étendre également le système de types pour traiter lesdites substitutions. Pour donner ici une idée de la manière dont cela est fait, l'on va employer la syntaxe du λx -calcul pour lui étendre les types simples.

Pour des raisons de cohérence, on s'attend à ce que le terme $A\langle x \leftarrow B \rangle$ ait le même type que le terme βx -équivalent $(\lambda x.A)B$. Or, ce terme se type comme suit :

$$\frac{\frac{\Gamma, x : \sigma \vdash A : \tau}{\Gamma \vdash \lambda x.A : \sigma \rightarrow \tau} \text{ (abs)} \quad \Gamma \vdash B : \sigma}{\Gamma \vdash (\lambda x.A)B : \tau} \text{ (app)}$$

et il semble naturel d'introduire en conséquence la figure de typage suivante pour typer les termes commençant par une substitution explicite :

$$\frac{\Gamma, x : \sigma \vdash A : \tau \quad \Gamma \vdash B : \sigma}{\Gamma \vdash (\lambda x.A)B : \tau} \text{ (substitution).}$$

Si l'on essaye de voir à quoi ce typage est associé *via* la correspondance de CURRY-HOWARD(-DE BRUIJN), on peut observer qu'en ôtant la décoration syntaxique, il se révèle être essentiellement la règle de logique suivante :

$$\frac{\Gamma, \sigma \vdash \tau \quad \Gamma \vdash \sigma}{\Gamma \vdash \tau}$$

en laquelle on reconnaît une coupure en déduction naturelle.

De nombreux travaux sont présents dans la littérature à propos de l'adjonction de la coupure aux systèmes de types du λ -calcul en relation à l'adjonction à la syntaxe de la substitution explicite. On pourra entre au citer principalement les articles [31] de R. DI COSMO et D. KESNER et [32] de R. DI COSMO, D. KESNER et E. POLONOVSKI, ainsi que les travaux de R. VERSTERGAARD ET J. WELLS dans [96] ou encore de H. HERBELIN dans [54]. Ces approches considèrent toutes des parties du λ -cube, et, pour la plupart, se placent dans le cadre des calculs à indices de DE BRUIJN. On peut également citer [77, 78, 7], C. MUÑOZ ET N. BJØRNER introduisent le système $\lambda\Pi_{\mathcal{L}}$, qui étend les types dépendants à un calcul de substitutions explicites à indices de DE BRUIJN. Par ailleurs, dans [30], S. LENGAND, P. LESCANNE, D. DOUGHERTY, M. DEZANI-CIANCAGLINI et S. VAN BAKEL étendent les types à intersection à une variante du λx -calcul.

On peut par ailleurs mentionner que la coupure n'est pas seulement une adjonction que l'on souhaite faire *ad hoc* aux systèmes de types purs pour prendre en charge la substitution explicite, elle présente un intérêt *per se*, et l'on peut d'ailleurs citer à cet effet [50], où F. GUTIÉRREZ ET B. RUIZ proposent un calcul avec coupure dans le cadre des systèmes de types purs, mais leur point de vue est ici quelque peu éloigné du nôtre, car leur calcul n'est pas à substitutions explicites, mais un calcul de séquents.

D'autres approches seront également exposées dans un historique présenté dans le chapitre suivant.

3. Systèmes de types purs

S. BERARDI ET J. TERLOUW ont, indépendamment l'un de l'autre, fourni dans leurs travaux en 1989 des méthodes générales permettant d'engendrer de manière systématique des systèmes de types à la CHURCH pour le λ -calcul. Ces méthodes ont abouti aux *systèmes de types purs*, qui sont un formalisme simple et élégant permettant de décrire de nombreux systèmes de types, parmi lesquels en particulier ceux dits du λ -cube, cette décomposition hiérarchique du calcul des constructions que l'on vient de présenter. Comme on l'a mentionné, cette décomposition a été introduite par H. BARENDREGT, lequel a donné ultérieurement dans [6] une présentation formelle et systématique des systèmes de types purs, ainsi que de leurs principales propriétés.

Dans cette partie, le lecteur trouvera une définition des systèmes de types purs, ainsi que la manière d'y construire le λ -cube, et enfin, les propriétés les plus remarquables de ces systèmes.

3.1. Définition et caractéristiques

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

Définition 1.3 : *système de types pur ; sorte, axiome, règle*

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

Les sortes d'un système de types pur sont en fait ses types « élémentaires », tandis que les axiomes indiquent les relations existant entre ces sortes. Ainsi, on peut distinguer deux catégories de sortes :

Définition 1.4 : *sortes typables/terminales*

Soit $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ un système de types pur. Soit σ une sorte de \mathfrak{T} . Cette sorte est dite *terminale* si $\{\sigma\} \times \mathcal{S} \cap \mathcal{A} = \emptyset$. Dans le cas contraire (*i.e.* si σ apparaît comme sujet dans un axiome), elle est dite *typable*.

La définition 1.3 caractérise un système de types par ses types élémentaires, les sortes, les relations existant entre ces types, les axiomes, et enfin des règles qui prendront leur sens plus tard puisqu'elles intéressent le calcul, en imposant des contraintes sur les dépendances autorisées dans les types composés.

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

Définition 1.5 : \mathcal{T} -expressions

Soient $\mathcal{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ un système de types pur, et \mathcal{U} un ensemble infini de variables. On définit alors l'ensemble $\mathcal{E}(\mathcal{T})$ des \mathcal{T} -expressions par la grammaire algébrique suivante, où les symboles σ et x décrivent respectivement les ensembles des sortes et des variables ($\sigma \in \mathcal{S}; x \in \mathcal{U}$) :

E ::=	σ	(sorte)
	x	(variable)
	E E	(application)
	$\lambda x:E.E$	(abstraction)
	$\Pi x:E.E$	(quantification)

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

Définition 1.6 : variables libres/liées pour les systèmes de types purs implicites

Soit \mathcal{T} un système de types pur. Soit M une \mathcal{T} -expression.

On définit l'ensemble $fv(M)$ des *variables libres* de M inductivement comme suit :

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

De façon analogue, on définit l'ensemble $bv(M)$ des *variables liées* de M :

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

On définit également la classique α -conversion :

Définition 1.7 : α -conversion

Soit \mathcal{T} un système de types pur. L' α -conversion \equiv_α est définie sur $(\mathcal{E}(\mathcal{T}))^2$ par :

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

Il est aisé de vérifier que \equiv_α est une relation de congruence sur les \mathcal{T} -expressions.

L' α -conversion étant une congruence, on peut considérer les expressions à α -conversion près – on prend en effet ce parti plutôt que celui de la considérer informellement comme une opération administrative à effectuer à la volée. On raisonne alors non plus sur les expressions, mais sur leurs classes d'équivalence modulo α -conversion, que l'on appelle les *termes* du calcul :

Définition 1.8 : $\lambda\mathfrak{T}$ -calcul

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

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

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

Définition 1.9 : β -réduction, β -conversion

Soit \mathfrak{T} un système de types pur. La β -réduction $\xrightarrow{\beta}$ est la relation de réduction sur les \mathfrak{T} -expressions induite par la règle :

$$(\beta) \quad (\lambda x:A.B)C \longrightarrow B\{x \leftarrow C\}$$

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

Les notions *supra* constituent l'aspect calculatoire des systèmes de types purs. Sur cela vient se greffer un aspect logique, par le biais de jugements de typage, qui peuvent être interprétés *via* la correspondance de CURRY-HOWARD(-DE BRUIJN) comme des assertions logiques. Le typage s'effectue par le moyen suivant :

Définition 1.10 : *assertion, contexte, jugement (valide) de typage*

Soit \mathfrak{T} un système de types pur.

On appelle *assertion de typage* un couple de \mathfrak{T} -expressions noté $S : P$, dont les élément S et P sont appelé respectivement son *sujet* et son *prédictat*.

On appelle *contexte de typage* une suite finie⁶ d'assertions de types dont les sujets sont des variables distinctes. L'ensemble de ces variables est appelé le *domaine* du contexte, et leur suite ordonnée, son *support*. Le contexte vide est noté $[]$, et la concaténation de contextes de domaines disjoints est notée par une virgule.

On appelle *jugement de typage* une expression de la forme $\Gamma \vdash_{\mathfrak{Z}} M : N$, où Γ est un contexte de typage et $M : N$ une assertion de typage. Un jugement est dit *valide* s'il dérive de l'application des figures⁷ d'inférence présentées dans la table 1 *supra*. L'indice \mathfrak{Z} peut être omis s'il n'y a pas d'ambiguïté.

$$\begin{array}{c}
\frac{(\sigma, \tau) \in \mathcal{A}}{\vdash \sigma : \tau} \text{ (axiome)} \\
\\
\frac{\Gamma \vdash A : \rho \quad \Gamma, x : A \vdash B : \sigma \quad (\rho, \sigma, \tau) \in \mathcal{R}}{\Gamma \vdash \Pi x:A.B : \tau} \text{ (règle)} \\
\\
\frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A} \text{ (hypothèse)} \\
\\
\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B} \text{ (affaiblissement)} \\
\\
\frac{\Gamma \vdash (\Pi x:A.B) : \sigma \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x:A.M : (\Pi x:A.B)} \text{ (\Pi - introduction)} \\
\\
\frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\{x \leftarrow N\}} \text{ (\Pi - élimination)} \\
\\
\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \sigma \quad A \equiv_{\beta} B}{\Gamma \vdash M : B} \text{ (conversion)}
\end{array}$$

TAB. 1 – *Figures d'inférence valides pour les systèmes de types purs (implicites)*

Définition 1.11 : *contexte bien formé*

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

⁶et non un ensemble : comme on l'a déjà expliqué, le fait que termes et types ne soient pas des catégories syntaxiques distinctes engendre des dépendances entre variables, ce qui impose que les contextes de typage soient ordonnés

⁷comme indiqué dans la note p. 24, on parle de figures plutôt que de règles, ce dernier terme étant réservé aux règles (triplets de sortes) d'un système de types pur

3.2. Propriétés générales

On va maintenant rappeler diverses propriétés générales des systèmes de types purs, desquelles on ne donnera pas de démonstrations, car celles-ci peuvent être trouvées dans [6], auquel pourra se référer le lecteur désireux de les consulter.

Lemme 1.3 : *Initialisation pour les systèmes de types purs implicites*

Soient Γ un contexte bien formé et $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ un système de types pur.

- (i) Si $(\sigma : \tau) \in \mathcal{A}$, alors $\Gamma \vdash \sigma : \tau$.
- (ii) Si $(u : V) \in \Gamma$, alors $\Gamma \vdash u : V$.

Lemme 1.4 : *Engendrement pour les systèmes de types purs implicites*

Soient Ξ un contexte et M, T deux $\lambda\mathfrak{T}$ -termes tels que $\Xi \vdash M : T$. Alors :

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

Ce lemme est fort utile car il permet la reconstruction de jugements de typage à partir de leur conclusion.

Lemme 1.5 : *Substitution pour les systèmes de types purs implicites*

Soit \mathfrak{T} un système de types pur.

Soient $\Gamma, x : A, \Delta \vdash_{\mathfrak{T}} M : B$ un jugement de typage valide et N un \mathfrak{T} -terme tels que $\Gamma \vdash_{\mathfrak{T}} N : A$ soit un jugement de typage valide.

Alors le jugement de typage $\Gamma, \Delta\{x \leftarrow N\} \vdash_{\mathfrak{T}} M\{x \leftarrow N\} : B\{x \leftarrow N\}$ est valide.

Du point de vue des substitutions explicites, ce lemme semble d'une importance cruciale, car il touche vraiment à la partie délicate de l'explicitation, pour des raisons que l'on va exposer.

D'une part, il est fondamental dans la démonstration de la réduction du sujet pour les systèmes de types purs – il permet en effet d'intervertir l'ordre dans lequel

sont effectuées deux coupures. On peut remarquer d'ailleurs un rapport net avec le problème de composition des substitutions exposé à la partie précédente. En effet, ce lemme permet, étant donné un jugement de la forme $\Gamma, x : U, y : V \vdash M : W$, de choisir si l'on effectue d'abord une coupure sur x puis sur y , ce qui produira un jugement de la forme $\Gamma \vdash M\{x \leftarrow X\}\{y \leftarrow Y\} : T$, ou inversement, ce qui produira un jugement de la forme $\Gamma \vdash M\{y \leftarrow Y\}\{x \leftarrow X\{y \leftarrow Y\}\} : T$. On voit que la différence entre les deux correspond à une composition.

D'autre part, l'énoncé du lemme laisse penser qu'il sera difficile à retranscrire avec des substitutions explicites, car la réduction du sujet est nécessaire pour effectuer des opérations d'interversion de substitutions indispensables pour raisonner inductivement, puisque, comme on l'a indiqué, l'on n'a à disposition que la x -convertibilité pour résoudre le problème de la composition.

On se retrouve donc ici avec une apparente interdépendance entre la substitution et la réduction du sujet, ce qui laisse présager de difficultés dans l'adaptation de ce résultat. Ce point sera abordé plus longuement dans le chapitre II consacré aux systèmes de types purs avec substitutions explicites.

Théorème 1.6 : *Correction pour les systèmes de types purs implicites*

Soit \mathfrak{T} un système de types pur.

Soient Γ un contexte de typage et A, B deux \mathfrak{T} -termes tels que le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide.

Alors B est une sorte ou il existe une sorte σ telle que $\Gamma \vdash_{\mathfrak{T}} B : \sigma$ soit valide.

Ce théorème, qui se démontre aisément par une induction sur le jugement, est de première importance, car il garantit en fait que les types apparaissant dans un jugement valide non seulement sont de la forme voulue, *i.e.* des sortes (types élémentaires) ou des quantifications (types composés), mais également qu'ils sont d'une certaine façon conformes à la spécification fournie dans le système de types pur \mathfrak{T} : en effet, intuitivement, ce théorème garantit qu'un type composé a nécessairement été introduit par l'emploi de la figure d'inférence (règle), où la prémisse $(\rho, \sigma, \tau) \in \mathcal{R}$ impose cette conformité aux spécifications du système.

L'adaptation de ce théorème dans le cas des substitutions explicites sera difficile, pour diverses raisons déjà envisagées par R. BLOO dans sa thèse de doctorat [9], sur lesquelles on reviendra en détail dans le chapitre II consacré aux systèmes à substitutions explicites. On verra en particulier la nécessité d'approfondir cette notion de type « correct », qui intervient par ailleurs souvent comme prémisse dans les figures d'inférence de typage.

Le corollaire suivant du théorème de correction des types découle directement de la définition des sortes terminales (1.4) :

Corollaire :

Soit \mathfrak{T} un système de types pur.

Soient Γ un contexte de typage et A, B deux \mathfrak{T} -termes tels que le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide.

Alors soit B est une sorte terminale, soit il existe une sorte σ telle que $\Gamma \vdash_{\mathfrak{T}} B : \sigma$ soit valide.

Ce corollaire peut sembler n'être qu'une simple paraphrase du théorème de correction des types (cf. tc-ipts). Toutefois, il établit une distinction fondamentale en isolant le cas problématique des sortes terminales. En effet, ainsi que l'indiquait déjà R. BLOO dans [9], c'est ce cas particulier qui, seul, posera problème pour établir les conditions de correction des types dans le cas des substitutions explicites, comme on le verra plus tard.

Théorème 1.7 : *Réduction du sujet pour les systèmes de types purs implicites*

Soit \mathfrak{T} un système de types pur.

Soient Γ un contexte de typage et A, A', B trois \mathfrak{T} -termes tels que le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide et $A \xrightarrow{\beta} A'$.

Alors le jugement de typage $\Gamma \vdash_{\mathfrak{T}} A' : B$ est valide.

Le théorème précédent est une propriété fondamentale des systèmes de types. En effet, les propriétés telles que le type d'un terme semblent devoir être indépendante de l'évolution calculatoire de ce terme – lorsqu'en programmation une expression est de type entier, on s'attend raisonnablement à ce qu'elle ne devienne pas une fonction sur les booléens au cours de son évaluation.

Cette propriété sera difficile à obtenir dans le cas des substitutions explicites ; en particulier, on verra que dans certains formalismes proposés auparavant pour étendre les systèmes de types purs aux calculs à substitutions explicites, la réduction du sujet n'est pas satisfaite.

Cette difficulté est, comme on l'a mentionné plus haut, liée à la difficulté d'établir le lemme de substitution dans le cas explicite, du fait que le passage aux substitutions explicites transforme une égalité syntaxique entre termes de même forme \mathbf{x} -normale en une simple $\beta\mathbf{x}$ -convertibilité, ce qui complique considérablement le passage à l'induction.

Ce point sera largement discuté dans le chapitre II consacré aux systèmes de types purs à substitutions explicites.

3.3. Construction du λ -cube en systèmes de types purs

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

Définition 1.12 : *système de types pur élémentaire (complet)*

Étant donné un ensemble non vide de sortes \mathcal{S} , on appelle *règle élémentaire* tout triplet de la forme (σ, τ, τ) , ce que l'on dénote en abrégé par $[\sigma, \tau]$. L'ensemble des règles élémentaires formées avec les éléments de \mathcal{S} se note $\mathcal{R}_{\mathcal{S}}^e$.

Un système de types pur $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ est dit *élémentaire* si $\mathcal{R} \subseteq \mathcal{R}_{\mathcal{S}}^e$ et *élémentaire complet* si $\mathcal{R} = \mathcal{R}_{\mathcal{S}}^e$.

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

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

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

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

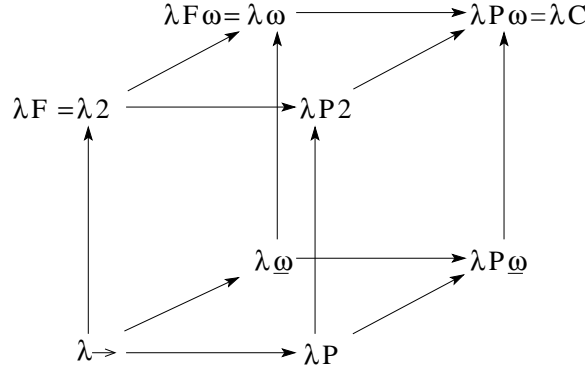


FIG. 3 – Le λ -cube de BARENDREGT

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

Comme on l'a mentionné précédemment dans les systèmes de types purs, chaque sorte caractérise une espèce de types. Or, le calcul des constructions, qui permet la construction de tout genre de dépendance envisageable entre termes et types, ne sépare pas les expressions à valeur de termes et les expressions à valeur de types, qui sont toutes des \mathfrak{T} -expressions.

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

Par ailleurs, chaque règle élémentaire permet de construire une espèce de dépendance entre termes et types :

- $(*, *, *)$ permet le typage de termes dépendant de termes ;
- $(\square, *, *)$ permet le typage de types dépendant de termes ;
- $(\square, \square, \square)$ permet le typage de types dépendant de types ;
- $(*, \square, \square)$ permet le typage de termes dépendant de types.

De manière plus détaillée, le λ -cube des systèmes de types purs se compose des systèmes suivants :

- $\lambda \rightarrow = (\{*, \square\}, \{* : \square\}, \{(*, *, *)\})$;
- $\lambda \underline{\omega} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (\square, \square, \square)\})$;
- $\lambda \underline{2} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (\square, *, *)\})$;
- $\lambda \underline{P} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (*, \square, \square)\})$;
- $\lambda \underline{\omega} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (\square, *, *), (\square, \square, \square)\})$;
- $\lambda \underline{P2} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (\square, *, *), (*, \square, \square)\})$;
- $\lambda \underline{P\underline{\omega}} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (\square, \square, \square), (*, \square, \square)\})$;
- $\lambda \underline{P\underline{\omega}} = (\{*, \square\}, \{* : \square\}, \{(*, *, *), (\square, *, *), (\square, \square, \square), (*, \square, \square)\})$.

Il est facile de voir cela lorsque l'on regarde la figure d'inférence (règle) :

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

Si l'on considère les systèmes de types en question dans leur présentation classique, on se rend compte que le quantificateur Π représente diverses constructions selon la règle à laquelle il est associé. Ainsi, avec la règle $(*, *, *)$ seule, on ne peut construire que les types équivalant à un type \rightarrow (dans $(\Pi x:A^*.B^*) : *$, B ne peut, du fait du jeu des sortes, pas contenir d'occurrence réelle de x , et cette expression équivaut $A \rightarrow B$.) De même, la règle $(*, \square, \square)$ permet par exemple la

construction de types comme $\Pi x:*\cdot\Box.x^*$, ce qui représente une quantification universelle comme dans le système F de Girard. La règle (\Box, \Box, \Box) permet d'écrire des fonctions sur des types, de manière très analogue aux fonctions sur les termes que permet d'écrire $(*, *, *)$. Quant à la règle $(*, \Box, \Box)$, c'est elle qui permet d'introduire des termes dépendants de types.

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

Définition 1.13 : *normalisation forte*

Un système de types pur \mathfrak{T} est dit *fortement normalisant* s'il satisfait la propriété énoncée ici :

soient Γ un contexte et $A : B$ une assertion tels que que le jugement $\Gamma \vdash_{\mathfrak{T}} A : B$ soit valide. Alors ni A ni B n'admet de chaîne infinie de β -réduction.

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

Cette propriété de normalisation forte, peut, tout comme celle de réduction du sujet, être considérée comme fondamentale, en particulier dans une optique de programmation, où l'on pourrait la formuler comme suit : l'évaluation d'un programme correct (*i.e.* d'un terme typable) doit s'achever en un temps fini.

On verra plus tard que ceci pose également problème dans le cas explicite, car dans ce formalisme apparaîtront nécessairement des termes analogues à des sortes mais réductibles et potentiellement non fortement normalisants. Ce point également sera évoqué plus tard.

Théorème 1.8 : *Normalisation forte dans les s.t.p. implicites du λ -cube*

Les systèmes de types purs du λ -cube ($\lambda \rightarrow$, $\lambda_{\underline{\omega}}$, $\lambda 2$, λP , $\lambda \omega$, $\lambda P 2$, $\lambda P_{\underline{\omega}}$, $\lambda P \omega$) sont fortement normalisants.

CHAPITRE II

Systemes de types purs explicites

— SOMMAIRE —

1. Définition	43
1.1. Historique	43
1.2. Syntaxe et réduction	44
1.3. Typage	59
2. Particularités remarquables	72
2.1. La figure d'inférence d'expansion	72
2.2. Le prédicat « pseudo-sortie »	79
2.3. Ordre sur les dérivations	85
3. Propriétés	88
3.1. Lemmes de dérivation	88
3.2. Réduction du sujet	95
3.3. Conservation de la normalisation forte	123

1. Définition

1.1. Historique

Comme on l’a déjà mentionné plus tôt, divers travaux considèrent l’extension de diverses parties du λ -cube, pour la plupart dans des calculs à indices de De Bruijn. On pourra citer principalement les articles [31] de R. DI COSMO et D. KESNER et [32] de R. DI COSMO, D. KESNER et E. POLONOVSKI, respectivement en 1997 et 2003 ainsi que les travaux de R. VERSTERGAARD ET J. WELLS dans [96] en 2001 ou encore de H. HERBELIN en 2001 dans [54], ainsi que [77, 78, 7], où en 2000-2001, C. MUÑOZ et N. BJØRNER introduisent le système $\lambda\Pi_{\mathcal{L}}$, qui étend les types dépendants à un calcul de substitutions explicites à indices de De Bruijn.

Une autre approche au problème de la coupure pour les systèmes de types purs peut être trouvée dans l’article [12] de 1996 de R. BLOO, F. KAMAREDDINE et R. NEDERPELT, ou encore dans l’article [87] de 1994 de P. SEVERI et E. POLL. Ces travaux considèrent des systèmes de types purs avec des *définitions*. Les définitions ressemblent aux substitutions en ce sens qu’elles consistent également en l’assignation d’un terme à une variable, mais au contraire de celles-ci, elles sont placées dans le contexte de typage, et, comme elles ne font, par conséquent, pas partie de la syntaxe des termes, elles sont assez éloignées des substitutions explicites.

Enfin, il convient de mentionner divers autres travaux considérant des substitutions explicites. M. STRECKER, dans sa thèse de doctorat [89] de 1999, considère également des substitutions, mais attachées uniquement à des méta-variables, car, comme il l’indique, « le calcul des termes devient plus complexe si la substitution explicite peut survenir dans des positions arbitraires dans les termes, et ne sont pas attachées à des méta-variables ». Quant à L. MAGNUSSON, elle introduit en 1995 dans sa thèse de doctorat [72] un calcul de substitutions explicites dans la théorie des types de Martin-Löf. En 1999, M.O. STEHR ET J. MESEGUER introduisent une forme de systèmes de types purs avec substitutions explicites, mais dans le cadre de la *membership equational logic*.

Les travaux les plus proches de ceux qui vont être présentés ici sont ceux de R. BLOO, qui dans sa thèse [9] en 1997 introduit des systèmes de types purs basés sur le $\lambda\mathbf{x}$ -calcul, lesquels sont ensuite développés dans son article [10] en 2001. Comme on l’a dit, ces systèmes sont basés sur la syntaxe du $\lambda\mathbf{x}$ -calcul, présenté au chapitre précédent. Le lecteur pourra se reporter au § 1.3 pour de plus amples précisions.

Les figures de typages de R. BLOO sont présentées dans la table 2 *infra*.

$$\begin{array}{c}
\frac{(\sigma, \tau) \in \mathcal{A}}{\vdash \sigma : \tau} \text{ (axiome)} \\
\\
\frac{\Gamma \vdash A : \rho \quad \Gamma, x : A \vdash B : \sigma \quad (\rho, \sigma, \tau) \in \mathcal{R}}{\Gamma \vdash \Pi x:A.B : \tau} \text{ (règle)} \\
\\
\frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A} \text{ (hypothèse)} \\
\\
\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B} \text{ (affaiblissement)} \\
\\
\frac{\Gamma \vdash (\Pi x:A.B) : \sigma \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x:A.M : (\Pi x:A.B)} \text{ (\Pi - introduction)} \\
\\
\frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\{x \leftarrow N\}} \text{ (\Pi - élimination)} \\
\\
\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\{x \leftarrow N\}} \text{ (coupure)} \\
\\
\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \sigma \quad A \equiv_{\beta} B}{\Gamma \vdash M : B} \text{ (conversion)}
\end{array}$$

TAB. 2 – Figures d'inférence de type valides pour les systèmes de types purs explicites de R. BLOO

On peut noter en particulier que ce système traite de façon différente les substitutions survenant en sujet ou en prédicat dans les assertions de typage : elles sont explicites en partie gauche et implicites en partie droite, ce qui peut sembler contestable. En outre, ce système présente le problème majeur de ne pas vérifier la réduction du sujet, point sur lequel on reviendra plus tard.

1.2. Syntaxe et réduction

La définition formelle basique des systèmes de types purs reste inchangée : par système de types pur explicite, on entend toujours un triplet $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ de sortes, axiomes et règles, qui constitue en fait la spécification du système, à laquelle vont

s'ajouter un calcul sous-jacent et un système d'inférence de types, qui, eux, seront distincts de ceux des systèmes de types purs pour calculs à substitutions implicites.

Le fait que cette définition reste inchangée permettra par la suite d'établir une correspondance canonique entre un système de types pur explicite d'une part et, d'autre part, le système correspondant – *i.e.* ayant la même spécification formelle – dans les systèmes usuels (implicites).

Définition 2.1 : $\lambda\mathcal{T}\mathbf{x}$ -expressions

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

$E ::=$	σ	(sorte)
	x	(variable)
	γ_x	(γ -constante)
	EE	(application)
	$\lambda x:E.E$	(abstraction)
	$\Pi x:E.E$	(quantification)
	$E\langle x \leftarrow E \rangle$	(substitution)

où σ et x décrivent respectivement les ensembles des sortes et des variables (*i.e.* : $\sigma \in \mathcal{S}$; $x \in \mathcal{U}$). Étant donné un terme M , on désigne par $ov(M)$ l'ensemble des variables occurrant dans M .

Les éléments de la forme γ_x peuvent être vus comme des « variables insubstituables », mais, d'un point de vue de réécriture, il s'agit de constantes. Leur introduction est nécessaire pour pouvoir considérer le calcul de réduction associé comme un système de réécriture. Ce point sera abordé plus loin.

On peut, selon la manière usuelle, définir les notions de sous-terme et de sous-terme propre :

Définition 2.2 : *sous-terme (propre)*

On introduit deux notions, \subseteq et \subset , respectivement « être un sous-terme de » et « être un sous-terme propre de », conjointement comme suit :

$\overline{M \subseteq M}$		$\frac{M \subset N}{\overline{M \subseteq N}}$	
$\frac{M \subseteq A}{\overline{M \subset \Pi x:A.B}}$	$\frac{M \subseteq B}{\overline{M \subset \Pi x:A.B}}$	$\frac{M \subseteq A}{\overline{M \subset \lambda x:A.B}}$	$\frac{M \subseteq B}{\overline{M \subset \lambda x:A.B}}$
$\frac{M \subseteq A}{\overline{M \subset AB}}$	$\frac{M \subseteq B}{\overline{M \subset AB}}$	$\frac{M \subseteq A}{\overline{M \subset A\langle x \leftarrow B \rangle}}$	$\frac{M \subseteq B}{\overline{M \subset A\langle x \leftarrow B \rangle}}$

Traditionnellement, l'on établit la distinction entre variables libres et liées d'une expression, comme on l'a vu dans le cas du λ -calcul ou des systèmes de types purs à substitutions implicites. Il est possible également de définir ici ces deux espèces de variables, de manière analogue au cas implicite, sachant que $M\langle x \leftarrow N \rangle$ a les mêmes variables libres et liées que l'expression $\Pi x:N.M$ (i.e. x est liée dans M mais pas dans N) :

Définition 2.3 : *variables libres/liées dans les systèmes de types purs explicites*
Soit \mathfrak{T} un système de types pur. Soit M une $\mathfrak{T}\mathbf{x}$ -expression.

On définit l'ensemble $fv(M)$ des variables *libres* de M inductivement comme suit :

$$\begin{aligned}
fv(\sigma) &= \emptyset \\
fv(x) &= \{x\} \\
fv(\gamma_x) &= \emptyset \\
fv(\Pi x:L.M) &= (fv(M) \setminus \{x\}) \cup fv(L) \\
fv(\lambda x:L.M) &= (fv(M) \setminus \{x\}) \cup fv(L) \\
fv(MN) &= fv(M) \cup fv(N) \\
fv(M\langle x \leftarrow L \rangle) &= (fv(M) \setminus \{x\}) \cup fv(L)
\end{aligned}$$

De même, on définit l'ensemble $bv(M)$ des variables *liées* de M comme suit :

$$\begin{aligned}
bv(\sigma) &= \emptyset \\
bv(x) &= \emptyset \\
bv(\gamma_x) &= \emptyset \\
bv(\Pi x:L.M) &= bv(M) \cup bv(L) \cup \{x\} \\
bv(\lambda x:L.M) &= bv(M) \cup bv(L) \cup \{x\} \\
bv(MN) &= bv(M) \cup bv(N) \\
bv(M\langle x \leftarrow L \rangle) &= bv(M) \cup bv(L) \cup \{x\}
\end{aligned}$$

On remarque que les deux ensembles $fv(M)$ et $bv(M)$ recouvrent $ov(M)$.

Toutefois, cette distinction, fondamentale dans le cas implicite, n'est pas totalement adaptée dans le cas de substitutions explicites, et une autre définition, distinguant variables *accessibles* et *inaccessibles*, semble plus appropriée.

La notion de variable accessible a été introduite par Lengrand *et al.* dans [30] sous le nom d'*available variables* (mot à mot, variables disponibles). Elle est définie comme suit :

Définition 2.4 : *variables accessibles/inaccessibles*

Soit \mathfrak{T} un système de types pur. Soit M une $\mathfrak{T}\mathbf{x}$ -expression.

On définit l'ensemble $av(M)$ des variables *accessibles* de M inductivement ainsi :

$$\begin{aligned}
av(\sigma) &= \emptyset \\
av(x) &= \{x\} \\
av(\gamma_x) &= \emptyset \\
av(\Pi x:L.M) &= (av(M) \setminus \{x\}) \cup av(L) \\
av(\lambda x:L.M) &= (av(M) \setminus \{x\}) \cup av(L) \\
av(MN) &= av(M) \cup av(N) \\
av(M\langle x \leftarrow N \rangle) &= \begin{cases} (av(M) \setminus \{x\}) \cup av(N) & \text{si } x \in av(M) \\ av(M) & \text{si } x \notin av(M). \end{cases}
\end{aligned}$$

De même, on définit l'ensemble $uv(M)$ des variables *inaccessibles* de M ainsi :

$$\begin{aligned}
uv(\sigma) &= \emptyset \\
uv(x) &= \emptyset \\
uv(\gamma_x) &= \emptyset \\
uv(\Pi x:L.M) &= uv(M) \cup uv(L) \cup \{x\} \\
uv(\lambda x:L.M) &= uv(M) \cup uv(L) \cup \{x\} \\
uv(MN) &= uv(M) \cup uv(N) \\
uv(M\langle x \leftarrow N \rangle) &= \begin{cases} uv(M) \cup uv(N) \cup \{x\} & \text{si } x \in uv(M) \\ uv(M) \cup uv(N) \cup \{x\} & \text{si } x \notin uv(M). \end{cases}
\end{aligned}$$

On remarque que les deux ensembles $av(M)$ et $uv(M)$ recouvrent $ov(M)$.

Ces notions seront commentées après la définition des notions de réduction.

Définition 2.5 : α -conversion

On définit sur l'ensemble des expressions $\mathcal{E}\mathbf{x}(\mathfrak{T})$ la relation \equiv_α dite α -conversion inductivement comme suit :

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

Muni de cette relation, il est possible de définir la notion de terme du calcul.

Définition 2.6 : $\lambda\mathfrak{X}$ -termes

Il est clair que la relation \equiv_α est une congruence sur les \mathfrak{X} -expressions, si bien que l'on peut définir l'ensemble quotient $\mathcal{T}_{\mathfrak{X}}(\mathfrak{X}) = \mathcal{E}\mathfrak{X}(\mathfrak{X}) / \equiv_\alpha$, auquel s'étendent canoniquement les opération de $\mathcal{E}\mathfrak{X}(\mathfrak{X})$.

Les éléments de $\mathcal{T}_{\mathfrak{X}}(\mathfrak{X}) = \mathcal{E}\mathfrak{X}(\mathfrak{X}) / \equiv_\alpha$ sont appelés les *termes* du $\lambda\mathfrak{X}$ -calcul.

Un terme est dit *pur* s'il ne contient pas de substitution.

Dans la suite, les termes étant, sauf mention expresse du contraire, considérés à l' α -conversion près, l'on adoptera de nouveau la convention de Barendregt : deux variables liées distinctes ont des noms distincts, et une variable ne peut apparaître à la fois accessible et liée.

On peut maintenant définir sur ces termes une notion de réduction. De même que les systèmes de types purs dans le cas implicite reprenaient la sémantique opérationnelle du λ -calcul, la notion de réduction utilisée se base sur celle de $\lambda\mathfrak{X}$.

Définition 2.7 : *notion de réduction et de conversion \mathfrak{x} , $\beta\mathfrak{x}$*

Soient les règles de réduction suivantes :

$$\begin{array}{lll}
\text{(B)} & (\lambda x:A.B) C & \xrightarrow{B} B\langle x \leftarrow C \rangle \\
\text{(quant)} & (\Pi y:A.B)\langle x \leftarrow C \rangle & \xrightarrow{X} \Pi y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle \\
\text{(abs)} & (\lambda y:A.B)\langle x \leftarrow C \rangle & \xrightarrow{X} \lambda y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle \\
\text{(app)} & (A B)\langle x \leftarrow C \rangle & \xrightarrow{X} A\langle x \leftarrow C \rangle B\langle x \leftarrow C \rangle \\
\text{(subst)} & x\langle x \leftarrow N \rangle & \xrightarrow{X} N \\
\text{(gc)} & M\langle x \leftarrow N \rangle & \xrightarrow{X} M\{x \leftarrow \gamma_x\} \quad \text{si } x \notin \text{av}(M) \\
\text{(comp)} & A\langle x \leftarrow X \rangle\langle y \leftarrow Y \rangle & \xrightarrow{C} A\langle y \leftarrow Y \rangle\langle x \leftarrow X\langle y \leftarrow Y \rangle \rangle
\end{array}$$

À partir de ces règles, on définit diverses notions de réduction :

- la \mathfrak{x} -réduction, notée $\xrightarrow{\mathfrak{x}}$, est définie comme la clôture contextuelle de \xrightarrow{X} ;
- la $\beta\mathfrak{x}$ -réduction, notée $\xrightarrow{\beta\mathfrak{x}}$, est définie comme la clôture contextuelle de l'union de \xrightarrow{X} et de \xrightarrow{B} ;

ainsi que les relations d'équivalence associées :

- la \mathfrak{x} -conversion, notée $\equiv_{\mathfrak{x}}$, est définie comme la clôture réflexive, symétrique et transitive de $\xrightarrow{\mathfrak{x}}$;
- la $\beta\mathfrak{x}$ -conversion, notée $\equiv_{\beta\mathfrak{x}}$, est définie comme la clôture réflexive, symétrique et transitive de $\xrightarrow{\beta\mathfrak{x}}$.

Remarque :

On peut faire divers commentaires à ces définitions :

- (i) Il peut sembler paradoxal, dans un système à substitutions explicites, de voir apparaître une substitution implicite dans la règle (gc). Toutefois, il s'agit en fait ici d'un renommage de variable – et les renommages implicites de ce genre sont déjà utilisés dans l' α -conversion⁸ – destiné à « tuer » la variable x . Ce procédé sera commenté prochainement.
- (ii) Il convient également d'observer que la règle (comp) ne saurait être ajoutée à la x -réduction, car elle induit visiblement une notion de réduction non normalisante. Toutefois, on peut se passer de cette règle dans la mesure où ses deux membres sont x -joignables, comme on le verra plus tard (lemme de composition *infra*).

Par ailleurs, on définit en outre sur les termes une notion de *propagation* comme suit :

Définition 2.8 : *propagation*

On définit une notion de *recul*, notée \rightsquigarrow , au moyen des figures suivantes :

$$\begin{array}{c} \overline{M \rightsquigarrow M} \text{ (réflexivité)} \\ \hline \overline{M\langle u \leftarrow V \rangle \langle x \leftarrow N \rangle \rightsquigarrow M\langle x \leftarrow N \rangle \langle u \leftarrow V \rangle \langle x \leftarrow N \rangle} \text{ (composition)} \\ \hline \overline{M\langle u \leftarrow V \rangle \langle x \leftarrow N \rangle \rightsquigarrow M\langle x \leftarrow N \rangle \langle u \leftarrow V \{x \leftarrow \gamma_x\} \rangle} \text{ (intersion)} \\ \hline \overline{M\langle u \leftarrow V \rangle \langle x \leftarrow N \rangle \rightsquigarrow M\langle x \leftarrow N \rangle \langle u \leftarrow W \rangle \quad M\langle x \leftarrow N \rangle \rightsquigarrow P} \text{ (séquence)} \\ \hline M\langle u \leftarrow V \rangle \langle x \leftarrow N \rangle \rightsquigarrow P\langle u \leftarrow W \rangle \end{array}$$

On définit ensuite comme suit la relation de *propagation*, notée \longrightarrow_χ , au moyen des figures suivantes :

$$\begin{array}{c} \overline{x \notin av(V)} \\ \hline M\langle x \leftarrow N \rangle \longrightarrow_\chi M\{x \leftarrow \gamma_x\} \text{ (gc)} \end{array} \quad \begin{array}{c} \overline{M\langle x \leftarrow N \rangle \rightsquigarrow P} \\ \hline M\langle x \leftarrow N \rangle \longrightarrow_\chi P \text{ (recul)} \end{array}$$

$$\begin{array}{c} \overline{M\langle x \leftarrow N \rangle \rightsquigarrow (\Pi x:U_1.U_2)\langle x \leftarrow N \rangle \cdots \langle u_k \leftarrow W_k \rangle \cdots \quad U_i\langle x \leftarrow N \rangle \longrightarrow_\chi U'_i} \\ \hline M\langle x \leftarrow N \rangle \longrightarrow_\chi (\Pi x:U'_1.U'_2) \cdots \langle u_k \leftarrow W_k \rangle \cdots \end{array} \text{ (\Pi-retard)}$$

$$\begin{array}{c} \overline{M\langle x \leftarrow N \rangle \rightsquigarrow (\lambda x:U_1.U_2)\langle x \leftarrow N \rangle \cdots \langle u_k \leftarrow W_k \rangle \cdots \quad U_i\langle x \leftarrow N \rangle \longrightarrow_\chi U'_i} \\ \hline M\langle x \leftarrow N \rangle \longrightarrow_\chi (\lambda x:U'_1.U'_2) \cdots \langle u_k \leftarrow W_k \rangle \cdots \end{array} \text{ (\lambda-retard)}$$

⁸Naturellement, dans un système à α -conversion implicite, il conviendrait d'expliciter ce renommage, mais ces considérations sont ici hors de propos.

$$\frac{M\langle x \leftarrow N \rangle \rightsquigarrow ((\lambda y:U_1.U_2)U_3)\langle x \leftarrow N \rangle \cdots \langle u_k \leftarrow W_k \rangle \cdots \quad U_i\langle x \leftarrow N \rangle \longrightarrow_{\chi} U'_i}{M\langle x \leftarrow N \rangle \longrightarrow_{\chi} ((\lambda y:U'_1.U'_2)U'_3)\cdots \langle u_k \leftarrow W_k \rangle \cdots} \text{ (@-retard)}$$

$$\frac{M\langle x \leftarrow N \rangle \rightsquigarrow (U_1 U_2)\langle x \leftarrow N \rangle \cdots \langle u_k \leftarrow W_k \rangle \cdots \quad U_i\langle x \leftarrow N \rangle \longrightarrow_{\chi} U'_i}{M\langle x \leftarrow N \rangle \longrightarrow_{\chi} (U'_1 U'_2)\cdots \langle u_k \leftarrow W_k \rangle \cdots} \text{ (A-retard)}$$

avec, dans ce dernier cas, U_1 ne commençant pas par un λ .

On peut noter que cette relation n'est pas une congruence, mais une forme de réduction de tête. Elle aura un rôle important dans le système de typage. On peut décrire en mots l'effet de cette relation de réduction comme suit : on propage la substitution de tête, soit en l'ôtant alors qu'elle peut être éliminée par un (gc), soit en la poussant aussi loin que l'on veut par composition, sans permettre la propagation dans l'opérande d'une substitution.

Propriétés des \mathbf{x} -réduction et $\beta\mathbf{x}$ -réduction

On a évoqué plus haut un lemme énonçant que la \mathbf{x} -conversion conserve les variables accessibles. Le voici :

Lemme 2.1 : *Conservation des variables accessibles*

Soient U et V deux termes.

Si $U \equiv_{\mathbf{x}} V$, alors $av(U) = av(V)$.

Démonstration.

Il suffit de montrer que si $U \xrightarrow{\mathbf{x}} V$, alors $av(U) = av(V)$, ce que l'on fait par cas :

- a. (quant) : $(\Pi y:A.B)\langle x \leftarrow C \rangle \xrightarrow{\mathbf{x}} \Pi y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle$
 $av((\Pi y:A.B)\langle x \leftarrow C \rangle) = (av(A) \cup (av(B) \setminus \{y\})) \cup (av(C) \setminus \{x\})$
 $= (av(A) \cup (av(C) \setminus \{x\})) \cup ((av(B) \setminus \{y\}) \cup (av(C) \setminus \{x\}))$
 $= av(\Pi y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle).$
- b. (abs) : $(\lambda y:A.B)\langle x \leftarrow C \rangle \xrightarrow{\mathbf{x}} \lambda y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle$
 $av((\lambda y:A.B)\langle x \leftarrow C \rangle) = (av(A) \cup (av(B) \setminus \{y\})) \cup (av(C) \setminus \{x\})$
 $= (av(A) \cup (av(C) \setminus \{x\})) \cup ((av(B) \setminus \{y\}) \cup (av(C) \setminus \{x\}))$
 $= av(\lambda y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle).$
- c. (app) : $(A B)\langle x \leftarrow C \rangle \xrightarrow{\mathbf{x}} A\langle x \leftarrow C \rangle B\langle x \leftarrow C \rangle$
 $av((A B)\langle x \leftarrow C \rangle) = (av(A) \cup av(B)) \cup (av(C) \setminus \{x\})$
 $= (av(A) \cup (av(C) \setminus \{x\})) \cup (av(B) \cup (av(C) \setminus \{x\}))$
 $= av(A\langle x \leftarrow C \rangle B\langle x \leftarrow C \rangle).$
- d. (subst) : $x\langle x \leftarrow N \rangle \xrightarrow{\mathbf{x}} N$
Par définition des variables accessibles, $av(x\langle x \leftarrow N \rangle) = av(N)$.
- e. (gc) : $M\langle x \leftarrow N \rangle \xrightarrow{\mathbf{x}} M\{x \leftarrow \gamma_x\}$ si $x \notin av(M)$
Par définition des variables accessibles, d'une part $av(M\langle x \leftarrow N \rangle) = av(M)$

et d'autre part $av(\mathbf{M}\langle x \leftarrow \gamma_x \rangle) = av(\mathbf{M}) \setminus \{x\} = av(\mathbf{M})$ car $x \notin av(\mathbf{M})$. Et donc $av(\mathbf{M}\langle x \leftarrow \mathbf{N} \rangle) = av(\mathbf{M}\langle x \leftarrow \gamma_x \rangle)$.

Dans tous les cas, on obtient bien que $\xrightarrow{\mathbf{x}}$ conserve les variables accessibles. Il suffit ensuite de passer à la congruence pour étendre cette propriété à $\xrightarrow{\mathbf{x}}$, sachant qu'il est aisé de montrer que si $av(\mathbf{M}) = av(\mathbf{N})$, alors pour tout $\mathbf{C}[\]$, $av(\mathbf{C}[\mathbf{M}]) = av(\mathbf{C}[\mathbf{N}])$, puis de passer à la clôture réflexive, symétrique et transitive pour l'étendre à $\equiv_{\mathbf{x}}$. Q.E.D.

Par ailleurs, comme il est dit *supra*, la \mathbf{x} -réduction vérifie la propriété suivante :

Proposition 2.2 : Normalisation forte de la \mathbf{x} -réduction

La \mathbf{x} -réduction est fortement normalisante.

Démonstration.

Pour montrer la normalisation forte de la \mathbf{x} -réduction, il suffit de définir sur les termes une fonction de poids w à valeurs dans \mathbb{N} vérifiant que pour chaque axiome $L \xrightarrow{\mathbf{x}} R$ de la \mathbf{x} -réduction, on a $w_L > w_R$. On rappelle que les axiomes – donnés à la définition 2.7 – de cette réduction sont les suivants :

$$\begin{array}{ll} (\text{quant}) & (\Pi y:A.B)\langle x \leftarrow C \rangle \xrightarrow{\mathbf{x}} \Pi y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle; \\ (\text{abs}) & (\lambda y:A.B)\langle x \leftarrow C \rangle \xrightarrow{\mathbf{x}} \lambda y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle; \\ (\text{app}) & (A B)\langle x \leftarrow C \rangle \xrightarrow{\mathbf{x}} A\langle x \leftarrow C \rangle B\langle x \leftarrow C \rangle; \\ (\text{subst}) & x\langle x \leftarrow N \rangle \xrightarrow{\mathbf{x}} N; \\ (\text{gc}) & \mathbf{M}\langle x \leftarrow \mathbf{N} \rangle \xrightarrow{\mathbf{x}} \mathbf{M}\{x \leftarrow \gamma_x\}. \end{array}$$

On propose la fonction de poids w suivante :

- $w(x) = 1$;
- $w(\gamma_x) = 1$;
- $w(\sigma) = 1$;
- $w(A\langle x \leftarrow B \rangle) = w(A) \times (w(B) + 1)$;
- $w(\Pi x:A.B) = w(A) + w(B) + 1$;
- $w(\lambda x:A.B) = w(A) + w(B) + 1$;
- $w(AB) = w(A) + w(B) + 1$.

On peut traiter ensemble les cas *(quant)*, *(abs)*, *(app)*. Dans ces trois cas, l'on a $w(L) = (w(A) + w(B) + 1)(w(C) + 1) = (w(A) + w(B)) \times (w(C) + 1) + w(C) + 1$ d'une part et $w(R) = w(A) \times (w(C) + 1) + w(B) \times (w(C) + 1) = (w(A) + w(B)) \times (w(C) + 1)$ d'autre part, si bien que $w(L) - w(R) = w(C) + 1 > 0$.

Pour *(subst)*, on a $w(L) = w(x) \times (w(N) + 1) = w(N) + 1$ d'une part, et $w(R) = w(N)$ d'autre part, si bien que $w(L) - w(R) = 1 > 0$.

Pour *(gc)*, l'on a $w(L) = w(\mathbf{M}) \times (w(\mathbf{N}) + 1)$ d'une part, et $w(R) = w(\mathbf{M}\{x \leftarrow \gamma_x\}) = w(\mathbf{M})$, d'autre part, si bien que $w(L) - w(R) = w(\mathbf{M}) \times w(\mathbf{N}) > 0$.

Dans tous les cas, on a bien que le poids du membre droit est strictement inférieur

à celui du membre gauche.

Et donc la \mathbf{x} -réduction est bien fortement normalisante.

Q.E.D.

Par ailleurs, on peut également montrer la confluence de la \mathbf{x} -réduction :

Lemme 2.3 : *Confluence de la \mathbf{x} -réduction*

La \mathbf{x} -réduction est confluente.

Démonstration.

Comme la \mathbf{x} -réduction est fortement normalisante, il suffit de montrer que toutes ses paires critiques sont joignables pour montrer sa confluence. On considère donc les paires critiques du système, qui sont nécessairement formées au moyen d'une règle de (gc), et sont donc au nombre de trois :

1°) $((\Pi y:U.V)\{x \leftarrow \gamma_x\}, \Pi y:U\langle x \leftarrow W \rangle.V\langle x \leftarrow W \rangle)$, issue de $(\Pi y:U.V)\langle x \leftarrow W \rangle$.

Par la convention de Barendregt, $y \neq x$, et donc, comme $x \notin \text{av}(\Pi y:U.V)$, $x \notin \text{av}(U)$ et $x \notin \text{av}(V)$.

Si bien que $\Pi y:U\langle x \leftarrow W \rangle.V\langle x \leftarrow W \rangle \xrightarrow{\frac{2}{x}} \Pi y:U\{x \leftarrow \gamma_x\}.V\{x \leftarrow \gamma_x\}$, et ce second membre se récrit $\Pi y:U.V\{x \leftarrow \gamma_x\}$, donc la paire critique est joignable.

2°) $((\lambda y:U.V)\{x \leftarrow \gamma_x\}, \lambda y:U\langle x \leftarrow W \rangle.V\langle x \leftarrow W \rangle)$, issue de $(\lambda y:U.V)\langle x \leftarrow W \rangle$.

Par la convention de Barendregt, $y \neq x$, et donc, comme $x \notin \text{av}(\lambda y:U.V)$, $x \notin \text{av}(U)$ et $x \notin \text{av}(V)$.

Si bien que $\lambda y:U\langle x \leftarrow W \rangle.V\langle x \leftarrow W \rangle \xrightarrow{\frac{2}{x}} \lambda y:U\{x \leftarrow \gamma_x\}.V\{x \leftarrow \gamma_x\}$, et ce second membre se récrit $\lambda y:U.V\{x \leftarrow \gamma_x\}$, donc la paire critique est joignable.

3°) $((U V)\{x \leftarrow \gamma_x\}, U\langle x \leftarrow W \rangle V\langle x \leftarrow W \rangle)$, issue de $(U V)\langle x \leftarrow W \rangle$.

Comme $x \notin \text{av}(UV)$, $x \notin \text{av}(U)$ et $x \notin \text{av}(V)$. Si bien que $U\langle x \leftarrow W \rangle V\langle x \leftarrow W \rangle \xrightarrow{\frac{2}{x}} U\{x \leftarrow \gamma_x\} V\{x \leftarrow \gamma_x\}$, et ce second membre se récrit $(UV)\{x \leftarrow \gamma_x\}$, donc la paire critique est joignable.

Toutes les paires critiques sont donc joignables.

Q.E.D.

La convergence de la \mathbf{x} -réduction, résultant des deux lemmes *supra*, permet en outre de montrer le résultat suivant :

Lemme 2.4 : *Composition*

Soient A, B, C trois termes et x, y deux variables telles que $x \notin \text{av}(B) \cup \text{av}(C) \cup \{y\}$.

Alors $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \equiv_x A\langle y \leftarrow C \rangle\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle$.

(On peut remarquer que la convention de Barendregt rend inutile la vérification de la condition sur x.)

Démonstration.

Quitte à effectuer une \mathbf{x} -normalisation, qui n'influe pas sur la \mathbf{x} -convertibilité car la \mathbf{x} -réduction est confluente, on peut supposer que A est une forme \mathbf{x} -normale.

On procède par induction sur A .

Si A est une variable, plusieurs cas se présentent :

- $A = x$. Dans ce cas, d'une part $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} B\langle y \leftarrow C \rangle$ et d'autre part $A\langle y \leftarrow C \rangle\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} B\langle y \leftarrow C \rangle$.
- $A = y$. Dans ce cas, d'une part $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} \xrightarrow{x} C$ et d'autre part, comme $x \notin \text{av}(C)$, $A\langle y \leftarrow C \rangle\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} C\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} C$.
- $A = z \neq x, y$. Dans ce cas, les deux termes se réduisent à z .

Si $A = QR$, alors on a que $A\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} \xrightarrow{x} Q\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle R\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle$, et $A\langle y \leftarrow C \rangle\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle \xrightarrow{x} \xrightarrow{x} Q\langle y \leftarrow C \rangle\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle R\langle y \leftarrow C \rangle\langle x \leftarrow B \rangle\langle y \leftarrow C \rangle$. On conclut alors aisément par hypothèse d'induction.

Si $A = \Pi w:Q.R$ ou $A = \lambda w:Q.R$, on procède comme pour le cas précédent. Q.E.D.

Corollaire 2.5 : Propagation et \mathbf{x} -réduction

Soient M et N deux termes tels que $M \xrightarrow{\chi} N$. Alors $M \equiv_x N$.

Démonstration.

On commence par montrer la propriété pour la relation de recul \rightsquigarrow , en considérant chaque inférence la définissant :

- (réflexivité) : ce point est trivial ;
- (composition) : ce point découle du lemme précédent ;
- (intersion) : ce point découle du lemme précédent et de l'axiome de réduction (gc) ;
- (séquence) : ce point est direct par induction et transitivité de \equiv_x .

On peut maintenant étendre la propriété à la propagation : si la propagation est un (gc), le point est trivial ; si c'est un recul, il découle de ce que l'on vient de montrer ; dans les autres éventualités, on conclut par induction avec en plus l'axiome de \mathbf{x} -réduction adapté à la forme du terme. Q.E.D.

La \mathbf{x} -réduction étant convergente, l'on peut définir la forme \mathbf{x} -normale d'un terme M , notée $\mathbf{x}(M)$, et établir ainsi une relation avec la réduction implicite :

Proposition 2.6 : Réduction implicite

Les assertions suivantes sont vraies :

- (i) soit M un terme pur ; si $M \xrightarrow{B} M'$, alors $M \xrightarrow{\beta} \mathbf{x}(M')$;
- (ii) soit M un terme pur ; si $M \xrightarrow{\beta} M'$, alors $M \xrightarrow{\beta} \xrightarrow{x}^* M'$;
- (iii) pour tout M , si $M \xrightarrow{B} M'$, alors $\mathbf{x}(M) \xrightarrow{\beta}^* \mathbf{x}(M')$.
- (iv) pour tout M , si $M \xrightarrow{\beta_x} M'$, alors $\mathbf{x}(M) \xrightarrow{\beta}^* \mathbf{x}(M')$.

Démonstration.

Ces quatre assertions se démontrent facilement une fois observé que la forme \mathbf{x} -normale de M peut être définie inductivement comme suit :

- $\mathbf{x}(\sigma) = \sigma ; \mathbf{x}(x) = x ;$
- $\mathbf{x}(\Pi x:L.M) = \Pi x:\mathbf{x}(L).\mathbf{x}(M) ;$
- $\mathbf{x}(\lambda x:L.M) = \lambda x:\mathbf{x}(L).\mathbf{x}(M) ;$
- $\mathbf{x}(MN) = (\mathbf{x}(M))(\mathbf{x}(N)) ;$
- $\mathbf{x}(M\langle x\leftarrow N \rangle) = (\mathbf{x}(M))\{x\leftarrow \mathbf{x}(N)\}.$

Q.E.D.

On peut alors en déduire le corollaire suivant :

Corollaire 2.7 : *Transposition de réduction*

Soient M et N deux termes purs. Alors $M \xrightarrow{\beta^*} N$ si, et seulement si, $M \xrightarrow{\beta^{\mathbf{x}}} N$.

Démonstration.

Le sens direct découle directement de la proposition 2.6 (ii), par récurrence sur la longueur de la réduction.

Réciproquement, par la proposition 2.6 (iv), $U \xrightarrow{\beta^{\mathbf{x}}} V$ implique $\mathbf{x}(U) \xrightarrow{\beta^} \mathbf{x}(V)$, et l'on peut alors montrer $\mathbf{x}(M) \xrightarrow{\beta^*} \mathbf{x}(N)$ par récurrence sur la longueur de la réduction. Or un terme pur est égal à sa forme \mathbf{x} -normale, et l'on peut donc en déduire que $M \xrightarrow{\beta^*} N$.*

Q.E.D.

Après la \mathbf{x} -réduction, on peut s'intéresser à la $\beta\mathbf{x}$ -réduction. Tout d'abord, celle-ci ne fait pas apparaître de variables accessibles :

Lemme 2.8 : *Inclusion des variables accessibles*

Soient U et V deux termes.

Si $U \xrightarrow{\beta^{\mathbf{x}}} V$, alors $av(U) \supseteq av(V)$.

Démonstration.

Vu le lemme 2.1, il suffit de montrer que si $U \xrightarrow{B} V$, alors $av(U) \supseteq av(V)$.

On considère donc (B) : $(\lambda y:A.M)N \xrightarrow{B} M\langle x\leftarrow N \rangle$.

On a d'une part $av((\lambda y:A.M)N) = av(A) \cup av(M) \setminus \{x\} \cup av(N)$ et d'autre part :

- *si $x \in av(M)$, $av(M\langle x\leftarrow N \rangle) = av(M) \setminus \{x\} \cup av(N) \subseteq av((\lambda y:A.M)N)$;*
- *si $x \in av(N)$, $av(M\langle x\leftarrow N \rangle) = av(M) = av(M) \setminus \{x\} \subseteq av((\lambda y:A.M)N)$.*

Dans tous les cas, on obtient bien que \xrightarrow{B} n'augmente pas les variables accessibles. En outre, on a montré avec le lemme 2.1 que \xrightarrow{X} conserve les variables accessibles, donc a fortiori ne les augmente pas. Il suffit ensuite de passer à la congruence pour étendre cette propriété à $\xrightarrow{\beta^{\mathbf{x}}}$ (en effet, il est aisé de montrer que si $av(M) \supseteq av(N)$, alors pour tout $C[]$, $av(C[M]) \supseteq av(C[N])$ puis à la clôture réflexive et transitive pour l'étendre à $\xrightarrow{\beta^}$.*

Q.E.D.

On peut maintenant montrer la confluence de la $\beta\mathbf{x}$ -réduction :

Lemme 2.9 : *Confluence de la $\beta\mathbf{x}$ -réduction*

La $\beta\mathbf{x}$ -réduction est confluente.

Démonstration.

On montre aisément la confluence de \xrightarrow{B} , réduction engendrée par l'axiome (B). On peut alors montrer la confluence de la $\beta\mathbf{x}$ -réduction par la méthode d'interprétation de Hardin et Lévy (cf. [51]). On considère donc le diagramme suivant :

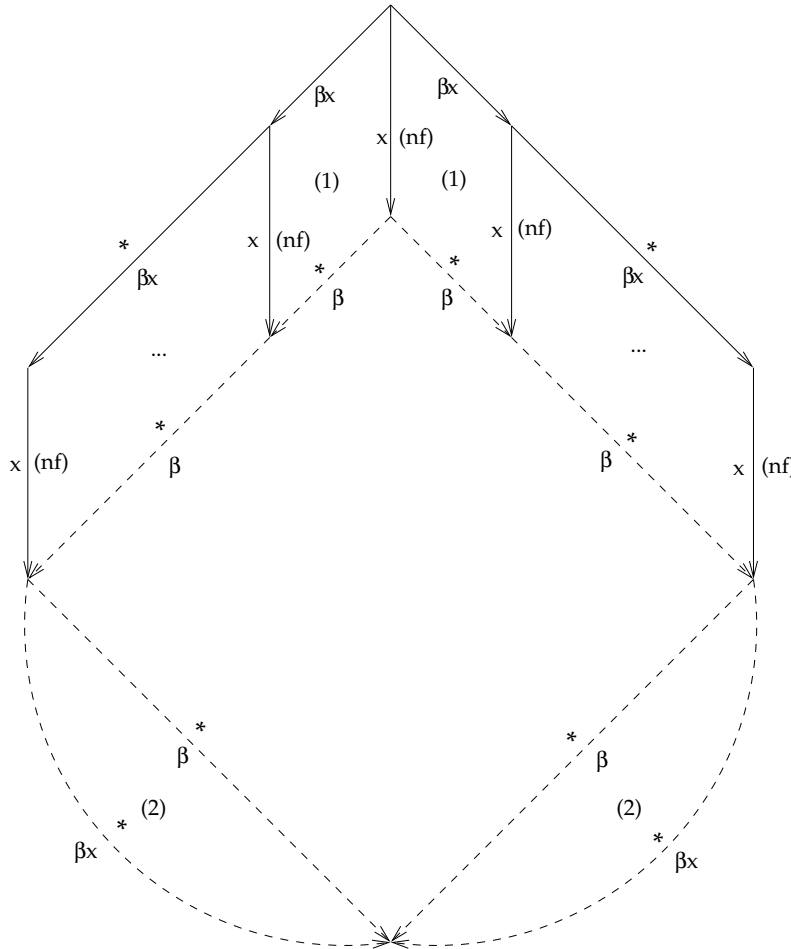


FIG. 4 – Confluence de $\beta\mathbf{x}$ par interprétation

où l'on désigne par (1) l'application de la proposition 2.6 (iii) (qui s'emploie récurremment), et par (2) celle du corollaire 2.7. La flèche $\xrightarrow{\frac{(nf)}{x}}$ désigne la \mathbf{x} -normalisation. On déduit de ce diagramme que la confluence du calcul implicite implique celle du calcul explicite. Q.E.D.

En outre, il est clair qu'au contraire de la \mathbf{x} -réduction, la $\beta\mathbf{x}$ -réduction n'est pas normalisante, même faiblement. Un contre-exemple possible est $\Omega = \omega\omega$ où l'on a $\omega = (\lambda u:p.uu)$ avec p une variable. On peut montrer que $\Omega \xrightarrow{\beta\mathbf{x}}^* M$ implique que $M \xrightarrow{\mathbf{x}}^* \Omega$. On en déduit que Ω n'admet aucune réduction finie : en effet, s'il existe une forme $\beta\mathbf{x}$ -normale N telle que $\Omega \xrightarrow{\beta\mathbf{x}}^* N$, d'où $N \xrightarrow{\mathbf{x}}^* \Omega$, d'où, N étant $\beta\mathbf{x}$ -normal, $N = \Omega$, mais Ω n'est pas $\beta\mathbf{x}$ -normal, contradiction. Et donc la $\beta\mathbf{x}$ -réduction n'est pas faiblement normalisante, ni, *a fortiori*, fortement.

Variables libres ou accessibles, et γ -constantes

L'on va ici expliquer pourquoi l'on prend en considération la notion de variable accessible plutôt que libre, et montrer les problèmes qui en découlent, que l'on résout par l'introduction des γ -constantes.

Tout d'abord, on constate que, pour un terme sans substitution explicite, ni γ -constantes, la notion de variable accessible (respectivement inaccessible) correspond exactement à celle de variable libre (respectivement liée), ce qui explique que cette nouvelle distinction ne soit jamais apparue dans le cadre du λ -calcul à substitutions implicites. Par ailleurs, il est évident que, quoique différenciées dans le cas général des termes à substitutions explicites, ces notions sont néanmoins reliées. Ainsi, une variable accessible est-elle nécessairement libre, et, par complémentarité, une variable liée, inaccessible.

Faut-il donc croire que la notion de variable accessible ou inaccessible soit la distinction la plus pertinente, et que la notion de variable libre ou liée n'est qu'un héritage du cas implicite où ces deux notions sont confondues ? Cela n'est pas évident, car deux arguments s'opposent.

D'une part, la notion de variable accessible semble en effet plus pertinente que celle de variable libre dans de nombreux cas : en particulier, d'un point de vue calculatoire ou d'un point de vue de typage, il semble qu'une variable libre non accessible ne présente pas d'intérêt, car une substitution sur cette variable appliquée au terme ne pourra qu'être éliminée sans être jamais effectivement réalisée : dans le terme $M\langle x \leftarrow N \rangle$, si x n'est pas accessible dans M , N ne pourra pas apparaître dans la forme \mathbf{x} -normale du terme.

En outre, cet argument est conforté par un lemme énonçant que des termes \mathbf{x} -convertibles (*i.e.* représentant le même terme pur – *i.e.* sans substitutions explicites ; cette notion sera définie *infra*) ont les mêmes variables accessibles – tandis que la \mathbf{x} -conversion peut faire disparaître des variables libres non accessibles.

D'autre part, la notion de variable inaccessible ne semble pas pouvoir complètement supplanter la notion de variable liée. En effet, si l'on ne voulait considérer

que des variables inaccessibles, cela signifierait, *mutatis mutandis*, qu’il faudrait que l’ α -conversion identifiât deux expressions où l’on aurait changé une variable inaccessible. Et donc, selon ce principe, il faudrait que, de même que l’on identifie $x\langle x\leftarrow M \rangle$ et $y\langle y\leftarrow M \rangle$, ce qui semble pertinent (les variables ici sont inaccessibles et liées), l’on identifiât $u\langle v\leftarrow x\langle y\leftarrow M \rangle \rangle$ et $u\langle v\leftarrow y\langle y\leftarrow M \rangle \rangle$, ce qui ne semble pas pertinent, car cette conversion de nom changerait le comportement opérationnel du terme (ce qui provient du fait que ces variables sont inaccessibles mais libres). C’est pourquoi l’on choisit, pour la définition de l’ α -conversion, la notion de variable liée plutôt que de variable inaccessible.

Il semble donc qu’il convienne de répartir les variables en trois espèces : les variables accessibles, les variables liées, et les autres variables, qui sont inaccessibles mais libres, et qui pourraient porter le nom de variables *dissimulées*. Ces dernières variables posent toutefois un problème pour identifier le calcul à un système de réécriture.

En effet, si l’on considère la règle (gc) « traditionnelle », *i.e.* sans γ -constante, et où l’on remplace la notion de variable libre par celle de variable accessible, *i.e.* la règle telle qu’elle est introduite dans [30] :

$$M\langle x\leftarrow N \rangle \xrightarrow{(gc)} M \quad \text{si } x \notin av(M),$$

on peut voir que cette règle permet, dans certains cas, la libération de variables liées :

$$u\langle v\leftarrow w \rangle\langle w\leftarrow y \rangle \xrightarrow{(gc)} u\langle v\leftarrow w \rangle.$$

Ici, la variable w est libérée. Cela pose naturellement problème, car cela empêche le système ainsi formé (la variante de λx présentée dans [30]) d’être identifié à un système de réécriture⁹. En outre, cela rend la substitution implicite non compositionnelle, puisque l’on a d’une part :

$$u\langle v\leftarrow w \rangle\langle w\leftarrow y \rangle \longrightarrow u\langle v\leftarrow w \rangle$$

tandis que

$$(u\langle v\leftarrow w \rangle\langle w\leftarrow y \rangle)\{w\leftarrow t\} \not\rightarrow (u\langle v\leftarrow w \rangle)\{w\leftarrow t\} = u\langle v\leftarrow t \rangle,$$

ce qui est naturellement une source de problèmes.

⁹Ce problème est déjà présent dans [30], où il n’est pas abordé ; toutefois, l’on peut probablement l’y traiter comme ici par l’adjonction de constantes destinées à être substituées aux variables libérées.

Toutefois, l'on observe qu'une telle variable se trouve nécessairement en position inaccessible, et, bien mieux, qu'elle ne pourra pas être substituée car pour qu'une substitution s'y appliquât (et pût être effectuée au moyen d'une composition), il faudrait que celle-ci capturât la variable libérée, ce qui est interdit du fait de la convention de Barendregt. C'est ainsi que vient l'idée de dire que ces variables apparemment libérées restent en fait sous contrôle, et, pour matérialiser ce contrôle, on les considère comme des constantes et les représente par une construction γ_x (qui peut être implantée de différentes manières, mais que l'on peut se contenter de considérer comme une constante). Ainsi, l'on obtient que la variable, au lieu d'être libérée comme dans la version naïve de (gc), est rendue insubstituable, *i.e.* remplacée par une constante.

Tout ceci permet ensuite de raisonner sur les termes à substitutions explicites avec la notion de variable accessible, comme on raisonnait sur les termes à substitutions implicites avec celle de variable libre. Pour augmenter la transparence de cette nouvelle construction, on introduit la notation \bar{x} , qui désigne indifféremment la variable x ou la constante γ_x .

On montrera d'ailleurs *infra* un lemme indiquant la possibilité dans une certaine mesure d'interchanger variable (substituable) et γ -constante (insubstituable), ce qui justifie en quelque sorte ce raccourci de langage, qui permet de se rapprocher des calculs plus habituels.

On peut vérifier que la solution ainsi apportée pallie le problème de libération de variables – et permet donc bien d'identifier le calcul à un système de réécriture d'ordre supérieur – en montrant le lemme suivant :

Lemme 2.10 : *Non-libération de variables*

Si $U \xrightarrow{\beta_x^*} V$, alors $fv(U) \supseteq fv(V)$.

Démonstration.

Il suffit de montrer que si $U \xrightarrow{B} V$ ou $U \xrightarrow{X} V$, alors $fv(U) \supseteq fv(V)$, ce que l'on fait par cas :

a. (B) : $(\lambda y:A.M)N \xrightarrow{B} M\langle y \leftarrow N \rangle$.

$$\begin{aligned} fv((\lambda y:A.M)N) &= fv(A) \cup fv(M) \setminus \{y\} \cup fv(N) \\ &\supseteq fv(M) \setminus \{y\} \cup fv(N) \\ &= fv(M\langle y \leftarrow N \rangle); \end{aligned}$$

b. (quant) : $(\Pi y:A.B)\langle x \leftarrow C \rangle \xrightarrow{X} \Pi y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle$

$$\begin{aligned} fv((\Pi y:A.B)\langle x \leftarrow C \rangle) &= (fv(A) \cup (fv(B) \setminus \{y\})) \setminus \{x\} \cup fv(C) \\ &= (fv(A) \setminus \{x\} \cup fv(C)) \cup (fv(B) \setminus \{y\} \setminus \{x\} \cup fv(C)) \\ &= (fv(A) \setminus \{x\} \cup fv(C)) \cup (fv(B) \setminus \{x\} \cup fv(C)) \setminus \{y\} \quad (\boxtimes) \\ &= fv(\Pi y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle). \end{aligned}$$

- c. (abs) : $(\lambda y:A.B)\langle x \leftarrow C \rangle \xrightarrow{\bar{x}} \lambda y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle$

$$\begin{aligned} fv((\lambda y:A.B)\langle x \leftarrow C \rangle) &= (fv(A) \cup (fv(B) \setminus \{y\})) \setminus \{x\} \cup fv(C) \\ &= (fv(A) \setminus \{x\} \cup fv(C)) \cup (fv(B) \setminus \{y\} \setminus \{x\} \cup fv(C)) \\ &= (fv(A) \setminus \{x\} \cup fv(C)) \cup (fv(B) \setminus \{x\} \cup fv(C)) \setminus \{y\} \quad (\heartsuit) \\ &= fv(\lambda y:A\langle x \leftarrow C \rangle.B\langle x \leftarrow C \rangle). \end{aligned}$$
- d. (app) : $(A B)\langle x \leftarrow C \rangle \xrightarrow{\bar{x}} A\langle x \leftarrow C \rangle B\langle x \leftarrow C \rangle$

$$\begin{aligned} fv((A B)\langle x \leftarrow C \rangle) &= (fv(A) \cup (fv(B) \setminus \{y\})) \setminus \{x\} \cup fv(C) \\ &= (fv(A) \setminus \{x\} \cup fv(C)) \cup (fv(B) \setminus \{y\} \setminus \{x\} \cup fv(C)) \\ &= (fv(A) \setminus \{x\} \cup fv(C)) \cup (fv(B) \setminus \{x\} \cup fv(C)) \setminus \{y\} \quad (\heartsuit) \\ &= fv(A\langle x \leftarrow C \rangle B\langle x \leftarrow C \rangle). \end{aligned}$$
- e. (subst) : $x\langle x \leftarrow N \rangle \xrightarrow{\bar{x}} N$

$$fv(x\langle x \leftarrow N \rangle) = (\{x\} \setminus \{x\}) \cup fv(N) = fv(N).$$
- f. (gc) : $M\langle x \leftarrow N \rangle \xrightarrow{\bar{x}} M\{x \leftarrow \gamma_x\}$ si $x \notin av(M)$

$$fv(M\langle x \leftarrow N \rangle) = (fv(M) \setminus \{x\}) \cup (fv(N) \supseteq fv(M) \setminus \{x\}) = fv(M\{x \leftarrow \gamma_x\}).$$

Dans les calculs supra, le symbole (\heartsuit) indique que l'on passe à cette ligne en observant que les choix de noms de variables dans les termes doivent respecter la convention de Barendregt.

Dans tous les cas, l'on obtient bien que $\xrightarrow{\bar{x}}$ et \xrightarrow{B} ne libèrent pas de variables. Il suffit ensuite de passer à la congruence pour étendre à $\xrightarrow{\beta_x}$ cette propriété – en effet, il est aisé de montrer que si $fv(M) \supseteq fv(N)$, alors pour tout contexte $C[\]$, $fv(C[M]) \supseteq fv(C[N])$ – puis de passer à la clôture réflexive et transitive pour l'étendre à $\xrightarrow{\beta_x^*}$ afin de conclure. Q.E.D.

Il est intéressant de remarquer ici que les variables accessibles sont déjà introduites par R. Bloo dans [8] sous le nom de « *substitutable free variables* » (mot à mot, variables libres substituables), et évoquées dans [14], bien que cette notion ne soit ensuite pas exploitée et que les auteurs s'en tiennent à la notion de variable libre, ce qui leur évite les complications liées à la règle (gc) évoquées supra. Leur dénomination, toutefois, est spécialement intéressante lorsque l'on considère la discussion qui précède, où l'on indique précisément que les variables n'entrant pas dans cette catégorie sont, précisément, non substituables.

1.3. Typage

On va ici introduire diverses modifications à la notion de typage dans les systèmes de types purs afin de les adapter au cas des substitutions explicites. La pertinence de ces modifications sera discutée par la suite, et ce, en particulier à la section 2, consacrée aux propriétés remarquables du système de typage.

Définition 2.9 : *assertion, contexte de typage dans les systèmes explicites*

Soit \mathfrak{X} un système de types pur explicite.

Une *assertion de typage* est un couple noté $M : N$, dont le *sujet* M et le *prédicat* N appartiennent à $\overline{\Lambda\mathfrak{X}} = \Lambda\mathfrak{X} \uplus \{ \int \langle u_1 \leftarrow V_1 \rangle \cdots \langle u_n \leftarrow V_n \rangle / i \in \mathbb{N}, x_i \in \mathcal{U}, V_i \in \Lambda\mathfrak{X} \}$. L'élément additionnel \int , appelé *pseudo-sorte*, est un prédicat spécial introduit dans le cadre explicite pour des raisons que l'on expliquera plus tard.

Un *contexte de typage* Γ est une suite¹⁰ finie d'assertions de typage dont les prédicats sont des termes de $\Lambda\mathfrak{X}$ (\int exclu, donc – on reviendra plus tard sur ce point important) et dont les sujets sont des variables ou des γ -constantes. Son *domaine*, dénoté $\text{dom}(\Gamma)$ est l'ensemble des sujets des assertions de typage dont le contexte est constitué ; son *support* en est la suite ordonnée, *i.e.* l'image du contexte par la première projection canonique. Le contexte vide se note $()$, la concaténation de contextes est notée par une virgule, et l'on suppose lorsque l'on emploie cette notation que les deux contextes ainsi concaténés ont des domaines disjoints.

Avant de définir la notion de jugement de typage proprement dite, on peut commencer par donner quelques précisions sur les contextes, afin de pouvoir les manipuler plus aisément. L'on adopte tout d'abord la notation suivante :

Notation :

Si $\Gamma = (\bar{x}_i : A_i)_{1 \leq i \leq n}$, alors :

- (i) $\Gamma, \bar{x}_{n+1} : A_{n+1}$ dénote $(\bar{x}_i : A_i)_{1 \leq i \leq n+1}$;
- (ii) $\Gamma \setminus \bar{y}$ dénote $(\bar{x}_i : A_i)_{i \in I}$, où $I = \{i \in \llbracket 1, n \rrbracket / \bar{x}_i \neq \bar{y}\}$.

Ensuite, on s'intéresse à la notion de sous-contexte, pour laquelle, les contextes étant des suites et non des ensembles¹¹, on peut distinguer deux espèces.

Définition 2.10 : *relations entre contextes*

Soient $\Gamma = (e_i)_{1 \leq i \leq m}$ et $\Delta = (f_j)_{1 \leq j \leq n}$ deux contextes de typage.

- (i) On dit que Γ est une *restriction* de Δ , et Δ une *extension* de Γ , ce que l'on note $\Gamma \subseteq \Delta$, si Γ est une suite extraite de Δ , *i.e.* il existe $\varphi : \llbracket 1, m \rrbracket \rightarrow \llbracket 1, n \rrbracket$ strictement croissante telle que pour tout i de $\llbracket 1, m \rrbracket$, $e_i = f_{\varphi(i)}$;
- (ii) On dit que Γ est un *préfixe* de Δ , et Δ une *prolongation* de Γ , ce que l'on note $\Gamma \sqsubseteq \Delta$, si $m \leq n$ et pour tout i de $\llbracket 1, m \rrbracket$, $e_i = f_i$.

¹⁰et non un ensemble, comme cela était déjà le cas dans le cadre des systèmes de types purs implicites – cf. note 5 p. 32

¹¹dans le cas où les contextes sont des ensembles, on considère habituellement la notion de sous-contexte au sens de l'inclusion

On peut maintenant définir l'habituelle notion de jugement de typage :

Définition 2.11 : *jugement de typage dans les systèmes explicites*

Soit $\mathfrak{T}\mathbf{x}$ un système de types pur explicite. Un *jugement de typage* est, comme précédemment, une expression de la forme $\Gamma \vdash_{\mathfrak{T}\mathbf{x}} M : N$, où M et N sont respectivement appelés *sujet* et *prédicat* du jugement. Le jugement est dit *valide* s'il est obtenu par dérivation à partir des figures d'inférence énoncées dans la table 3 ci-dessous. L'indice $\mathfrak{T}\mathbf{x}$ sera omis dans les cas où il n'y a pas d'ambiguïté.

$$\begin{array}{c}
\frac{(\bar{\sigma}, \bar{\tau}) \in \bar{\mathcal{A}}}{\vdash \bar{\sigma} : \bar{\tau}} \text{ (axiome)} \\
\\
\frac{\Gamma \vdash A : \rho \quad \Gamma, x : A \vdash B : \sigma \quad (\rho, \sigma, \tau) \in \mathcal{R}}{\Gamma \vdash \Pi x:A.B : \tau} \text{ (règle)} \\
\\
\frac{\Gamma \vdash A : \sigma \quad \bar{x} \notin \text{dom}(\Gamma)}{\Gamma, \bar{x} : A \vdash \bar{x} : A} \text{ (hypothèse)} \\
\\
\frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad \bar{x} \notin \text{dom}(\Gamma)}{\Gamma, \bar{x} : C \vdash A : B} \text{ (affaiblissement)} \\
\\
\frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)} \text{ (introduction de } \Pi \text{)} \\
\\
\frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B \langle x \leftarrow N \rangle} \text{ (élimination de } \Pi \text{)} \\
\\
\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M \langle x \leftarrow N \rangle : B \langle x \leftarrow N \rangle} \text{ (coupure)} \\
\\
\frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad A \in \Lambda \mathfrak{T}\mathbf{x} \quad P \langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P \langle x \leftarrow N \rangle : B} \text{ (expansion)} \\
\\
\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta\mathbf{x}} B}{\Gamma \vdash M : B} \text{ (conversion)}
\end{array}$$

TAB. 3 – *Figures d'inférence valides pour les systèmes de types purs explicites*

Il y a par ailleurs un certain nombre de conditions tacites à observer, du fait que les termes intervenant dans un contexte sont des termes de $\Lambda \mathfrak{T}\mathbf{x}$, tandis que le

sujet et le prédicat du jugement sont des termes de $\overline{\Lambda\mathcal{T}\mathbf{x}}$. Ainsi, dans la figure d'hypothèse (respectivement d'affaiblissement), A (respectivement C) est un terme de $\Lambda\mathcal{T}\mathbf{x}$, et ne peut pas appartenir à $\overline{\Lambda\mathcal{T}\mathbf{x}} \setminus \Lambda\mathcal{T}\mathbf{x}$. Ou encore, dans les figures de coupure et d'expansion, N doit-il vérifier cette même condition. En fait, toutes ces conditions seront nécessairement satisfaites, comme le montrera une proposition sur l'usage de pseudo-sorte, dont on trouvera l'énoncé p. 83.

Lorsqu'on donnera des arbres d'inférence, les noms des figures employées pourront être abrégées comme suit : (axiome) – (\mathcal{A}) ; (règle) – (\mathcal{R}) ; (hypothèse) – (hyp) ; (affaiblissement) – (W) ; (introduction de Π) – (Π -I) ; (élimination de Π) – (Π -É) ; (coupure) – (\dagger) ; (expansion) – (\mathbf{x}) ; (conversion) – (\equiv) ;

On peut faire différentes observations à la définition *supra* :

- i. à côté des variables se trouvent maintenant des γ -constantes ;
- ii. un nouveau prédicat, \int , a été adjoint ;
- iii. la figure d'affaiblissement est légèrement modifiée ;
- iv. deux nouvelles figures d'inférence (coupure et expansion) sont adjointes.

Ces points ont déjà été ou seront bientôt discutés, mais l'on va d'ores et déjà indiquer quelques notions importantes liées.

i. La question des γ -constantes a déjà été abordée au § 1.2 *supra*.

ii. La fonction du prédicat \int sera évoquée longuement au § 2.1 *infra*, mais pour pouvoir le manipuler dès à présent et présenter les nouvelles figures de typage, on introduit dès maintenant les notations suivantes :

Notation :

$\overline{\mathcal{S}}$ est la réunion disjointe $\mathcal{S} \uplus \{\int\}$ et $\overline{\mathcal{A}}$ est la réunion disjointe $\mathcal{A} \uplus \{(\overline{\sigma}, \int) / \overline{\sigma} \in \overline{\mathcal{S}}\}$.

Cette simple notation donne déjà une idée de l'utilité de \int : si l'on songe que les deux ensembles ci-dessus sont destinés à se substituer à ceux qu'ils complètent, on voit que le prédicat \int se comporte comme une sorte et fournit au système de typage un moyen d'identifier une sorte. Cette idée intuitive est relativement juste ; elle sera développée par la suite.

La notion de sorte typable ou terminale, qui existait dans les systèmes implicites (cf. § 3.1 *supra*), conserve tout son sens, bien que les sortes terminales semblent avoir ici acquis un nouveau type. En fait, comme on l'a déjà indiqué (et cela apparaîtra de manière plus flagrante par la suite), le prédicat \int est plutôt un mécanisme d'identification qu'un vrai type, si bien qu'il n'ôte pas son sens à cette distinction, que l'on rappelle ici :

Définition 2.12 : *sortes typables/terminales*

Soit $\mathfrak{T} = (\mathcal{S}, \mathcal{A}, \mathcal{R})$ un système de types pur explicite. Une sorte σ est dite *terminale* si $\{\sigma\} \times \mathcal{S} \cap \mathcal{A} = \emptyset$ – ou encore si $(\{\sigma\} \times \overline{\mathcal{S}}) \cap \overline{\mathcal{A}} = \{(\sigma, f)\}$. Dans le cas contraire, elle est dite *typable*.

iii. Avant de parler de la figure d'affaiblissement, l'on peut rappeler que celle-ci est nécessaire – et ne peut pas être ôtée du système et établie comme lemme – dans des formalismes tels que les systèmes de types purs, du fait que les dépendances entre termes et types exigent un certain nombre de vérifications de cohérence qui doivent être effectuées au fur et à mesure que la dérivation grandit. Si l'on compare la figure d'affaiblissement donnée ici :

$$\frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma}{\Gamma, \bar{x} : C \vdash A : B}$$

avec celle usuelle dans les systèmes de types purs implicites (cf. *e.g.* [5]), on constate que la prémisses « faible », qui dans le cas des systèmes implicites était $\Gamma \vdash C : \sigma$, avec Γ le même contexte que celui de la prémisses « forte » $\Gamma \vdash A : B$, est devenue $\Delta \vdash C : \sigma$, avec Δ une restriction de Γ . Ce changement repose sur l'idée qu'il n'est pas nécessaire de pouvoir ordonner totalement les assertions du contexte de typage : il suffit de s'assurer qu'elles ne contiennent pas de dépendance cyclique. Ceci rend les dérivations plus simples, au sens où cela évite un grand nombre d'affaiblissements superflus, qui seraient requis pour « compléter » Δ afin d'atteindre Γ . Cette simplification des dérivations sera très utile au moment de s'assurer que les dérivations obtenues en remontant un arbre à partir de sa conclusion sont en un certain sens plus petites que la conclusion de l'arbre, car la nouvelle figure facilite considérablement le déplacement de cette opération « administrative » (en tout cas sans influence sur le sujet et le prédicat du jugement) qu'est l'affaiblissement. Ce point sera abordé au § 2.3 *infra*.

iv. La figure d'inférence (coupure) est celle classiquement introduite pour typer les termes munis d'une substitution explicite. On peut étendre la correspondance de Curry-Howard à cette figure d'inférence en l'interprétant en logique intuitionniste comme une coupure – d'où son nom. En effet, si l'on oublie les sujets, on lit essentiellement une coupure :

$$\frac{\Gamma, A \vdash B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

On obtient ainsi la correspondance dite de Curry-Howard-De Bruijn (cf. chapitre I § 2.3 à propos de cette correspondance).

En outre, cette figure d'inférence semble naturelle, dans la mesure où l'on peut considérer que dans le cas des substitutions explicites, les termes $(\lambda x:A.M)N$ et $M\langle x \leftarrow N \rangle$ sont des preuves identiques de leur types, puisqu'ils aboutissent à la même forme \mathbf{x} -normale, *i.e.* se réduisent à la même preuve sans coupure. Or le premier des termes susdits se type de la façon suivante :

$$\frac{\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash \Pi x:A.B : \tau}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)} \text{ (}\Pi\text{-I)}}{\Gamma \vdash (\lambda x:A.M)N : B\langle x \leftarrow N \rangle} \text{ (}\Pi\text{-É)}$$

Si l'on considère globalement cette figure d'inférence (en éliminant toutefois la deuxième prémisse, $\Gamma \vdash \Pi x:A.B : \tau$, rendue non nécessaire par le fait que le prédicat $\Pi x:A.B$ n'est pas formé, si bien qu'il est superflu de vérifier sa correction), on retrouve, en remplaçant $(\lambda x:A.M)N$ par $M\langle x \leftarrow N \rangle$, la figure de coupure.

Cette figure d'inférence correspond à celle nommée (substitution) par R. BLOO dans [9], hormis que les substitutions explicites apparaissent non seulement dans le sujet du jugement mais également dans son prédicat, ce qui n'était pas le cas chez R. BLOO, qui évoquait certaines difficultés liées aux sortes terminales imposant d'utiliser des substitutions implicites dans les prédicats. Ces difficultés sont ici traitées grâce à un mécanisme nécessitant (et justifiant) l'adjonction de \int , et seront présentées plus en détail dans la partie consacrée précisément à ce mécanisme. Cette extension à la partie prédictive de l'usage des substitutions explicites est également visible dans la figure d'élimination du quantificateur Π .

La figure d'inférence de (coupure) est également l'équivalent, avec des noms explicites, de celle appelée (Clos $_{\Pi}$) par C. MUÑOZ dans [77].

Une comparaison entre ces diverses figures sera effectuée au § 2.1 *infra*.

La seconde figure d'inférence ajoutée par rapport au cas implicite est celle d'(expansion), qui a un usage double. D'une part, elle est une généralisation partielle de la figure d'inférence de largage – intitulée *drop* en anglais – introduite dans [30]. Cette dernière figure fut originellement introduite pour permettre à l'équivalent pour le calcul $\lambda\mathbf{x}$ des types à intersection de caractériser la normalisation forte, comme le fait ce système dans le cas implicite. En effet, en ne prenant pas en compte certains termes appliquant des substitutions explicites sur des variables inaccessibles, le système employant uniquement la coupure pour typer les termes ayant au sommet une substitution ne permettait pas le typage de certains de ces termes toutefois fortement normalisants. Ainsi, cette figure de largage augmente la puissance du système en lui permettant de typer plus de termes. D'autre part, la figure d'(expansion) généralise plusieurs figures permettant de « faire remonter » des substitutions explicites. Une discussion plus précise des fonctions de

cette figure d'inférence sera effectuée ultérieurement au § 2.1, consacré à l'étude d'(expansion), où l'on en donnera un exemple d'utilisation.

On a déjà mentionné, à propos de l'affaiblissement, la nécessité dans laquelle on se trouvera de considérer une forme d'ordre sur les dérivations, cet ordre reposant sur leur structure de manière à étendre la notion, usuelle, de sous-arbre propre. Cet ordre sera introduit au § 2.3, mais l'on va d'ores et déjà indiquer quels sont les éléments caractéristiques d'une dérivation sur lesquels on se reposera pour le définir :

Définition 2.13 : *nœuds principaux et sous-arbres d'une dérivation*

Soit (ϖ) une dérivation. On lui associe un entier $\mathcal{P}(\varpi)$ et un ensemble de sous-arbres $\mathcal{S}(\varpi)$, définis inductivement comme suit :

$$\begin{array}{ll}
\text{si } (\varpi) = \frac{\overline{\sigma} \in \overline{\mathcal{S}}}{\overline{\sigma} : \int}^{(\mathcal{A})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = 0 \\ \mathcal{S}(\varpi) = \emptyset \end{cases} \\
\text{si } (\varpi) = \frac{(\sigma, \tau) \in \mathcal{A}}{\sigma : \tau}^{(\mathcal{A})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = 1 \\ \mathcal{S}(\varpi) = \emptyset \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\varpi}^{(\mathcal{R})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi_1) + \mathcal{P}(\pi_2) + 1 \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2) \end{cases} \\
\text{si } (\varpi) = \frac{(\pi)}{\varpi}^{(\text{hyp})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi) + 1 \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi) \cup \{\pi\} \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma}^{(\text{w})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi_1) \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \{\pi_2\} \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\varpi}^{(\Pi - 1)} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi_1) + \mathcal{P}(\pi_2) + 1 \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2) \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\varpi}^{(\Pi - \acute{E})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi_1) + \mathcal{P}(\pi_2) + 1 \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2) \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\varpi}^{(\dagger)} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi_1) + \mathcal{P}(\pi_2) + 1 \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2) \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\varpi}^{(\text{x})} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\pi_1) + \mathcal{P}(\pi_2) \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2) \end{cases} \\
\text{si } (\varpi) = \frac{(\pi_1) \quad (\pi_2)}{\varpi}^{(\equiv)} & \text{alors } \begin{cases} \mathcal{P}(\varpi) = \max(\mathcal{P}(\pi_1), 1) + \mathcal{P}(\pi_2) \\ \mathcal{S}(\varpi) = \mathcal{S}(\pi_1) \cup \mathcal{S}(\pi_2) \end{cases}
\end{array}$$

Comme on l'a fait remarquer à propos de la figure d'affaiblissement, dans les systèmes de types purs existent des dépendances complexes entre variables, entre autres dans les contextes. On va maintenant donner quelques définitions concernant ces dépendances, à la suite de quoi on exposera certains éléments remarquables des dérivations, avant de s'intéresser de plus près aux contextes.

Dans toute la suite de ce chapitre, on s'efforcera de respecter les conventions d'écriture suivantes : les lettres latines minuscules désignent des variables ; les lettres latines majuscules, des termes ; les lettres grecques minuscules, des sortes (ou, lorsqu'elles sont surmontées d'une barre, une sorte ou le prédicat pseudo-sorte) ; les lettres grecques majuscules, des contextes de typage.

Dépendances entre variables

Les dépendances entre variables revêtent une importance particulière requérant une attention spéciale. En effet, le fait que les variables assignées dans le contexte puissent intervenir dans les prédicats suivants – ce qui d'ailleurs impose l'ordonnancement¹² des contextes – crée certaines contraintes pour la vérification de la correction des types employés.

Les dépendances en question sont décrites par le lemme suivant :

Lemme 2.11 : *Dépendances dans les systèmes de types purs explicites*

Soient $\Xi = \bar{u}_1 : V_1, \dots, \bar{u}_n : V_n$ un contexte de typage et $X : Y$ une assertion tels que le jugement $\Xi \vdash X : Y$ soit valide. Alors :

- (i) $av(X) \cup av(Y) \subseteq \text{dom}(\Xi)$;
- (ii) $\forall i \in \llbracket 1, n \rrbracket (av(V_i) \subseteq \{\bar{u}_j / 1 \leq j < i\})$.

Démonstration.

L'on procède par induction structurelle sur l'arbre de dérivation du jugement $\Xi \vdash X : Y$ en discriminant selon la dernière figure d'inférence employée :

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

- (i) *Ce point est évident car $av(X) \cup av(Y) = \emptyset = \text{dom}(\Xi)$.*
- (ii) *Ce point est trivial car Ξ est vide.*

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

¹²au moins partiel, comme on l'a fait remarquer précédemment

- (i) $av(X) \cup av(Y) = av(\Pi x:A.B) \cup av(\tau) = av(A) \cup (av(B) \setminus \{x\})$. Or l'application de l'hypothèse d'induction (i) aux deux prémisses donne respectivement que $av(A) \subseteq \text{dom}(\Gamma)$ et $av(B) \subseteq \text{dom}(\Gamma) \cup \{x\}$, ce qui implique $av(B) \setminus \{x\} \subseteq \text{dom}(\Gamma)$. Et donc $av(A) \cup (av(B) \setminus \{x\}) \subseteq \text{dom}(\Gamma)$.
- (ii) Ce point résulte directement de l'application de l'hypothèse d'induction (ii) à la prémisse $\Gamma \vdash A : \rho$.

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma \quad \bar{x} \notin \text{dom}(\Gamma)}{\Gamma, \bar{x} : A \vdash \bar{x} : A}$$

- (i) L'application de l'hypothèse d'induction (i) donne $av(A) \subseteq \text{dom}(\Gamma)$. Par ailleurs, deux cas se présentent :
- 1°) $\bar{x} = x$, et $av(A) \cup av(\bar{x}) = av(A) \cup \{x\} \subseteq \text{dom}(\Gamma) \cup \{x\} = \text{dom}(\Gamma, \bar{x} : A)$;
- 2°) $\bar{x} = \gamma_x$, et $av(A) \cup av(\bar{x}) = av(A) \subseteq \text{dom}(\Gamma) \cup \{\gamma_x\} = \text{dom}(\Gamma, \bar{x} : A)$.
- (ii) Soit $\bar{u}_i \in \text{dom}(\Xi) = \text{dom}(\Gamma, \bar{x} : A)$. Il y a deux possibilités.
- 1°) $\bar{u}_i \in \text{dom}(\Gamma)$, auquel cas la propriété découle directement de l'application de l'hypothèse d'induction (ii) à la prémisse $\Gamma \vdash A : \sigma$.
- 2°) $\bar{u}_i = \bar{x}$, auquel cas $\forall_i = A$; or par l'hypothèse d'induction (i) appliquée à $\Gamma \vdash A : \sigma$, l'on obtient que $av(A) \subseteq \text{dom}(\Gamma) = \{\bar{u}_j / 1 \leq j < i\}$.

$$d. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad \bar{x} \notin \text{dom}(\Gamma)}{\Gamma, \bar{x} : C \vdash A : B}$$

- (i) L'application de l'hypothèse d'induction (i) à la prémisse $\Gamma \vdash A : B$ donne $av(A) \cup av(B) \subseteq \text{dom}(\Gamma)$, ce qui implique le résultat désiré car $\text{dom}(\Gamma) \subseteq \text{dom}(\Gamma, \bar{x} : C) = \text{dom}(\Xi)$.
- (ii) Soit $\bar{u}_i \in \text{dom}(\Xi) = \text{dom}(\Gamma, \bar{x} : C)$. Il y a deux possibilités :
- 1°) $\bar{u}_i \in \text{dom}(\Gamma)$, auquel cas la propriété découle directement de l'application de l'hypothèse d'induction (ii) à la prémisse $\Gamma \vdash A : B$.
- 2°) $\bar{u}_i = \bar{x}$, auquel cas $\forall_i = C$; or par l'hypothèse d'induction (i) appliquée à $\Delta \vdash C : \sigma$, on a $av(C) \subseteq \text{dom}(\Delta)$. L'application dom étant un morphisme de l'ensemble des contextes muni de l'ordre partiel \subseteq sur $(\mathfrak{P}(\mathcal{V} \cup \{\gamma_u/u \in \mathcal{V}\}), \subseteq)$, l'on a $\text{dom}(\Delta) \subseteq \text{dom}(\Gamma) = \{\bar{u}_j / 1 \leq j < i\}$, et l'on tire de tout cela que $av(C) \subseteq \{\bar{u}_j / 1 \leq j < i\}$.

$$e. \text{ (introduction de } \Pi) : \frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

- (i) $av(\lambda x:A.M) \cup av(\Pi x:A.B) = av(A) \cup ((av(B) \cup av(M)) \setminus \{x\})$. Appliquée à la prémisse $\Gamma, x : A \vdash M : B$, l'hypothèse d'induction (i) indique que $av(B) \cup av(M) \subseteq \text{dom}(\Gamma) \cup \{x\}$ – d'où $(av(B) \cup av(M)) \setminus \{x\} \subseteq \text{dom}(\Gamma)$ –

tandis que l'hypothèse d'induction (ii) appliquée à cette même prémisse donne en particulier que $av(A) \subseteq \text{dom}(\Gamma)$. De tout ceci, l'on tire bien que $av(\lambda x:A.M) \cup av(\Pi x:A.B) \subseteq \text{dom}(\Gamma)$.

(ii) Ce point résulte directement de l'application de l'hypothèse d'induction (ii) à la prémisse $\Gamma \vdash \Pi x:A.B : \tau$.

$$f. \quad (\text{élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

(i) $av(MN) \cup av(B\langle x \leftarrow N \rangle) = av(M) \cup av(N) \cup (av(B) \setminus \{x\})$. Si l'on applique l'hypothèse d'induction (i) aux deux prémisses, on obtient respectivement que $av(M) \cup av(A) \cup (av(B) \setminus \{x\}) \subseteq \text{dom}(\Gamma)$ et $av(N) \cup av(A) \subseteq \text{dom}(\Gamma)$. De tout ceci, l'on déduit que $av(M) \cup av(N) \cup (av(B) \setminus \{x\}) \subseteq \text{dom}(\Gamma)$.

(ii) Ce point résulte directement de l'application de l'hypothèse d'induction (ii) à l'une quelconque des deux prémisses.

$$g. \quad (\text{coupure}) : \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

(i) $av(M\langle x \leftarrow N \rangle) \cup av(B\langle x \leftarrow N \rangle) \subseteq av(N) \cup ((av(M) \cup av(B)) \setminus \{x\})$. Si l'on applique l'hypothèse d'induction (i) aux deux prémisses, on obtient respectivement que $av(M) \cup av(B) \subseteq \text{dom}(\Gamma) \cup \{x\}$ – d'où l'on obtient que $(av(M) \cup av(B)) \setminus \{x\} \subseteq \text{dom}(\Gamma)$ – et $av(N) \cup av(A) \subseteq \text{dom}(\Gamma)$. De tout ceci, l'on déduit que $av(N) \cup ((av(M) \cup av(B)) \setminus \{x\}) \subseteq \text{dom}(\Gamma)$.

(ii) Ce point résulte directement de l'application de l'hypothèse d'induction (ii) à la prémisse $\Gamma \vdash N : A$.

$$h. \quad (\text{expansion}) : \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

(i) Ce point résulte directement de l'application de l'hypothèse d'induction à la prémisse $\Gamma \vdash M : B$ (en effet, d'après le lemme 2.1, les sujets de la prémisse et de la conclusion, étant \mathbf{x} -joignables, ont les mêmes variables accessibles).

(ii) Ce point résulte directement de l'application de l'hypothèse d'induction (ii) à la prémisse $\Gamma \vdash M : B$.

$$i. \quad (\text{conversion}) : \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta \mathbf{x}} B}{\Gamma \vdash M : B}$$

(i) En appliquant l'hypothèse d'induction (i) aux deux prémisses, l'on obtient respectivement que $av(M) \subseteq \text{dom}(\Gamma)$ et que $av(B) \subseteq \text{dom}(\Gamma)$, ce qui démontre ce point.

(ii) *Ce point résulte directement de l'application de l'hypothèse d'induction (ii) à une quelconque des deux prémisses.* Q.E.D.

Par ailleurs, comme on le disait *supra*, il est possible, dans une certaine mesure, d'interchanger une variable (substituable) et une γ -constante (insubstituable), en particulier dans le cas où elles sont libres, comme l'indique le lemme suivant :

Lemme 2.12 : *Interversion de variable et γ -constante*

Soient y et γ_y respectivement une variable et une γ -constante ; étant donné un contexte $\Xi = (\bar{u}_i : V_i)_{1 \leq i \leq n}$, on note $\Xi\{y \leftarrow \gamma_y\}$ le contexte $(\bar{u}'_i : V_i\{y \leftarrow \gamma_y\})_{1 \leq i \leq n}$ avec $\bar{u}'_i = \bar{u}_i$ si $\bar{u}_i \neq y$ et $\bar{u}'_i = \gamma_y$ sinon ; de manière analogue on définit $\Xi\{\gamma_y \leftarrow y\}$.

On a alors d'une part :

(i) si $(\pi) - \Xi \vdash X : Y$, alors $(\varpi) - \Xi\{y \leftarrow \gamma_y\} \vdash X\{y \leftarrow \gamma_y\} : Y\{y \leftarrow \gamma_y\}$;
de surcroît, $\mathcal{S}(\varpi) = \mathcal{S}(\pi)\{y \leftarrow \gamma_y\}$;

et d'autre part :

(ii) si $(\pi) - \Xi \vdash X : Y$, alors $(\varpi) - \Xi\{y \leftarrow \gamma_y\} \vdash X\{\gamma_y \leftarrow y\} : Y\{\gamma_y \leftarrow y\}$;
de surcroît, $\mathcal{S}(\varpi) = \mathcal{S}(\pi)\{\gamma_y \leftarrow y\}$;

avec, dans tous les cas, $\mathcal{P}(\varpi) = \mathcal{P}(\pi)$.

Démonstration.

Il suffit de prendre la dérivation et de remplacer chaque u par γ_u , ou réciproquement. La convention de Barendregt assure que les variables libérées au fur et à mesure que l'on remonte dans la dérivation sont distinctes de u . Q.E.D.

Dans toute la suite, on pourra omettre les barres indiquant que l'on a affaire à une variable ou une γ -constante en l'absence d'ambiguïté.

Contextes de typage

Certaines manipulations des contextes de typages sont nécessaires à la démonstration de divers résultats, ce pour quoi l'on introduit les notions suivantes.

En premier lieu, du fait des rapports de dépendance existant entre les différentes variables d'un contexte, il est nécessaire de définir le critère suivant :

Définition 2.14 : *contexte bien formé ; justification*

Soit $\Gamma = (\bar{x}_1 : A_1, \dots, \bar{x}_n : A_n)$ un contexte de typage.

Γ est dit *bien formé*, ce que l'on note $\Gamma \vdash$, si pour tout i de $\llbracket 1, n \rrbracket$, il existe k_i dans $\llbracket 1, i - 1 \rrbracket$ et σ_i dans \mathcal{S} tels que : $(\bar{x}_1 : A_1, \dots, \bar{x}_{k_i} : A_{k_i}) \vdash A_i : \sigma_i$.

En outre, si Γ est bien formé, on dit que l'ensemble des dérivations de la forme $(\bar{x}_1 : A_1, \dots, \bar{x}_{k_i} : A_{k_i}) \vdash A_i : \sigma_i$ susmentionnées est une *justification* de Γ .

Par convention, l'on fixe que le contexte vide admet pour justification l'ensemble vide. On peut par ailleurs noter qu'un contexte non vide admet en général plusieurs justifications distinctes.

Par exemple, soient σ et τ deux sortes telles que σ soit typable par τ terminale et (σ, σ, σ) soit une règle (e.g., dans le λ -cube, $\sigma = *$ et $\tau = \square$).

Les contextes suivants sont alors bien formés :

- le contexte vide $()$, et $\mathcal{J}(() \vdash) = \emptyset$;
- $(a : \sigma, b : \sigma, x : a)$, et $\mathcal{J}((a : \sigma, b : \sigma, x : a) \vdash) = \{ \vdash \sigma : \tau; a : \sigma \vdash a : \sigma \}$;
- $(a : \sigma, b : \sigma, x : a, y : b, z : a)$
 $\mathcal{J}((a : \sigma, b : \sigma, x : a, y : b, z : a) \vdash) = \{ \vdash \sigma : \tau; a : \sigma \vdash a : \sigma; b : \sigma \vdash b : \sigma \}$;
- $(a : \sigma, b : \sigma, x : a, f : \Pi y : a. b)$
 $\mathcal{J}((a : \sigma, b : \sigma, x : a, f : \Pi y : a. b) \vdash) = \{ \vdash \sigma : \tau; a : \sigma \vdash a : \sigma; a : \sigma, b : \sigma \vdash \Pi y : a. b : \sigma \}$;

tandis que $(a : \tau, \dots)$ n'est pas bien formé.

Par ailleurs, le contexte $(a : \sigma, h : \Pi u : a. \sigma)$ sera bien formé si, et seulement si, il existe un axiome de la forme $\sigma : \xi$ et une règle de la forme (σ, ξ, ν) . Pour rester dans l'exemple des systèmes du λ -cube avec $\sigma = *$, ce serait le cas dans le calcul des constructions (en prenant $\xi = \nu = \square$) mais pas dans les types simples.

Remarque :

Tout préfixe d'un contexte de typage bien formé est bien formé. En outre, il admet alors une justification incluse dans celle du contexte dont il est issu.

Cette propriété est en revanche évidemment fautive pour un sous-contexte.

On peut naturellement s'assurer que tous les contextes intervenant dans la dérivation d'un jugement de typage valide sont bien formés, comme l'énonce la propriété suivante :

Proposition 2.13 : Bonne formation

Soit Ξ un contexte de typage tel qu'il existe une assertion $X : Y$ tels que la dérivation $(\varpi) - \Xi \vdash X : Y$ soit valide. Alors Ξ est bien formé. En outre, il existe une justification de Ξ incluse dans $\mathcal{S}(\varpi)$.

Démonstration.

L'on va démontrer que $\Xi \vdash X : Y$ implique $\Xi \vdash$ en procédant par induction structurelle sur l'arbre de dérivation du jugement $\Xi \vdash X : Y$ en discriminant selon la dernière figure d'inférence employée, et en construisant parallèlement une justification $\mathcal{J}(\Xi \vdash)$.

- a. (axiome) : le contexte étant vide, sa bonne formation est trivialement satisfaite. En outre, on vérifie bien que $\mathcal{J}(\Xi \vdash) = \mathcal{S}(\varpi) = \emptyset$.

$$b. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma}{\Gamma, \bar{x} : A \vdash \bar{x} : A}$$

Par hypothèse d'induction, $\Gamma \vdash$ et $\mathcal{J}(\Gamma \vdash) \subseteq \mathcal{S}(\pi)$, où (π) désigne l'unique prémisses. En outre, la dernière vérification nécessaire est donnée par la prémisses $(\pi) - \Gamma \vdash A : \sigma$ elle-même. On obtient donc par définition de la bonne formation que $\Gamma, \bar{x} : A \vdash$, et l'on vérifie bien de surcroît la condition sur la vérification : en effet, l'on obtient $\mathcal{J}(\Gamma, x : A \vdash) \subseteq \mathcal{S}(\pi) \cup \{\pi\}$.

$$c. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma}{\Gamma, \bar{x} : C \vdash A : B}$$

Ce cas est analogue au précédent.

d. toutes les autres figures sont de la forme :

$$\frac{\dots \quad \Xi \vdash U : V \quad \dots}{\Xi \vdash X : Y}$$

avec, si l'on désigne par (π) la prémisses ainsi isolée, $\mathcal{S}(\varpi) \supseteq \mathcal{S}(\pi)$. Par hypothèse d'induction appliquée à (π) , $\Xi \vdash$ et $\mathcal{J}(\Xi \vdash) \subseteq \mathcal{S}(\pi)$, donc a fortiori $\mathcal{J}(\Xi \vdash) \subseteq \mathcal{S}(\varpi)$.

On vérifie donc bien dans tous les cas la bonne formation du contexte avec la condition requise pour la justification. Q.E.D.

Par ailleurs, pour manipuler par la suite les contextes, on aura usage de la notation suivante :

Notation :

Soient $\Gamma = (\bar{x}_i : A_i)_{1 \leq i \leq n}$ et $\Delta = (\bar{x}_i : B_i)_{1 \leq i \leq n}$ deux contextes de même support.

(i) Soit \xrightarrow{R} une relation de réduction sur $\Lambda \mathfrak{X} \mathbf{x}$. On écrit $\Gamma \xrightarrow{R} \Delta$ si pour tout contexte n -adique $C[\]$, $C[A_1, \dots, A_n] \xrightarrow{RC} C[B_1, \dots, B_n]$.

(ii) Soit \equiv_R une relation d'équivalence $\Lambda \mathfrak{X} \mathbf{x}$. On écrit $\Gamma \equiv_R \Delta$ si pour tout i de $\llbracket 1, n \rrbracket$, $A_i \equiv_R B_i$. On peut noter que la relation obtenue de cette façon est de manière générale une relation d'équivalence sur l'ensemble des contextes de même support.

Il convient de prendre garde à ces manipulations, car elles ne préservent pas nécessairement la bonne formation des contextes. Par exemple, on démontrera que la $\beta \mathbf{x}$ -réduction préserve la bonne formation du contexte, mais pas la $\beta \mathbf{x}$ -expansion : en effet, $z : *$ est bien formé, et $(\lambda x : * . *)y \xrightarrow{\beta \mathbf{x}} *$, mais $z : (\lambda x : * . *)y$ n'est pas bien formé (cf. lemme 2.11).

Remarque :

Pour la $\beta \mathbf{x}$ -réduction, la définition (i) équivaut à dire qu'il existe i de $\llbracket 1, n \rrbracket$ tel que $A_i \xrightarrow{\beta \mathbf{x}} B_i$ et pour tout j de $\llbracket 1, n \rrbracket \setminus \{i\}$, $A_j = B_j$.

2. Particularités remarquables

2.1. La figure d'inférence d'expansion

L'introduction de cette figure d'inférence :

$$\frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B} \text{ (expansion)}$$

répond à différents problèmes liés aux substitutions explicites.

D'une part, et c'est là la principale motivation à cette introduction, elle permet de résoudre le problème d'*intersion d'hypothèses* exposé *infra*. On suppose que l'on a ici typé le terme $(\lambda x:A.B)\langle y \leftarrow C \rangle$ dans un contexte Γ de façon naturelle en partant d'un typage de B dans un contexte $\Gamma, y : D, x : A$ au moyen de l'application successive d'une coupure et d'une introduction de Π . Si l'on souhaite ensuite, dans une optique de réduction du sujet par exemple, typer dans ce même contexte le même terme où la substitution a été propagée $\lambda x:A\langle y \leftarrow C \rangle.B\langle y \leftarrow C \rangle$, il est alors nécessaire d'invertir l'ordre dans lequel les deux hypothèses $y : D$ et $x : A$ sont éliminées : il faut tout d'abord effectuer la coupure, ce qui correspond à l'hypothèse y , puis effectuer une λ -abstraction, ce qui correspond à l'hypothèse x . Mais le problème dans les systèmes de types purs est que les contextes sont des listes ordonnées, et non de simples ensembles, ce qui est justifié par la possibilité que dans A occurre librement y , ce qui implique d'avoir préalablement à la déclaration de A comme type pour x une déclaration de type portant sur y afin de s'assurer de la correction des types et de la bonne formation du contexte, indispensable à la validité des dérivations. Il n'est donc ici pas possible d'invertir tout bonnement l'ordre de ces deux hypothèses à l'intérieur du contexte, comme on le ferait dans d'autres systèmes de types.

Dans le cas des substitutions implicites, ce problème est résolu à l'aide d'un *lemme de substitution*, lequel établit que si $\Gamma, y : D, \Delta \vdash M : N$ et $\Gamma \vdash C : D$, alors $\Gamma, \Delta\{y \leftarrow C\} \vdash M\{y \leftarrow C\} : N\{y \leftarrow C\}$. Ce procédé permet donc d'utiliser prématurément une hypothèse vouée à servir de pivot à une coupure, en répercutant sur les hypothèses ultérieures et sur le jugement en cours les effets de la coupure en question sous la forme d'une substitution. On résout ainsi le problème d'intersion.

Cependant, lorsque l'on se replace dans le cadre des substitutions explicites, on se heurte à l'impossibilité de transposer directement ce lemme : en effet, si, dans l'énoncé *supra*, M est par exemple une abstraction $\lambda x:P.Q$, le terme auquel on souhaite aboutir pour transposer la conclusion du lemme, $(\lambda x:P.Q)\langle y \leftarrow C \rangle$,

et le terme produit par le raisonnement inductif, $\lambda x:P\langle y\leftarrow C\rangle.Q\langle y\leftarrow C\rangle$, ne sont pas, comme leurs pendants l'étaient dans le cas des substitutions implicites, syntaxiquement égaux, mais simplement convertibles mutuellement en un pas de \mathbf{x} -réduction.

Afin que la réduction du sujet soit satisfaite dans les systèmes de types purs explicites, il faut permettre d'une manière ou d'une autre l'égalité de traitement de ces termes, en prévoyant un procédé permettant de faire « remonter » les substitutions – car le traitement inductif des jugements au moyen d'un lemme similaire au lemme de substitution ne pourra que les faire « descendre » dans le terme.

Une première possibilité envisagée fut d'associer à chaque figure d'inférence correspondant à une construction syntaxique inductive du terme (*i.e.* les figures de formation, d'introduction et d'élimination du Π , d'hypothèse et de coupure) une figure permettant de faire remonter une substitution – ainsi, l'on associerait *e.g.* à la figure d'élimination du Π la figure suivante :

$$\frac{\Gamma \vdash M\langle u\leftarrow U\rangle : (\Pi x:A.B) \quad \Gamma \vdash N\langle u\leftarrow U\rangle : A}{\Gamma \vdash (MN)\langle u\leftarrow U\rangle : B\langle x\leftarrow N\rangle}$$

mais cette solution présente de nombreux défauts, dont le principal est qu'elle double le nombre de figures d'inférences structurelles, et les nouvelles figures semblent maladroites car visiblement construites *ad hoc* pour remplir un besoin particulier. Il semblait donc plus approprié de construire une figure d'inférence spécifique pour traiter le problème en intégrant sous forme d'une nouvelle figure le procédé requis, d'où l'idée d'une figure d'inférence d'expansion, qui s'appliquerait à un jugement dont le sujet est issu par réduction de tête de la propagation d'une substitution, la finalité de la figure étant d'effectuer une opération, que l'on pourrait presque qualifier d'administrative, consistant à remonter en surface la substitution considérée. On obtiendrait ainsi une figure d'inférence de la forme :

$$\frac{\Gamma \vdash M : B \quad P\langle x\leftarrow N\rangle \xrightarrow{\gamma} M}{\Gamma \vdash P\langle x\leftarrow N\rangle : B}$$

où la relation $\xrightarrow{\gamma}$ de la seconde prémisse peut être n'importe laquelle des axiomes relationnels de propagation structurelle de substitution, nommément (quant), (abs), (app), (subst) et (comp). L'application de cette figure d'inférence immédiatement après l'une des figures structurelles avec la relation de réduction adaptée – (quant), (abs), (app), (subst) et (comp) correspondant chacun à l'une des cinq figures structurelles, respectivement la règle de formation, l'introduction et l'élimination du Π , l'hypothèse et la coupure – permet de simuler chacune des figures d'inférence supplémentaires évoquées *supra*.

D'autre part, on se rend compte qu'en étendant la figure d'inférence présentée *supra* à la règle de (gc), cette figure inclut la notion de largage, introduite dans [30] dans le cadre d'une extension au calcul à substitutions explicites λx du système D des types à intersection :

$$\frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad M \langle x \leftarrow N \rangle \xrightarrow{(gc)} M}{\Gamma \vdash M \langle x \leftarrow N \rangle : B} \text{ (largage).}$$

Cette figure d'inférence permet de typer plus de termes, ce qui explique son introduction dans le cadre du système D. En effet, il existe des termes non divergents mais *a priori* non typables du fait de contraintes artificiellement imposées par des substitutions « vaines » (*i.e.* portant sur des variables inaccessibles) rendant impossible le typage alors que ces substitutions, si leur opérande diverge pas, sont vouées à disparaître au cours de la x -normalisation¹³. Du point de vue du système D, dont l'un des principaux points forts est le fait qu'il caractérise les termes fortement normalisants, cette figure est donc indispensable. D. DOUGHERTY, P. LESCANNE ET S. LENGAND donnent l'exemple suivant (D est un système de types à la CURRY, *i.e.* sans annotations de type, cf. chapitre 1 p. 24) : $M = ((\lambda y.z)(xx)) \langle x \leftarrow \omega \rangle$; $M' = z \langle y \leftarrow xx \rangle \langle x \leftarrow \omega \rangle$. Le premier n'est pas fortement normalisant, le second l'est, mais comme l'application d'une coupure dans le typage de M' aurait exactement le même effet que l'application successive d'une introduction et d'une élimination de flèche dans le typage de M , sans la figure de largage ces deux termes sont soit tous deux typables, soit tous deux non typables, car les arbres permettant d'inférer leurs types, certes construits avec des figures d'inférence différentes, partagent exactement les mêmes prémisses. On voit ici que la figure d'inférence d'expansion utilisée avec l'axiome relationnel (gc) pallie ce problème (on se place ici dans le cas où $x \notin fv(P)$ pour simplifier) :

$$\frac{\Gamma \vdash P : B \quad \Delta \vdash N : A \quad P \langle x \leftarrow N \rangle \xrightarrow{(gc)}_x P}{\Gamma \vdash P \langle x \leftarrow N \rangle : B}$$

et l'on comprend l'introduction de la prémisse $\Delta \vdash N : A$: en effet, cette prémisse peut sembler superflue dans le cas où la propagation appliquée n'est pas (gc), car dans ce cas, la substitution $\langle x \leftarrow N \rangle$ apparaît déjà dans M , et il est aisé de montrer que dans ce cas N est typable, tandis que dans le cas du largage considéré ici, il est nécessaire d'ajouter cette prémisse, précisément pour conserver cette typabilité. Le fait que Δ soit indépendant de Γ peut sembler une complication inutile, dans la mesure où une même variable peut être introduite avec deux types n'ayant

¹³laquelle x -réduction est, pour mémoire, fortement normalisante

rien à voir dans les deux contextes. Si l'on souhaite éviter cela, il est possible d'ajouter la condition que $\Delta \supseteq \Gamma$, ce qui n'affecte pas les propriétés essentielles du système, et évite le problème mentionné. On constate par ailleurs que la relation de propagation ici présentée est légèrement plus forte que la propagation suggérée par les considérations exposées *supra*. Cette extension, motivée à l'origine par des considérations techniques, se trouve avoir du sens, comme on le verra un peu plus loin en abordant cette figure du point de vue de la synthèse de type.

Comme on le disait *supra* en évoquant le problème de l'interversion des hypothèses, un cas typique posant ce problème est celui de la réduction du sujet. En effet, l'on verra ultérieurement que la figure d'inférence d'expansion confère au système la propriété de réduction du sujet, laquelle propriété n'est pas automatique dans le cas des systèmes à substitutions explicites. En particulier, dans les systèmes introduits par R. BLOO dans [9], cette propriété n'est pas vérifiée, et l'on peut trouver dans [9, 10] le contre-exemple suivant :

$$(\lambda x:a.(\lambda z:a.z)x)\langle a \leftarrow b \rangle \xrightarrow[\beta x]{*} \lambda x:b.((\lambda z:a.z)x)\langle a \leftarrow b \rangle.$$

Avant de développer ce contre-exemple, on rappelle que les systèmes introduits par R. BLOO, comme vu dans la partie précédente, sont fort semblables aux systèmes ci-présentés, hormis en trois points, lesquels sont, premièrement, qu'ils n'introduisent point le prédicat particulier de pseudo-sorte (qui fera l'objet de développements ultérieurs, mais n'est pas utile ici) ; deuxièmement, qu'ils font intervenir les substitutions explicites uniquement dans les sujets des jugements de type, tandis que dans leurs prédicats, ils conservent les substitutions implicites (en particulier pour les figures d'inférence d'élimination du Π et de coupure)¹⁴ ; troisièmement, qu'ils ne comportent pas d'autre figure d'inférence que celle de coupure pour typer les termes commençant par une substitution, et ne présentent donc point la figure d'expansion.

On va ici donner quelques arbres d'inférence de type pour soutenir le propos, et, afin d'éviter de surcharger lesdits arbres, on laissera en guise de feuilles des jugements valides de la forme $\dots, u : V, \dots \vdash u : V$, dont la dérivation au moyen d'axiomes, d'hypothèses et d'affaiblissements est simple. En outre, les dérivations de bonne formation de Π aisées à construire seront omises, remplacées par le symbole (\mathcal{R}) surmontant trois points, et le symbole \blacktriangledown représentera la feuille (au sens ci-dessus) $b:*, \dots \vdash b:*$, vérification survenant dans diverses coupures, conversions, et expansions dans les arbres présentés *infra*.

¹⁴En donnant les arbres dans la version de R. BLOO, l'on fera néanmoins apparaître de manière explicite la substitution dans le prédicat par ces figures avant de la réaliser par conversion.

Si l'on considère donc la réduction donnée *supra*, il est aisé de constater que son membre gauche est typable, aussi bien dans les systèmes de R. Bloo que dans ceux définis dans la présente, pourvu que la hiérarchie de sortes du système permette la construction des types Π -quantifiés requis, ce qui est déjà le cas dans des systèmes aussi simples que $\lambda \rightarrow \mathbf{x}$, comme le montre l'arbre de dérivation suivant :

$$\begin{array}{c}
 \begin{array}{l}
 \text{(R)} \\
 \vdots \quad b : *, a : *, x : a, z : a \vdash z : a \\
 \frac{}{b : *, a : *, x : a \vdash \lambda z : a. z : \Pi z : a. a} \text{(\Pi-I)} \\
 \frac{b : *, a : *, x : a \vdash z : a}{b : *, a : *, x : a \vdash (\lambda z : a. z)x : a \langle z \leftarrow x \rangle} \text{(\Pi-É)} \\
 \frac{\vdots}{b : *, a : *, x : a \vdash (\lambda z : a. z)x : a} \text{(\equiv)} \\
 \frac{b : *, a : * \vdash (\lambda x : a. (\lambda z : a. z)x) : \Pi x : a. a}{b : * \vdash (\lambda x : a. (\lambda z : a. z)x) \langle a \leftarrow b \rangle : (\Pi x : a. a) \langle a \leftarrow b \rangle} \text{(\Pi-I)} \\
 \frac{\quad}{b : * \vdash (\lambda x : a. (\lambda z : a. z)x) \langle a \leftarrow b \rangle : \Pi x : b. b} \text{(\dagger)} \\
 \frac{\quad}{\quad} \text{(\equiv)}
 \end{array}
 \end{array}$$

DÉR. 1 – Typage dans $\lambda \rightarrow \mathbf{x}$ (toutes versions) de $(\lambda x : a. (\lambda z : a. z)x) \langle a = b \rangle$

Le membre droit de la réduction *supra*, en revanche, n'est pas typable, et l'on peut facilement, à défaut de démontrer rigoureusement cette impossibilité, en distinguant le motif, en considérant une tentative de typage dudit terme, par exemple dans $\lambda \rightarrow \mathbf{x}$, qui est pourtant l'un des systèmes de types purs parmi les plus simples que l'on puisse imaginer. On obtient en effet l'arbre de synthèse de type suivant :

$$\begin{array}{c}
 \begin{array}{l}
 \text{(R)} \\
 \vdots \quad b : *, x : b, a : *, z : a \vdash z : a \\
 \frac{}{b : *, x : b, a : * \vdash \lambda z : a. z : \Pi z : a. a} \quad \frac{}{b : *, x : b, a : * \vdash x : a} \\
 \frac{b : *, x : b, a : * \vdash (\lambda z : a. z)x : a \langle z \leftarrow x \rangle}{b : *, x : b, a : * \vdash (\lambda z : a. z)x : a} \text{(\equiv)} \\
 \frac{b : *, x : b, a : * \vdash (\lambda z : a. z)x : a}{b : *, x : b \vdash ((\lambda z : a. z)x) \langle a \leftarrow b \rangle : a \langle a \leftarrow b \rangle} \text{(\dagger)} \\
 \frac{\vdots}{b : *, x : b \vdash ((\lambda z : a. z)x) \langle a \leftarrow b \rangle : b} \text{(\equiv)} \\
 \frac{\quad}{b : * \vdash \lambda x : b. ((\lambda z : a. z)x) \langle a \leftarrow b \rangle : \Pi x : b. b} \text{(\Pi-I)}
 \end{array}
 \end{array}$$

DÉR. 2 – Tentative d'inférence de type dans $\lambda \rightarrow \mathbf{x}$ (version de R. Bloo) pour $\lambda x : b. ((\lambda z : a. z)x) \langle a = b \rangle$

Le symbole \dagger dans l'arbre *supra* indique l'endroit où la synthèse de type aboutit à un conflit. En effet, à cet endroit, on a dans le contexte $x : b$, et l'on a besoin

de dériver l’assertion $x : a$ – cette contrainte provient du bas de l’arbre. Toutefois, l’information que a et b peuvent être identifiés a été perdue lors de l’application de la figure de coupure trois nœuds plus bas. La figure d’inférence d’expansion résout ce problème. On donne ici en guise d’exemple une dérivation dans $\lambda \rightarrow$:

$$\begin{array}{c}
 \frac{b : *, x : b, z : a \langle a \leftarrow b \rangle, a : * \vdash z : a \langle a \leftarrow b \rangle \quad \nabla}{b : *, x : b, z : a \langle a \leftarrow b \rangle \vdash z \langle a \leftarrow b \rangle : a \langle a \leftarrow b \rangle} \quad (\dagger) \\
 \frac{\vdots \quad \frac{b : *, x : b, z : a \langle a \leftarrow b \rangle \vdash z \langle a \leftarrow b \rangle : b}{b : *, x : b \vdash \lambda z : a \langle a \leftarrow b \rangle . z \langle a \leftarrow b \rangle : \Pi z : b . b} \quad (\Pi\text{-I})}{b : *, x : b \vdash (\lambda z : a . z) \langle a \leftarrow b \rangle : \Pi z : b . b} \quad (\equiv) \quad \frac{b : *, x : b, a : * \vdash x : b \quad \nabla}{b : *, x : b \vdash x \langle a \leftarrow b \rangle : b \langle a \leftarrow b \rangle} \quad (\dagger)}{b : *, x : b \vdash (\lambda z : a . z) \langle a \leftarrow b \rangle : \Pi z : b . b} \quad (\equiv) \\
 \frac{b : *, x : b \vdash (\lambda z : a . z) \langle a \leftarrow b \rangle : \Pi z : b . b}{b : *, x : b \vdash (\lambda z : a . z) \langle a \leftarrow b \rangle x \langle a \leftarrow b \rangle : b \langle z \leftarrow x \langle a \leftarrow b \rangle \rangle} \quad (\Pi\text{-É}) \\
 \frac{\vdots \quad \frac{b : *, x : b \vdash (\lambda z : a . z) \langle a \leftarrow b \rangle x \langle a \leftarrow b \rangle : b}{b : *, x : b \vdash (\lambda z : a . z) x \langle a \leftarrow b \rangle : b} \quad (\equiv)}{b : * \vdash \lambda x : b . ((\lambda z : a . z) x) \langle a \leftarrow b \rangle : \Pi x : b . b} \quad (\Pi\text{-I})
 \end{array}$$

DÉR. 3 – Typage dans $\lambda \rightarrow \mathbf{x}$ de $(\lambda x : b . (\lambda z : a . z) x) \langle a := b \rangle$

On voit donc que le problème de R. BLOO est ici résolu.

Dans [77], C. MUÑOZ présente une manière de conserver la réduction du sujet pour des types dépendants dans un calcul à indice de De Bruijn $\lambda\phi$, qui est une variante de $\lambda\sigma$. Ceci est obtenu grâce à une modification de la notion de réduction, qui serait, dans un calcul à noms, exprimée comme suit :

$$\begin{array}{l}
 (\lambda y : A . B) \langle x \leftarrow C \rangle \longrightarrow \lambda y : A \langle x \leftarrow C \rangle . B \langle y : A, x \leftarrow C \rangle \\
 (\Pi y : A . B) \langle x \leftarrow C \rangle \longrightarrow \Pi y : A \langle x \leftarrow C \rangle . B \langle y : A, x \leftarrow C \rangle .
 \end{array}$$

Dans la substitution $\langle y : A, x \leftarrow C \rangle$, on remarque la présence de $y : A$, qui est ce que l’on nomme un *rappel de type* (*type reminder*). Sans entrer dans les détails, cette précaution permet de considérer dans $B \langle y : A, x \leftarrow C \rangle$ que y est de type A et non $A \langle x \leftarrow C \rangle$, comme on le voit dans les figures d’inférence de type, dont on donne ici une version dans un calcul à noms :

$$\frac{\Gamma, x : U \vdash y : V \{x \leftarrow X\} \quad \Gamma \vdash X : U}{\Gamma, x : U, \langle y : V \rangle \vdash y : V} \quad (\text{Cons})$$

$$\frac{\Gamma, x : U, \langle y : V \rangle \vdash A : B \quad \Gamma \vdash X : U}{\Gamma \vdash A \langle y : V, x \leftarrow X \rangle : B \{x \leftarrow X\}} \quad (\text{Clos}).$$

Cette construction permet également de résoudre le problème de la réduction du sujet, mais elle pose néanmoins certains problèmes, en particulier liés à la figure

(Clos), que C. MUÑOZ mentionne dans la conclusion de [77]. Ces problèmes sont résolus par la figure d'expansion).

Une autre approche pour obtenir des systèmes de types purs explicites avec réduction du sujet est donnée à la fin de [9], s'inspirant largement de l'article [12] de R. BLOO, F. KAMAREDDINE et R. NEDERPELT, où ceux-ci introduisent une notion de systèmes de types purs avec *définitions*. Une définition est une substitution placée dans le contexte de typage. Dans ce système, la substitution est introduite d'abord dans le contexte, au moyen des figures suivantes :

$$\frac{\Gamma \vdash A : \sigma \quad \Gamma \vdash B : A \quad x \notin \text{dom}(\Gamma)}{\Gamma, \langle x \leftarrow B \rangle \vdash x : A}$$

$$\frac{\Gamma \vdash A : \sigma \quad \Gamma \vdash B : A \quad \Gamma \vdash M : T \quad x \notin \text{dom}(\Gamma)}{\Gamma, \langle x \leftarrow B \rangle \vdash M : T}$$

Après quoi, elle peut être introduite dans les termes :

$$\frac{\Gamma, \langle x \leftarrow B \rangle \vdash M : T}{\Gamma, \langle x \leftarrow B \rangle \vdash M \langle x \leftarrow B \rangle : T \langle x \leftarrow B \rangle}$$

En outre, il est nécessaire d'adapter la figure de conversion :

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \sigma \quad \Gamma \vdash A \cong B}{\Gamma \vdash M : B}$$

avec $\Gamma \vdash A \cong B$ signifiant que A et B sont joignables par β x-conversion aux substitutions survenant dans Γ près.

Cette approche résout de manière assez satisfaisant le problème, mais elle présente le désavantage de compliquer les notions liées aux contextes et les règles les manipulant.

Par ailleurs, on pourrait objecter que la figure d'inférence d'expansion n'est pas satisfaisante du point de vue de l'inférence de type, car elle effectue une forme d'expansion du sujet. Mais en fait, c'est tout le contraire, car si l'on se place du point de vue de la synthèse de type, les figures d'inférence doivent être lues de bas en haut et non de haut en bas. Et si l'on considère ce sens de lecture, on peut lire la figure d'expansion comme suit : si l'on veut typer un terme quelconque commençant par au moins une substitution explicite, on peut pousser d'un cran l'une des substitutions situées « en surface » du terme, et si l'on trouve un type pour le terme obtenu, et que l'opérande de la substitution poussée était bien typable, alors on peut donner au terme initial le type du terme obtenu en poussant la substitution.

On retrouve ici l'idée somme tout naturelle et intuitive de voir les substitutions explicites comme des substitutions « paresseuses », qui restent en place dans le terme et ne sont propagées que lorsque cela s'avère nécessaire. Si l'on reprend les deux dérivations *supra*, l'on peut voir une parfaite illustration de cette notion : en effet, la tentative de typage de $(\lambda x:a.(\lambda z:a.z)x)\langle a \leftarrow b \rangle$ dans la dérivation 2 au moyen d'une coupure ayant échoué, l'on revient en arrière et, dans la dérivation 3, l'on effectue, au lieu d'une conversion et de la coupure, une expansion. On peut alors, comme dans la dérivation 2, tenter une élimination de Π . Cette fois-ci, comme la substitution a été poussée d'un cran, l'information que a et b sont interchangeable est conservée, et l'on parvient à typer $x\langle a \leftarrow b \rangle : b$, au lieu de se heurter, comme précédemment, à un blocage provenant du conflit entre les deux contraintes $x : a$ et $x : b$, l'information de synonymie entre a et b ayant été perdue.

Comme on le voit illustré dans cet exemple, il semble que ce mécanisme puisse présenter un intérêt certain dans le cadre de la synthèse de type. En particulier, il traite le problème évoqué au chapitre 1 de la question délicate de la composition de la substitution. On voit en fait d'où vient la nécessité d'avoir deux figures pour traiter les termes commençant par une substitution explicite : lorsque l'on se trouve face à un terme commençant par une substitution, deux cas se présentent : soit on peut la traiter tout de suite au moyen d'une coupure, soit il faut retarder cette substitution au moyen d'une expansion. Ceci est patent dans l'exemple *supra*. Si l'on reprend la dérivation 2 (qui peut également être vue comme une tentative de dérivation dans les systèmes présentés ici), on voit que l'on a effectué une coupure, mais que l'on est parvenu à un blocage, tandis que dans la dérivation 3, on reprend le même typage, mais l'on fait au lieu de cette coupure une expansion pour retarder l'application de cette substitution, le typage peut aboutir.

On pourrait également éprouver quelque crainte à la vue du fait que l'on permette par la figure d'(expansion) d'effectuer des compositions de substitutions, car on pourrait ainsi construire des arbres infinis. Mais en fait, il serait parfaitement inutile de l'appliquer deux fois à une même paire de substitution, car on reviendrait alors à la même position relative des variables substituées en ayant simplement compliqué les termes. Il faut donc admettre que cette figure d'(expansion) peut, de même que la conversion ou l'affaiblissement, être répétée inutilement un grand nombre de fois, et se remettre à la clairvoyance de l'utilisateur – ou, le cas échéant, de la procédure de typage – pour que cela ne soit pas le cas. L'on pourrait également envisager de se reposer, non plus sur λx , mais sur un calcul muni d'une notion de composition déterministe, par exemple comme ceux introduits par R. DAVID ET B. GUILLAUME mentionnés au chapitre 1.

2.2. Le prédicat « pseudo-sorte »

La nécessaire adjonction du pseudo-type \int provient du fait que dans les systèmes explicites introduits *supra*, les substitutions explicites sont appliquées non seulement aux expressions intervenant comme sujet mais également comme prédicat d'un jugement de typage – au contraire, par exemple, des systèmes introduits par R. BLOO, lesquels, comme on peut le voir *e.g.* dans [9] appliquent, comme on l'a dit précédemment, des substitutions implicites aux dernières.

Ce choix – délibéré, et motivé par les diverses considérations sur les mérites comparés des substitutions explicites et implicites mentionnées *supra* – implique la possibilité de voir survenir d'autres types, et en particulier des types de la forme $\sigma \langle x_1 \leftarrow N_1 \rangle \dots \langle x_k \leftarrow N_k \rangle$. On reconnaît ici, sous des substitutions que l'on a précédemment qualifiées de vaines (car elles portent sur des variables inaccessibles), une sorte, et il s'avère naturellement nécessaire de pouvoir identifier d'une façon ou d'une autre tous les termes composés d'une sorte « cachée » sous ses substitutions vaines et la sorte en question.

Il semble alors tout naturel de vouloir utiliser pour cette identification la figure d'inférence de conversion de types, déjà existante dans les systèmes de types purs usuels, qui s'écrit comme suit :

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

En effet, on peut vouloir appliquer cette figure au cas où les deux termes A et B représentent (*i.e.* sont équivalents à) une même sorte.

Toutefois, on se heurte à la vérification de correction imposée par la prémisse $\Gamma \vdash B : \sigma$. Le rôle, dans les systèmes de types purs usuels, de cette prémisse est d'assurer que seuls des termes représentant effectivement des types, que l'on appelle types corrects, surviennent comme prédicat d'un jugement.

Dans les systèmes de types purs usuels (implicites), on entend habituellement par type correct un terme typable par une sorte, ou encore une sorte elle-même (cas des sortes terminales). Dans le cadre des substitutions implicites, on est sûr que l'on aura jamais à effectuer de conversion sur une sorte terminale, comme le montre le lemme suivant :

Proposition 2.14 : *Types convertibles à une sorte terminale*

Soit \mathfrak{T} un système de types pur implicite. Soient Γ un contexte de typage, $A : B$ d'une assertion et ξ une sorte terminale tels que $\Gamma \vdash_{\mathfrak{T}} A : B$ et $B \equiv_{\beta} \xi$. Alors $B = \xi$.

Démonstration.

L'on procède par induction sur la dérivation du jugement $\Gamma \vdash A : B$ en discriminant selon la dernière figure employée.

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

Ce cas est évident.

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

Ce cas est évident.

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Ce cas est trivial car si $A \equiv_{\beta} \xi$, alors, comme ξ est β -irréductible, on a $A \xrightarrow{\beta}^* \xi$, et donc, par le théorème de réduction du sujet 1.7, $\Gamma \vdash \xi : \sigma$, et par engendrement $(\xi, \sigma) \in \mathcal{A}$, ce qui contredit le fait que ξ soit terminale.

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

Ce cas est évident par hypothèse d'induction.

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

Ce cas est trivial car $\Pi x:A.B$ ne peut être β -convertible à une sorte.

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

Dans ce cas, par le théorème de correction des types 1.6, comme $(\Pi x:A.B)$ n'est pas une sorte, $\Gamma \vdash (\Pi x:A.B) : \tau$. On applique alors le lemme d'engendrement pour obtenir qu'il existe une sorte σ telle que $\Gamma, x : A \vdash B : \sigma$. On a alors, par le lemme de substitution 1.5, que $\Gamma \vdash B\{x \leftarrow N\} : \sigma$. On conclut alors comme dans le cas c. que si $B\{x \leftarrow N\} \equiv_{\beta} \xi$, alors $(\xi, \sigma) \in \mathcal{A}$, et donc ξ n'est pas terminale, et la propriété est triviale.

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

Ce cas est évident par hypothèse d'induction.

Q.E.D.

On voit donc dans ce cas qu'il est inutile d'appliquer la figure de conversion dans le cas où le prédicat est une sorte terminale dans le cadre des substitutions implicites, c'est pourquoi il n'est pas nécessaire de prévoir dans la figure de conversion le cas où B est une sorte non typable.

Toutefois, le problème particulier soulevé *supra* montre bien que, dans le cas de substitutions explicites, ce problème d'identification de termes mutuellement

convertibles s'étend aux sortes terminales. Comme on l'a vu, il semble naturel de réaliser cette identification comme un cas particulier de l'application de la conversion, puisque celle-ci permet déjà d'effectuer dans un autre cadre une opération analogue. Mais pour permettre ceci, il appert nécessaire d'étendre la prémisse $\Gamma \vdash B : \sigma$. La question qui se pose ensuite est de savoir comment la rendre à la fois assez large pour résoudre le problème ci-dessus, et assez stricte pour préserver la correction.

La première extension qu'il semble naturel de considérer au vu de la discussion précédente est la prémisse $(\Gamma \vdash B : \sigma) \vee (B \in \mathcal{S})$. Cependant, cette prémisse n'est pas satisfaisante car elle produit une figure de conversion dirigée. En effet, elle permet certes de passer d'un jugement de la forme $\Gamma \vdash A : \sigma \langle x_1 \leftarrow N_1 \rangle \dots \langle x_k \leftarrow N_k \rangle$ à un jugement de la forme $\Gamma \vdash A : \sigma$, elle ne permet pas le passage dans l'autre sens, ce qui n'est pas satisfaisant car, contrairement à ce que l'on pourrait croire, ledit sens peut être nécessaire, car l'on peut être amené à faire apparaître des substitutions vaines (*e.g.* pour former un type produit dans une optique de réduction du sujet), et non toujours à les faire disparaître.

On peut alors envisager une extension plus large en disant que si B n'est pas typable par une sorte, alors il doit effectivement être une sorte, mais non être au sens de la trop stricte égalité syntaxique, mais être au sens de la plus souple β x -conversion, ce que traduit $(\Gamma \vdash B : \sigma) \vee (B \equiv_{\beta x} \tau)$. Toutefois, il est aisé de se rendre compte que cette prémisse ne convient pas car elle est trop large. En effet, elle permet l'apparition non contrôlée de termes, car aucune condition n'est imposée sur les opérandes des substitutions vaines dont elle permet l'apparition. Et cette remarque reste un obstacle dans le cas où l'on se contenterait d'une simple x -conversion en adoptant la variante $(\Gamma \vdash B : \sigma) \vee (B \equiv_x \tau)$. En effet, dans les deux cas, l'on peut effectuer à partir d'un quelconque jugement valide de sorte $\Gamma \vdash M : \sigma$ la construction suivante :

$$\frac{\begin{array}{c} \vdots \\ \Gamma \vdash M : \sigma \end{array} \quad \sigma \langle x \leftarrow \Omega \rangle \equiv_{(\beta)x} \sigma}{\Gamma \vdash M : \sigma \langle x \leftarrow \Omega \rangle} \text{ (conversion)}$$

et faire ainsi survenir arbitrairement dans les jugements de sortes (qui sont nombreux, car nécessaires à chaque introduction de type) un terme divergeant. La nouvelle prémisse considérée ne peut donc convenir.

Les petits exemples que l'on vient de voir montrent bien qu'il est nécessaire de permettre une forme de x -conversion afin de pouvoir non seulement retrancher des substitutions vaines, mais aussi d'en ajouter, pourvu que l'opérande de la substitution soit contrôlé – et par contrôlé, il est naturel dans ce contexte d'entendre

typable. On ramène ainsi le problème à la question suivante : comment exprimer que B est une sorte à laquelle a été appliquée une succession de substitutions d'opérandes typables ?

La caractérisation syntaxique des termes susmentionnés n'étant pas évidente, on peut avoir l'idée d'introduire un mécanisme de vérification sous forme d'un jugement indiquant que B a la forme requise, ce pour quoi l'on introduit le symbole \int , qui, de même qu'une sorte caractérise intuitivement une classe de types (comme $*$ et \square dans le λ -cube caractérisent les types de termes et les types de types), caractérisera d'une certaine façon les types sortes, indépendamment de la hiérarchie de celles-ci. Ainsi, par un jugement de la forme $\Gamma \vdash B : \int$, on voudra indiquer que B est un représentant possible d'une sorte quelconque. Afin de dériver ces jugements, on peut, à partir d'un axiome de la forme $\sigma : \int$, appliquer successivement coupures, affaiblissements et expansions pour ajouter les substitutions adéquates. On remarque en particulier que la notion de largage suivante (provenant de [30]) :

$$\frac{\Gamma \vdash A : B \quad \Delta \vdash C : D \quad A \langle x \leftarrow C \rangle \xrightarrow{(gg)} A}{\Gamma \vdash A \langle x \leftarrow C \rangle : B} \text{ (largage)}$$

qui est capturée par la figure d'inférence d'expansion (cf. § 2.1), permet en pratique de dériver les vérifications requises pour les termes considérés.

En outre, la notion de pseudo-sorte trouve une justification supplémentaire du fait qu'elle possède une signification en terme de correction des types, qui sera abordée au paragraphe suivant.

Par ailleurs, il est nécessaire ici de préciser que, bien que pour des raisons de commodité l'on manipule cette pseudo-sorte comme une sorte, en particulier à l'aide des figures d'inférence d'axiome et d'expansion, il ne s'agit pas d'une sorte, et il n'est pas envisageable de l'ajouter à l'ensemble des sortes, car il est notoire que les systèmes comportant la sorte de toute les sortes sont en général incohérents. Cet écueil est ici évité car ce prédicat \int ne peut pas intervenir dans l'ensemble des règles, et ne peut pas non plus apparaître comme prédicat d'une assertion de typage élément d'un contexte (on voit en effet que les figures d'inférence (hypothèse) et (affaiblissement), seules à agrandir le contexte, vérifient que le type introduit est typable par une vraie sorte – ce que n'est pas la pseudo-sorte. En fait, on peut même garantir que l'usage de \int n'échappe pas au cadre qu'on lui a fixé par une proposition dont la démonstration sera donnée ultérieurement dans ce chapitre, p. 122, du fait qu'elle utilise des résultats qui seront présentés dans les prochains paragraphes, mais dont voici l'énoncé :

Proposition 2.15 : *Usage de pseudo-sorte*

Soient Γ un contexte et $M : B$ une assertion tels que $\Gamma \vdash M : B$. Alors :

- (i) si $B \equiv_{\beta x} \int$, alors il existe $\bar{\sigma} \in \bar{\mathcal{S}}$, $x_1, \dots, x_k \in \mathcal{U}$, $N_1, \dots, N_k \in \Lambda \mathfrak{X}$ des termes tels que $M = \bar{\sigma} \langle x_1 \leftarrow N_1 \rangle \cdots \langle x_k \leftarrow N_k \rangle$.
- (ii) si $M \equiv_{\beta x} \int$, alors $B \equiv_{\beta x} \int$.

Correction des types

Ces jugements de prédicat \int sont tous destinés à assurer que le terme qui leur sert de sujet est une sorte déguisée, ce qui est un rôle bien distinct de celui des autres jugements. Pour pouvoir les distinguer, on introduit la définition suivante, leur donnant un nom :

Définition 2.15 : *jugement de sorte*

On dit qu'un jugement $\Gamma \vdash M : B$ est un *jugement de sorte* si $B \equiv_{\beta x} \int$.

Ce type de jugement est indispensable, car on ne saurait se passer de sortes terminales, mais, dans le cas implicite, celles-ci ne sont pas traitées car, pour ainsi dire, il ne peut rien leur arriver tant leur usage est limité – elles ne peuvent apparaître que comme type dans une vérification de la forme $\Gamma \vdash A : \tau$ précédant l'introduction de A dans un contexte dans une figure d'hypothèse ou d'affaiblissement, l'introduction d'un type produit dans le cas où A en est un, ou la conversion d'un type équivalent à A en A , et elles ne peuvent pas servir de prédicat dans un contexte de typage. Ce dernier point reste d'ailleurs vrai dans le cas explicite ; simplement, le système contient un mécanisme supplémentaire pour traiter ces sortes, mais la différence demeure entre une sorte typable uniquement par \int (qui sera terminale), et une sorte typable également par une autre sorte. Les deux correspondent à une notion de correction des types, mais avec une nuance, semblable à celle entre être une sorte et être typable par une sorte dans le système implicite. On pourrait d'ailleurs envisager, à la lumière de ceci, de se restreindre aux axiomes de la forme (σ, τ) et (ρ, \int) où ρ est une sorte terminale. Toutefois, il est plus simple de conserver tous les axiomes de prédicat \int , et c'est ce que l'on fera.

Le résultat suivant, le théorème de correction des types, est indispensable dans les systèmes où, comme c'est le cas pour les systèmes de types purs, on n'établit pas de distinction syntaxique entre termes et types. Ce théorème suppose une caractérisation des termes ayant un comportement de type (ici, la typabilité par une sorte ou pseudo-sorte – cf. paragraphe précédent), et établit que tout terme apparaissant comme prédicat dans une dérivation est correct en ce sens.

Théorème 2.16 : *Correction des types des systèmes de types purs explicites*
 Soit $\Gamma \vdash A : B$ un jugement de type valide. Alors : $\exists \bar{\sigma} \in \bar{\mathcal{S}}, \Gamma \vdash B : \bar{\sigma}$.

Démonstration.

Ce lemme requiert le lemme d'engendrement qui sera démontré au paragraphe suivant, et sera démontré à sa suite.

2.3. Ordre sur les dérivations

Les particularités des substitutions explicites font, comme on l'a dit précédemment, qu'il est parfois nécessaire, dans certains raisonnements par induction, de reconstituer la dérivation – ce que l'on fait au moyen du classique lemme d'engendrement¹⁵ – puis d'appliquer l'hypothèse d'induction aux dérivations ainsi obtenues. Toutefois, celles-ci ne sont en général pas des sous-arbres de la dérivation principale considérée, ce qui met en péril la bonne fondation de l'induction, et indique en tout cas que la simple induction structurelle sur les dérivations ne suffit plus. Toutefois, ces dérivations sont bien en un sens plus petites que la dérivation principale, et il semble donc bien qu'il soit possible d'effectuer une induction bien fondée dans ce cas, pourvu qu'on formalise de manière correcte, *i.e.* en introduisant un ordre bien fondé sur les dérivations de typage.

La principale difficulté est le fait que le lemme d'engendrement supprime dans les dérivations une figure structurelle, mais repousse plus haut dans l'arbre de dérivation, et éventuellement duplique, les éventuelles figures d'affaiblissement suivant cette dernière, et, par voie de conséquence, également les figures conduisant à la dérivation de la prémisse faible de ces figures. Cela empêche de considérer simplement le nombre de figures employées dans la dérivation, ce qui eût été un moyen simple d'obtenir un ordre bien fondé sur les dérivations. Cette difficulté rend également de considérer un ordre multi-ensemble induit par une préséance plus faible pour l'affaiblissement que pour les autres figures. De même pour l'ordre récursif de chemin, pourtant souvent efficace dans ce genre de contexte : le fait que l'on ne puisse savoir ni combien de fois les affaiblissements seront dupliqués ni à quelle hauteur (*i.e.* derrière combien de nœuds) dans l'arbre ils seront déplacés empêchent d'employer cette approche.

Une première chose à faire pour résoudre ce problème de duplication de sous-arbre est de « curryfier » l'affaiblissement, *i.e.* de ne pas le considérer comme un nœud binaire, mais comme un nœud unaire indicé par une dérivation :

¹⁵Le lecteur peut se reporter à l'énoncé du lemme d'engendrement 1.4 pour les systèmes de types purs implicites afin de se remémorer la manière dont il permet de remonter les dérivations

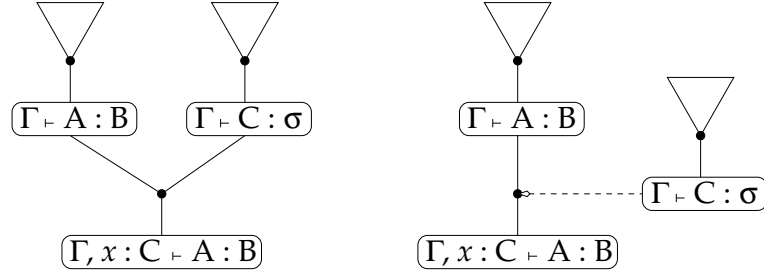


FIG. 5 – « Curryfication » de l'affaiblissement

On peut, à partir de là, identifier une sorte de « trace principale » dans une dérivation, et s'intéresser plus spécifiquement aux nœuds intervenant sur celle-ci en les comptant, en omettant toutefois les conversions – c'est pourquoi l'on introduit le nombre $\mathcal{P}(\cdot)$ dans la définition 2.13. Toutefois, l'on ne peut pas omettre les dérivations « auxiliaires », puisque, afin d'obtenir un ordre bien fondé, l'on doit naturellement conserver la propriété qu'une sous-dérivation propre est strictement inférieure à la dérivation dans laquelle elle apparaît – c'est pourquoi on en garde trace dans l'ensemble $\mathcal{S}(\cdot)$ dans la définition 2.13. En outre, toujours pour préserver cette propriété de sous-dérivation propre, on doit également prendre en compte le cas particulier des conversions.

L'ordre que l'on va définir tient compte de toutes ces composantes :

Définition 2.16 : *ordre de simplicité sur les dérivations*

Soient (π) et (ϖ) deux dérivations. On dit que (π) est *plus simple* que (ϖ) , ce que l'on note $(\pi) \triangleleft (\varpi)$, si l'une des trois conditions suivantes est remplie :

- (i) $(\pi) \triangleleft_{\mathcal{S}} (\varpi)$, i.e. $(\pi) \in \mathcal{S}(\varpi)$;
- (ii) $(\pi) \triangleleft_{\mathcal{N}} (\varpi)$, i.e. $\mathcal{S}(\pi) \subseteq \mathcal{S}(\varpi) \wedge \mathcal{P}(\pi) < \mathcal{P}(\varpi)$
- (iii) $(\pi) \triangleleft_{\mathcal{I}} (\varpi)$, i.e. $\mathcal{S}(\pi) \subseteq \mathcal{S}(\varpi) \wedge \mathcal{P}(\pi) = \mathcal{P}(\varpi) \wedge (\pi)$ est une prémisse de (ϖ) .

On note $\triangleleft_{\mathcal{P}}$ l'union de $\triangleleft_{\mathcal{N}}$ et $\triangleleft_{\mathcal{I}}$. La clôture transitive de \triangleleft , que l'on dénotera par $<$, est nommée *ordre de simplicité* sur les dérivations.

Naturellement, pour pouvoir utiliser la relation $<$ dans des raisonnements par induction, il est nécessaire de montrer qu'il s'agit d'un ordre strict bien fondé :

Proposition 2.17 : *Bonne fondation de l'ordre de simplicité*

Les assertions suivantes sont vraies :

- (i) Si $\varpi_0 \triangleright_{\mathcal{P}} \cdots \triangleright_{\mathcal{P}} \varpi_k \triangleright_{\mathcal{S}} \pi$, alors $(\pi) \in \mathcal{S}(\varpi_0)$.
- (ii) Il n'existe pas de chaîne infinie $\pi_0 \triangleright \pi_1 \triangleright \cdots \triangleright \pi_i \triangleright \cdots$.
- (iii) La relation $<$ est un ordre strict bien fondé.

Démonstration.

(i) On raisonne par récurrence sur k .

Si $k = 0$, c'est évident par définition.

On suppose la propriété vraie pour k . Soit alors $\varpi_0 \triangleright_{\mathcal{P}} \cdots \triangleright_{\mathcal{P}} \varpi_{k+1} \triangleright_{\mathcal{S}} \pi$; on a, par hypothèse de récurrence, que $(\pi) \in \mathcal{S}(\varpi_1)$. En outre, $\mathcal{S}(\varpi_1) \subseteq \mathcal{S}(\varpi_0)$, donc $(\pi) \in \mathcal{S}(\varpi_0)$.

(ii) Soit $(\pi_i)_{i \in \mathbb{N}}$ une suite infinie telle que pour tout $i \in \mathbb{N}$, $\pi_i \triangleright \pi_{i+1}$. Pour tout $i \in \mathbb{N}$, soit $\pi_i \triangleright_{\mathcal{S}} \pi_{i+1}$, soit $\pi_i \triangleright_{\mathcal{N}} \pi_{i+1}$, soit $\pi_i \triangleright_{\mathcal{I}} \pi_{i+1}$. Plusieurs cas se présentent :

1°) Il y a une infinité de $\triangleright_{\mathcal{S}}$. On a alors : $\pi_0 \triangleright_{\mathcal{P}}^{k_0} \triangleright_{\mathcal{S}} \pi_{i_1} \triangleright_{\mathcal{P}}^{k_1} \triangleright_{\mathcal{S}} \pi_{i_2} \triangleright_{\mathcal{P}}^{k_2} \triangleright_{\mathcal{S}} \cdots$, et l'on obtient donc grâce au point (i) une suite $(\pi_{i_j})_{j \in \mathbb{N}}$ telle que pour tout j , $\pi_{i_{j+1}}$ soit élément de $\mathcal{S}(\pi_{i_j})$, donc en particulier un sous-arbre propre de π_{i_j} , et l'on obtient une suite infinie strictement décroissante au sens de la relation de sous-arbre, ce qui est impossible.

2°) À partir d'un certain rang n , il n'y a plus de $\triangleright_{\mathcal{S}}$ mais il y a une infinité de $\triangleright_{\mathcal{N}}$. On a alors : $\pi_n \triangleright_{\mathcal{I}}^{k_0} \triangleright_{\mathcal{N}} \pi_{n+i_1} \triangleright_{\mathcal{I}}^{k_0} \triangleright_{\mathcal{N}} \pi_{n+i_2} \triangleright_{\mathcal{I}}^{k_0} \triangleright_{\mathcal{N}} \cdots$, et l'on obtient donc ainsi une suite $(\mathcal{P}(\pi_{n+i_j}))_{j \in \mathbb{N}}$, qui est une suite infinie strictement décroissante d'entiers naturels, ce qui est impossible.

3°) À partir d'un certain rang n , il n'y a plus que des $\triangleright_{\mathcal{I}}$. Dans ce cas, la suite $(\pi_{n+j})_{j \in \mathbb{N}}$ est une suite infinie strictement décroissante au sens de la relation de sous-arbre, ce qui est impossible.

Dans tous les cas, l'on est parvenu à une impossibilité.

(iii) L'irréflexivité de cette relation découle de celle, évidente, de \triangleleft , et sa transitivité découle directement de sa définition. Sa bonne fondation découle du point (ii).

Cet ordre sera beaucoup utilisé dans les raisonnements par induction liés aux systèmes de types purs explicites. On peut se demander pourquoi l'on a besoin de ce raffinement, au lieu de se contenter de l'induction structurelle basée sur l'ordre de sous-arbre, utilisée usuellement dans les systèmes à substitutions implicites ou dans d'autres systèmes à substitutions explicites comme [9, 13, 77, 68]. Cela est essentiellement dû à deux facteurs : d'une part, des problèmes spécifiques liés aux substitutions explicites intervenant dans la démonstration de réduction du sujet, et d'autre part le choix fait ici d'employer des substitutions explicites partout, y compris en prédicat d'assertion de type.

En ce qui concerne le premier point, la difficulté vient du fait que la coupure « saute » une étape par rapport à une introduction et élimination de Π consécutives, ce qui fait que l'on a besoin d'utiliser le lemme d'engendrement pour reconstituer le typage un peu plus haut, afin d'obtenir les prémisses dont on a besoin pour

la coupure. Dans les autres systèmes explicites, cela ne pose pas de problème, car on a besoin des résultats du lemme d'engendrement tels quels. Mais dans les systèmes présentés ici, du fait que l'on est obligé de faire la démonstration de réduction du sujet avec une notion de réduction parallèle (ceci est dû à la figure d'expansion, ce point sera abordé au § 3.2), on peut avoir besoin d'appliquer une hypothèse d'induction à ces résultats.

En ce qui concerne le second point, ce choix oblige de faire grand usage de la figure de conversion afin de récupérer les types d'origine, toujours par exemple dans l'optique de réduction du sujet. Et de même que pour le point précédent, cela peut rendre nécessaire d'appliquer une hypothèse d'induction aux résultats du théorème de correction des types.

On remarque en particulier que cet ordre contient l'ordre de sous-arbre propre, ce qui sera utile lors des raisonnements par induction à venir. Par ailleurs, afin de préciser les choses, on peut donner quelques propriétés supplémentaires :

Proposition 2.18 : *Propriétés liées à l'ordre de simplicité*

Soit $(\varpi) - \Xi \vdash X : Y$ une dérivation. Si $\mathcal{P}(\varpi) = 0$, alors $X \in \overline{\mathcal{S}}$ et $Y = \int$.

Démonstration.

On raisonne par induction structurelle sur (ϖ) . La dernière figure employée ne peut être que (axiome), (affaiblissement) ou (conversion) :

- a. (axiome) : la propriété est évidente ;
- b. (affaiblissement) : soit $(\pi) - \Gamma \vdash X : Y$ la première prémisse ; on a alors $\mathcal{P}(\pi) = \mathcal{P}(\varpi) = 0$, et il suffit d'appliquer l'hypothèse d'induction à (π) ;
- c. (conversion) : soit $(\pi) - \Xi \vdash X : Y'$ avec $Y' \equiv_{\beta_x} Y$ la première prémisse ; on a alors $\mathcal{P}(\pi) \leq \varpi = 0$, et l'on peut donc appliquer l'hypothèse d'induction à (π) , d'où $Y \equiv_{\beta_x} Y' = \int$;

et donc dans tous les cas, l'on obtient bien que $Y = \int$.

Q.E.D.

3. Propriétés

Dans ce paragraphe, on étudie des propriétés générales des systèmes de types purs explicites définis ci-dessus. En particulier, on démontre qu'ils vérifient une notion de correction des types – à propos de cette notion, on pourra se reporter à la section relative au prédicat pseudo-sortie – et la réduction du sujet. Auparavant, on établit quelques lemmes utiles pour reconstituer l'arbre de dérivation d'un jugement valide.

3.1. Lemmes de dérivation

Maintenant viennent les lemmes de dérivation, et pour commencer un résultat d'affaiblissement, énonçant que l'on peut rajouter des hypothèses dans le contexte :

Lemme 2.19 : Affaiblissement

Soient Θ, Ξ, Σ trois contextes, w une variable, X, Y, Z trois termes, ν une sorte tels que les dérivations $(\varpi) - \Theta, \Xi \vdash X : Y$ et $(\pi) - \Sigma \vdash Z : \nu$ soient des jugements valides, et que $\Sigma \subseteq \Theta$. Alors $(\varpi') - \Theta, w : Z, \Xi \vdash X : Y$ est un jugement valide. En outre, $\mathcal{P}(\varpi') = \mathcal{P}(\varpi)$ et $\mathcal{S}(\varpi') = \mathcal{S}(\varpi) \cup \{w\}$.

Démonstration.

On procède par induction structurelle sur la dérivation (ϖ) .

On peut traiter à part le cas où $\Xi = ()$, qui se résout par l'application d'une simple figure d'affaiblissement. Sinon, on distingue différents cas selon la dernière figure appliquée dans (ϖ) :

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

On est nécessairement dans le cas où $\Xi = ()$.

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

On appelle les deux prémisses respectivement (π_1) et (π_2) . Par hypothèse d'induction, $(\pi'_1) - \Theta, w : Z, \Xi \vdash A : \rho$ et $(\pi'_2) - \Theta, w : Z, \Xi, x : A \vdash B : \sigma$, et $\mathcal{P}(\pi'_i) = \mathcal{P}(\pi_i)$ et $\mathcal{S}(\pi'_i) = \mathcal{S}(\pi_i) \cup \{w\}$, et l'on peut alors conclure par une figure de règle.

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

On appelle la prémisse (π_0) . Par hypothèse d'induction, l'on dérive alors $(\pi'_0) - \Theta, w : Z, \Xi \vdash A : \sigma$, avec $\mathcal{P}(\pi'_0) = \mathcal{P}(\pi_0)$ et $\mathcal{S}(\pi'_0) = \mathcal{S}(\pi_0) \cup \{w\}$, et l'on peut conclure par une figure d'hypothèse.

$$d. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

On appelle les deux prémisses respectivement (π_1) et (π_2) . On peut supposer que l'on n'est pas dans le cas où $\Xi = ()$. Par hypothèse d'induction, l'on a $(\pi'_1) - \Theta, w : Z, \Xi' \vdash A : B$ avec $\Xi = \Xi', x : C$, et l'on obtient alors que $\mathcal{P}(\pi'_1) = \mathcal{P}(\pi_1)$ et $\mathcal{S}(\pi'_1) = \mathcal{S}(\pi_1) \cup \{w\}$, et l'on peut conclure au moyen d'une figure d'affaiblissement :

$$\frac{(\pi'_1) \quad (\pi_2)}{\Theta, w : Z, \Xi', x : C \vdash A : B}^{(w)}.$$

$$e. \text{ (introduction de } \Pi) : \frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

Ce cas est analogue au cas b.

$$f. \text{ (élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

Le cas est analogue au cas b.

$$g. \text{ (coupure) : } \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

Le cas est analogue au cas b.

$$h. \text{ (expansion) : } \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

Le cas est analogue au cas b.

$$i. \text{ (conversion) : } \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

Le cas est analogue au cas b.

Q.E.D.

Le lemme suivant permet d'initialiser un arbre de dérivation :

Lemme 2.20 : *Initialisation pour les systèmes de types purs explicites*

Soit Γ un contexte ayant une justification $\mathcal{J}(\Gamma \vdash)$. Alors :

- (i) si $(\bar{\sigma} : \bar{\tau}) \in \overline{\mathcal{A}}$, alors $(\varpi) - \Gamma \vdash \bar{\sigma} : \bar{\tau}$, et $\begin{cases} \mathcal{P}(\varpi) = \mathcal{P}(\bar{\sigma} : \bar{\tau}) \\ \mathcal{S}(\varpi) = \mathcal{J}(\Gamma \vdash) \end{cases}$;
- (ii) si $\Gamma = \Theta, \bar{x} : A, \Xi$, alors $(\varpi) - \Gamma \vdash \bar{x} : A$.

Démonstration.

Les deux points découlent directement du lemme précédent. Pour le point (i), la condition sur $\mathcal{P}(\varpi)$ et $\mathcal{S}(\varpi)$ est vérifiée car lorsque l'on applique répétitivement le lemme précédent, $\mathcal{P}(\varpi)$ ne change pas, et $\mathcal{S}(\varpi)$ s'enrichit à chaque fois d'une dérivation appartenant à $\mathcal{J}(\Gamma \vdash)$.

Q.E.D.

Le lemme suivant, dit d'engendrement, est fondamental, car il est particulièrement utile lorsque l'on souhaite raisonner sur la structure d'un terme typable : en effet, il permet de reconstituer étape par étape la dérivation et le prédicat d'un jugement de typage en fonction de la structure de son sujet. On peut se concentrer ainsi sur les figures « importantes » (*i.e.* structurelles) en n'ayant pas à se préoccuper des figures « administratives » (*i.e.* n'affectant pas le sujet). Toutefois, la contrepartie de cela est que les arbres obtenus ne seront pas de vrais sous-arbres, et l'on voit ici l'utilité de l'ordre de simplicité, qui permettra néanmoins de les considérer comme des arbres plus petits que celui dont ils sont issus afin de pouvoir raisonner par induction sur l'arbre que l'on souhaiterait avoir plutôt que sur celui qui est donné.

Lemme 2.21 : *Engendrement pour les systèmes de types purs explicites*

Soient Ξ un contexte et $M : T$ une assertion tels que la dérivation $(\varpi) - \Xi \vdash M : T$ soit valide. Alors :

- (i) si $M = \bar{\sigma} \in \bar{\mathcal{S}}$ alors $\exists \bar{\tau} \in \bar{\mathcal{S}}$
 $(T \equiv_{\beta x} \bar{\tau} \wedge (\bar{\sigma} : \bar{\tau}) \in \mathcal{A})$
- (ii) si $M = \bar{x}$ alors $\exists \tau \in \mathcal{S} \exists U \in \Lambda \mathfrak{I}_x$
 $((\pi_0) - \Xi \vdash U : \tau \wedge (\bar{x} : U) \in \Xi \wedge T \equiv_{\beta x} U)$
- (iii) si $M = \Pi x : A.B$ alors $\exists (\rho, \sigma, \tau) \in \mathcal{R}$
 $((\pi_1) - \Xi \vdash A : \rho \wedge (\pi_2) - \Xi, x : A \vdash B : \sigma \wedge T \equiv_{\beta x} \tau)$
- (iv) si $M = \lambda x : A.B$ alors $\exists \tau \in \mathcal{S} \exists C \in \Lambda \mathfrak{I}_x$
 $((\pi_1) - \Xi \vdash (\Pi x : A.C) : \tau \wedge (\pi_2) - \Xi, x : A \vdash B : C \wedge T \equiv_{\beta x} \Pi x : A.C)$
- (v) si $M = A B$ alors $\exists C, D \in \Lambda \mathfrak{I}_x$
 $((\pi_1) - \Xi \vdash A : (\Pi x : C.D) \wedge (\pi_2) - \Xi \vdash B : C \wedge T \equiv_{\beta x} D \langle x \leftarrow B \rangle)$

En outre, dans tous les cas, pour chaque i , $(\pi_i) \triangleleft_{\mathcal{N}} (\varpi)$.

Démonstration.

Lorsque l'on considère les formes des sujets des différentes figures d'inférence, on voit que la dernière figure appliquée dans le typage de $\Xi \vdash M : T$ ne peut être que (affaiblissement), (conversion), ou la figure adaptée à la forme de M – dans l'ordre, de (i) à (v) : (axiome), (hypothèse), (règle), (introduction de Π), (élimination de Π). En outre, comme (affaiblissement) et (conversion) ne changent pas le sujet, et qu'il faut bien que le celui-ci ait été introduit à un moment quelconque, nécessairement la figure que l'on qualifiait d'adaptée a dû être employée.

Plus formellement, l'on va montrer par induction sur la dérivation de $\Xi \vdash M : T$ que la propriété souhaitée est vérifiée. L'on discrimine selon la dernière figure appliquée. Seules trois figures ont pu être appliquées : la figure adaptée, celle d'affaiblissement ou celle de conversion.

- 1°) Si c'est la figure adaptée, les prémisses de la figure sont les conclusions auxquelles on veut aboutir, et la propriété est donc satisfaite. En outre, comme, si (π_i) est une prémisses de ϖ pour l'une des cinq règles considérées, on a (cf. définition 2.13) que $\mathcal{P}(\pi_i) < \mathcal{P}(\varpi)$ et $\mathcal{S}(\pi) \subseteq \mathcal{S}(\varpi)$, i.e. $(\pi_i) \triangleleft_{\mathcal{N}} (\varpi)$.

- 2°) (affaiblissement) :
$$\frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad \bar{x} \notin \text{dom}(\Gamma)}{\Gamma, \bar{x} : C \vdash A : B}$$

On appelle respectivement (ϖ') et (ϖ'') les deux prémisses. On applique l'hypothèse d'induction à la prémisses (ϖ') , ce qui donne des résultats de la forme $(\pi'_i) - \Gamma, \Theta \vdash X : Y$, avec l'inégalité $(\pi'_i) \triangleleft_{\mathcal{N}} (\varpi')$, alors que l'on souhaite obtenir les mêmes résultats avec le contexte $\Gamma, \bar{x} : C, \Theta$. On utilise

alors le lemme d'affaiblissement 2.19 pour obtenir $(\pi_i) - \Gamma, \bar{x} : C, \Theta \vdash X : Y$. De plus, on a d'une part $\mathcal{P}(\pi_i) = \mathcal{P}(\pi'_i) < \mathcal{P}(\varpi') = \mathcal{P}(\varpi)$, et d'autre part $\mathcal{S}(\pi_i) = \mathcal{S}(\pi'_i) \cup \{(\varpi'')\} \subset \mathcal{S}(\varpi') \cup \{(\varpi'')\} = \mathcal{S}(\varpi)$, si bien que $(\pi_i) \triangleleft_N (\varpi)$.

$$3^\circ \text{ (conversion)} : \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

Dans ce cas, l'application à la prémisse $(\varpi') - \Gamma \vdash M : A$ de l'hypothèse d'induction satisfait la propriété, et ce, avec a fortiori les inégalités souhaitées, car $\mathcal{P}(\varpi') \leq \mathcal{P}(\varpi)$, et $\mathcal{S}(\varpi') \subseteq \mathcal{S}(\varpi)$.

Dans tous les cas, on obtient le résultat voulu.

Q.E.D.

On peut donc maintenant démontrer le théorème de correction des types, dont on rappelle l'énoncé, déjà donné p. 85 :

Théorème : *correction des types dans les systèmes de types purs explicites*

Soit $(\varpi) - \Gamma \vdash A : B$ un jugement de typage valide. Alors il existe $\bar{\sigma} \in \bar{\mathcal{S}}$ tel que $(\pi) - \Gamma \vdash B : \bar{\sigma}$. En outre, $(\pi) \triangleleft_N (\varpi)$ si $B \neq \int$, et $\mathcal{P}(\pi) = 0$ et $\mathcal{S}(\pi)$ égal à une justification de Γ sinon.

Démonstration.

L'on procède par induction sur l'ordre de simplicité, en discriminant selon la dernière figure employée dans la dérivation du jugement $\Gamma \vdash A : B$:

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

Ce cas est trivial, car $(\pi) - \vdash \bar{\tau} : \int$, et $\mathcal{P}(\pi) = 0$ et $\mathcal{S}(\pi) = \emptyset = \mathcal{S}(\varpi) = \mathcal{J}(\emptyset \vdash)$, ce qui donne bien les inégalités souhaitées.

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

Nécessairement, $\tau \neq \int$. Ce cas est évident : par le lemme d'initialisation 2.20, on obtient $(\pi) - \Gamma \vdash \tau : \int$ avec $\mathcal{P}(\pi) = 0 < \mathcal{P}(\varpi)$ et $\mathcal{S}(\pi) = \mathcal{J}(\Gamma \vdash)$, où $\mathcal{J}(\Gamma \vdash)$ est la justification de Γ donnée par le lemme de bonne formation 2.13, et vérifiant donc $\mathcal{J}(\Gamma \vdash) \subseteq \mathcal{S}(\varpi)$. Et donc, en rassemblant ces inégalités, l'on obtient que $(\pi) \triangleleft_N (\varpi)$.

$$c. \text{ (hypothèse)} : \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Nécessairement, $\tau \neq \int$. Ce cas est évident : la prémisse est exactement ce que l'on veut montrer, et l'on vérifie bien l'inégalité souhaitée par définition de $\mathcal{P}(\cdot)$, $\mathcal{S}(\cdot)$ et \triangleleft_N .

$$d. \text{ (affaiblissement)} : \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

On appelle respectivement (ϖ') et (ϖ'') les deux prémisses. Par hypothèse

d'induction appliquée à (ϖ') , on obtient $(\pi') - \Gamma \vdash B : \bar{\tau}$ et l'on peut appliquer une figure d'affaiblissement :

$$\frac{(\pi') \quad (\varpi')}{\Gamma \vdash B : \bar{\tau} \quad \Delta \vdash C : \sigma} \text{ (W)} \quad \frac{}{\Gamma, x : C \vdash B : \bar{\tau}}$$

pour obtenir une dérivation (π) avec $\mathcal{P}(\pi) = \mathcal{P}(\pi')$ et $\mathcal{S}(\pi) = \mathcal{S}(\pi') \cup \{\varpi''\}$, qui donne la conclusion voulue avec les inégalités souhaitées.

e. (introduction de Π) :
$$\frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

Ce cas est évident : la première prémisse est exactement ce que l'on veut montrer, et, comme $\Pi x:A.B \neq \int$, l'on vérifie bien l'inégalité souhaitée par définition de $\mathcal{P}(\cdot)$, $\mathcal{S}(\cdot)$ et \triangleleft_N .

f. (élimination de Π) :
$$\frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

On appelle respectivement (ϖ') et (ϖ'') les deux prémisses. Par hypothèse d'induction, on obtient $(\pi') - \Gamma \vdash (\Pi x:A.B) : \tau$ et $(\pi') \triangleleft_N (\varpi')$, donc, par le lemme d'engendrement (2.21), il existe σ telle que $(\pi'') - \Gamma, x : A \vdash B : \sigma$ et $(\pi'') \triangleleft_N (\pi')$, et donc $(\pi'') \triangleleft_N \triangleleft_N (\varpi')$, et a fortiori $(\pi'') \triangleleft_N (\varpi')$.

En outre, Γ est contexte d'un jugement valide, donc, par le lemme de bonne formation (2.13), on a $\Gamma \vdash$ avec la justification $\mathcal{J}(\Gamma \vdash) = \mathcal{S}(\varpi'')$, et donc, par le lemme d'initialisation (2.20) i., on a $(\psi) - \Gamma \vdash \sigma : \int$ avec $\mathcal{P}(\psi) = 0$ et $\mathcal{S}(\psi) = \mathcal{J}(\Gamma \vdash) = \mathcal{S}(\varpi'')$, et l'on effectue alors la dérivation (π) suivante :

$$\frac{\frac{(\pi'') \quad (\varpi'')}{\Gamma, x : A \vdash B : \sigma \quad \Gamma \vdash N : A} \text{ (†)} \quad (\psi)}{\Gamma \vdash B\langle x \leftarrow N \rangle : \sigma\langle x \leftarrow N \rangle} \text{ (‡)} \quad \frac{}{\Gamma \vdash \sigma : \int} \text{ (‡)}$$

avec $\mathcal{P}(\pi) = \mathcal{P}(\pi'') + \mathcal{P}(\varpi'') + 1 + \mathcal{P}(\psi) < \mathcal{P}(\varpi') + \mathcal{P}(\varpi'') + 1 = \mathcal{P}(\varpi)$ et $\mathcal{S}(\pi) = \mathcal{S}(\pi'') \cup \mathcal{S}(\varpi'') \cup \mathcal{S}(\psi) \subseteq \mathcal{S}(\varpi') \cup \mathcal{S}(\varpi'') = \mathcal{S}(\varpi)$; d'où l'on peut conclure que $(\pi) \triangleleft_N (\varpi)$.

g. (coupure) :
$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

On appelle respectivement (ϖ') et (ϖ'') les prémisses. Par hypothèse d'induction appliquée à (ϖ') , l'on obtient $(\pi') - \Gamma, x : A \vdash B : \bar{\sigma}$.

En outre, Γ est contexte d'un jugement valide, donc, par le lemme de bonne formation (2.13), on obtient qu'il existe une justification $\mathcal{J}(\Gamma \vdash)$ telle que

$\mathcal{J}(\Gamma \vdash) = \mathcal{S}(\varpi'')$, et donc, par le lemme d'initialisation (2.20) i., on obtient que $(\psi) - \Gamma \vdash \sigma : \int$ avec $\mathcal{P}(\psi) = 0$ et $\mathcal{S}(\psi) = \mathcal{J}(\Gamma \vdash) = \mathcal{S}(\varpi'')$. Deux cas se présentent :

1°) $B \neq \int$, et donc $(\pi') \triangleleft_{\mathcal{N}} (\varpi')$, et l'on effectue la dérivation (π) suivante :

$$\frac{\frac{\frac{(\pi') \quad (\varpi'')}{\Gamma, x : A \vdash B : \sigma \quad \Gamma \vdash N : A} \quad (\dagger)}{\Gamma \vdash B \langle x \leftarrow N \rangle : \sigma \langle x \leftarrow N \rangle} \quad (\psi) \quad \Gamma \vdash \sigma : \int}{\Gamma \vdash B \langle x \leftarrow N \rangle : \sigma} \quad (\equiv)$$

et l'on a comme précédemment $\mathcal{P}(\pi) = \mathcal{P}(\pi') + \mathcal{P}(\varpi'') + \mathcal{P}(\psi) + 1 < \mathcal{P}(\varpi)$, et $\mathcal{S}(\pi) = \mathcal{S}(\pi') \cup \mathcal{S}(\varpi'') \cup \mathcal{S}(\psi) \subseteq \mathcal{S}(\varpi') \cup \mathcal{S}(\varpi'') = \mathcal{S}(\varpi)$, d'où l'on peut conclure que $(\pi) \triangleleft_{\mathcal{N}} (\varpi)$.

2°) $B = \int$. Dans ce cas, d'après la proposition 2.13, il existe une justification $\mathcal{J}(\Gamma \vdash)$ telle que $\mathcal{S}(\psi) = \mathcal{J}(\Gamma \vdash) \subseteq \mathcal{S}(\varpi'') \subseteq \mathcal{S}(\varpi)$. Par le lemme d'initialisation 2.20 (i), l'on obtient $(\psi) - \Gamma \vdash \int : \int$ avec $\mathcal{P}(\psi) = 0$ et $\mathcal{S}(\psi) = \mathcal{J}(\Gamma \vdash)$. On peut alors obtenir la dérivation (π) suivante :

$$\frac{\frac{(\psi) \quad (\varpi'')}{\Gamma \vdash \int : \int \quad \Gamma \vdash N : A} \quad (\dagger)}{\Gamma \vdash \int \langle x \leftarrow N \rangle : \int} \quad (\times)$$

On remarque que l'expansion ici est légitime, car comme A apparaît dans le contexte de la première prémisse, il s'agit nécessairement d'un terme de $\Lambda \mathfrak{I} \mathbf{x}$. Par ailleurs, $\mathcal{S}(\psi) = \mathcal{J}(\Gamma \vdash) \subseteq \mathcal{S}(\varpi'') \subseteq \mathcal{S}(\varpi)$. On en déduit $\mathcal{S}(\pi) = \mathcal{S}(\psi) \cup \mathcal{S}(\varpi'') \subseteq \mathcal{S}(\varpi)$, et l'on obtient également l'inégalité souhaitée du fait que $\mathcal{P}(\pi) = \mathcal{P}(\psi) + \mathcal{P}(\varpi'') = \mathcal{P}(\varpi'') < \mathcal{P}(\varpi)$.

$$h. \text{ (expansion) : } \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad A \in \Lambda \mathfrak{I} \mathbf{x} \quad P \langle x \leftarrow N \rangle \longrightarrow_{\mathbf{x}} M}{\Gamma \vdash P \langle x \leftarrow N \rangle : B}$$

Ce cas est évident : l'hypothèse d'induction appliquée à la première prémisse donne exactement ce que l'on veut montrer.

$$i. \text{ (conversion) : } \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta \mathbf{x}} B}{\Gamma \vdash M : B}$$

Ce cas est évident : la seconde prémisse est exactement ce que l'on veut montrer. Q.E.D.

Remarque :

Avec le lemme d'engendrement et le théorème de correction des types énoncés ici, on voit que l'on peut, dans un raisonnement par induction sur l'ordre de simplicité,

appliquer l'hypothèse d'induction à des dérivations issues de celle considérée par toute combinaison des trois opérations suivantes :

- sélection d'un sous-arbre propre ;
- application du lemme d'engendrement ;
- application du théorème de correction des types.

3.2. Réduction du sujet

Ce paragraphe contient un résultat fondamental, dont on entend habituellement pouvoir disposer dans un système de types pour pouvoir le considérer « approprié ». Il indique que le comportement calculatoire d'un terme n'a pas d'influence sur son type, et comme on l'a indiqué au § 2.1, il s'en faut de beaucoup que l'établissement de cette propriété dans le cadre des substitutions explicites soit évident. La démonstration étant assez lourde, on va en donner tout d'abord une idée générale.

Comme on l'a dit au § 2.3, l'une des difficultés réside dans la figure d'expansion. En effet, celle-ci a pour effet, dans certains cas de « mettre en commun » des sous-termes, comme dans l'exemple suivant :

$$\frac{\Gamma \vdash U\langle x \leftarrow W \rangle V\langle x \leftarrow W \rangle : Y \quad \Delta \vdash W : T}{\Gamma \vdash (UV)\langle x \leftarrow W \rangle : Y}$$

Dans ce cas, si la réduction a lieu dans le sous-terme W – *i.e.* est de la forme $(UV)\langle x \leftarrow W \rangle \xrightarrow{x} (UV)\langle x \leftarrow W' \rangle$ – il est nécessaire de faire plusieurs réductions dans la prémisse pour parvenir à $U\langle x \leftarrow W' \rangle V\langle x \leftarrow W' \rangle$ qui permet de réappliquer la figure pour conclure, et l'hypothèse d'une réduction simple ne permet donc pas de passer à l'induction. On va donc devoir employer, contrairement à l'usage, une notion de réduction parallèle, et l'on ne pourra par conséquent pas séparer les réductions effectuées dans le contexte de celles effectuées dans le sujet. C'est là, comme on l'a déjà indiqué au § 2.3, la motivation pour l'introduction de l'ordre de simplicité : en effet, avec une réduction parallèle ayant lieu à la fois dans le contexte et dans le sujet, on ne peut pas laisser tels quels les résultats du lemme d'engendrement ou du théorème de correction des types, mais on doit leur appliquer l'hypothèse d'induction. Or ces résultats ne sont pas en général des sous-arbres de la dérivation considérée, bien qu'ils semblent intuitivement plus petits. L'ordre de simplicité est précisément la notion convenant pour indiquer en quoi ils sont plus petits.

Avant d'entamer la démonstration de réduction du sujet proprement dite, il est nécessaire d'établir certains lemmes. On commence par un lemme permettant d'effectuer des conversions dans le contexte.

Lemme 2.22 : *Conversion de contexte*

Soit $\Theta \vdash X : Y$ un jugement de typage, et soit Θ' un contexte bien formé tel que $\Theta \equiv_{\beta x} \Theta'$. Alors $\Theta' \vdash X : Y$.

Démonstration.

On procède par induction sur la dérivation de $\Theta \vdash X : Y$.

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

Le cas est trivial : $\Theta = \Theta' = ()$.

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

Par hypothèse d'induction, $\Gamma' \vdash A : \rho$. On en déduit par ailleurs en particulier que $\Gamma', x : A$ est bien formé, ce qui permet d'appliquer l'hypothèse d'induction à l'autre prémisse, ce qui donne $\Gamma', x : A \vdash B : \sigma$. On conclut alors par l'application de la figure de règle.

$$c. \text{ (hypothèse)} : \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

On a $\Theta' = \Gamma', x : A'$. Θ' est bien formé donc $\Gamma' \vdash A' : \tau$, ce que l'on note (1). Par ailleurs, par hypothèse d'induction, $\Gamma' \vdash A : \sigma$, d'où, par le lemme d'affaiblissement, $\Gamma', x : A' \vdash A : \sigma$, ce que l'on note (2). On effectue alors la dérivation suivante :

$$\frac{\frac{(1)}{\Gamma', x : A' \vdash x : A'} \text{ (hyp)} \quad (2)}{\Gamma', x : A' \vdash x : A} \text{ (}\equiv\text{)}$$

qui permet de conclure.

$$d. \text{ (affaiblissement)} : \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

On a $\Theta' = \Gamma', x : C'$. Θ' est bien formé donc $\Gamma' \vdash C' : \tau$, ce que l'on note (1). Par ailleurs, par hypothèse d'induction, $\Gamma' \vdash A : B$, d'où, par une simple figure d'affaiblissement, $\Gamma', x : C' \vdash A : B$.

$$e. \text{ (introduction de } \Pi \text{)} : \frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

Pour conclure directement par hypothèse d'induction comme au cas b., il suffit de montrer que le contexte $\Gamma', x : A$ est bien formé. Or, par hypothèse

d'induction, $\Gamma' \vdash \Pi x:A.B : \tau$, et il suffit d'appliquer le lemme d'engendrement pour avoir $\Gamma' \vdash A : \rho$.

$$f. \quad (\text{élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

Ce cas est direct par hypothèse d'induction.

$$g. \quad (\text{coupure}) : \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

Pour conclure directement par hypothèse d'induction comme au cas b., il suffit de montrer que le contexte $\Gamma', x : A$ est bien formé. Or, par hypothèse d'induction, $\Gamma' \vdash N : A$, donc d'une part Γ' est bien formé, et il suffit alors d'appliquer d'autre part le théorème de correction des types à cette assertion pour obtenir que $\Gamma', x : A$ est bien formé et conclure.

$$h. \quad (\text{expansion}) : \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

Ce cas est direct par hypothèse d'induction.

$$i. \quad (\text{conversion}) : \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

Ce cas est direct par hypothèse d'induction.

Q.E.D.

Les lemmes suivants concernent les substitutions, qui sont ici explicites et non plus implicites, selon l'optique que l'on a déjà évoquée plus haut.

Lemme 2.23 : (Gc)-expansion

Soient $\Gamma \vdash C[M\{y \leftarrow \gamma_y\}] : B$ et $\Delta \vdash N : A$ deux jugements de typage valides, où y est une variable non accessible dans M . Alors $\Gamma \vdash C[M\langle y \leftarrow N \rangle] : B$.

On va en fait démontrer une propriété équivalente par induction à l'aide d'une hypothèse d'induction légèrement plus forte.

Pour ce faire, on introduit une notion de réduction, la (gc)-expansion parallèle indicée, notée $\overleftarrow{(gc) // N}$, où N est un terme, comme suit :

- $\overleftarrow{(gc) // N}$ contient l'égalité ;
- si $x \notin \text{av}(M)$, alors $M\{x \leftarrow \gamma_x\} \overleftarrow{(gc) // N} M\langle x \leftarrow N \rangle$;
- si $A \overleftarrow{(gc) // N} A_1$ et $B \overleftarrow{(gc) // N} B_1$, alors $\Pi x:A.B \overleftarrow{(gc) // N} \Pi x:A_1.B_1$, $A B \overleftarrow{(gc) // N} A_1 B_1$, $\lambda x:A.B \overleftarrow{(gc) // N} \lambda x:A_1.B_1$ et $A\langle x \leftarrow B \rangle \overleftarrow{(gc) // N} A_1\langle x \leftarrow B_1 \rangle$.

On peut maintenant énoncer la propriété que l'on souhaite montrer : on suppose que $\Xi \vdash X : Y$ et $\Theta \vdash Z : W$; alors le fait que Ξ_1 et X_1 soient tels que $\Xi \overleftarrow{(gc) // Z} \Xi_1$ et $X \overleftarrow{(gc) // Z} X_1$ implique que $\Xi_1 \vdash X_1 : Y$.

Il est clair que le lemme de (gc)-expansion *supra* en est un corollaire.

Démonstration.

On procède par induction sur la dérivation de $\Xi \vdash X : Y$, avec l'ordre de simplicité, en discriminant selon la dernière figure employée.

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

Ce cas est trivial.

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

Deux cas se présentent :

1°) $X_1 = \Pi x:A_1.B_1$. Alors, par hypothèse d'induction appliquée aux prémisses, $\Gamma_1 \vdash A_1 : \rho$ d'une part et $\Gamma_1, x : A_1 \vdash B_1 : \sigma$, d'autre part et l'on peut conclure directement par une figure de règle.

2°) $X_1 = (\Pi x:A.B)\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle$. Alors, par hypothèse d'induction appliquée aux prémisses, $\Gamma_1 \vdash A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \rho$, ce que l'on note (1) et $\Gamma_1, x : A\langle y \leftarrow Z \rangle \vdash B\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \sigma$, ce que l'on note (2). En outre, $\Theta \vdash Z : W$, ce que l'on note (3). L'on peut alors effectuer la dérivation suivante :

$$\frac{\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash \Pi x:A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.B\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \tau}{} \text{(R)}}{(3)} \quad (x)}{\Gamma_1 \vdash (\Pi x:A.B)\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \tau}}$$

qui permet de conclure.

$$c. \text{ (hypothèse)} : \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Nécessairement, $X_1 = x\langle z \leftarrow Z \rangle$ ou $X_1 = x$. On a alors $\Xi_1 = \Gamma_1, x : A_1$. Par hypothèse d'induction appliquée aux prémisses, d'une part $\Gamma_1 \vdash A_1 : \sigma$, que l'on note (1), et d'autre part $\Gamma_1 \vdash A : \sigma$, que l'on note (2). On construit alors l'arbre suivant :

$$\frac{\frac{(1)}{\Gamma_1, x : A_1 \vdash x : A_1} \text{(hyp)} \quad (2)}{\Gamma_1, x : A_1 \vdash x : A} \text{(=)}$$

Si $X_1 = x$, on peut conclure. Sinon, $X_1 = x\langle y \leftarrow Z \rangle$; dans ce cas, et l'on conclut alors par une figure d'expansion avec une propagation de type (gc).

$$d. \text{ (affaiblissement)} : \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

L'on a alors $\Xi_1 = \Gamma_1, x : C_1$, et, par hypothèse d'induction appliquée aux

prémises, $\Gamma_1 \vdash A_1 : B_1$ et $\Delta_1 \vdash C_1 : \sigma$, et l'on effectue alors la dérivation suivante :

$$\frac{\Gamma_1 \vdash A_1 : B_1 \quad \Delta_1 \vdash C_1 : \sigma \quad \Delta_1 \subseteq \Gamma_1}{\Gamma_1, x : C_1 \vdash A_1 : B_1} \text{(w)}$$

qui permet de conclure.

e. (introduction de Π) :
$$\frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

Deux cas se présentent :

1°) $X_1 = \lambda x:A_1.M_1$. Alors, par hypothèse d'induction appliquée aux prémisses, l'on a d'une part $\Gamma_1 \vdash \Pi x:A.B : \tau$, ce que l'on note (1), et d'autre part $\Gamma_1 \vdash \Pi x:A_1.B : \tau$, ce que l'on note (2), et $\Gamma_1, x : A_1 \vdash M_1 : B$, ce que l'on note (3). On peut alors effectuer la dérivation suivante :

$$\frac{\frac{(2) \quad (3)}{\Gamma_1 \vdash \lambda x:A_1.M_1 : \Pi x:A_1.B} \text{(\Pi-1)} \quad (1)}{\Gamma_1 \vdash \lambda x:A_1.M_1 : \Pi x:A.B} \text{(\equiv)}$$

2°) $X_1 = (\lambda x:A.B)\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle$. Alors, par hypothèse d'induction appliquée aux prémisses, l'on a d'une part $\Gamma_1 \vdash (\Pi x:A.B) : \tau$, ce que l'on note (1), et d'autre part les deux jugements $\Gamma_1 \vdash \Pi x:A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.B : \tau$ et $\Gamma_1, x : A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B$, d'où, par une introduction de Π , $\Gamma_1 \vdash \lambda x:A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \Pi x:A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.B$, ce que l'on note (2). En outre, $\Theta \vdash Z : W$, ce que l'on note (3). On peut alors effectuer la dérivation suivante :

$$\frac{\frac{(2) \quad (1)}{\Gamma_1 \vdash \lambda x:A\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \Pi x:A.B} \text{(\equiv)} \quad (3)}{\Gamma_1 \vdash \lambda x:A.M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \Pi x:A.B} \text{(x)}$$

f. (élimination de Π) :
$$\frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

Trois cas se présentent :

1°) $X_1 = M_1 N_1$. Alors, par hypothèse d'induction appliquée aux prémisses, $\Gamma_1 \vdash M_1 : \Pi x:A.B$, ce que l'on note (1), et $\Gamma_1 \vdash N_1 : A$, ce que l'on note (2). En outre, par le théorème de correction des types, $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, assertion à laquelle on peut appliquer l'hypothèse d'induction, d'où $\Gamma_1 \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, ce que l'on note (3). On conclut alors comme suit :

$$\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash M_1 N_1 : B\langle x \leftarrow N_1 \rangle} \text{(\Pi-É)} \quad (3)}{\Gamma_1 \vdash M_1 N_1 : B\langle x \leftarrow N \rangle} \text{(\equiv)}$$

2°) $X_1 = (MN)\langle y \leftarrow Z \rangle$ avec M n'étant pas une abstraction. Alors, par hypothèse d'induction appliquée aux deux prémisses, l'on obtient les jugements $\Gamma_1 \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \Pi x:A.B$, et $\Gamma_1 \vdash N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : A$, qui, au moyen d'une élimination de Π , se combinent pour donner le jugement $\Gamma_1 \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B\langle x \leftarrow N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \rangle$, ce que l'on note (1). En outre, par le théorème de correction des types, $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, assertion à laquelle on peut appliquer l'hypothèse d'induction, d'où $\Gamma_1 \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, ce que l'on note (2). Par ailleurs, $\Theta \vdash Z : W$ ce que l'on note (3). L'on peut alors effectuer la dérivation suivante :

$$\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B\langle x \leftarrow N \rangle}^{(=)} (3)}{\Gamma_1 \vdash (MN)\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B\langle x \leftarrow N \rangle}^{(x)}$$

qui permet de conclure.

3°) $X_1 = ((\lambda z:C.D)N)\langle y \leftarrow Z \rangle$. Par le lemme d'engendrement appliqué à la première prémisses, l'on obtient $\Gamma, z : C \vdash D : E$ et $\Gamma \vdash \Pi z:C.E : \rho$ avec $\Pi z:C.E \equiv_{\beta x} \Pi x:A.B$. On peut appliquer l'hypothèse d'induction aux deux jugements précédents et à la seconde prémisses, et l'on obtient par ce moyen $\Gamma_1, z : C\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \vdash D\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : E$, noté (1), $\Gamma_1 \vdash \Pi z:C\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.E : \rho$, noté (2), et $\Gamma_1 \vdash N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : A$, noté (3). En outre, on peut appliquer le théorème de correction des types puis l'hypothèse d'induction à la première prémisses, ce qui donne le jugement $\Gamma_1 \vdash \Pi x:A.B : \bar{v}$, que l'on note (4). L'on peut alors effectuer la dérivation suivante :

$$\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash \lambda z:C\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.D\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \Pi z:C\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.E}^{(\Pi-I)} (4)}{\Gamma_1 \vdash \lambda z:C\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.D\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : \Pi x:A.B}^{(=)}}{\Gamma_1 \vdash (\lambda z:C\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle.D\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle)N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B\langle x \leftarrow N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \rangle}^{(\Pi\text{-É})}$$

qui permet de conclure, comme au cas précédent, par une conversion et une expansion.

g. (coupure) : $\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$

Deux cas se présentent :

1°) $X_1 = M_1\langle x \leftarrow N_1 \rangle$. Alors, par hypothèse d'induction appliquée aux prémisses, $\Gamma_1, x : A \vdash M_1 : B$, ce que l'on note (1), et $\Gamma_1 \vdash N_1 : A$, ce

que l'on note (2). En outre, par le théorème de correction des types, $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, assertion à laquelle on peut appliquer l'hypothèse d'induction, d'où $\Gamma_1 \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, ce que l'on note (3). On conclut alors comme suit :

$$\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash M_1\langle x \leftarrow N_1 \rangle : B\langle x \leftarrow N_1 \rangle}^{(\dagger)} \quad (3)}{\Gamma_1 \vdash M_1\langle x \leftarrow N_1 \rangle : B\langle x \leftarrow N \rangle}^{(\equiv)}$$

2°) $X_1 = M\langle x \leftarrow N \rangle\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle$. Alors, par hypothèse d'induction appliquée aux deux prémisses, l'on obtient que $\Gamma_1, x:A \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B$ et que $\Gamma_1 \vdash N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : A$, d'où, par une coupure, on tire le jugement $\Gamma_1 \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle\langle x \leftarrow N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \rangle : B\langle x \leftarrow N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \rangle$, que l'on note (1). En outre, par le théorème de correction des types, $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, d'où, par hypothèse d'induction, $\Gamma_1 \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, ce que l'on note (2). Par ailleurs, $\Theta \vdash Z : W$ ce que l'on note (3). L'on peut alors effectuer la dérivation suivante :

$$\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash M\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle\langle x \leftarrow N\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle \rangle : B\langle x \leftarrow N \rangle}^{(\equiv)} \quad (3)}{\Gamma_1 \vdash M\langle x \leftarrow N \rangle\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B\langle x \leftarrow N \rangle}^{(x)}$$

qui permet de conclure.

h. (expansion) :
$$\frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_\chi M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

Deux cas se présentent :

1°) $X_1 = P_1\langle x \leftarrow N_1 \rangle$ On peut montrer que si $P\langle x \leftarrow N \rangle \longrightarrow_\chi M$, alors il existe M_1 tel que $P_1\langle x \leftarrow N_1 \rangle \longrightarrow_\chi M_1$ et $M_1 \xleftarrow{(gc)/Z} M$ (attendu que l'on peut supposer par la convention de Barendregt que $x \notin \text{av}(Z)$). On a alors par hypothèse d'induction appliquée aux prémisses que $\Gamma_1 \vdash M_1 : B$, ce que l'on note (1), et $\Gamma_1 \vdash N_1 : A$, ce que l'on note (2). On conclut alors comme suit :

$$\frac{(1) \quad (2) \quad P_1\langle x \leftarrow N_1 \rangle \longrightarrow_\chi M_1}{\Gamma_1 \vdash P_1\langle x \leftarrow N_1 \rangle : B}$$

2°) $X_1 = P\langle x \leftarrow N \rangle\langle y \leftarrow Z \rangle$. Alors, par hypothèse d'induction, $\Gamma_1 \vdash M : B$, ce que l'on note (1), et $\Gamma_1 \vdash N : A$, ce que l'on note (2). En outre, $\Theta \vdash Z : W$, ce que l'on note (3), et l'on peut ensuite conclure comme suit :

$$\frac{\frac{(1) \quad (2)}{\Gamma_1 \vdash P\langle x \leftarrow N \rangle : B}^{(x)} \quad (3)}{\Gamma_1 \vdash P\langle x \leftarrow N \rangle\{\gamma_y \leftarrow y\}\langle y \leftarrow Z \rangle : B}^{(x)}$$

- i. (conversion) :
$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

 Par hypothèse d'induction, $\Gamma_1 \vdash M_1 : A$ et $\Gamma_1 \vdash B : \bar{\sigma}$. On dérive alors :

$$\frac{\Gamma_1 \vdash M_1 : A \quad \Gamma_1 \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma_1 \vdash M_1 : B} (\equiv),$$

ce qui permet de conclure.

Q.E.D.

Lemme 2.24 : *Substitution explicite*

Soient Θ et Ξ deux contextes, u une variable et W, X, Y, Z quatre termes tels que $\Theta, u : W, \Xi \vdash X : Y$ et $\Theta \vdash Z : W$. Alors $\Theta, \Xi \langle u \leftarrow Z \rangle \vdash X \langle u \leftarrow Z \rangle : Y \langle u \leftarrow Z \rangle$.

Dans une optique d'allégement des arbres d'inférence *infra*, l'on introduit les abréviations suivantes : le symbole \blacktriangledown représente la vérification $\Theta \vdash Z : W$ (par exemple dans les expansions), et le symbole ∇ représente une prémisse de la forme $\Delta \vdash \xi : \int$, où ξ est une sorte – cette vérification, conséquence immédiate du lemme d'initialisation (2.20), survient en général comme prémisse d'une conversion.

Démonstration.

On va en fait montrer la propriété légèrement plus forte suivante : on suppose que $\Theta, u : W, \Xi \vdash X : Y$ et $\Sigma \vdash Z : W$ et $\Theta \subseteq \Sigma$; alors $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash X \langle u \leftarrow Z \rangle : Y \langle u \leftarrow Z \rangle$. On procède par induction en employant l'ordre de simplicité.

- a. (axiome) :
$$\frac{(\bar{\sigma}, \bar{\tau}) \in \bar{\mathcal{A}}}{\vdash \bar{\sigma} : \bar{\tau}}$$

Le contexte étant vide, ce point est trivial.

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

Par hypothèse d'induction appliquée aux deux prémisses, on obtient d'une part que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \rho \langle u \leftarrow Z \rangle$, ce que l'on notera (1) et d'autre part que $\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle$, ce que l'on notera (2).

En outre, $\Gamma \vdash$ en tant que contexte d'une prémisse, et donc, par le lemme d'initialisation (2.20), l'on obtient $(\psi) - \Gamma \vdash \tau : \int$ avec $\mathcal{P}(\psi) = 0$, et l'on peut donc appliquer l'hypothèse d'induction, ce qui produit le jugement $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash \tau \langle u \leftarrow Z \rangle : \int \langle u \leftarrow Z \rangle$, auquel on peut appliquer une figure de conversion pour obtenir $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash \tau \langle u \leftarrow Z \rangle : \int$, que l'on notera (3).

On conclut alors au moyen de l'arbre d'inférence suivant :

$$\frac{\frac{\frac{(1) \quad \nabla}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \rho} (\equiv)}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash \Pi x : A \langle u \leftarrow Z \rangle . B \langle u \leftarrow Z \rangle : \tau} (\mathcal{R}) \quad \frac{(2) \quad \nabla}{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \sigma} (\equiv)}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\Pi x : A . B) \langle u \leftarrow Z \rangle : \tau} (\mathcal{X})}{\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\Pi x : A . B) \langle u \leftarrow Z \rangle : \tau \langle u \leftarrow Z \rangle}{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)} (\text{hyp})} (\equiv)} (3)$$

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Il y a deux cas :

1°) soit $u \neq x$, auquel cas par hypothèse d'induction appliquée à la pré-misse, on obtient que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle$, et l'on conclut comme suit :

$$\frac{\frac{\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \sigma} (\equiv)}{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash x : A \langle u \leftarrow Z \rangle} (\text{hyp}) \quad \nabla \quad x \langle u \leftarrow Z \rangle \longrightarrow_x x}{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash x \langle u \leftarrow Z \rangle : A \langle u \leftarrow Z \rangle} (\mathcal{X});$$

2°) soit $u = x$, auquel cas $\Theta = \Gamma, \Xi = ()$; d'une part par le lemme d'affaiblissement appliqué à la prémisse $\Sigma, x : A \vdash x : A$ et d'autre part l'hypothèse $\Sigma \vdash Z : W$ se récrit $\Sigma \vdash Z : A$, et l'on conclut ainsi :

$$\frac{\Sigma, x : A \vdash x : A \quad \Sigma \vdash Z : A}{\Sigma \vdash x \langle x \leftarrow Z \rangle : A \langle x \leftarrow Z \rangle} (\dagger).$$

$$d. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

Il y a plusieurs cas :

1°) soit $u \neq x$ et $u \in \text{dom}(\Delta)$ – i.e. $\Delta = \Theta', u : S, \Xi'$ avec $\Xi' \subseteq \Xi$ et $\Theta' \subseteq \Theta \subseteq \Sigma -$, auquel cas par hypothèse d'induction appliquée aux deux prémisses, on obtient d'une part que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle$ et d'autre part que $\Sigma, \Xi' \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle$ et l'on construit alors l'arbre de dérivation suivant :

$$\frac{\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle}{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : C \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle} (\mathcal{W}) \quad \frac{\Sigma, \Xi' \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle}{\Sigma, \Xi' \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma} (\equiv)}{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : C \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle} (\mathcal{W})$$

afin de conclure ;

2°) soit $u \neq x$ et $u \notin \text{dom}(\Delta)$ – i.e. $\Delta = \Theta', \Xi'$ avec $\Xi' \subseteq \Xi$ et $\Theta' \subseteq \Theta \subseteq \Sigma -$, auquel cas d'une part par hypothèse d'induction appliquée à la première prémisse, on obtient $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle$ et d'autre part par les lemmes d'affaiblissement et de (gc)-expansion appliqués à la

seconde prémisses, on obtient $\Sigma, \Xi' \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma$ et l'on effectue la dérivation suivante :

$$\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle \quad \Sigma, \Xi' \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma}{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : C \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle} \text{(w)}$$

afin de conclure ;

3°) soit $u = x$, auquel cas d'une part le lemme d'affaiblissement donne $\Sigma, x : C \vdash A : B$ et d'autre part l'hypothèse $\Sigma \vdash Z : W$ se réécrit $\Sigma \vdash Z : C$, et l'on conclut par l'application d'une coupure :

$$\frac{\Sigma, x : C \vdash A : B \quad \Sigma \vdash Z : C}{\Sigma \vdash A \langle x \leftarrow Z \rangle : B \langle x \leftarrow Z \rangle} \text{(\ddagger)}$$

e. (introduction de Π) : $\frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$

Le lemme d'engendrement (2.21) appliqué à la prémisses $\Gamma \vdash (\Pi x:A.B) : \tau$ indique qu'il existe une règle $(\rho, \sigma, \tau) \in \mathcal{R}$ telle que $\Gamma \vdash A : \rho$ et que $\Gamma, x : A \vdash B : \sigma$, et que ces deux assertions sont plus petites pour l'ordre de simplicité.

On peut appliquer l'hypothèse d'induction à ces deux jugements, ce qui donne d'une part que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \rho \langle u \leftarrow Z \rangle$ et d'autre part que $\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle$. On en déduit par la même inférence que dans le cas b.i. (antépénultième conclusion de l'arbre) que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\Pi x:A \langle u \leftarrow Z \rangle). B \langle u \leftarrow Z \rangle : \tau$, ce que l'on désignera par (1).

En outre, on peut appliquer à la prémisses $\Gamma \vdash \Pi x:A.B : \tau$ l'hypothèse d'induction, ce qui donne $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\Pi x:A.B) \langle u \leftarrow Z \rangle : \tau \langle u \leftarrow Z \rangle$, et, par une figure de conversion, on obtient que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\Pi x:A.B) \langle u \leftarrow Z \rangle : \tau$, ce que l'on désignera par (2).

Ces deux points étant établis, on applique l'hypothèse d'induction à la prémisses $\Gamma, x : A \vdash M : B$, ce qui donne l'assertion de typage suivante : $\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash M \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle$. On construit ensuite l'arbre suivant :

$$\frac{\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash M \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle \quad (1)}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\lambda x:A \langle u \leftarrow Z \rangle). M \langle u \leftarrow Z \rangle : (\Pi x:A \langle u \leftarrow Z \rangle). B \langle u \leftarrow Z \rangle} \text{(\Pi-I)}}{\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\lambda x:A \langle u \leftarrow Z \rangle). M \langle u \leftarrow Z \rangle : (\Pi x:A \langle u \leftarrow Z \rangle). B \langle u \leftarrow Z \rangle \quad (x)}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\lambda x:A \langle u \leftarrow Z \rangle). M \langle u \leftarrow Z \rangle : (\Pi x:A \langle u \leftarrow Z \rangle). B \langle u \leftarrow Z \rangle} \text{(2)}} \text{(\equiv)}$$

qui permet de conclure.

$$f. \text{ (élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

On distingue deux cas, selon que M commence ou non par un λ . On traite d'abord le cas où M ne commence pas par un λ .

En appliquant l'hypothèse d'induction aux deux prémisses, on obtient d'une part $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle : (\Pi x:A.B)\langle u \leftarrow Z \rangle$, que l'on note (1), et d'autre part $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash N\langle u \leftarrow Z \rangle : A\langle u \leftarrow Z \rangle$, que l'on note (2). Par ailleurs, le théorème de correction des types (2.16) appliqué à la première prémisses donne que $\Gamma \vdash (\Pi x:A.B) : \tau$, si bien que par le lemme d'engendrement, il existe une règle $(\rho, \sigma, \tau) \in \mathcal{R}$ telle que $\Gamma \vdash A : \rho$ et $\Gamma, x : A \vdash B : \sigma$, et l'on peut appliquer à ces deux dérivations l'hypothèse d'induction, et l'on obtient ainsi deux jugements : d'une part $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash A\langle u \leftarrow Z \rangle : \rho\langle u \leftarrow Z \rangle$ et d'autre part $\Sigma, \Xi\langle u \leftarrow Z \rangle, x:A\langle u \leftarrow Z \rangle \vdash B\langle u \leftarrow Z \rangle : \sigma\langle u \leftarrow Z \rangle$, que l'on peut traiter similairement par une conversion, d'où, respectivement, $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash A\langle u \leftarrow Z \rangle : \rho$ et $\Sigma, \Xi\langle u \leftarrow Z \rangle, x:A\langle u \leftarrow Z \rangle \vdash B\langle u \leftarrow Z \rangle : \sigma$. On a alors :

$$\frac{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash A\langle u \leftarrow Z \rangle : \rho \quad \Sigma, \Xi\langle u \leftarrow Z \rangle, x : A\langle u \leftarrow Z \rangle \vdash B\langle u \leftarrow Z \rangle : \sigma}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash (\Pi x:A\langle u \leftarrow Z \rangle.B\langle u \leftarrow Z \rangle) : \tau} \text{ (R)}$$

dont on désigne la conclusion par (3), et l'on a alors :

$$\frac{\frac{(1) \quad (3)}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle : (\Pi x:A\langle u \leftarrow Z \rangle.B\langle u \leftarrow Z \rangle)}{(\equiv)} \quad (2)}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle N\langle u \leftarrow Z \rangle : B\langle u \leftarrow Z \rangle\langle x \leftarrow N\langle u \leftarrow Z \rangle \rangle} \text{ (}\Pi\text{-É)}$$

dont on note la conclusion (4).

Par ailleurs, $B\langle u \leftarrow Z \rangle\langle x \leftarrow N\langle u \leftarrow Z \rangle \rangle \equiv_{\beta x} B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle$ comme on le sait d'après le lemme de composition (2.4). Il suffit donc de montrer que le type $B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle$ est correct dans le contexte $\Sigma, \Xi\langle u \leftarrow Z \rangle$, et l'on pourra conclure par une conversion. Or d'après le théorème de correction des types (2.16) appliqué à $\Gamma \vdash MN : B\langle x \leftarrow N \rangle$, on obtient $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, à laquelle on peut appliquer l'hypothèse d'induction, afin d'obtenir le jugement $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{\sigma}\langle u \leftarrow Z \rangle$, d'où l'on tire la dérivation :

$$\frac{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{\sigma}\langle u \leftarrow Z \rangle \quad \nabla}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{\sigma}} \text{ (}\equiv\text{)}$$

dont on note la conclusion (5). L'on peut alors conclure :

$$\frac{\frac{(4) \quad (5)}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle N\langle u \leftarrow Z \rangle : B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle} {(\equiv)} \quad \blacktriangledown}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash MN\langle u \leftarrow Z \rangle : B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle} \text{ (x).$$

Si maintenant M commence par un λ , il suffit de remonter un cran de plus dans la dérivation au moyen du lemme d'engendrement pour faire descendre d'un cran de plus la substitution explicite, afin de pouvoir la faire remonter comme dans la première éventualité au moyen d'une expansion.

$$g. \text{ (coupure)} : \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

On applique l'hypothèse d'induction aux deux prémisses, ce qui donne d'une part que $\Sigma, \Xi\langle u \leftarrow Z \rangle, x : A\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle : B\langle u \leftarrow Z \rangle$, ce que l'on note (1), et d'autre part que $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash N\langle u \leftarrow Z \rangle : A\langle u \leftarrow Z \rangle$, ce que l'on note (2).

En appliquant au jugement $\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle$ le théorème de correction des types (2.16), on obtient $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{v}$, auquel on peut appliquer l'hypothèse d'induction, d'où $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{v}\langle u \leftarrow Z \rangle$. On effectue alors la simple conversion suivante :

$$\frac{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{v}\langle u \leftarrow Z \rangle \quad \nabla}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{v}} \text{ (}\equiv\text{)}$$

dont on note la conclusion (3).

Avec tous ces éléments, on peut construire la dérivation suivante :

$$\frac{\frac{\frac{(1)}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle\langle x \leftarrow N\langle u \leftarrow Z \rangle \rangle : B\langle u \leftarrow Z \rangle\langle x \leftarrow N\langle u \leftarrow Z \rangle \rangle} \quad \frac{(2)}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash N\langle u \leftarrow Z \rangle : A\langle u \leftarrow Z \rangle}}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : B\langle u \leftarrow Z \rangle\langle x \leftarrow N\langle u \leftarrow Z \rangle \rangle} \text{ (}\dagger\text{)} \quad \nabla \text{ (x)}}{\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle} \text{ (3)}} \text{ (}\equiv\text{)}$$

dont la conclusion est ce que l'on voulait montrer.

$$h. \text{ (expansion)} : \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_{\chi} M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

On distingue deux cas :

1°) La propagation est un (gc), i.e. $M = P\{x \leftarrow \gamma_x\}$. Dans ce cas, par hypothèse d'induction, $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash P\{x \leftarrow \gamma_x\}\langle u \leftarrow Z \rangle : B\{x \leftarrow \gamma_x\}\langle u \leftarrow Z \rangle$, et on a alors $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash P\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : B\langle u \leftarrow Z \rangle$ par le lemme de (gc)-expansion (2.23).

2°) Sinon, par hypothèse d'induction, $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle : B\langle u \leftarrow Z \rangle$, et, comme par la convention de Barendregt on peut supposer que $x \notin \text{fv}(Z)$, on a $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash M\langle u \leftarrow Z \rangle\langle x \leftarrow N \rangle : B\langle u \leftarrow Z \rangle$ par le lemme de (gc)-expansion (2.23). On vérifie bien que $P\langle u \leftarrow Z \rangle\langle x \leftarrow N \rangle \longrightarrow_{\chi} M\langle u \leftarrow Z \rangle$, et l'on peut donc dériver $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash P\langle u \leftarrow Z \rangle\langle x \leftarrow N \rangle : B\langle u \leftarrow Z \rangle$. par une figure d'expansion. En outre, par la convention de Barendregt, $u \notin \text{fv}(N)$, d'où $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash P\langle u \leftarrow Z \rangle\langle x \leftarrow N\langle u \leftarrow Z \rangle \rangle : B\langle u \leftarrow Z \rangle$ par le lemme de (gc)-expansion (2.23). On peut alors conclure par une figure d'expansion

(puisque Z est typable) que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash P \langle x \leftarrow N \rangle \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle$, qui est le jugement auquel on voulait aboutir.

$$i. \text{ (conversion) : } \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

Par hypothèse d'induction appliquée aux deux prémisses, on obtient d'une part le jugement $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash M \langle u \leftarrow Z \rangle : A \langle u \leftarrow Z \rangle$ et d'autre part le jugement $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \bar{\sigma} \langle u \leftarrow Z \rangle$, d'où par une figure de conversion l'on tire $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \bar{\sigma}$. De $A \equiv_{\beta x} B$, l'on tire par congruence que $A \langle u \leftarrow Z \rangle \equiv_{\beta x} B \langle u \leftarrow Z \rangle$, et l'on peut conclure par une conversion :

$$\frac{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash M \langle u \leftarrow Z \rangle : A \langle u \leftarrow Z \rangle \quad \Sigma, \Xi \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \bar{\sigma}}{\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash M \langle u \leftarrow Z \rangle : B \langle u \leftarrow Z \rangle} \text{ (}\equiv\text{)}$$

Q.E.D.

Le résultat suivant ressemble quelque peu au précédent, mais prend en compte les cas où la variable substituée n'est pas accessible.

Lemme 2.25 : *Substitution de variable inaccessible*

Soient Θ et Ξ deux contextes, u une variable et W, X, Y, Z quatre termes tels que $\Theta, u : W, \Xi \vdash X : Y$ et $\Theta \vdash Z : W$ et $u \notin \text{av}(X)$. Alors $\Theta, \hat{\Xi} \vdash X : Y \langle u \leftarrow Z \rangle$, où, étant donné un terme T , \hat{T} désigne T si $u \notin \text{av}(T)$ et $T \langle u \leftarrow Z \rangle$ sinon.

Démonstration.

On va en fait montrer la propriété légèrement plus forte suivante : on suppose que $\Theta, u : W, \Xi \vdash X : Y$ et $\Sigma \vdash Z : W$ avec $\Theta \subseteq \Sigma$; alors $\Sigma, \hat{\Xi} \vdash X : Y \langle u \leftarrow Z \rangle$. On procède par induction en employant l'ordre de simplicité.

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

Le contexte étant vide, ce point est trivial.

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

Par hypothèse d'induction appliquée aux deux prémisses, on obtient d'une part que $\Sigma, \hat{\Xi} \vdash A : \rho \langle u \leftarrow Z \rangle$, ce que l'on notera (1) et d'autre part que $\Sigma, \hat{\Xi}, x : A \vdash B : \sigma \langle u \leftarrow Z \rangle$, ce que l'on notera (2). En outre, $\Sigma, \hat{\Xi}$ est bien formé, et donc, par le lemme d'initialisation (2.20), l'on a $\Sigma, \hat{\Xi} \vdash \tau : \int$, ce que l'on note (3). On conclut alors au moyen de l'arbre d'inférence suivant :

$$\frac{\frac{(1) \quad \nabla}{\Sigma, \hat{\Xi} \vdash A : \rho} \text{ (}\equiv\text{)} \quad \frac{(2) \quad \nabla}{\Sigma, \hat{\Xi}, x : A \vdash B : \sigma} \text{ (}\equiv\text{)}}{\Sigma, \hat{\Xi} \vdash \Pi x : A. B : \tau} \text{ (}\mathcal{R}\text{)} \quad \frac{(3) \quad \Theta \vdash Z : W}{\Sigma, \hat{\Xi} \vdash \tau \langle u \leftarrow Z \rangle : \int} \text{ (x)}}{\Sigma, \hat{\Xi} \vdash \Pi x : A. B : \tau \langle u \leftarrow Z \rangle} \text{ (}\equiv\text{)}$$

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Nécessairement, $u \notin x$ et $\Xi = \Xi_1, x : A$. Deux cas se présentent :

- 1°) $u \notin \text{av}(A)$. Dans ce cas, par hypothèse d'induction, $\Sigma, \hat{\Xi}_1 \vdash A : \sigma \langle u \leftarrow Z \rangle$, d'où, par une figure de conversion, $\Sigma, \hat{\Xi}_1 \vdash A : \sigma$, ce que l'on note (1). En outre, par une figure d'expansion, $\Sigma, \hat{\Xi}_1 \vdash A \langle u \leftarrow Z \rangle : \sigma$, ce que l'on note (2). On construit alors l'arbre d'inférence suivant :

$$\frac{\frac{(1)}{\Sigma, \hat{\Xi}_1, x : A \vdash x : A} \text{ (hyp)}}{\Sigma, \hat{\Xi}_1, x : A \vdash x : A \langle u \leftarrow Z \rangle} \text{ (}\equiv\text{)}$$

qui permet de conclure.

- 2°) $u \in \text{av}(A)$. Dans ce cas, par le lemme de substitution explicite appliqué à la prémisse, on obtient $\Sigma, \Xi_1 \langle u \leftarrow Z \rangle \vdash A \langle u \leftarrow Z \rangle : \sigma$. Par ailleurs, par correction des types (2.16) et hypothèse d'induction appliqués successivement à la prémisse, on obtient que $\Sigma, \hat{\Xi}_1$ est le contexte d'un type valide, donc est bien formé, et comme $\Sigma, \hat{\Xi}_1 \equiv_{\beta x} \Sigma, \Xi_1 \langle u \leftarrow Z \rangle$, on a par le lemme de conversion de contexte (2.22) que $\Sigma, \hat{\Xi}_1 \vdash A \langle u \leftarrow Z \rangle : \sigma$, et l'on conclut alors par une simple figure d'hypothèse.

$$d. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

On distingue quatre cas :

- 1°) $u = x$, auquel cas, par le théorème de correction des types (2.16), $\Gamma \vdash B : \bar{\tau}$, d'où, par une figure d'expansion, $\Gamma \vdash B \langle u \leftarrow Z \rangle : \bar{\tau}$, et l'on peut appliquer à la première prémisse une figure de conversion pour obtenir $\Gamma \vdash A : B \langle u \leftarrow Z \rangle$, ce qui est le résultat voulu.
- 2°) $u \neq x$, d'où $\Xi = \Xi_1, x : C$, et $u \notin \text{dom}(\Delta)$, ce qui implique par le lemme de dépendances entre variables que $u \notin \text{av}(C)$. Dans ce cas, par hypothèse d'induction, $\Sigma, \hat{\Xi}_1 \vdash A : B \langle u \leftarrow Z \rangle$, d'où, par une figure d'affaiblissement, $\Sigma, \hat{\Xi}_1, x : C \vdash A : B \langle u \leftarrow Z \rangle$, ce qui est le résultat voulu.
- 3°) $u \neq x$, d'où $\Xi = \Xi_1, x : C$, $u \notin \text{av}(C)$ et $u \in \text{dom}(\Delta)$ i.e. Δ est de la forme $\Delta_1, u : W, \Delta_2$ avec $\Delta_1 \subseteq \Theta$ et $\Delta_2 \subseteq \Xi_1$. Par hypothèse d'induction, $\Sigma, \hat{\Xi}_1 \vdash A : B \langle u \leftarrow Z \rangle$ et $\Sigma \hat{\Delta}_2 \vdash C : \sigma \langle u \leftarrow Z \rangle$, d'où, par une conversion, $\Sigma, \hat{\Delta}_2 \vdash C : \sigma$. Comme $\Sigma, \hat{\Delta}_2 \subseteq \Sigma, \hat{\Xi}$, on peut conclure par une figure d'affaiblissement.
- 4°) $u \neq x$, d'où $\Xi = \Xi_1, x : C$, $u \in \text{av}(C)$ et $\Delta = \Delta_1, u : W, \Delta_2$ avec $\Delta_1 \subseteq \Theta$ et $\Delta_2 \subseteq \Xi_1$. Dans ce cas, d'une part par hypothèse d'induction, $\Sigma, \hat{\Xi}_1 \vdash A : B \langle u \leftarrow Z \rangle$ et d'autre part par le lemme de substitution expli-

cite, $\Sigma, \Delta_2 \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle$, d'où, par une figure de conversion, $\Sigma, \Delta_2 \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma$. On a alors, par le lemme d'affaiblissement, $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash C \langle u \leftarrow Z \rangle : \sigma$, d'où, par le lemme de conversion de contexte (2.22) que $\Sigma, \hat{\Xi} \vdash C \langle u \leftarrow Z \rangle : \sigma$. On conclut alors par une figure d'affaiblissement.

$$e. \text{ (introduction de } \Pi) : \frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

Le lemme d'engendrement (2.21) appliqué à la prémisse $\Gamma \vdash (\Pi x:A.B) : \tau$ indique qu'il existe une règle $(\rho, \sigma, \tau) \in \mathcal{R}$ telle que $\Gamma \vdash A : \rho$ et que $\Gamma, x : A \vdash B : \sigma$, et que ces deux assertions sont plus petites pour l'ordre de simplicité. En appliquant l'hypothèse d'induction à la première, on obtient $\Sigma, \hat{\Xi} \vdash A : \rho \langle u \leftarrow Z \rangle$, d'où, par conversion, $\Sigma, \hat{\Xi} \vdash A : \rho$, ce que l'on note (1), et qui implique la bonne formation du contexte $\Sigma, \hat{\Xi}, x : A$. Par ailleurs, en appliquant le lemme de substitution explicite au deuxième jugement supra, l'on obtient $\Sigma, \Xi \langle u \leftarrow Z \rangle, x : A \langle u \leftarrow Z \rangle \vdash B \langle u \leftarrow Z \rangle : \sigma \langle u \leftarrow Z \rangle$, d'où, par une conversion et le lemme de conversion de contexte (2.22), $\Sigma, \hat{\Xi}, x : A \vdash B \langle u \leftarrow Z \rangle : \sigma$, ce que l'on note (2). On dérive alors :

$$\frac{(1) \quad (2)}{\Sigma, \hat{\Xi} \vdash (\Pi x:A.B \langle u \leftarrow Z \rangle) : \tau} \text{ (R)}$$

dont on note la conclusion (3). En outre, par une figure d'expansion, l'on obtient que $\Sigma, \hat{\Xi} \vdash (\Pi x:A.B) \langle u \leftarrow Z \rangle : \tau$, ce que l'on note (4).

Par ailleurs, par hypothèse d'induction appliquée à la deuxième prémisse, on a $\Sigma, \hat{\Xi}, x : A \vdash M : B \langle u \leftarrow Z \rangle$, ce que l'on note (5). On effectue alors la dérivation suivante :

$$\frac{\frac{(3) \quad (5)}{\Sigma, \hat{\Xi} \vdash (\lambda x:A.M) : (\Pi x:A.B \langle u \leftarrow Z \rangle)} \text{ (}\Pi\text{-1)} \quad (4)}{\Sigma, \hat{\Xi} \vdash (\lambda x:A.M) : (\Pi x:A.B) \langle u \leftarrow Z \rangle} \text{ (}\equiv\text{)}$$

qui permet de conclure.

$$f. \text{ (élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B \langle x \leftarrow N \rangle}$$

Par hypothèse d'induction, $\Sigma, \hat{\Xi} \vdash M : (\Pi x:A.B) \langle u \leftarrow Z \rangle$ et $\Sigma, \hat{\Xi} \vdash N : A \langle u \leftarrow Z \rangle$, ce que l'on note respectivement (1) et (2). Par ailleurs, on applique à la première prémisse successivement le théorème de correction des types, le lemme d'engendrement, le lemme de substitution explicite et une figure de règle pour obtenir que $\Sigma, \Xi \langle u \leftarrow Z \rangle \vdash (\Pi x:A \langle u \leftarrow Z \rangle). B \langle u \leftarrow Z \rangle : \tau$, d'où l'on déduit par le lemme de conversion de contexte (2.22) le jugement de typage

$\Sigma, \hat{\Xi} \vdash (\Pi x:A\langle u \leftarrow Z \rangle. B\langle u \leftarrow Z \rangle) : \tau$, ce que l'on note (3). On effectue alors la dérivation suivante :

$$\frac{\frac{(1) \quad (3)}{\Sigma, \hat{\Xi} \vdash M : (\Pi x:A\langle u \leftarrow Z \rangle. B\langle u \leftarrow Z \rangle)}^{(\equiv)} \quad (2)}{\Sigma, \hat{\Xi} \vdash MN : B\langle u \leftarrow Z \rangle\langle x \leftarrow N \rangle} \text{ (}\Pi\text{-É)}$$

à la conclusion de laquelle on applique successivement le théorème de correction des types (2.16) et une figure d'expansion, ce qui, comme par la convention de Barendregt on peut supposer que $x \notin \text{av}(Z)$, donne que $\Sigma, \hat{\Xi} \vdash B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle : \bar{\sigma}$. On en déduit par une conversion le résultat souhaité, à savoir $\Sigma, \hat{\Xi} \vdash MN : B\langle x \leftarrow N \rangle\langle u \leftarrow Z \rangle$.

$$g. \text{ (coupure)} : \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

On distingue deux cas :

1°) $u \in \text{av}(A)$. Dans ce cas, par hypothèse d'induction, on obtient les jugements de typage $\Sigma, \hat{\Xi}, x : A\langle u \leftarrow Z \rangle \vdash M : B\langle u \leftarrow Z \rangle$ et $\Sigma, \hat{\Xi} \vdash N : A\langle u \leftarrow Z \rangle$. Par une figure de coupure, on obtient $\Sigma, \hat{\Xi} \vdash M\langle x \leftarrow N \rangle : B\langle u \leftarrow Z \rangle\langle x \leftarrow N \rangle$, et l'on conclut exactement comme au cas précédent.

2°) $u \notin \text{av}(A)$. Dans ce cas, par hypothèse d'induction, on obtient les jugements $\Sigma, \hat{\Xi}, x : A \vdash M : B\langle u \leftarrow Z \rangle$ et $\Sigma, \hat{\Xi} \vdash N : A\langle u \leftarrow Z \rangle$, ce que l'on note respectivement (1) et (2). Le contexte $\Sigma, \hat{\Xi}, x : A$ est bien formé, donc $\Sigma, \hat{\Xi} \vdash A : \sigma$, ce que l'on note (3). L'on construit alors l'arbre de dérivation suivant :

$$\frac{\frac{(2) \quad (3)}{\Sigma, \hat{\Xi} \vdash N : A}^{(\equiv)} \quad (1)}{\Sigma, \hat{\Xi} \vdash M\langle x \leftarrow N \rangle : B\langle u \leftarrow Z \rangle\langle x \leftarrow N \rangle}^{(\dagger)}$$

et l'on conclut ensuite exactement comme au cas précédent.

$$h. \text{ (expansion)} : \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_{\chi} M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

Ce cas est direct par induction, car la propagation étant incluse dans la \mathbf{x} -équivalence, elle préserve les variables accessibles comme l'assure le lemme 2.1.

$$i. \text{ (conversion)} : \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta \mathbf{x}} B}{\Gamma \vdash M : B}$$

Par hypothèse d'induction, $\Sigma, \hat{\Xi} \vdash M : B\langle u \leftarrow Z \rangle$. Par ailleurs, par le lemme de substitution explicite, $\Sigma, \Xi\langle u \leftarrow Z \rangle \vdash B\langle u \leftarrow Z \rangle : \sigma\langle u \leftarrow Z \rangle$, d'où, par une figure de conversion et le lemme de conversion de contexte (2.22), l'on tire le jugement $\Sigma, \hat{\Xi} \vdash B\langle u \leftarrow Z \rangle : \sigma$. On peut alors conclure par une simple figure de conversion.

Q.E.D.

Théorème 2.26 : *Réduction du sujet pour β_x*

Soient Γ un contexte et $A : B$ une assertion tels que $\Gamma \vdash A : B$. Soient Γ' et A' tels que $\Gamma \xrightarrow{\beta_x^*} \Gamma'$ et $A \xrightarrow{\beta_x^*} A'$. Alors $\Gamma' \vdash A' : B$.

La démonstration de ce théorème étant relativement lourde, elle sera faite en deux temps : l'on étudiera d'abord la x -réduction, puis la clôture contextuelle de la B -réduction. Dans les deux cas, comme indiqué au début de ce paragraphe, l'on fera la démonstration non pour la réduction simple, mais pour une réduction parallèle la contenant. Dans une optique d'allègement des arbres d'inférence de cette démonstration, l'on adopte les abréviations suivantes : les symboles \blacktriangledown et ∇ représentent respectivement les vérifications $\Theta \vdash Z : W$ (par exemple dans les expansions), et $\Delta \vdash \xi : \int$, où ξ est une sorte (typiquement dans les conversions).

L'on va tout d'abord introduire les deux réductions parallèles, et montrer une propriété de commutation partielle avec la propagation.

Définition 2.17 : *x -réduction fortement parallèle, B -réduction parallèle*

On définit conjointement une notion de X -réduction groupée, notée $\overline{\text{XX}}$, et une notion de x -réduction fortement parallèle, notée $\overline{\text{XX}}$, inductivement au moyen des figures suivantes, où $\frac{(x)}{X}$ désigne l'union des axiomes (quant), (app), (abs), (gc) de la X -réduction :

$$\begin{array}{c} \overline{\text{XX}} \text{ (réflexivité)} \\ \frac{}{M \overline{\text{XX}} M} \end{array} \quad \frac{M \overline{\text{XX}} M' \quad M' \langle x \leftarrow N' \rangle \xrightarrow{\frac{(x)}{X}} M'' \quad N \xrightarrow{\text{x//}} N'}{M \langle x \leftarrow N \rangle \overline{\text{XX}} M''} \text{ (groupe)}$$

$$\frac{M \xrightarrow{X} M'}{M \xrightarrow{\text{x//}} M'} \text{ (X-réd.)} \quad \frac{A \xrightarrow{\text{x//}} A' \quad B \xrightarrow{\text{x//}} B'}{\Pi x : A.B \xrightarrow{\text{x//}} \Pi x : A'.B'} \text{ (\Pi-ctx.)} \quad \frac{A \xrightarrow{\text{x//}} A' \quad B \xrightarrow{\text{x//}} B'}{\lambda x : A.B \xrightarrow{\text{x//}} \lambda x : A'.B'} \text{ (\lambda-ctx.)}$$

$$\frac{M \overline{\text{XX}} M'}{M \xrightarrow{\text{x//}} M'} \text{ (XX-réd.)} \quad \frac{A \xrightarrow{\text{x//}} A' \quad B \xrightarrow{\text{x//}} B'}{A B \xrightarrow{\text{x//}} A' B'} \text{ (A-ctx.)} \quad \frac{A \xrightarrow{\text{x//}} A' \quad B \xrightarrow{\text{x//}} B'}{A \langle x \leftarrow B \rangle \xrightarrow{\text{x//}} A' \langle x \leftarrow B' \rangle} \text{ (S-ctx.)}$$

On définit en outre une notion de B -réduction parallèle $\overline{\text{B//}}$ comme suit :

$$\overline{\overline{\text{B//}}} \text{ (groupe)}$$

$$\frac{A \overline{\text{B//}} A' \quad B \overline{\text{B//}} B'}{\Pi x : A.B \overline{\text{B//}} \Pi x : A'.B'} \text{ (\Pi-ctx.)} \quad \frac{A \overline{\text{B//}} A' \quad B \overline{\text{B//}} B'}{\lambda x : A.B \overline{\text{B//}} \lambda x : A'.B'} \text{ (\lambda-ctx.)}$$

$$\frac{A \overline{\text{B//}} A' \quad B \overline{\text{B//}} B'}{A B \overline{\text{B//}} A' B'} \text{ (A-ctx.)} \quad \frac{A \overline{\text{B//}} A' \quad B \overline{\text{B//}} B'}{A \langle x \leftarrow B \rangle \overline{\text{B//}} A' \langle x \leftarrow B' \rangle} \text{ (S-ctx.)}$$

Intuitivement, la réduction $\xrightarrow{x//}$ déplace en parallèle des blocs de substitution, tandis que $\xrightarrow{B//}$ est une simple réduction parallèle. On montre ensuite un résultat de commutation partielle avec la propagation pour ces réductions :

Lemme 2.27 : *Semi-commutation avec la propagation*

Soit $P\langle x \leftarrow N \rangle$ un terme. Alors :

- (i) on suppose que $P\langle x \leftarrow N \rangle \rightarrow_\chi M$ et que $P\langle x \leftarrow N \rangle \xrightarrow{x//} P'\langle x \leftarrow N' \rangle$
– alors il existe M' tel que $P'\langle x \leftarrow N' \rangle \rightarrow_\chi M'$ et $M \xrightarrow{x//} M'$;
- (ii) on suppose que $P\langle x \leftarrow N \rangle \rightarrow_\chi M$ et que $P\langle x \leftarrow N \rangle \xrightarrow{B//} P'\langle x \leftarrow N' \rangle$
– alors il existe M' tel que $P'\langle x \leftarrow N' \rangle \rightarrow_\chi M'$ et $M \xrightarrow{B//} M'$.

Démonstration.

On procède par induction.

- (i) On distingue trois cas selon la nature de la propagation $P\langle x \leftarrow N \rangle \rightarrow_\chi M$:
 - a. Il s'agit d'un (gc), auquel cas $x \notin \text{av}(P)$ et $M = P\{x \leftarrow \gamma_x\}$. On a alors $x \notin \text{av}(P')$, et l'on pose $M' = M'\{x \leftarrow \gamma_x\}$.
 - b. Il s'agit d'un recul. Les termes prennent alors les formes suivantes :
 $P = T\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_n \leftarrow V_n \rangle$ avec T ne commençant pas par une substitution, et $M = T\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_j \leftarrow V_j \rangle \langle x \leftarrow N \rangle \langle u_{j+1} \leftarrow W_{j+1} \rangle \cdots \langle u_n \leftarrow W_n \rangle$ avec $W_k = V_k\langle x \leftarrow N \rangle$ ou $V_k\{x \leftarrow \gamma_x\}$. L'on considère la forme de T .
 - 1°) $T = \lambda y:Q.R$. Il y a plusieurs possibilités pour P' :
 - α . $P' = \lambda y:Q'.R'\langle u_1 \leftarrow V'_1 \rangle \cdots \langle u_n \leftarrow V'_n \rangle$. Dans cette éventualité, l'on pose $M' = \lambda y:Q'.R'\langle u_1 \leftarrow V'_1 \rangle \cdots \langle u_i \leftarrow V'_i \rangle \langle x \leftarrow N' \rangle \langle u_{i+1} \leftarrow W'_{i+1} \rangle \cdots \langle u_n \leftarrow W'_n \rangle$, où $W'_k = V'_k\langle x \leftarrow N' \rangle$ ou $W'_k = V'_k$ selon la forme de W_k .
 - β . $P' = (\lambda y:Q\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_i \leftarrow V_i \rangle.R\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_i \leftarrow V_i \rangle)\langle u_{i+1} \leftarrow V'_{i+1} \rangle \cdots \langle u_n \leftarrow V'_n \rangle$ avec $i \leq j$. Si l'on est dans cette éventualité, l'on pose $M' = (\lambda y:Q\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_i \leftarrow V_i \rangle.R\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_i \leftarrow V_i \rangle)\langle u_{i+1} \leftarrow V'_{i+1} \rangle \cdots \langle u_j \leftarrow V'_j \rangle \langle x \leftarrow N' \rangle \langle u_{j+1} \leftarrow W'_{j+1} \rangle \cdots \langle u_n \leftarrow W'_n \rangle$, où $W'_k = V'_k\langle x \leftarrow N' \rangle$ ou $W'_k = V'_k$ selon la forme de W_k .
 - γ . $P' = (\lambda y:Q\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_i \leftarrow V_i \rangle.R\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_i \leftarrow V_i \rangle)\langle u_{i+1} \leftarrow V'_{i+1} \rangle \cdots \langle u_n \leftarrow V'_n \rangle$ avec $i > j$. Placé dans cette éventualité, l'on pose $M' = (\lambda y:Q\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_j \leftarrow V_j \rangle \langle x \leftarrow N' \rangle \langle u_{j+1} \leftarrow U_{j+1} \rangle) \cdots \langle u_i \leftarrow U_i \rangle.R\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_j \leftarrow V_j \rangle \langle x \leftarrow N' \rangle \langle u_{j+1} \leftarrow U_{j+1} \rangle) \cdots \langle u_i \leftarrow U_i \rangle \langle u_{i+1} \leftarrow W'_{i+1} \rangle \cdots \langle u_n \leftarrow W'_n \rangle$ où l'on a d'une part que $U_k = V_k\langle x \leftarrow N' \rangle$ ou $U_k = V_k$ et d'autre part que $W'_k = V'_k\langle x \leftarrow N' \rangle$ ou $W'_k = V'_k$ selon la forme de W_k .
 - 2°) $T = \Pi y:Q.R$ ou QR avec Q soit ne commençant pas par une λ -abstraction soit de la forme $\lambda y:P.S$. Ces cas se traitent de manière analogue (simplement, dans le dernier, l'on a trois sous-termes au lieu de deux).
 - 3°) $T = x$. Ce cas est trivial.

- c. *Il s'agit d'un retard. On a alors* $P = T\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_n \leftarrow V_n \rangle$, *et il y a plusieurs possibilités suivant la forme de* T , *qui sont toutes analogues. On traite ici le cas où* T *est de la forme* $\lambda y:Q.R$. *Dans cette éventualité, on a* $P' = (\lambda y:Q'.R')\langle u_1 \leftarrow V'_1 \rangle \cdots \langle u_n \leftarrow V'_n \rangle$ *et* $M = (\lambda y:Q_1.R_1)\langle u_1 \leftarrow W_1 \rangle \cdots \langle u_n \leftarrow W_n \rangle$ *avec* $Q\langle x \leftarrow N \rangle \rightarrow_x Q_1$ *et* $R\langle x \leftarrow N \rangle \rightarrow_x R_1$. *Par hypothèse d'induction, l'on trouve* Q'' *et* R'' *tels que* $Q'\langle x \leftarrow N' \rangle \rightarrow_x Q''$ *et* $Q_1 \xrightarrow{x//} Q''$ *(et id. pour* $R'')$. *On pose alors* $M' = \lambda y:Q''.R''\langle u_1 \leftarrow W'_1 \rangle \cdots \langle u_n \leftarrow W'_n \rangle$.
- (ii) *On distingue trois cas selon la nature de la propagation* $P\langle x \leftarrow N \rangle \rightarrow_x M$:
- a. *Il s'agit d'un (gc), auquel cas* $x \notin \text{av}(P)$ *et* $M = P\{x \leftarrow \gamma_x\}$. *On a alors* $x \notin \text{av}(P')$, *et l'on pose* $M' = P'\{x \leftarrow \gamma_x\}$.
- b. *Il s'agit d'un recul. Les termes prennent alors les formes suivantes :* $P = T\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_n \leftarrow V_n \rangle$ *avec* T *ne commençant pas par une substitution, et* $M = T\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_j \leftarrow V_j \rangle \langle x \leftarrow N \rangle \langle u_{j+1} \leftarrow W_{j+1} \rangle \cdots \langle u_n \leftarrow W_n \rangle$ *avec* $W_k = V_k\langle x \leftarrow N \rangle$ *ou* $V_k\{x \leftarrow \gamma_x\}$ *si* $x \notin \text{av}(V_k)$. *On a* $P' = T'\langle u_1 \leftarrow V'_1 \rangle \cdots \langle u_n \leftarrow V'_n \rangle$. *On pose alors* $M' = T'\langle u_1 \leftarrow V'_1 \rangle \cdots \langle u_j \leftarrow V'_j \rangle \langle x \leftarrow N' \rangle \langle u_{j+1} \leftarrow W'_{j+1} \rangle \cdots \langle u_n \leftarrow W'_n \rangle$ *avec* $W'_k = V'_k\langle x \leftarrow N' \rangle$ *ou* $V'_k\{x \leftarrow \gamma_x\}$ *selon la forme de* W_k .
- c. *Il s'agit d'un retard. On a alors* $P = T\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_n \leftarrow V_n \rangle$ *d'une part, et d'autre part* $P' = T'\langle u_1 \leftarrow V'_1 \rangle \cdots \langle u_n \leftarrow V'_n \rangle$, *et il y a plusieurs possibilités suivant la forme de* T :
- 1°) T *est de la forme* $\lambda y:Q.R$. *Dans cette éventualité, on a* $T' = \lambda y:Q'.R'$ *d'une part, et d'autre part* $M = (\lambda y:Q_1.R_1)\langle u_1 \leftarrow W_1 \rangle \cdots \langle u_n \leftarrow W_n \rangle$ *avec* $Q\langle x \leftarrow N \rangle \rightarrow_x Q_1$ *et* $R\langle x \leftarrow N \rangle \rightarrow_x R_1$. *Par hypothèse d'induction, l'on trouve* Q'' *et* R'' *tels que* $Q'\langle x \leftarrow N' \rangle \rightarrow_x Q''$ *et* $Q_1 \xrightarrow{x//} Q''$ *(et id. pour* $R'')$, *et l'on pose* $M' = (\lambda y:Q''.R'')\langle u_1 \leftarrow W'_1 \rangle \cdots \langle u_n \leftarrow W'_n \rangle$.
- 2°) T *est de la forme* $\Pi y:Q.R$ *ou* QR *avec* Q *ne commençant pas par une abstraction. Cette éventualité est analogue à la précédente.*
- 3°) T *est de la forme* $(\lambda y:Q.R)S$, *et on peut distinguer alors deux possibilités selon la forme de* T' .
- Si* $T' = (\lambda y:Q'.R')S'$, *le raisonnement est analogue au précédent.*
- Sinon,* $T' = R\langle y \leftarrow S \rangle$, *et* $M = ((\lambda y:Q_1.R_1)S_1)\langle u_1 \leftarrow W_1 \rangle \cdots \langle u_n \leftarrow W_n \rangle$ *avec* $Q\langle x \leftarrow N \rangle \rightarrow_x Q_1$ *(et id. pour* R *et* $S)$. *Par hypothèse d'induction, l'on trouve* R' *et* S' *tels que* $R\langle x \leftarrow N' \rangle \rightarrow_x R'$ *et* $R_1 \xrightarrow{x//} R'$ *(et id. pour* $S')$, *et l'on pose* $M' = (R'\langle y \leftarrow S' \rangle)\langle u_1 \leftarrow W'_1 \rangle \cdots \langle u_n \leftarrow W'_n \rangle$.

Dans tous les cas, on trouve M' qui convient.

Q.E.D.

L'on peut maintenant passer à la réduction du sujet proprement dite. Afin de simplifier cette démonstration, l'on va la séparer en deux, et commencer par la montrer pour la x -réduction avant de passer à la βx -réduction complète. La dé-

monstration pour la \mathbf{x} -réduction est en fait la plus délicate, du fait des complications introduites par la figure d'expansion. L'on va donc montrer le lemme dont l'énoncé suit :

Lemme 2.28 : *Réduction du sujet pour \mathbf{x}*

Soient Γ un contexte et $A : B$ une assertion tels que $\Gamma \vdash A : B$. Soient Γ' et A' tels que $\Gamma \xrightarrow{*}_{\mathbf{x}} \Gamma'$ et $A \xrightarrow{*}_{\mathbf{x}} A'$. Alors $\Gamma' \vdash A' : B$.

Démonstration.

On va en fait démontrer la propriété plus forte suivante :

soient Σ, X, Y tels que $\Sigma \vdash X : Y$; soient Σ' et X' tels que $\Sigma \xrightarrow{x//} \Sigma'$ et $X \xrightarrow{x//} X'$. Alors $\Sigma' \vdash X' : Y$.

L'on procède par induction sur la dérivation de $\Sigma \vdash X : Y$ avec l'ordre de simplicité en discriminant selon la dernière figure employée.

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

La propriété est triviale, car les sortes sont \mathbf{x} -irréductibles, et le contexte est vide.

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

On a ici $\Sigma' = \Gamma'$, et $X' = \Pi x:A'.B'$. Par hypothèse d'induction appliquée aux deux prémisses, on obtient que $\Gamma' \vdash A' : \rho$ et $\Gamma', x : A' \vdash B' : \sigma$ (en effet, $\Gamma, x : A \xrightarrow{x//} \Gamma', x : A'$), ce qui permet d'appliquer la figure de règle :

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

qui permet de conclure.

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Nécessairement, x étant $\beta\mathbf{x}$ -irréductible, $X' = x$. En outre, $\Sigma' = \Gamma', x : A'$ avec $A \xrightarrow{x//} A'$. Par hypothèse d'induction appliquée de deux façons, on obtient d'une part $\Gamma' \vdash A' : \sigma$ et d'autre part $\Gamma' \vdash A : \sigma$. Il suffit alors de construire l'arbre d'inférence suivant :

$$\frac{\frac{\Gamma' \vdash A' : \sigma}{\Gamma' \vdash x : A'} \text{ (hyp)} \quad \Gamma' \vdash A : \sigma \quad A \equiv_{\beta\mathbf{x}} A'}{\Gamma' \vdash x : A} \text{ (}\equiv\text{)}$$

pour conclure.

$$d. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

On a $\Sigma' = \Gamma', x : C'$ et $X' = A'$. Par hypothèse d'induction appliquée aux deux prémisses, on obtient que $\Gamma' \vdash A' : B$ et $\Delta' \vdash C' : \sigma$, et l'on conclut par une figure d'affaiblissement :

$$\frac{\Gamma' \vdash A' : B \quad \Delta' \vdash C' : \sigma \quad \Delta' \subseteq \Gamma'}{\Gamma', x : C' \vdash A' : B} \text{ (w)}$$

$$e. \text{ (introduction de } \Pi) : \frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

On a $\Sigma' = \Gamma'$ et $X' = \lambda x:A'.M'$. Par hypothèse d'induction appliquée aux deux prémisses, on obtient d'une part que $\Gamma' \vdash (\Pi x:A'.B) : \tau$ et d'autre part que $\Gamma', x : A' \vdash M' : B$. En outre, en appliquant différemment l'hypothèse d'induction à la première prémisse, on obtient que $\Gamma' \vdash (\Pi x:A.B) : \tau$, et l'on conclut comme suit :

$$\frac{\frac{\Gamma' \vdash (\Pi x:A'.B) : \tau \quad \Gamma', x : A' \vdash M' : B}{\Gamma' \vdash (\lambda x:A'.M') : (\Pi x:A'.B)} \text{ (}\Pi\text{-I)}}{\Gamma' \vdash (\lambda x:A'.M') : (\Pi x:A.B)} \text{ (}\equiv\text{)}$$

$$f. \text{ (élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

On a $\Sigma' = \Gamma'$ et $X' = M'N'$. Par hypothèse d'induction appliquée aux deux prémisses, $\Gamma' \vdash M' : \Pi x:A.B$ et $\Gamma' \vdash N' : A$.

En outre, par le théorème de correction des types (2.16), il existe $\bar{\sigma}$ telle que $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, et l'on peut aussi appliquer l'hypothèse d'induction à cette assertion, ce qui donne $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$.

On effectue alors la dérivation suivante :

$$\frac{\frac{\Gamma' \vdash M' : \Pi x:A.B \quad \Gamma' \vdash N' : A}{\Gamma' \vdash M'N' : B\langle x \leftarrow N' \rangle} \text{ (}\Pi\text{-É)}}{\Gamma' \vdash M'N' : B\langle x \leftarrow N \rangle} \text{ (}\equiv\text{)}$$

qui permet de conclure.

$$g. \text{ (coupure) : } \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

On a $\Sigma' = \Gamma'$, mais plusieurs cas se présentent pour X' .

1°) $X' = M'\langle x \leftarrow N' \rangle$. Alors, par hypothèse d'induction, $\Gamma', x : A \vdash M' : B$ et $\Gamma' \vdash N' : A$. En outre, par le théorème de correction des types (2.16), on a que $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, auquel on peut appliquer l'hypothèse d'induction pour obtenir $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$. On conclut alors comme suit :

$$\frac{\frac{\Gamma', x : A \vdash M' : B \quad \Gamma' \vdash N' : A}{\Gamma' \vdash M'\langle x \leftarrow N' \rangle : B\langle x \leftarrow N' \rangle} \text{ (}\ddagger\text{)} \quad \Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}}{\Gamma' \vdash M'\langle x \leftarrow N' \rangle : B\langle x \leftarrow N \rangle} \text{ (}\equiv\text{)}.$$

2°) $M = \Pi y : P.R$ et $X' = \Pi y : P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle$.

On applique à la prémisse $\Gamma, x : A \vdash M : B$ le lemme d'engendrement (2.21), ce qui donne $(\rho, \sigma, \tau) \in \mathcal{R}$ vérifiant que $\Gamma, x : A \vdash P : \rho$, $\Gamma, x : A, y : P \vdash R : \sigma$ et $B \equiv_{\beta x} \tau$. Par le lemme de substitution explicite (2.24), l'on obtient alors d'une part que $\Gamma' \vdash P\langle x \leftarrow N' \rangle : \rho\langle x \leftarrow N \rangle$ et d'autre part $\Gamma', y : P\langle x \leftarrow N' \rangle \vdash R\langle x \leftarrow N' \rangle : \sigma\langle x \leftarrow N' \rangle$ et l'on peut ensuite effectuer la dérivation suivante :

$$\frac{\frac{\Gamma' \vdash P\langle x \leftarrow N' \rangle : \rho\langle x \leftarrow N \rangle \quad \nabla \quad \Gamma', y : P\langle x \leftarrow N' \rangle \vdash R\langle x \leftarrow N' \rangle : \sigma\langle x \leftarrow N' \rangle \quad \nabla}{\Gamma' \vdash P\langle x \leftarrow N' \rangle : \rho \quad \Gamma', y : P\langle x \leftarrow N' \rangle \vdash R\langle x \leftarrow N' \rangle : \sigma} \text{ (}\equiv\text{)}}{\Gamma' \vdash \Pi y : P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle : \tau} \text{ (}\mathcal{R}\text{)}.$$

Par ailleurs, par le théorème de correction des types (2.16), il existe \bar{v} tel que $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{v}$. On peut appliquer l'hypothèse d'induction à cette assertion, d'où $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}$. Comme $B\langle x \leftarrow N \rangle \equiv_{\beta x} \tau\langle x \leftarrow N \rangle \equiv_{\beta x} \tau$, l'on peut effectuer la dérivation suivante :

$$\frac{\Gamma' \vdash \Pi y : P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle : \tau \quad \Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}}{\Gamma' \vdash \Pi y : P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle : B\langle x \leftarrow N \rangle} \text{ (}\equiv\text{)}.$$

3°) $M = \lambda y : P.Q$ et $X' = \lambda y : P\langle x \leftarrow N' \rangle.Q\langle x \leftarrow N' \rangle$.

On applique à la prémisse $\Gamma, x : A \vdash M : B$ le lemme d'engendrement (2.21) et l'on obtient ainsi qu'il existe une sorte τ et un terme R tels que $\Gamma, x : A \vdash \Pi y : P.R : \tau$, $\Gamma, x : A, y : P \vdash Q : R$ et $B \equiv_{\beta x} \Pi y : P.R$. De même qu'au sous-cas précédent, on déduit du premier de ces jugements que $\Gamma' \vdash \Pi y : P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle : \tau$. Le second, par le lemme de substitution explicite, donne $\Gamma', y : P\langle x \leftarrow N' \rangle \vdash Q\langle x \leftarrow N' \rangle : R\langle x \leftarrow N' \rangle$ (sachant que l'on a bien par hypothèse d'induction que $\Gamma \vdash N' : A$). Par une introduction de Π , l'on obtient alors que $\Gamma' \vdash \lambda y : P\langle x \leftarrow N' \rangle.Q\langle x \leftarrow N' \rangle : \Pi y : P\langle x \leftarrow N' \rangle.Q\langle x \leftarrow N' \rangle$. Enfin, de même qu'au sous-cas précédent, on montre que $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}$, et l'on peut alors obtenir par une conversion $\Gamma' \vdash \lambda y : P\langle x \leftarrow N' \rangle.Q\langle x \leftarrow N' \rangle : B\langle x \leftarrow N \rangle$, ce qui permet de conclure.

4°) $M = QS$ et $X' = Q\langle x \leftarrow N' \rangle S\langle x \leftarrow N' \rangle$.

On applique à la prémisse $\Gamma, x : A \vdash M : B$ le lemme d'engendrement (2.21) et l'on obtient ainsi qu'il existe deux termes P et R tels que $\Gamma, x : A \vdash Q : \Pi x: P.R$ et $\Gamma, x : A \vdash S : P$ et $B \equiv_{\beta x} R\langle y \leftarrow S \rangle$. Par le lemme de substitution explicite appliqué aux jugements précédents, l'on obtient $\Gamma' \vdash Q\langle x \leftarrow N' \rangle : (\Pi x: P.R)\langle x \leftarrow N' \rangle$ et $\Gamma' \vdash S\langle x \leftarrow N' \rangle : P\langle x \leftarrow N' \rangle$.

En outre, de même que dans les deux sous-cas précédents, l'on peut montrer que $\Gamma' \vdash \Pi x: P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle : \tau$, ce que l'on note (1).

L'on peut ensuite effectuer la dérivation suivante :

$$\frac{\frac{\Gamma' \vdash Q\langle x \leftarrow N' \rangle : (\Pi x: P.R)\langle x \leftarrow N' \rangle \quad (1)}{\Gamma' \vdash Q\langle x \leftarrow N' \rangle : \Pi x: P\langle x \leftarrow N' \rangle.R\langle x \leftarrow N' \rangle} \quad (\equiv)}{\Gamma' \vdash Q\langle x \leftarrow N' \rangle S\langle x \leftarrow N' \rangle : R\langle x \leftarrow N' \rangle\langle y \leftarrow S\langle x \leftarrow N' \rangle \rangle} \quad (\Pi\text{-É}).$$

En outre, comme aux deux sous-cas précédents, $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}$ est dérivable. Or $B\langle x \leftarrow N \rangle \equiv_{\beta x} R\langle y \leftarrow S \rangle\langle x \leftarrow N' \rangle \equiv_{\beta x} R\langle x \leftarrow N' \rangle\langle y \leftarrow S\langle x \leftarrow N' \rangle \rangle$ d'après le lemme de composition (2.4), et l'on peut donc ensuite effectuer la dérivation suivante :

$$\frac{\Gamma' \vdash Q\langle x \leftarrow N' \rangle S\langle x \leftarrow N' \rangle : R\langle x \leftarrow N' \rangle\langle y \leftarrow S\langle x \leftarrow N' \rangle \rangle \quad \Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}}{\Gamma' \vdash Q\langle x \leftarrow N' \rangle S\langle x \leftarrow N' \rangle : B\langle x \leftarrow N \rangle} \quad (\equiv).$$

5°) $M = x$ et $X' = N$.

De la prémisse $\Gamma, x : A \vdash M : B$, on déduit, par le lemme d'engendrement (2.21), que $A \equiv_{\beta x} B$, et, par le théorème de correction des types, que $\Gamma, x : A \vdash B : \bar{v}$. On peut appliquer à ce jugement le lemme de substitution explicite, ce qui donne $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}\langle x \leftarrow N \rangle$.

De plus, en appliquant à la prémisse $\Gamma \vdash N : A$ l'hypothèse d'induction, l'on obtient que $\Gamma' \vdash N : A$.

On a en outre $A \equiv_{\beta x} B$, ce qui implique que $A\langle x \leftarrow N \rangle \equiv_{\beta x} B\langle x \leftarrow N \rangle$, et, comme par le lemme de dépendances entre variables $x \notin \text{av}(A)$, on en déduit que $A \equiv_{\beta x} B\langle x \leftarrow N \rangle$.

On peut effectuer la dérivation suivante :

$$\frac{\Gamma' \vdash N : A \quad \frac{\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}\langle x \leftarrow N \rangle \quad \nabla}{\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{v}} \quad (\equiv)}{\Gamma' \vdash N : B\langle x \leftarrow N \rangle} \quad (\equiv) \quad A \equiv_{\beta x} B\langle x \leftarrow N \rangle \quad (\equiv)$$

pour parvenir à la conclusion souhaitée.

6°) $x \notin \text{av}(\mathbf{M})$ et $X' = \mathbf{M}\{x \leftarrow \gamma_x\}$.

Dans ce cas, par hypothèse d'induction appliquée aux deux prémisses, $\Gamma', x : A \vdash \mathbf{M} : B$ et $\Gamma' \vdash N : A$, d'où, par le lemme de substitution de variable inaccessible (2.25), $\Gamma' \vdash \mathbf{M} : B\langle x \leftarrow N \rangle$, d'où, par le lemme d'interversion 2.12, $\Gamma'\{x \leftarrow \gamma_x\} \vdash \mathbf{M}\{x \leftarrow \gamma_x\} : B\langle x \leftarrow N \rangle\{x \leftarrow \gamma_x\}$. En outre, Γ' est bien formé, et par le théorème de correction des types, $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, donc, par le lemme de conversion de contexte (2.22) et par une figure de conversion, $\Gamma' \vdash \mathbf{M}\{x \leftarrow \gamma_x\} : B\langle x \leftarrow N \rangle$, ce qui est le résultat voulu.

7°) $\mathbf{M} = f(\mathbf{Q}, \mathbf{R})\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle$, avec f étant la Π -quantification, la λ -abstraction, ou l'application, et $X' = f(\mathbf{Q}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle, \mathbf{R}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle)$. L'application de l'hypothèse d'induction donne $\Gamma', x : A \vdash f(\mathbf{Q}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle, \mathbf{R}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle) : B$ et $\Gamma' \vdash N : A$. On applique alors successivement le lemme d'engendrement, le lemme de substitution explicite et, selon le cas, la figure de règle, de Π -introduction ou de Π -élimination, ce qui donne $\Gamma' \vdash X' : B\langle x \leftarrow N \rangle$, qui est le résultat souhaité.

$$h. \text{ (expansion) : } \frac{\Gamma \vdash \mathbf{M} : B \quad \Delta \vdash N : A \quad \mathbf{P}\langle x \leftarrow N \rangle \longrightarrow_{\chi} \mathbf{M}}{\Gamma \vdash \mathbf{P}\langle x \leftarrow N \rangle : B}$$

On a $\Sigma' = \Gamma'$. Pour X' , deux cas se présentent :

1°) $X' = \mathbf{P}'\langle x \leftarrow N' \rangle$. Dans ce cas, par le lemme 2.27, qu'il existe \mathbf{M}' tel que $\mathbf{P}'\langle x \leftarrow N' \rangle \longrightarrow_{\chi} \mathbf{M}'$ et $\mathbf{M} \xrightarrow{x//} \mathbf{M}'$. On applique alors l'hypothèse d'induction aux prémisses pour obtenir respectivement $\Gamma' \vdash \mathbf{M}' : B$, que l'on note (1), et $\Delta \vdash N' : A$, que l'on note (2), et l'on peut alors conclure :

$$\frac{(1) \quad (2) \quad \mathbf{P}'\langle x \leftarrow N' \rangle \longrightarrow_{\chi} \mathbf{M}'}{\Gamma' \vdash \mathbf{P}'\langle x \leftarrow N' \rangle : B} \text{ (x)}$$

2°) $X' = f(\mathbf{Q}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle, \mathbf{R}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle)$, où $\mathbf{P} = f(\mathbf{Q}, \mathbf{R})\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle$, f étant un constructeur parmi la Π -quantification, la λ -abstraction et l'application.

Dans ce cas, $\mathbf{M} \xrightarrow{x//} f(\mathbf{Q}', \mathbf{R}')$ avec $\mathbf{Q}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle \longrightarrow_{\chi} \mathbf{Q}'$ (et id. avec \mathbf{R}). Par hypothèse d'induction, on a $\Gamma' \vdash f(\mathbf{Q}', \mathbf{R}') : B$. On traite ici le cas de l'application, les autres sont analogues. Par le lemme d'engendrement, $\Gamma' \vdash \mathbf{Q}' : \Pi y : C.D$ et $\Gamma' \vdash \mathbf{R}' : C$, avec $D\langle y \leftarrow \mathbf{R}' \rangle \equiv_{\beta x} B$, d'où, au moyen d'une figure d'expansion, l'on tire respectivement les jugements $\Gamma' \vdash \mathbf{Q}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle : \Pi y : C.D$, que l'on note (1), et $\Gamma' \vdash \mathbf{R}\langle u_1 \leftarrow V_1 \rangle \cdots \langle u_k \leftarrow V_k \rangle\langle x \leftarrow N \rangle : C$, que l'on note (2). On effectue

alors la dérivation suivante :

$$\frac{\frac{(1) \quad (2)}{\Gamma' \vdash X' : D\langle y \leftarrow R' \rangle} \text{(\Pi-É)}}{\Gamma' \vdash X' : B} \text{(\equiv)}$$

qui permet de conclure.

3°) $x \notin \text{av}(\mathbf{P})$ et $X' = \mathbf{P}\{x \leftarrow \gamma_x\}$. Alors il est clair par la convention de Barrendregt et d'après la définition de la propagation que $M \xrightarrow{x//} \mathbf{P}\{x \leftarrow \gamma_x\}$, on conclut en appliquant l'hypothèse d'induction à la première prémisse.

4°) $X = x\langle x \leftarrow \mathbf{N} \rangle$ et $X' = \mathbf{N}$. Ce cas est impossible car on ne peut appliquer de propagation à $x\langle x \leftarrow \mathbf{N} \rangle$.

i. (conversion) :
$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

Par hypothèse d'induction appliquée aux deux prémisses, $\Gamma' \vdash M' : A$ et $\Gamma' \vdash B : \bar{\sigma}$, et il suffit d'une simple conversion :

$$\frac{\Gamma' \vdash M' : A \quad \Gamma' \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma' \vdash M' : B} \text{(\equiv)}$$

pour conclure.

Q.E.D.

On peut maintenant étendre le résultat à l'axiome de réduction restant :

Lemme 2.29 : Réduction du sujet pour B

Soient Σ, X, Y tels que $\Sigma \vdash X : Y$; soient Σ' et X' tels que $\Sigma \xrightarrow{B//} \Sigma'$ et $X \xrightarrow{B//} X'$. Alors $\Sigma' \vdash X' : Y$.

Il est clair que le théorème de réduction du sujet sera un corollaire du lemme 2.28 et de l'énoncé *supra*.

Démonstration.

L'on procède par induction sur l'ordre de simplicité sur la dérivation $\Sigma \vdash X : Y$, en discriminant selon la dernière figure employée.

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

La propriété est triviale, car les sortes sont irréductibles et le contexte, vide.

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

On a ici $\Sigma' = \Gamma'$, et $X' = \Pi x:A'.B'$. Par hypothèse d'induction appliquée aux deux prémisses, on obtient que $\Gamma' \vdash A' : \rho$ et $\Gamma', x : A' \vdash B' : \sigma$ (en effet,

$\Gamma, x : A \xrightarrow{\mathbb{B}} \Gamma', x : A'$, ce qui permet de conclure par la figure de règle :

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

c. (hypothèse) :
$$\frac{\Gamma \vdash A : \sigma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : A \vdash x : A}$$

Nécessairement, x étant irréductible, $X' = x$. En outre, on a $\Sigma' = \Gamma', x : A'$ avec $A \xrightarrow{\mathbb{B}} A'$. Par hypothèse d'induction appliquée de deux façons, on obtient d'une part $\Gamma' \vdash A' : \sigma$ et d'autre part $\Gamma' \vdash A : \sigma$. On conclut ainsi :

$$\frac{\frac{\Gamma' \vdash A' : \sigma}{\Gamma' \vdash x : A'} \text{ (hyp)} \quad \Gamma' \vdash A : \sigma \quad A \equiv_{\beta x} A'}{\Gamma' \vdash x : A} \text{ (}\equiv\text{)}.$$

d. (affaiblissement) :
$$\frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

On a $\Sigma' = \Gamma', x : C'$ et $X' = A'$. Par hypothèse d'induction appliquée aux deux prémisses, on obtient que $\Gamma' \vdash A' : B$ et $\Delta' \vdash C' : \sigma$, et l'on conclut :

$$\frac{\Gamma' \vdash A' : B \quad \Delta' \vdash C' : \sigma \quad \Delta' \subseteq \Gamma'}{\Gamma', x : C' \vdash A' : B} \text{ (w)}.$$

e. (introduction de Π) :
$$\frac{\Gamma \vdash (\Pi x:A.B) : \tau \quad \Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

On a $\Sigma' = \Gamma'$ et $X' = \lambda x:A'.M'$. Par hypothèse d'induction appliquée aux deux prémisses, on obtient d'une part que $\Gamma' \vdash (\Pi x:A'.B) : \tau$ et d'autre part que $\Gamma', x : A' \vdash M' : B$. En outre, en appliquant différemment l'hypothèse d'induction à la première prémisse, on a $\Gamma' \vdash (\Pi x:A.B) : \tau$, et l'on conclut :

$$\frac{\frac{\Gamma' \vdash (\Pi x:A'.B) : \tau \quad \Gamma', x : A' \vdash M' : B}{\Gamma' \vdash (\lambda x:A'.M') : (\Pi x:A'.B)} \text{ (}\Pi\text{-I)}}{\Gamma' \vdash (\lambda x:A'.M') : (\Pi x:A.B)} \text{ (}\equiv\text{)}.$$

f. (élimination de Π) :
$$\frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

On a $\Sigma' = \Gamma'$. Concernant X' , deux cas se présentent :

1°) $X' = M'N'$, auquel cas par hypothèse d'induction appliquée aux deux prémisses, $\Gamma' \vdash M' : \Pi x:A.B$ et $\Gamma' \vdash N' : A$.

En outre, par le théorème de correction des types (2.16), il existe $\bar{\sigma}$ telle que $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, et l'on peut aussi appliquer l'hypothèse d'induction à cette assertion, ce qui donne $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$.

On peut ensuite conclure directement par une figure d'élimination de Π et une conversion :

$$\frac{\frac{\Gamma' \vdash M' : \Pi x:A.B \quad \Gamma' \vdash N' : A}{\Gamma' \vdash M'N' : B\langle x \leftarrow N' \rangle} (\Pi\text{-É}) \quad \Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}}{\Gamma' \vdash M'N' : B\langle x \leftarrow N \rangle} (\equiv).$$

2°) $M = \lambda x:R.Q$ et $X' = Q\langle x \leftarrow N \rangle$. Dans ce cas, par le lemme d'engendrement (2.21) appliqué à la première prémisse, on obtient P et τ tels que $\Pi x:R.P \equiv_{\beta x} \Pi x:A.B$ et $\Gamma, x : R \vdash Q : P$ et $\Gamma \vdash \Pi x:R.P : \tau$, assertions auxquelles on peut appliquer l'hypothèse d'induction. En appliquant l'hypothèse d'induction à la première de ces assertions, ainsi qu'à la prémisse $\Gamma \vdash N : A$, l'on obtient respectivement $\Gamma', x : R \vdash Q : P$ et $\Gamma' \vdash N : A$.

Par ailleurs, le lemme d'engendrement (2.21) appliqué la seconde de ces assertions indique entre autres qu'il existe une sorte ρ telle que $\Gamma \vdash R : \rho$, et l'on peut alors appliquer l'hypothèse d'induction à cette assertion, ce qui donne $\Gamma' \vdash R : \rho$.

En outre, par le théorème de correction des types (2.16), $\Gamma \vdash B\langle x \leftarrow N \rangle : \tau$, et l'on peut appliquer l'hypothèse d'induction à cette assertion, ce qui donne $\Gamma' \vdash B\langle x \leftarrow N \rangle : \tau$.

Sachant que nécessairement $B\langle x \leftarrow N \rangle \equiv_{\beta x} P\langle x \leftarrow N \rangle$ et $R \equiv_{\beta x} A$, on peut ensuite conclure comme suit :

$$\frac{\frac{\Gamma', x : R \vdash Q : P \quad \frac{\Gamma' \vdash N : A \quad \Gamma' \vdash R : \rho}{\Gamma' \vdash N : R} (\equiv)}{\Gamma' \vdash Q\langle x \leftarrow N \rangle : P\langle x \leftarrow N \rangle} (\dagger) \quad \Gamma' \vdash B\langle x \leftarrow N \rangle : \tau}{\Gamma' \vdash Q\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle} (\equiv).$$

g. (coupure) : $\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$

On a $\Sigma' = \Gamma'$, et $X' = M'\langle x \leftarrow N' \rangle$. Dans ce cas, par hypothèse d'induction, $\Gamma', x : A \vdash M' : B$ et $\Gamma' \vdash N' : A$. En outre, par le théorème de correction des types (2.16), on a que $\Gamma \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$, auquel on peut appliquer l'hypothèse d'induction pour obtenir $\Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}$. On conclut ainsi :

$$\frac{\frac{\Gamma', x : A \vdash M' : B \quad \Gamma' \vdash N' : A}{\Gamma' \vdash M'\langle x \leftarrow N' \rangle : B\langle x \leftarrow N' \rangle} (\dagger) \quad \Gamma' \vdash B\langle x \leftarrow N \rangle : \bar{\sigma}}{\Gamma' \vdash M'\langle x \leftarrow N' \rangle : B\langle x \leftarrow N \rangle} (\equiv).$$

$$h. \text{ (expansion)} : \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

On a $\Sigma' = \Gamma'$, et $X' = P'\langle x \leftarrow N' \rangle$. Par le lemme 2.27, il existe M' tel que $P'\langle x \leftarrow N' \rangle \longrightarrow_x M'$ et $M \xrightarrow{B//} M'$. On applique alors l'hypothèse d'induction aux prémisses pour obtenir respectivement $\Gamma' \vdash M' : B$, que l'on note (1), et $\Delta \vdash N' : A$, que l'on note (2), et l'on peut alors conclure :

$$\frac{(1) \quad (2) \quad P'\langle x \leftarrow N' \rangle \longrightarrow_x M'}{\Gamma' \vdash P'\langle x \leftarrow N' \rangle : B} \text{ (x)}$$

$$i. \text{ (conversion)} : \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

Par hypothèse d'induction appliquée aux deux prémisses, $\Gamma' \vdash M' : A$ et $\Gamma' \vdash B : \bar{\sigma}$, et il suffit d'une simple conversion :

$$\frac{\Gamma' \vdash M' : A \quad \Gamma' \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma' \vdash M' : B} \text{ (}\equiv\text{)}$$

pour conclure.

Q.E.D.

Remarque :

La propriété duale, l'expansion du sujet, n'est pas satisfaite. Il suffit de considérer l'expansion suivante : $x \xleftarrow{\beta x} x\langle y \leftarrow zz \rangle$. En effet, x est trivialement typable. En revanche, $x\langle y \leftarrow zz \rangle$ n'est pas typable, puisqu'il est impossible de typer zz .

Par ailleurs, on peut maintenant montrer la proposition 2.15 énoncée plus haut, dont on rappelle le contenu :

Proposition :

Soient Γ un contexte et $M : B$ une assertion tels que $\Gamma \vdash M : B$. Alors :

- (i) si $M \equiv_{\beta x} \int$, alors $B \equiv_{\beta x} \int$.
- (ii) si $B \equiv_{\beta x} \int$, alors il existe $\bar{\sigma} \in \bar{\mathcal{S}}$, $x_1, \dots, x_k \in \mathcal{U}$, $N_1, \dots, N_k \in \Lambda \mathfrak{X}$ des termes tels que $M = \bar{\sigma}\langle x_1 \leftarrow N_1 \rangle \cdots \langle x_k \leftarrow N_k \rangle$.

Démonstration.

- (i) si $M \equiv_{\beta x} \int$, alors $M \xrightarrow{bx^*} \int$, et donc, par le théorème de réduction du sujet, $\Gamma \vdash \int : B$. De ceci découle, par le lemme d'engendrement, que $B \equiv \bar{\sigma}$ avec $(\int, \bar{\sigma}) \in \mathcal{A}$, et $\bar{\sigma}$ ne peut être que \int .
- (ii) On procède par induction sur la dérivation de $\Gamma \vdash M : B$, en discriminant selon la dernière figure appliquée :
 - (axiome), (affaiblissement), (conversion) : ces cas sont évidents ;

- (règle), (introduction de Π) : ces cas sont triviaux car $B \not\equiv_{\beta x} \int$;
- (hypothèse) : ce cas est trivial car si $B \equiv_{\beta x} \int$, alors par le point (i), on a une sorte σ telle que $\sigma \equiv_{\beta x} \int$, ce qui est impossible.
- (coupure), (expansion) : ces cas sont évidents par hypothèse d'induction ;
- (élimination de Π) : par le lemme d'engendrement, on a $\Gamma, u : W \vdash X : \sigma$ avec $B = X\langle u \leftarrow V \rangle$ et $\Gamma \vdash V : W$, d'où, par une (coupure), $\Gamma \vdash B : \sigma$. Or, comme $B \equiv_{\beta x} \int$, on a $B \xrightarrow{\beta x} \int$, et donc, par le théorème de réduction du sujet, $\Gamma \vdash \int : \sigma$, ce qui est impossible ; le cas est trivial.

Les deux points sont donc satisfaits.

Q.E.D.

3.3. Conservation de la normalisation forte

On a déjà mentionné plus haut qu'il existe une correspondance canonique entre systèmes de types purs explicites et systèmes de types purs implicites : en effet, ces deux espèces de systèmes étant définies formellement uniquement par leur triplet de sortes, axiomes et règles, on peut les apparier en fonction de cela.

Cette partie montre qu'entre les deux systèmes ainsi définis existe une similitude importante : en effet, on va montrer qu'un système de types pur explicite est fortement normalisant si, et seulement si, son correspondant implicite l'est.

Pour ce faire, on va employer dans un premier temps l'ordre récursif de chemin (cf. chapitre I, § 1.3, p. 21) pour montrer la préservation de la normalisation forte, puis l'on étudiera la normalisation forte.

Préservation de la normalisation forte

Étant donnée une relation de réduction \xrightarrow{R} définie sur un ensemble E , on note \mathcal{N}_R^V l'ensemble des termes de E n'admettant pas de R -réduction infinie. On commence par définir un ensemble de $\Lambda\mathfrak{I}\mathbf{x}$ -termes :

$$\mathcal{F} = \{M \in \Lambda\mathfrak{I}\mathbf{x} / \forall N \subseteq M (\mathbf{x}(N) \in \mathcal{N}_\beta^V)\}$$

Définition 2.18 : algèbre de termes \mathcal{T}_Σ

On considère la signature infinie $\Sigma = \{\circ, \mathfrak{Q}, \mathfrak{Q}\} \cup \{\mathfrak{A}^n\}_{n \in \mathbb{N}} \cup \{\mathfrak{S}^n\}_{n \in \mathbb{N}}$, où \circ est une constante, \mathfrak{Q} et \mathfrak{Q} des symboles binaires, et les \mathfrak{A}^n et \mathfrak{S}^n des symboles binaires. On définit la préséance suivante sur Σ :

$$\dots \triangleright \mathfrak{S}^{n+1} \triangleright \mathfrak{A}^{n+1} \triangleright \mathfrak{S}^n \triangleright \mathfrak{A}^n \triangleright \dots \triangleright \mathfrak{S}^0 \triangleright \mathfrak{A}^0 \triangleright \mathfrak{Q} \triangleright \mathfrak{Q} \triangleright \circ.$$

On définit également \mathcal{T}_Σ l'algèbre de termes engendrée par la signature Σ , et l'on considère sur ses termes l'ordre récursif de chemin \leq_{rpo} engendré par la préséance \triangleright et le statut multi-ensemble pour tous les symboles de fonction.

Afin de pouvoir utiliser l'ordre récursif de chemin, il est nécessaire de traduire les termes de \mathcal{F} dans l'algèbre \mathcal{T}_Σ :

Définition 2.19 : *traduction de \mathcal{F} dans \mathcal{T}_Σ*

On définit la traduction $|\cdot|_\Sigma$ des termes de \mathcal{F} dans \mathcal{T}_Σ :

$$\begin{aligned} |\sigma|_\Sigma &= \circ \\ |x|_\Sigma &= \circ \\ |\gamma_x|_\Sigma &= \circ \\ |\Pi x:A.B|_\Sigma &= \mathfrak{Q}(|A|_\Sigma, |B|_\Sigma) \\ |\lambda x:A.B|_\Sigma &= \mathfrak{L}(|A|_\Sigma, |B|_\Sigma) \\ |A B|_\Sigma &= \mathfrak{A}^n(|A|_\Sigma, |B|_\Sigma) \quad \text{où } n = \nu(\mathbf{x}(MN)) \\ |A\langle x \leftarrow B \rangle|_\Sigma &= \mathfrak{S}^n(|A|_\Sigma, |B|_\Sigma) \quad \text{où } n = \nu(\mathbf{x}(M\langle x \leftarrow N \rangle)) \end{aligned}$$

où $\nu(M)$ est la longueur maximale d'une β -réduction commençant par M – qui est bien défini car $M \in \mathcal{F} \subseteq \mathcal{N}_\beta^\forall$.

On peut ensuite montrer que la $\beta\mathbf{x}$ -réduction sur les $\Lambda\mathfrak{T}\mathbf{x}$ -termes se traduit dans l'algèbre \mathcal{T}_Σ par une décroissance stricte pour l'ordre récursif de chemin :

Lemme 2.30 : *Transposition de réduction*

Soit $X \in \mathcal{F}$. On suppose que $X \xrightarrow{\beta\mathbf{x}} X'$. Alors :

- (i) $X' \in \mathcal{F}$ (et donc $|X'|_\Sigma$ est bien définie ;
- (ii) $|X|_\Sigma >_{\text{rpo}} |X'|_\Sigma$.

Démonstration.

- (i) Cette propriété se montre aisément par induction sur la structure de X .
- (ii) On procède par réduction sur la structure de X . On ne montre que les cas où la réduction a lieu à la racine dans X , les autres cas se traitant aisément en appliquant l'hypothèse d'induction :

$$a. \quad X = (\lambda x:A.M)N \xrightarrow{\beta\mathbf{x}} M\langle x \leftarrow N \rangle = X'.$$

$$\begin{aligned} \text{Alors } |X|_\Sigma &= \mathfrak{A}^n(\mathfrak{L}(|A|_\Sigma, |M|_\Sigma), |N|_\Sigma) \\ &>_{\text{rpo}} \mathfrak{S}^{n-1}(|M|_\Sigma, |N|_\Sigma) \\ &= |X'|_\Sigma. \end{aligned}$$

Les longueurs maximales de réduction dans les deux cas diffèrent de 1 car $\mathbf{x}(X) = (\lambda x:\mathbf{x}(A).\mathbf{x}(M))\mathbf{x}(N) \xrightarrow{\beta} \mathbf{x}(M)\{x \leftarrow \mathbf{x}(N)\} = \mathbf{x}(X)$.

$$b. \quad X = (\Pi y:A.B)\langle x \leftarrow N \rangle \xrightarrow{\beta\mathbf{x}} \Pi y:A\langle x \leftarrow N \rangle.B\langle x \leftarrow N \rangle = X'.$$

$$\begin{aligned} \text{Alors } |X|_{\Sigma} &= \mathfrak{S}^n(\mathfrak{Q}(|A|_{\Sigma}, |B|_{\Sigma}), |N|_{\Sigma}) \\ &>_{\text{RPO}} \mathfrak{Q}(\mathfrak{S}^i(|A|_{\Sigma}, |N|_{\Sigma}), \mathfrak{S}^j(|B|_{\Sigma}, |N|_{\Sigma})) \\ &= |X'|_{\Sigma}. \end{aligned}$$

On a $n = i + j$ car $\mathbf{x}(X) = \Pi y:\mathbf{x}(A\langle x \leftarrow N \rangle).\mathbf{x}(B\langle x \leftarrow N \rangle)$.

$$c. \quad X = (\lambda y:A.B)\langle x \leftarrow N \rangle \xrightarrow{\beta_{\mathbf{x}}} \lambda y:A\langle x \leftarrow N \rangle.B\langle x \leftarrow N \rangle = X'.$$

$$\begin{aligned} \text{Alors } |X|_{\Sigma} &= \mathfrak{S}^n(\mathfrak{Q}(|A|_{\Sigma}, |B|_{\Sigma}), |N|_{\Sigma}) \\ &>_{\text{RPO}} \mathfrak{Q}(\mathfrak{S}^i(|A|_{\Sigma}, |N|_{\Sigma}), \mathfrak{S}^j(|B|_{\Sigma}, |N|_{\Sigma})) \\ &= |X'|_{\Sigma}. \end{aligned}$$

On a $n = i + j$ car $\mathbf{x}(X) = \Pi y:\mathbf{x}(A\langle x \leftarrow N \rangle).\mathbf{x}(B\langle x \leftarrow N \rangle)$.

$$d. \quad X = (A B)\langle x \leftarrow N \rangle \xrightarrow{\beta_{\mathbf{x}}} A\langle x \leftarrow N \rangle B\langle x \leftarrow N \rangle = X'.$$

$$\begin{aligned} \text{Alors } |X|_{\Sigma} &= \mathfrak{S}^n(\mathfrak{Q}^k(|A|_{\Sigma}, |B|_{\Sigma}), |N|_{\Sigma}) \\ &>_{\text{RPO}} \mathfrak{Q}^n(\mathfrak{S}^i(|A|_{\Sigma}, |N|_{\Sigma}), \mathfrak{S}^j(|B|_{\Sigma}, |N|_{\Sigma})) \\ &= |X'|_{\Sigma}. \end{aligned}$$

On a d'une part $k \leq n$ car si $\mathbf{x}(AB) \xrightarrow{\beta} T$, alors par compositionnalité de la β -réduction, $\mathbf{x}(AB\langle x \leftarrow N \rangle) = \mathbf{x}(AB)\{x \leftarrow \mathbf{x}(N)\} \xrightarrow{\beta} T\{x \leftarrow \mathbf{x}(N)\}$, et d'autre part $i + j \leq n$ car si $\mathbf{x}(A\langle x \leftarrow N \rangle) \xrightarrow{i/\beta} A'$ et $\mathbf{x}(B\langle x \leftarrow N \rangle) \xrightarrow{j/\beta} B'$, alors $\mathbf{x}(A\langle x \leftarrow N \rangle B\langle x \leftarrow N \rangle) \xrightarrow{i+j/\beta} A'B'$.

$$e. \quad X = x\langle x \leftarrow N \rangle \xrightarrow{\beta_{\mathbf{x}}} N = X'.$$

$$\text{Alors } |X|_{\Sigma} = \mathfrak{S}^n(\circ, |N|_{\Sigma}) >_{\text{RPO}} |N|_{\Sigma} = |X'|_{\Sigma}.$$

$$f. \quad X = M\langle x \leftarrow N \rangle \xrightarrow{\beta_{\mathbf{x}}} M\{x \leftarrow \gamma_x\} = X'.$$

$$\text{Alors } |X|_{\Sigma} = \mathfrak{S}^n(|M|_{\Sigma}, |N|_{\Sigma}) >_{\text{RPO}} |M|_{\Sigma} = |X'|_{\Sigma}.$$

En effet, $|M|_{\Sigma} = |M\{x \leftarrow \gamma_x\}|_{\Sigma}$ car $|x|_{\Sigma} = |\gamma_x|_{\Sigma} = \circ$.

Q.E.D.

Corollaire 2.31 : Préservation de la forte normalisation

Soit M un $\Lambda\mathfrak{T}\mathbf{x}$ -terme.

(i) Si $M \in \mathcal{F}$, alors $M \in \mathcal{N}_{\beta_{\mathbf{x}}}^{\vee}$.

(ii) Si M ne contient pas de substitution explicite et $M \in \mathcal{N}_{\beta}^{\vee}$, alors $M \in \mathcal{N}_{\beta_{\mathbf{x}}}^{\vee}$.

Démonstration.

(i) On suppose que $M \in \mathcal{F}$ et $M \notin \mathcal{N}_{\beta_{\mathbf{x}}}^{\vee}$. $M \notin \mathcal{N}_{\beta_{\mathbf{x}}}^{\vee}$ donc il existe une chaîne infinie de réduction de la forme $M \xrightarrow{\beta_{\mathbf{x}}} M_1 \xrightarrow{\beta_{\mathbf{x}}} M_2 \xrightarrow{\beta_{\mathbf{x}}} \cdots \xrightarrow{\beta_{\mathbf{x}}} M_k \xrightarrow{\beta_{\mathbf{x}}} \cdots$. Comme $M \in \mathcal{F}$, le lemme de transposition 2.30 donne (i) que tout terme M_k est dans \mathcal{F} – donc peut être traduit via $|\cdot|_{\Sigma}$ –, et (ii) que l'on a une suite infinie décroissante $|M|_{\Sigma} >_{\text{RPO}} |M_1|_{\Sigma} >_{\text{RPO}} |M_2|_{\Sigma} >_{\text{RPO}} \cdots >_{\text{RPO}} |M_k|_{\Sigma} >_{\text{RPO}} \cdots$, ce qui contredit la bonne fondation de $>_{\text{RPO}}$ (cf. chapitre 1).

(ii) Si $M \in \mathcal{N}_{\beta}^{\vee}$, alors tous ses sous-termes sont également dans $\mathcal{N}_{\beta}^{\vee}$, et donc $M \in \mathcal{F}$, et l'on peut alors conclure par le point précédent.

Q.E.D.

Traduction implicite/explicite et normalisation forte

Lemme 2.32 : Traduction entre systèmes implicites et explicites

Soient \mathfrak{T} un système de types pur implicite, et \mathfrak{T}_x son système explicite associé. Soit $\Xi \vdash_{\mathfrak{T}_x} Y : Z$ un jugement valide avec $Y \notin_{\beta_x} \int$ et $Z \notin_{\beta_x} \int$. Alors $\mathbf{x}(\Xi) \vdash_{\mathfrak{T}} \mathbf{x}(Y) : \mathbf{x}(Z)$ est un jugement valide (où $\mathbf{x}(\Xi)$ est le contexte Ξ où $\mathbf{x}(\cdot)$ est appliquée à chaque prédicat).

Démonstration.

Par induction sur la dérivation de $\Xi \vdash Y : Z$.

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

La propriété est triviale.

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

Par hypothèse d'induction, $\mathbf{x}(\Gamma) \vdash \mathbf{x}(A) : \rho$ et $\mathbf{x}(\Gamma), x : \mathbf{x}(A) \vdash \mathbf{x}(B) : \sigma$ (puisque $\rho, \sigma \notin_{\beta_x} \int$). On peut appliquer une figure de (règle) à ces jugements, d'où $\mathbf{x}(\Gamma) \vdash \Pi x:\mathbf{x}(A).\mathbf{x}(B) : \tau$, i.e. $\mathbf{x}(\Gamma) \vdash \mathbf{x}(\Pi x:A.B) : \mathbf{x}(\tau)$.

$$c. \text{ (hypothèse) : } \frac{\Gamma \vdash A : \sigma}{\Gamma, x : A \vdash x : A}$$

Par hypothèse d'induction, $\mathbf{x}(\Gamma) \vdash \mathbf{x}(A) : \sigma$ (puisque $\sigma \notin_{\beta_x} \int$) et l'on peut conclure par une figure d'(hypothèse) que $\mathbf{x}(\Gamma), x : \mathbf{x}(A) \vdash x : \mathbf{x}(A)$.

$$d. \text{ (affaiblissement) : } \frac{\Gamma \vdash A : B \quad \Delta \vdash C : \sigma \quad \Delta \subseteq \Gamma \quad x \notin \text{dom}(\Gamma)}{\Gamma, x : C \vdash A : B}$$

Si $B \in_{\beta_x} \int$, la propriété est triviale. Sinon, par hypothèse d'induction, $\mathbf{x}(\Gamma) \vdash \mathbf{x}(A) : \mathbf{x}(B)$ et $\mathbf{x}(\Delta) \vdash \mathbf{x}(C) : \sigma$, et il suffit d'appliquer une figure d'(affaiblissement) pour conclure que $\mathbf{x}(\Gamma), x : \mathbf{x}(C) \vdash \mathbf{x}(A) : \mathbf{x}(B)$.

$$e. \text{ (introduction de } \Pi) : \frac{\Gamma \vdash (\Pi x:A.B) : \sigma \quad \Gamma, x : A \vdash M : B \quad x \notin \text{dom}(\Gamma)}{\Gamma \vdash (\lambda x:A.M) : (\Pi x:A.B)}$$

Par le lemme d'engendrement 2.21 appliqué à $\Gamma \vdash (\Pi x:A.B) : \sigma$, il existe $\rho \in \mathcal{S}$ telle que $\Gamma \vdash B : \rho$, d'où, par la proposition 2.15 (i), $B \notin_{\beta_x} \int$. De plus, $\Pi x:A.B \notin_{\beta_x} \int$. On obtient donc par hypothèse d'induction d'une part $\mathbf{x}(\Gamma) \vdash \mathbf{x}(\Pi x:A.B) : \mathbf{x}(\sigma)$, qui se récrit $\mathbf{x}(\Gamma) \vdash \Pi x:\mathbf{x}(A).\mathbf{x}(B) : \sigma$, et d'autre part $\mathbf{x}(\Gamma), x : \mathbf{x}(A) \vdash \mathbf{x}(M) : \mathbf{x}(B)$. Par une figure d'introduction de Π , l'on obtient $\mathbf{x}(\Gamma) \vdash (\lambda x:\mathbf{x}(A).\mathbf{x}(M)) : (\Pi x:\mathbf{x}(A).\mathbf{x}(B))$, ce qui se récrit $\mathbf{x}(\Gamma) \vdash \mathbf{x}(\lambda x:A.M) : \mathbf{x}(\Pi x:A.B)$.

$$f. \text{ (élimination de } \Pi) : \frac{\Gamma \vdash M : (\Pi x:A.B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B\langle x \leftarrow N \rangle}$$

De même que B dans le cas précédent, l'on montre que $A \notin_{\beta_x} \int$; de plus,

$\Pi x:A.B \not\equiv_{\beta x} \int$. Donc par hypothèse d'induction, $\mathbf{x}(\Gamma) \vdash \mathbf{x}(N) : \mathbf{x}(A)$ et $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M) : \mathbf{x}(\Pi x:A.B)$, ce qui se récrit $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M) : (\Pi x:\mathbf{x}(A).\mathbf{x}(B))$. D'où, par une figure d'élimination de Π , $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M)\mathbf{x}(N) : \mathbf{x}(B)[x \leftarrow \mathbf{x}(N)]$, ce qui se récrit $\mathbf{x}(\Gamma) \vdash \mathbf{x}(MN) : \mathbf{x}(B\langle x \leftarrow N \rangle)$.

$$g. \text{ (coupure)} : \frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash M\langle x \leftarrow N \rangle : B\langle x \leftarrow N \rangle}$$

Par hypothèse d'induction, $\mathbf{x}(\Gamma), x : \mathbf{x}(A) \vdash \mathbf{x}(M) : \mathbf{x}(B)$ d'une part et $\mathbf{x}(\Gamma) \vdash \mathbf{x}(N) : \mathbf{x}(A)$ d'autre part. On applique alors le lemme de substitution 1.5, ce qui donne le jugement $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M)[x \leftarrow \mathbf{x}(N)] : \mathbf{x}(B)[x \leftarrow \mathbf{x}(N)]$, dans lequel on reconnaît bien $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M\langle x \leftarrow N \rangle) : \mathbf{x}(B\langle x \leftarrow N \rangle)$.

$$h. \text{ (expansion)} : \frac{\Gamma \vdash M : B \quad \Delta \vdash N : A \quad P\langle x \leftarrow N \rangle \longrightarrow_x M}{\Gamma \vdash P\langle x \leftarrow N \rangle : B}$$

On vérifie aisément que $P\langle x \leftarrow N \rangle \longrightarrow_x M$ implique que $\mathbf{x}(P\langle x \leftarrow N \rangle) = \mathbf{x}(M)$, et la propriété est évidente par hypothèse d'induction.

$$i. \text{ (conversion)} : \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : \bar{\sigma} \quad A \equiv_{\beta x} B}{\Gamma \vdash M : B}$$

$B \not\equiv_{\beta x} \int$ donc, comme $A \equiv_{\beta x} B$, $A \not\equiv_{\beta x} \int$. Donc par hypothèse d'induction, $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M) : \mathbf{x}(A)$. Deux cas se présentent alors :

1°) Soit il existe $\rho \neq \int$ tel que $\Gamma \vdash B : \rho$. Il est aisé de montrer que si $\Gamma \vdash B : \int [h]$ et $\Gamma \vdash B : \rho$, alors $\Gamma \vdash B : \rho [h]$; et l'on peut donc appliquer l'hypothèse d'induction à $\Gamma \vdash B : \rho [k-1]$, et conclure ensuite par une figure de (conversion).

2°) Soit $\bar{\sigma} = \int$ et il n'existe aucune sorte $\rho \in \mathcal{S}$ telle que $\Gamma \vdash B : \rho$. Alors, comme $\Gamma \vdash B : \int$, par la proposition 2.15 (ii), il existe une sorte τ telle que $B = \tau\langle x_1 \leftarrow N_1 \rangle \cdots \langle x_k \leftarrow N_k \rangle$ avec tous les N_i typables. Or $A \equiv_{\beta x} B$, donc $\mathbf{x}(A) \equiv_{\beta} \mathbf{x}(B) = \tau$. On suppose que τ est typable, i.e. qu'il existe une sorte $v \in \mathcal{S}$ telle que $(\tau, v) \in \mathcal{A}$. Alors, comme $B = \tau\langle x_1 \leftarrow N_1 \rangle \cdots \langle x_k \leftarrow N_k \rangle$ avec tous les N_i typables, par le lemme d'initialisation 2.20 et k applications de la figure d'(expansion), $\Gamma \vdash B : v$, contradiction. Donc τ est terminale, et donc par la proposition 2.14, comme $\mathbf{x}(\Gamma) \vdash \mathbf{x}(M) : \mathbf{x}(A)$ et $\mathbf{x}(A) \equiv_{\beta} \mathbf{x}(B) = \tau$, on a $\mathbf{x}(A) = \tau = \mathbf{x}(B)$, et l'on conclut. Q.E.D.

La propriété de préservation de forte normalisation étant établie, il manque peu de choses pour passer à celle de forte normalisation, que l'on énonce comme suit à l'état de conjecture :

Conjecture 2.33 : *Normalisation forte*

Soit \mathfrak{X} un système de types pur explicite, et \mathfrak{I} le système implicite correspondant. \mathfrak{X} est fortement normalisant si, et seulement si, \mathfrak{I} l'est.

On peut en fait montrer aisément le sens direct de cette équivalence, c'est sur le sens réciproque que porte la conjecture.

Démonstration.

On suppose que $\mathfrak{T}\mathbf{x}$ est fortement normalisant.

Soit X_0 un terme typable dans \mathfrak{T} et admettant une chaîne de réduction infinie $X_0 \xrightarrow{\beta} X_1 \xrightarrow{\beta} \cdots \xrightarrow{\beta} X_n \xrightarrow{\beta} X_{n+1} \xrightarrow{\beta} \cdots$. A fortiori X_0 est typable dans $\mathfrak{T}\mathbf{x}$. Or, un pas de β -réduction peut être simulé par plusieurs pas de $\beta\mathbf{x}$ -réduction, et l'on a donc $X_0 \xrightarrow{\beta\mathbf{x}^+} X_1 \xrightarrow{\beta\mathbf{x}^+} \cdots \xrightarrow{\beta\mathbf{x}^+} X_n \xrightarrow{\beta\mathbf{x}^+} X_{n+1} \xrightarrow{\beta\mathbf{x}^+} \cdots$, ce qui contredit le fait que $\mathfrak{T}\mathbf{x}$ est fortement normalisant. Q.E.D.

Pour le sens réciproque, on fait habituellement usage de deux stratégies. La première, introduite par R. BLOO, repose sur la propriété de typabilité des sous-termes d'un terme typable :

On suppose que \mathfrak{T} est fortement normalisant. Soit M un $\Lambda\mathfrak{T}\mathbf{x}$ -terme tel qu'il existe Γ et B tels que $\Gamma \vdash_{\mathfrak{T}\mathbf{x}} M : B$. Soit N un sous-terme de M , par typabilité des sous-termes, il existe alors Δ et A tels que $\Delta \vdash_{\mathfrak{T}\mathbf{x}} N : A$, et donc $\Delta \vdash_{\mathfrak{T}\mathbf{x}}(N) : \mathbf{x}(A)$, et donc $N \in \mathcal{N}_{\beta}^{\vee}$ (puisque \mathfrak{T} est fortement normalisant). Et donc, par définition, $M \in \mathcal{F}$, et donc, par le lemme 2.31, $M \in \mathcal{N}_{\beta\mathbf{x}}^{\vee}$, ce qui montre que $\mathfrak{T}\mathbf{x}$ est fortement normalisant.

La seconde, introduite par H. HERBELIN et développée par E. POLONOVSKI dans [80], consiste à essayer de trouver une traduction des termes impurs dans les termes purs telle que le terme pur se $\beta\mathbf{x}$ -réduise à un terme convertible en un certain sens au terme impur, afin que la normalisation forte du terme pur puisse être « héritée » par le terme impur (en effet, l'ensemble des termes fortement normalisants est clos par réduction).

L'application de ces deux stratégies est rendue très difficile par la figure d'expansion. La conjecture de typabilité des sous-termes *supra* semble en effet raisonnable, mais la présence des deux figures de coupure et d'expansion permettant de typer les substitutions rend la chose très complexe. Par ailleurs, la définition d'une traduction des termes impurs dans les termes purs est également plus difficile à concevoir du fait de la possibilité que la propagation dans une figure d'expansion soit l'application de la règle de (gc).

CHAPITRE III

Systemes de types purs et séquents classiques

— SOMMAIRE —

1. Le $\tilde{\lambda}\mu\tilde{\mu}$-calcul	131
1.1. Historique	131
1.2. Syntaxe et réduction	132
1.3. Séquents et typage	134
2. Systemes de types purs pour le $\tilde{\lambda}\mu\tilde{\mu}$-calcul	137
2.1. Syntaxe et réduction	137
2.2. Typage	142
2.3. Correction des types	145
3. Propriétés remarquables	146
3.1. Lemmes d'inférence de types	147
3.2. Réduction du sujet	149
3.3. Normalisation forte	151

1. Le $\bar{\lambda}\mu\bar{\mu}$ -calcul

1.1. Historique

La problématique générale dans laquelle on se place est la suivante : la correspondance de Curry-Howard(-De Bruijn), que l'on a déjà présentée au chapitre I § 2.1, a été à son origine développée dans le cadre de la logique intuitionniste. Toutefois, l'observation que certaines tautologies de la logique classique évoquent le typage de certains opérateurs de contrôle conduisit T. GRIFFIN en 1990 dans [49] à entamer l'extension de cette correspondance à la logique classique. Parmi les pistes les plus prometteuses dans cette optique, on peut noter le $\lambda\mu$ -calcul de M. PARIGOT (cf. en particulier [79]), dont le système de types associé est basé sur un formalisme de déduction naturelle pour la logique classique.

Une approche alternative, qui est celle que l'on va appliquer ici, est celle du $\bar{\lambda}\mu\bar{\mu}$ -calcul, introduit dans [21] en 2000 par P.-L. CURIEN ET H. HERBELIN, dont les termes représentent des dérivations dans un calcul de séquent, tandis que la réduction sur ces termes représentent l'élimination des coupures. L'un des avantages de cette approche par les calculs de séquents est sa symétrie intrinsèque, qui peut être mise en parallèle avec la symétrie inhérente de la logique, et spécialement de la logique classique. Dans [21], le $\bar{\lambda}\mu\bar{\mu}$ -calcul est introduit avec un système de types simples, et ses versions en appel par valeur et en appel par nom sont encodées en λ -calcul simplement typé *via* une traduction CPS – ce qui montre en particulier la forte normalisation de ce système de types.

Le $\bar{\lambda}\mu\bar{\mu}$ -calcul a par ailleurs été enrichi en calcul à substitutions explicites. On pourra en particulier se reporter à l'article [81] d'E. POLONOVSKI, où celui-ci étudie la forte normalisation du $\bar{\lambda}\mu\bar{\mu}$ -calcul pour la réduction sans restriction, puis étend cette propriété à une variante de son calcul obtenue par l'adjonction de substitutions explicites. L'une des importantes difficultés posées par ce problème est que cette réduction n'est pas confluente, du fait d'une paire critique. La méthode exposée dans [81] fait usage d'une notion de « candidats de symétries », trouvant son origine dans [4] de F. BARBANERA ET S. BERARDI.

Par ailleurs, dans l'article [33] de D. DOUGHERTY, S. GHILEZAN ET P. LESCANNE, on trouve une étude des propriétés computationnelles du $\bar{\lambda}\mu\bar{\mu}$ -calcul non typé, et en particulier une caractérisation des termes fortement normalisants (en appel par valeur ou par nom) au moyen d'un système de types à intersection et union, sous forme d'un calcul de séquents analogue à celui des types simples définis dans [21].

Enfin, un travail allant dans la même direction que celui présenté ici est le calcul F_{ω}^C , introduit par S. LENGREND ET A. MIQUEL dans [69] et repris par S. Lengrand dans sa thèse [68], qui est une extension du système de types pur $F\omega$. Au contraire de l'approche suivie ici, qui est résolument syntaxique, leur approche est plus sémantique : partant du constat que les types dépendants compliquent nécessairement beaucoup les choses lorsque l'on parle de logique classique, car ils peuvent introduire de la non-confluence dans les formules, ils considèrent le système $F\omega$, qui est le plus grand système du λ -cube ne contenant pas les types dépendants, et où l'on peut donc séparer syntaxiquement les types et les termes. Ils étendent ce système de manière à obtenir un calcul de séquents, et montrent la cohérence du système obtenu.

1.2. Syntaxe et réduction

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

Définition 3.1 : $\bar{\lambda}\mu\bar{\mu}$ -calcul ; *appelants, appelés, capsules*

Soient \mathcal{X} et \mathcal{V} deux ensembles infinis dénombrables de variables appelées respectivement *variables d'appelant* et *variables d'appelé*. On définit alors trois ensembles d'expressions au moyen de la grammaire algébrique suivante :

$$\begin{array}{lll}
 \mathbf{R} & ::= & x \quad (\text{variable d'appelant}) \\
 & | & \lambda x.R \quad (\lambda\text{-abstraction}) \\
 & | & \mu\alpha.C \quad (\mu\text{-abstraction}) \\
 \\
 \mathbf{E} & ::= & \alpha \quad (\text{variable d'appelé}) \\
 & | & \mathbf{R}\bullet\mathbf{E} \quad (\text{empilement}) \\
 & | & \bar{\mu}x.C \quad (\bar{\mu}\text{-abstraction}) \\
 \\
 \mathbf{C} & ::= & \langle \mathbf{R} \parallel \mathbf{E} \rangle \quad (\text{capsule})
 \end{array}$$

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

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

- \mathbf{R} décrit les *appelants* (parfois aussi appelés termes) ;
- \mathbf{E} décrit les *appelés* (parfois aussi appelés environnements) ;
- \mathbf{C} décrit les *capsules* (parfois aussi appelées commandes) ;

On observe qu'il existe deux espèces différentes de variables : les variables dites respectivement *d'appelant*, qui décrivent l'ensemble \mathcal{X} , et *d'appelé*, qui décrivent l'ensemble \mathcal{V} . Par convention, l'on dénotera respectivement les variables d'appelant et d'appelé par des lettres minuscules latines et grecques.

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

Définition 3.2 : *variables libres/liées pour $\bar{\lambda}\mu\bar{\mu}$*

On définit les variables d'appelant (respectivement d'appelé) libres ou liées d'une expression comme appartenant à une partie de \mathcal{X} (respectivement \mathcal{V}) définie par induction sur l'expression.

Les ensembles des variables d'appelant libres, des variables d'appelant liées, des variables d'appelé libres, des variables d'appelé liées de M sont respectivement dénotées par $fv_{\mathcal{X}}(M)$, $bv_{\mathcal{X}}(M)$, $fv_{\mathcal{V}}(M)$, $bv_{\mathcal{V}}(M)$.

Les variables d'appelant libres sont définies comme suit :

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

et les variables liées comme suit :

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

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

Parmi tous les termes exposés ci-dessus, les expressions essentielles de $\bar{\lambda}\mu\bar{\mu}$, *i.e.* celles sur lesquelles se font les calculs, sont les capsules. Le lecteur désireux de trouver de plus amples explications sur la notion de calcul sur les capsules de $\bar{\lambda}\mu\bar{\mu}$ pourra en trouver dans [33], où le comportement opérationnel de ces termes

est décrit en détail. On se contentera ici de donner leur sémantique opérationnelle, *i.e.* la notion de réduction employée.

Définition 3.3 : *réduction pour $\bar{\lambda}\mu\tilde{\mu}$*

La réduction, notée \rightarrow , est définie comme la clôture contextuelle de l'union des trois axiomes de réduction suivants :

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

On donne ici non les règles originelles données dans [21], mais une variante dont la différence réside dans l'axiome (λ) , qui, dans [21], a la forme suivante :

$$\langle \lambda x.M \parallel N \bullet E \rangle \rightarrow \langle N \parallel \tilde{\mu} x.(M \parallel E) \rangle$$

Toutefois, il est clair que la différence entre ces deux axiomes réside dans l'application d'un axiome $(\tilde{\mu})$.

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

On constate que la notion de réduction ainsi obtenue n'est pas confluente. En effet, si l'on applique les deux dernières règles à la capsule $\langle \mu \xi.C_1 \parallel \tilde{\mu} x.C_2 \rangle$, on obtient la paire critique $(C_1[\tilde{\mu} x.C_2/\xi], C_2[\mu \xi.C_1/x])$, qui n'est pas joignable, comme on peut le voir sur le simple contre-exemple suivant : on considère la capsule $C = \langle \mu \xi.\langle a \parallel \alpha \rangle \parallel \tilde{\mu} x.\langle b \parallel \beta \rangle \rangle$, où a, b (respectivement α, β) sont des variables d'appelant (respectivement d'appelés) distinctes. On observe d'une part que $C \xrightarrow{(\mu)} \langle a \parallel \alpha \rangle$ et d'autre part que $C \xrightarrow{(\tilde{\mu})} \langle b \parallel \beta \rangle$. Ces deux capsules étant de toute évidence irréductibles et distinctes, donc injoignables, il est impossible de clore le diagramme.

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

1.3. Séquents et typage

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

Soient un ensemble Φ de formules définies sur une signature donnée. On appelle *séquent* un couple de parties de Φ noté $\Gamma \vdash \Delta$. Un séquent s'interprète comme suit : la conjonction des formules de Γ implique la disjonction des formules de Δ . Les séquents se dérivent au moyens d'axiomes, d'introductions gauches et droites pour chaque connecteur de la signature des formules, et de coupures. On donne ici comme exemple dans la table 4 les figures de dérivation valides du calcul des séquents sur les variables propositionnels et l'implication \rightarrow :

$$\begin{array}{c}
 \overline{\Gamma, \alpha \vdash \alpha, \Delta} \text{ (axiome)} \\
 \\
 \frac{\Gamma \vdash \alpha, \Delta \quad \Gamma, \beta \vdash \Delta}{\Gamma, \alpha \rightarrow \beta \vdash \Delta} \text{ (}\rightarrow\text{-G)} \qquad \frac{\Gamma, \alpha \vdash \beta, \Delta}{\Gamma \vdash \alpha \rightarrow \beta, \Delta} \text{ (}\rightarrow\text{-D)} \\
 \\
 \frac{\Gamma \vdash \alpha, \Delta \quad \Gamma, \alpha \vdash \Delta}{\Gamma \vdash \Delta} \text{ (coupure)}
 \end{array}$$

TABLE 4 – Calcul des séquents sur la signature $\{\rightarrow\}$

On remarque dans ces figures de dérivation qu'une formule est en général distinguée : la formule où a lieu l'introduction du connecteur. Si l'on considère maintenant le calcul des séquents dans une perspective de traitement automatique, il semble important de repérer cette formule, d'où l'idée d'introduire une notion de *formule active*, que l'on va définir ici. Étant donné l'ensemble Φ de formules, on construit l'ensemble $\hat{\Phi} = \Phi \times \{0, 1\}$, dont les éléments sont de deux espèces : les formules inactives, de la forme $(\varphi, 0)$ et les formules actives, de la forme $(\varphi, 1)$. Plutôt que d'écrire ces couples, on adopte la convention d'écriture suivante :

Convention :

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

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

tuelle formule active joue un rôle particulier dans la dérivation : dans toute figure d'introduction, la formule où le connecteur a été introduit est active et dérive d'une formule active dans les prémisses. En outre, la coupure fait disparaître les formules actives, et de nouvelles figures en font apparaître dans un séquent n'en ayant pas. Pour poursuivre l'exemple précédent, on donne dans la table 5 les figures de dérivation valides du calcul des séquents à formules actives sur les variables propositionnels et l'implication \rightarrow :

$$\begin{array}{c}
\frac{}{\Gamma, \alpha \vdash \alpha, \Delta} \text{(ax-G)} \qquad \frac{}{\Gamma, \alpha \vdash \alpha, \Delta} \text{(ax-D)} \\
\frac{\Gamma \vdash \alpha, \Delta \quad \Gamma, \beta \vdash \Delta}{\Gamma, \alpha \rightarrow \beta \vdash \Delta} \text{(}\rightarrow\text{-G)} \qquad \frac{\Gamma, \alpha \vdash \beta, \Delta}{\Gamma \vdash \alpha \rightarrow \beta, \Delta} \text{(}\rightarrow\text{-D)} \\
\frac{\Gamma, \alpha \vdash \Delta}{\Gamma, \alpha \vdash \Delta} \text{(act-G)} \qquad \frac{\Gamma \vdash \alpha, \Delta}{\Gamma \vdash \alpha, \Delta} \text{(act-D)} \\
\frac{\Gamma \vdash \alpha, \Delta \quad \Gamma, \alpha \vdash \Delta}{\Gamma \vdash \Delta} \text{(coupure)}
\end{array}$$

TAB. 5 – Calcul des séquents à formule active sur la signature $\{\rightarrow\}$

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

Définition 3.4 : *assertion, contexte, jugement (valide) de typage dans $\bar{\lambda}\mu\tilde{\mu}$*

On appelle *assertion de typage* un couple d'expressions noté $M : R$, où M (nommé *sujet*) peut être un appelant ou un appelé, et R (nommé *prédicat*) est un appelant. Une assertion peut être ou non *active*.

On appelle *contexte de typage* un ensemble d'assertions de typage non actives dont les sujets sont des variables distinctes de même nature. Si ces variables sont des variables d'appelant, on parle de *contexte gauche*. Si ce sont des variables d'appelé, on parle de *contexte droit*. Le contexte vide est noté \emptyset , et la concaténation de contextes d'un même côté et de supports disjoints est notée par une virgule.

On appelle *jugement de typage* une expression de l'une des trois formes suivantes, Γ et Δ étant respectivement des contextes gauche et droit de typage :

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

Un jugement est dit *valide* s'il dérive de l'application des figures¹⁶ d'inférence présentées dans la table 6.

$$\begin{array}{c}
\frac{}{\Gamma; \alpha : A \vdash \alpha : A, \Delta} \text{(ax-G)} \qquad \frac{}{\Gamma, x : A \vdash x : A; \Delta} \text{(ax-D)} \\
\frac{\Gamma \vdash R : A; \Delta \quad \Gamma; E : B \vdash \Delta}{\Gamma; R \bullet E : A \rightarrow B \vdash \Delta} \text{(}\rightarrow\text{-G)} \qquad \frac{\Gamma, x : A \vdash R : B; \Delta}{\Gamma \vdash \lambda x. R : A \rightarrow B; \Delta} \text{(}\rightarrow\text{-D)} \\
\frac{C : (\Gamma, x : A \vdash \Delta)}{\Gamma; \tilde{\mu}x. C : A \vdash \Delta} \text{(ax-G)} \qquad \frac{C : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \vdash \mu\alpha. C : A; \Delta} \text{(ax-D)} \\
\frac{\Gamma \vdash R : A; \Delta \quad \Gamma; E : A \vdash \Delta}{\langle R \parallel E \rangle : (\Gamma \vdash \Delta)} \text{(coupure)}
\end{array}$$

TAB. 6 – Figures d'inférence de type valides pour $\bar{\lambda}\mu\tilde{\mu}$ simplement typé

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

Dans cette partie, l'on va présenter dans un premier paragraphe une syntaxe du $\bar{\lambda}\mu\tilde{\mu}$ -calcul étendue pour les systèmes de types purs, puis, dans un deuxième paragraphe, le typage pour ce système, et enfin, dans un troisième paragraphe, on exposera une notion de bonne formation et de correction des types.

2.1. Syntaxe et réduction

On adopte ici la présentation du $\bar{\lambda}\mu\tilde{\mu}$ -calcul donnée dans [33], qui a été présentée dans la section précédente. Naturellement, il va être nécessaire d'étendre la syntaxe pour l'adapter aux nécessités des systèmes de types purs, qui imposent que termes et types ne soient pas syntaxiquement séparables.

La définition formelle des systèmes de types purs va, une nouvelle fois, rester inchangée : il s'agit d'une spécification du système sous la forme d'un triplet $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ de sortes, axiomes et règles, indiquant l'expressivité du système de types

¹⁶ On conserve encore ici cette dénomination inspirée de G. GENTZEN.

en précisant les espèces permises de dépendances, sans prendre en compte le calcul sous-jacent et le système d'inférence de types qui viendront s'adjoindre à cette spécification. Le calcul susmentionné est défini comme suit :

Définition 3.5 : \mathfrak{T} -appelants, \mathfrak{T} -appelés, \mathfrak{T} -capsules

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

$$\begin{array}{ll}
 \mathbf{R} & ::= \quad x \quad \quad \quad (\text{variable d'appelant}) \\
 & | \quad \lambda x:\mathbf{R}.\mathbf{R} \quad \quad (\lambda\text{-abstraction}) \\
 & | \quad \mu\alpha.\mathbf{C} \quad \quad \quad (\mu\text{-abstraction}) \\
 & | \quad \sigma \quad \quad \quad (\text{sorte}) \\
 & | \quad \Pi x:\mathbf{R}.\mathbf{R} \quad \quad (\Pi\text{-quantification}) \\
 \\
 \mathbf{E} & ::= \quad \alpha \quad \quad \quad (\text{variable d'appelé}) \\
 & | \quad \mathbf{R}\bullet\mathbf{E} \quad \quad \quad (\text{empilement}) \\
 & | \quad \tilde{\mu}x.\mathbf{C} \quad \quad \quad (\tilde{\mu}\text{-abstraction}) \\
 \\
 \mathbf{C} & ::= \quad \langle \mathbf{R} \parallel \mathbf{E} \rangle \quad \quad (\text{capsule})
 \end{array}$$

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

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

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

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

On remarque divers changements par rapport au $\bar{\lambda}\mu\tilde{\mu}$ -calcul usuel. D'une part, on est passé d'un calcul à la Curry à un calcul à la Church : les λ -abstractions sont maintenant annotées. En outre, on remarque l'adjonction au calcul de deux constructions de type appelant : les sortes et la Π -quantification.

Le choix a été fait, ici, de traiter comme des appelants les constructions de types. Il aurait également été possible d'envisager de les traiter comme appelés. Le choix a été fait ici de les considérer comme des appelants car, dans les systèmes de types purs, il n'y a pas de distinction entre les variables sur lesquelles

on peut effectuer une λ -abstraction et celles sur lesquelles on peut effectuer une Π -quantification, ce que l'on a souhaité préserver ici. Ce choix sera évoqué au paragraphe suivant, avec le typage.

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

Définition 3.6 : *variables libres/liées dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*
On définit les variables d'appelant (respectivement d'appelé) libres ou liées d'une expression comme appartenant à une partie de \mathcal{X} (respectivement \mathcal{V}) définie par induction sur l'expression. Les ensembles des variables d'appelant libres, des variables d'appelant liées, des variables d'appelé libres, des variables d'appelé liées de M sont respectivement dénotées par $fv_{\mathcal{X}}(M)$, $bv_{\mathcal{X}}(M)$, $fv_{\mathcal{V}}(M)$, $bv_{\mathcal{V}}(M)$.

Les variables d'appelant libres sont définies comme suit :

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

et les variables d'appelé libres comme suit :

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

tandis que les variables d'appelant liées sont définies comme suit :

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

- $bv_X(\mathbf{R}\bullet\mathbf{E}) = bv_X(\mathbf{R}) \cup bv_X(\mathbf{E})$;
- $bv_X(\tilde{\mu}x.C) = bv_X(C) \cup \{x\}$;
- $bv_X(\langle \mathbf{R} \parallel \mathbf{E} \rangle) = bv_X(\mathbf{R}) \cup bv_X(\mathbf{E})$.

et les variables d'appelé liées comme suit :

- $bv_Y(x) = \emptyset$;
- $fv_Y(\lambda x:A.R) = bv_Y(A) \cup bv_Y(R)$;
- $bv_Y(\mu\alpha.C) = bv_Y(C) \cup \{\alpha\}$;
- $fv_Y(\sigma) = \emptyset$;
- $fv_Y(\Pi x:A.B) = bv_Y(A) \cup bv_Y(B)$;
- $bv_Y(\alpha) = \emptyset$;
- $bv_Y(\mathbf{R}\bullet\mathbf{E}) = bv_Y(\mathbf{R}) \cup bv_Y(\mathbf{E})$;
- $bv_Y(\tilde{\mu}x.C) = bv_Y(C)$;
- $bv_Y(\langle \mathbf{R} \parallel \mathbf{E} \rangle) = bv_Y(\mathbf{R}) \cup bv_Y(\mathbf{E})$.

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

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

Définition 3.7 : réduction (à la racine) dans les s.t.p. pour $\bar{\lambda}\mu\tilde{\mu}$

La réduction, notée \rightarrow , est définie comme la clôture contextuelle de l'union des trois axiomes de réduction suivants :

- (λ) $\langle \lambda x:A.M \parallel \mathbf{N}\bullet\mathbf{E} \rangle \rightarrow \langle M[N/x] \parallel \mathbf{E} \rangle$
- (μ) $\langle \mu\alpha.C \parallel \mathbf{E} \rangle \rightarrow C[E/\alpha]$
- ($\tilde{\mu}$) $\langle \mathbf{R} \parallel \tilde{\mu}x.C \rangle \rightarrow C[\mathbf{R}/x]$

Une réduction est dite à la racine si elle est un axiome, *i.e.* la relation de réduction à la racine $\overrightarrow{\mathbf{R}}$ est définie comme l'union des trois axiomes (λ), (μ), ($\tilde{\mu}$).

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

Toutefois, il est possible d'invertir deux pas de réduction quand le redex impliqué dans le premier est inclus dans celui impliqué dans le second.

Lemme 3.1 : *Interversion de rédex*

Soient C et D deux capsules et $\mathcal{G}[\]$ un contexte tels que $C = \mathcal{G}[D]$. On suppose que $C \longrightarrow \mathcal{G}[D'] \xrightarrow{\mathcal{R}} C'$ avec $D \xrightarrow{\mathcal{R}} D'$. Alors $C \xrightarrow{\mathcal{R}}^* C'$.

Démonstration.

Il y a différents cas possibles :

a. $C = \langle \lambda x : \mathcal{H}[D].M \parallel N \bullet E \rangle$.

Alors $C' = \langle M[N/x] \parallel E \rangle$, et l'on a bien $C \xrightarrow{\mathcal{R}} \langle M[N/x] \parallel E \rangle = C'$.

b. $C = \langle \lambda x : A.\mathcal{H}[D] \parallel N \bullet E \rangle$.

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

c. $C = \langle \lambda x : A.M \parallel \mathcal{H}[D] \bullet E \rangle$.

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

d. $C = \langle \lambda x : A.M \parallel N \bullet \mathcal{H}[D] \rangle$.

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

e. $C = \langle \mu \alpha : \mathcal{H}[D] \parallel E \rangle$.

Alors $C' = \mathcal{H}[D'][E/\alpha]$, et l'on a bien $C \xrightarrow{\mathcal{R}} \mathcal{H}[D][E/\alpha] \longrightarrow C'$.

f. $C = \langle \mu \alpha : Z \parallel \mathcal{H}[D] \rangle$.

Alors $C' = Z[\mathcal{H}[D']/\alpha]$, et l'on a bien $C \xrightarrow{\mathcal{R}} Z[\mathcal{H}[D]/\alpha] \xrightarrow{*} C'$.

g. $C = \langle \mathcal{H}[D] \parallel \tilde{\mu}x.Z \rangle$.

Alors $C' = Z[\mathcal{H}[D']/x]$, et l'on a bien $C \xrightarrow{\mathcal{R}} Z[\mathcal{H}[D]/x] \xrightarrow{*} C'$.

h. $C = \langle R \parallel \tilde{\mu}x.\mathcal{H}[D] \rangle$.

Alors $C' = \mathcal{H}[D'][R/x]$, et l'on a bien $C \xrightarrow{\mathcal{R}} \mathcal{H}[D][R/x] \longrightarrow C'$.

Dans tous les cas, la réduction à la racine peut être faite en premier lieu. Q.E.D.

On en déduit la possibilité de réorganiser une réduction infinie en allant des capsules les plus extérieures aux les plus intérieures sans jamais remonter :

Corollaire 3.2 : *Réduction par l'extérieur*

Soit $(C_n)_{n \in \mathbb{N}}$ une chaîne de réduction infinie.

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

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

(iii) Si $(C_n)_{n \in \mathbb{N}}$ comprend un nombre fini non nul de pas de réduction à la racine et C_0 n'admet pas de réduction à la racine infinie, alors il existe $C' = \mathcal{G}[D_0]$ tel que $C_0 \xrightarrow{\dagger} C'$; et D_0 admet une réduction infinie comprenant un nombre non nul de pas de réduction à la racine.

Les contextes $\mathcal{G}[\]$ dans les assertions précédentes sont distincts de $[\]$.

Démonstration.

- (i) Un tel terme admet nécessairement un sous-terme propre ayant une réduction infinie. Les termes étant finis, on aboutit forcément à un sous-terme vérifiant la condition supplémentaire.
- (ii) On a $C_0 \xrightarrow{k} C_k \xrightarrow{\mathcal{R}} C_{k+1}$, où le pas ainsi isolé est le premier pas de réduction à la racine. On a alors, en appliquant le lemme précédent k fois, $C_0 \xrightarrow{\mathcal{R}} C'_1 \xrightarrow{*} C_{k+1} \longrightarrow \dots$. On poursuit ainsi la construction en appliquant le même processus à C'_1 pour obtenir une réduction à la racine de longueur non bornée, et l'on peut conclure.
- (iii) Soit j tel qu'à partir de C_j aucun pas de réduction ne soit à la racine. De même que précédemment, l'on obtient $C_0 \xrightarrow{\mathcal{R}^+} C'_k \xrightarrow{*} C_j \longrightarrow \dots$; une alternative se présente alors :
- si $C'_k \xrightarrow{*} C_j$ ne contient pas de pas de réduction à la racine, on peut alors conclure en appliquant le cas (i) à C'_k ;
 - sinon, on répète la construction précédente, ce qui augmente la taille de la réduction à la racine issue de C_0 .
- Comme C_0 n'admet pas de réduction à la racine infinie, nécessairement la première branche de l'alternative finit par être réalisée. Q.E.D.

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

Définition 3.8 : réduction imbriquée dans les systèmes de types purs pour $\bar{\lambda}\mu\bar{\mu}$
 Soit $C_0 \longrightarrow \dots \longrightarrow C_n$ une chaîne de réduction ; pour tout i , il existe un contexte $\mathcal{G}_i[\]$ tel que $C_i = \mathcal{G}_i[D_i]$ et $C_{i+1} = \mathcal{G}_i[D'_i]$ avec $D_i \xrightarrow{\mathcal{R}} D'_i$.
 La chaîne de réduction est dite *imbriquée* si elle vérifie la propriété suivante : pour tout i , il existe un contexte $\mathcal{H}_i[\]$ tel que $\mathcal{G}_{i+1}[\] = \mathcal{G}_i[\mathcal{H}_i[\]]$.

On voit donc que la méthode de réduction par l'extérieur présentée dans le corollaire *supra* transforme une réduction infinie quelconque en réduction infinie imbriquée. Cette dernière forme de réduction sera intéressante par la suite pour ce qui concerne la normalisation forte.

2.2. Typage

Les jugements de type des systèmes de types purs pour le $\bar{\lambda}\mu\bar{\mu}$ -calcul sont analogues à ceux des types simples : il s'agit de séquents à formule active, comme vu précédemment, qui peuvent donc être de trois formes, selon que leur formule

active est en partie gauche, en partie droite, ou absente. Ces trois formes correspondent aux trois espèces d'expressions intervenant dans le calcul. Il s'agit, tout comme dans le cas des types simples, respectivement du typage d'un appelé, d'un appelant ou d'une capsule.

Définition 3.9 : *assertion, contexte, jugement de typage des s.t.p. pour $\bar{\lambda}\mu\tilde{\mu}$*
Soit \mathfrak{T} un système de types pur. On appelle *assertion de typage* un couple de \mathfrak{T} -expressions noté $M : R$, où M peut être un appelant ou un appelé, et R est un appelant. Dans un tel couple, le premier élément est appelé *sujet* de l'assertion et le second, *prédicat* de l'assertion. Une assertion peut être ou non *active*.

On appelle *contexte gauche de typage* une suite finie – et non pas un ensemble – d'assertions de typage non actives dont les sujets sont des variables d'appelant distinctes. On appelle *contexte droit de typage* une suite finie – et non pas un ensemble – d'assertions de typage non actives dont les sujets sont des variables d'appelés distinctes. Pour ces deux espèces de contexte, le contexte vide est noté $[]$, et la concaténation de contextes de même côté et de supports disjoints est notée par une virgule.

On appelle *jugement de typage* une expression de l'une des trois formes suivante, Γ et Δ étant respectivement des contextes gauche et droit de typage :

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

Un jugement est dit *valide* s'il dérive de l'application des figures¹⁷ d'inférence présentées dans la table 7. L'indice \mathfrak{T} peut être omis s'il n'y a pas d'ambiguïté quant au système considéré.

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

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

¹⁷ On conserve ici cette dénomination inspirée de G. GENTZEN.

$$\begin{array}{c}
\frac{(\sigma, \tau) \in \mathcal{A}}{[] \vdash \sigma : \tau ; \emptyset} \text{ (axiome)} \\
\\
\frac{\Gamma \vdash A : \rho ; \Delta \quad \Gamma, x : A \vdash B : \sigma ; \Delta \quad (\rho, \sigma, \tau) \in \mathcal{A}}{\Gamma \vdash (\Pi x:A.B) : \tau ; \Delta} \text{ (règle)} \\
\\
\frac{\Gamma \vdash A : \sigma ; \Delta \quad A \notin \mathcal{S}}{\Gamma ; \alpha : A \vdash \Delta, \alpha : A} \text{ (hyp-G)} \qquad \frac{\Gamma \vdash A : \sigma ; \Delta}{\Gamma, x : A \vdash x : A ; \Delta} \text{ (hyp-D)} \\
\\
\frac{\Gamma ; E : B \vdash \Delta \quad \Gamma \vdash A : \sigma ; \Delta}{\Gamma, x : A ; E : B \vdash \Delta} \text{ (aff-g-G)} \qquad \frac{\Gamma \vdash R : B ; \Delta \quad \Gamma \vdash A : \sigma ; \Delta}{\Gamma, x : A \vdash R : B ; \Delta} \text{ (aff-g-D)} \\
\\
\frac{\Gamma ; E : B \vdash \Delta \quad \Gamma \vdash A : \sigma ; \Delta \quad A \notin \mathcal{S}}{\Gamma ; E : B \vdash \Delta, \alpha : A} \text{ (aff-d-G)} \qquad \frac{\Gamma \vdash R : B ; \Delta \quad \Gamma \vdash A : \sigma ; \Delta \quad A \notin \mathcal{S}}{\Gamma \vdash R : B ; \Delta, \alpha : A} \text{ (aff-d-D)} \\
\\
\frac{\Gamma \vdash (\Pi x:A.B) : \sigma ; \Delta \quad \Gamma ; E : B[R/x] \vdash \Delta \quad \Gamma \vdash R : A ; \Delta}{\Gamma ; R \bullet E : \Pi x:A.B \vdash \Delta} \text{ (\Pi-G)} \qquad \frac{\Gamma \vdash (\Pi x:A.B) : \sigma ; \Delta \quad \Gamma, x : A \vdash R : B ; \Delta}{\Gamma \vdash (\lambda x:A.R) : (\Pi x:A.B) ; \Delta} \text{ (\Pi-D)} \\
\\
\frac{\Gamma ; E : B \vdash \Delta \quad \Gamma \vdash A : \sigma ; \Delta \quad A \equiv B}{\Gamma ; E : A \vdash \Delta} \text{ (\equiv-G)} \qquad \frac{\Gamma \vdash R : B ; \Delta \quad \Gamma \vdash A : \sigma ; \Delta \quad A \equiv B}{\Gamma \vdash R : A ; \Delta} \text{ (\equiv-D)} \\
\\
\frac{C : (\Gamma, x : A \vdash \Delta) \quad \Gamma \vdash A : \sigma ; \Delta}{\Gamma ; \tilde{\mu}x.C : A \vdash \Delta} \text{ (\tilde{\mu})} \qquad \frac{C : (\Gamma \vdash \Delta, \alpha : A) \quad \Gamma \vdash A : \sigma ; \Delta}{\Gamma \vdash \mu\alpha.C : A ; \Delta} \text{ (\mu)} \\
\\
\frac{\Gamma \vdash R : A ; \Delta \quad \Gamma ; E : A \vdash \Delta}{\langle R \parallel E \rangle : (\Gamma \vdash \Delta)} \text{ (coupure)}
\end{array}$$

TAB. 7 – Figures d'inférence de type pour les systèmes de types purs pour $\tilde{\lambda}\tilde{\mu}\tilde{\mu}$

Dans les figures (μ) et ($\tilde{\mu}$), on peut observer particulièrement la présence d'une prémisse $\Gamma \vdash A : \sigma ; \Delta$. Cette prémisse joue un rôle en quelque sorte administratif : en effet, elle permet de s'assurer que les dépendances de variables au sein du couple de contextes (Γ, Δ) vont être correctes, bien que l'on ait ôté la dernière variable de l'un des contextes. Ce genre de vérification est assez naturel dans le cadre des systèmes de types purs, où vérifier que le type introduit est bien correct est une précaution fréquemment nécessaire. Ici, la difficulté vient de la plus grande complexité de la vérification de correction du fait de la présence de deux contextes, ce qui explique que bien que le type A ait déjà été introduit, il faut cependant vérifier que les dépendances soient correctes. Cette prémisse « de précaution » pourrait en fait être rendue superflue par un lemme de renforcement, indiquant que l'on peut supprimer du contexte les variables n'intervenant pas ailleurs dans le jugement de typage. Il suffirait alors d'avoir une condition interdisant que la variable supprimée apparaisse dans le reste du contexte de typage, où elle ne pourra être liée. Toutefois, ce genre de résultat est assez difficile à obtenir, comme l'a indiqué H.P. BARENDREGT, en particulier du fait de certaines figures présentes dans le système, en particulier celle de coupure, et surtout celles de conversion.

2.3. Correction des types

Comme il est habituel avec les systèmes de types purs, les notions de correction des types et de bonne formation des contextes revêtent une importance particulière. On introduit donc la définition suivante :

Définition 3.10 : *couple de contextes bien formé*

Un couple de contextes (Γ, Δ) où les contextes sont respectivement de la forme $\Gamma = [x_1 : T_1, \dots, x_n : T_n]$ et $\Delta = [\alpha_1 : A_1, \dots, \alpha_m : A_m]$ est dit *bien formé* si l'une des conditions suivantes est remplie :

- (i) Γ et Δ sont vides ($n = m = 0$) ;
- (ii) $[x_1 : T_1, \dots, x_{n-1} : T_{n-1}], \Delta$ est bien formé, et il existe une sorte σ telle que $[x_1 : T_1, \dots, x_{n-1} : T_{n-1}] \vdash T_n : \sigma ; \Delta$;
- (iii) $(\Gamma, [\alpha_1 : A_1, \dots, \alpha_{m-1} : A_{m-1}])$ est bien formé, $A_m \notin \mathcal{S}$ et il existe une sorte σ telle que $\Gamma \vdash A_m : \sigma ; [\alpha_1 : A_1, \dots, \alpha_{m-1} : A_{m-1}]$.

Naturellement, on peut montrer que les couples de contextes intervenant dans un jugement valablement dérivé sont bien formés :

Lemme 3.3 : *Bonne formation de couple*

Soit $\Gamma ; E : A \vdash \Delta$, $\Gamma \vdash R : A ; \Delta$ ou $C : (\Gamma \vdash \Delta)$ un jugement valide.

Alors (Γ, Δ) est un couple bien formé.

Démonstration.

On procède par induction en discriminant selon la dernière figure employée :

- a. (axiome) : ce cas est trivial.
- b. (règle), (Π -G), (Π -D), (\equiv -G), (\equiv -D), (coupure), (μ), ($\tilde{\mu}$) : ces cas sont évidents par hypothèse d'induction, car toutes ces figures ont une prémisse ayant le même couple de contextes que la conclusion.
- c. (hyp-D) :
$$\frac{\Gamma \vdash A : \sigma ; \Delta}{\Gamma, x : A \vdash x : A ; \Delta}$$

Par hypothèse d'induction, (Γ, Δ) est bien formé, et pour montrer que le couple $(\Gamma, x : A), \Delta)$ est bien formé, il suffit de trouver une sorte σ telle que $\Gamma \vdash A : \sigma ; \Delta$, et cela est assuré par la prémisse.
- d. (hyp-G) :
$$\frac{\Gamma \vdash A : \sigma ; \Delta \quad A \notin \mathcal{S}}{\Gamma ; \alpha : A \vdash \Delta, \alpha : A}$$

Par hypothèse d'induction, (Γ, Δ) est bien formé, et pour montrer que le couple $(\Gamma, (\Delta, \alpha : A))$ est bien formé, il suffit de vérifier que $A \notin \mathcal{S}$ et de trouver σ telle que $\Gamma \vdash A : \sigma ; \Delta$ et cela est assuré par les prémisses.
- e. (aff-d-D), (aff-d-G) : ces deux cas sont analogues à (hyp-D).
- f. (aff-g-D), (aff-g-G) : ces deux cas sont analogues à (hyp-G). Q.E.D.

À l'aide des quatre figures d'affaiblissement, l'on déduit le corollaire suivant :

Corollaire 3.4 : Correction des prédicats de contexte

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

Avec tout ceci, on peut également démontrer le théorème de correction des types, qui s'énonce de manière très analogue à celui des systèmes de types purs présentés au chapitre 1 :

Théorème 3.5 : Correction des types dans les s.t.p. pour $\bar{\lambda}\bar{\mu}\bar{\tilde{\mu}}$

Soit $\Gamma \vdash R : A ; \Delta$ ou $\Gamma ; E : A \vdash \Delta$ un jugement de type valide. Alors soit $A \in \mathcal{S}$, soit il existe une sorte $\sigma \in \mathcal{S}$ telle que $\Gamma \vdash A : \sigma ; \Delta$.

Démonstration.

Par induction sur la dérivation du jugement. Tous les cas sont évidents car, dans chaque cas, soit A est une sorte, soit la propriété $\Gamma \vdash A : \sigma ; \Delta$ est garantie par une prémisse. Q.E.D.

Il est intéressant de remarquer que, comme on l'avait déjà noté au chapitre précédent, on voit surgir à nouveaux deux notions de correction des types : l'une pour les prédicats de contexte, et l'une pour les prédicats d'assertion. Ces notions sont les mêmes que celles des systèmes de types purs usuels (cf. chapitre 1).

3. Propriétés remarquables

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

3.1. Lemmes d'inférence de types

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

Lemme 3.6 : *Initialisation pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*

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

- (i) si $(\sigma, \tau) \in \mathcal{A}$, alors $\Gamma \vdash \sigma : \tau ; \Delta$;
- (ii) si $(x : T) \in \Gamma$, alors $\Gamma \vdash x : T ; \Delta$;
- (iii) si $(\alpha : A) \in \Delta$, alors $\Gamma ; \alpha : A \vdash \Delta$.

Démonstration.

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

- $n = 0$. Dans ce cas, le point (i) est une application d'une figure d'(axiome) et les points (ii) et (iii) sont triviaux.
- $n > 0$. On a alors :
 - (i) On a, par hypothèse d'induction, $\Gamma' \vdash \sigma : \tau ; \Delta'$ avec, selon les cas, $\Gamma = \Gamma'$ et $\Delta = \Delta', \alpha : A$ ou bien $\Gamma = \Gamma', x : A$ et $\Delta = \Delta'$. En outre, par bonne formation, on a $\Gamma' \vdash A : \sigma ; \Delta'$ ce qui permet d'appliquer la figure d'affaiblissement adéquate pour obtenir $\Gamma \vdash \sigma : \tau ; \Delta$.
 - (ii) Soit on se trouve dans un cas analogue à celui de la propriété (i), soit on a $\Gamma = \Gamma', x : T$. Dans ce dernier cas, par bonne formation, on a $\Gamma' \vdash T : \sigma ; \Delta$, ce qui permet d'obtenir, par une figure d'(hyp-R), que $\Gamma, x : T \vdash x : T ; \Delta$.
 - (iii) Cette démonstration est symétrique de la précédente. Q.E.D.

Lemme 3.7 : *Engendrement pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*

Soit un jugement de typage valide.

- (i) Si le jugement est de la forme $\Gamma \vdash R : T ; \Delta$, alors R doit avoir l'une des formes suivantes :
 - a. $R = \sigma \in \mathcal{S}$. Dans ce cas, $T = \tau \in \mathcal{S}$ et $(\sigma, \tau) \in \mathcal{A}$.
 - b. $R = x \in \mathcal{X}$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et $(x : U) \in \Gamma$ telles que $\Gamma \vdash U : \tau ; \Delta$ et $T \equiv U$.

- c. $R = \Pi x:A.B$. Dans ce cas, $T = \tau \in \mathcal{S}$ et il existe des sortes ρ et σ telles que $(\rho, \sigma, \tau) \in \mathcal{R}$, $\Gamma \vdash A : \rho ; \Delta$ et $\Gamma, x : A \vdash B : \sigma ; \Delta$.
 - d. $R = \lambda x:A.N$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et un terme B tels que $\Gamma \vdash (\Pi x:A.B) : \tau ; \Delta$, $\Gamma, x : A \vdash N : B ; \Delta$ et $T \equiv \Pi x:A.B$
 - e. $R = \mu \alpha.C$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et un terme U tels que $C : (\Gamma \vdash \Delta, \alpha : U)$, $\Gamma \vdash U : \tau ; \Delta$ et $T \equiv U$.
- (ii) Si le jugement est de la forme $\Gamma ; E : T \vdash \Delta$, alors E doit avoir l'une des formes suivantes :
- a. $E = \alpha \in \mathcal{V}$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et $(\alpha : U) \in \Delta$ tels que $\Gamma ; U : \tau \vdash \Delta$ et $T \equiv U$.
 - b. $E = R \bullet F$. Dans ce cas, il existe $\tau \in \mathcal{S}$, des termes A et B et une variable $x \in \mathcal{X}$ telle que $\Gamma \vdash (\Pi x:A.B) : \tau ; \Delta$, $\Gamma \vdash R : A ; \Delta$, $\Gamma ; F : B[R/x] \vdash \Delta$ et $T \equiv \Pi x:A.B$
 - c. $E = \tilde{\mu}x.C$. Dans ce cas, il existe $\tau \in \mathcal{S}$ et un terme U tels que $C : (\Gamma, x : U \vdash \Delta)$, $\Gamma \vdash U : \tau ; \Delta$ et $T \equiv U$.
- (iii) Si le jugement est de la forme $C : (\Gamma \vdash \Delta)$ avec $C = \langle R \parallel E \rangle$, alors il existe $\tau \in \mathcal{S}$ et un terme A tel que $\Gamma \vdash R : A ; \Delta$ et $\Gamma ; E : A \vdash \Delta$.

Démonstration.

Bien que son énoncé semble plus complexe en raison du plus grand nombre de cas possibles, ce lemme se montre exactement comme son analogue pour les systèmes de types purs implicites présenté au chapitre 1. Q.E.D.

Lemme 3.8 : *Dépendances dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*

Les énoncés suivants sont vrais :

- (i) Soit $\Gamma \vdash X : T ; \Delta$ ou $\Gamma ; X : T \vdash \Delta$ ou $X : (\Gamma \vdash \Delta)$ un jugement. Alors :
 - a. toute variable libre de X est dans le domaine de Γ ou Δ selon qu'elle est d'appelant ou d'appelé ;
 - b. dans les deux premiers cas, toute variable libre de T est dans le domaine de Γ ou Δ selon qu'elle est d'appelant ou d'appelé.
- (ii) Soit (Γ, Δ) un couple bien formé. Alors :
 - a. si $\Gamma = \Gamma', x : T, \Gamma''$, alors toute variable libre de T est dans le domaine de Γ' ou de Δ selon sa nature ;
 - b. si $\Delta = \Delta', \alpha : A$, alors toute variable libre de A est dans le domaine de Γ ou de Δ' selon sa nature.

Démonstration.

- (i) *Le point a. se démontre facilement par induction sur la dérivation, et le point b. découle du point a. et du théorème de correction des types (3.5).*

(ii) On traite le point $a.$, le point $b.$ étant symétrique. On raisonne par induction sur la taille n du couple, i.e. le nombre total d'assertions des deux contextes. Si $n = 0$, la propriété est triviale. On suppose maintenant $n > 0$. Le couple étant bien formé, plusieurs cas se présentent :

- Γ est vide, et la propriété est triviale ;
 - $\Gamma = \Gamma', x : T$ avec $\Gamma' \vdash T : \sigma ; \Delta$ et (Γ', Δ) bien formé, et la propriété est évidente grâce au point $i.$;
 - $\Gamma = \Gamma_1, u : V$ avec $u \neq x$ et (Γ_1, Δ) bien formé, et l'on conclut alors directement par hypothèse d'induction ;
 - $\Delta = \Delta_1, \alpha : A$ avec (Γ, Δ_1) bien formé, et l'on conclut alors directement par hypothèse d'induction, car $\text{dom}(\Delta_1) \subseteq \text{dom}(\Delta)$.
- Dans tous les cas, l'on a que les variables libres de T sont dans le domaine de Γ' ou de Δ .

Par ailleurs, on peut observer la propriété suivante, donnant des indications sur la forme des types :

Lemme 3.9 : *Typage par une sorte pour $\bar{\lambda}\mu\tilde{\mu}$*

Soit $\Gamma \vdash A : T ; \Delta$ tel qu'il existe une sorte $\sigma \in \mathcal{S}$ telle que $T = \sigma$. Alors A est soit une sorte, soit une variable d'appelé, soit une Π -quantification.

Démonstration.

On procède par induction sur la dérivation de $\Gamma \vdash A : T ; \Delta$. La dernière figure appliquée ne peut être que (axiome), (règle), (hyp-D), (aff-g-D), (aff-d-D), (\equiv -D), ou (μ).

Les trois premiers cas sont triviaux.

Les trois suivants sont évidents par hypothèse d'induction (pour le cas (\equiv -D), cela découle du fait que $T \equiv \sigma$ implique $T = \sigma$). Dans le dernier cas, on a donc $\Gamma \vdash \mu\alpha.C ; \Delta$, d'où, par le lemme d'engendrement, il existe U tel que $A \equiv U$ et $C : (\Gamma \vdash \delta, \alpha : U)$. Mais alors $U \equiv \sigma$, et donc $U = \sigma$, ce qui est impossible car les variables d'appelé ne peuvent être déclarées avec une sorte comme type dans un jugement valide. Q.E.D.

3.2. Réduction du sujet

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

Lemme 3.10 : *Substitution dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$*

Soit un jugement de type valide.

- (i) Si ce jugement est de la forme $\Gamma \vdash R : T ; \Delta$, alors :
 - a. si $\Gamma = \Gamma', x : U, \Gamma''$ et $\Gamma' \vdash S : U ; \Delta_1$ avec Δ_1 préfixe de Δ , alors $\Gamma', \Gamma''[S/x] \vdash R[S/x] : T[S/x] ; \Delta[S/x]$;
 - b. si $\Delta = \Delta', \alpha : A, \Delta''$ et $\Gamma_1 ; F : A \vdash \Delta'$ avec Γ_1 préfixe de Γ , alors $\Gamma[F/\alpha] \vdash R[F/\alpha] : T[F/\alpha] ; \Delta', \Delta''[F/\alpha]$.
- (ii) Si ce jugement est de la forme $\Gamma ; E : T \vdash \Delta$, alors :
 - a. si $\Gamma = \Gamma', x : U, \Gamma''$ et $\Gamma' \vdash S : U ; \Delta_1$ avec Δ_1 préfixe de Δ , alors $\Gamma', \Gamma''[S/x] ; E[S/x] : T[S/x] \vdash \Delta[S/x]$;
 - b. si $\Delta = \Delta', \alpha : A, \Delta''$ et $\Gamma_1 ; F : A \vdash \Delta'$ avec Γ_1 préfixe de Γ , alors $\Gamma[F/\alpha] ; E[F/\alpha] : T[F/\alpha] \vdash \Delta', \Delta''[F/\alpha]$.
- (iii) Si le jugement est de la forme $C : (\Gamma \vdash \Delta)$ avec $C = \langle R \parallel E \rangle$, alors :
 - a. si $\Gamma = \Gamma', x : U, \Gamma''$ et $\Gamma' \vdash S : U ; \Delta_1$ avec Δ_1 préfixe de Δ , alors $C[S/x] : (\Gamma', \Gamma''[S/x] \vdash \Delta[S/x])$;
 - b. si $\Delta = \Delta', \alpha : A, \Delta''$ et $\Gamma_1 ; F : A \vdash \Delta'$ avec Γ_1 préfixe de Γ , alors $C[F/\alpha] : (\Gamma[F/\alpha] \vdash \Delta', \Delta''[F/\alpha])$.

Démonstration.

Par induction directe sur la dérivation du jugement. On traite ici seulement quelques cas typiques :

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

- a. Par hypothèse d'induction, l'on a $\Gamma', \Gamma''[S/x] \vdash A[S/x] : \rho ; \Delta[S/x]$ et $\Gamma', \Gamma''[S/x], y : A[S/x] \vdash B[S/x] : \sigma ; \Delta[S/x]$, et l'on peut alors conclure par une figure de (règle).
- b. Par hypothèse d'induction, l'on a $\Gamma[F/\alpha] \vdash A[F/\alpha] : \rho ; \Delta', \Delta''[F/\alpha]$ et $\Gamma[F/\alpha], y : A[F/\alpha] \vdash B[F/\alpha] : \sigma ; \Delta', \Delta''[F/\alpha]$, et l'on peut alors conclure par une figure de (règle).

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

- a. Par hypothèse d'induction, on a $\Gamma', \Gamma''[S/x] \vdash A[S/x] : \sigma ; \Delta[S/x]$, et, par une figure de (hyp-G), $\Gamma', \Gamma''[S/x] ; \beta : A[S/x] \vdash \Delta[S/x], \beta : A[S/x]$.
- b. Deux cas se présentent. Si $\beta \neq \alpha$, ce cas est direct par induction. Sinon, le résultat se déduit de l'hypothèse $\Delta = \Delta', \alpha : A, \Delta''$ par affaiblissement.

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

a. Par hypothèse d'induction, $\Gamma', \Gamma''[S/x] \vdash \mathbf{R}[S/x] : \mathbf{A}[S/x] ; \Delta[S/x]$ et $\Gamma', \Gamma''[S/x] ; \mathbf{E}[S/x] : \mathbf{A}[S/x] \vdash \Delta[S/x]$, et l'on conclut par une coupure.

b. Par hypothèse d'induction, $\Gamma[F/\alpha] \vdash \mathbf{R}[F/\alpha] : \mathbf{A}[F/\alpha] ; \Delta', \Delta''[F/\alpha]$ et $\Gamma[F/\alpha] ; \mathbf{E}[F/\alpha] : \mathbf{A}[F/\alpha] \vdash \Delta', \Delta''[F/\alpha]$, et l'on conclut par une coupure.

Les autres cas se traitent de manière analogue.

Q.E.D.

Théorème 3.11 : Réduction du sujet dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$
Si $C : (\Gamma \vdash \Delta)$ et $C \rightarrow C'$ alors $C' : (\Gamma \vdash \Delta)$.

Démonstration.

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

(λ) Dans ce cas, $\mathbf{R} = \lambda x:A.M$, $\mathbf{E} = \mathbf{N} \bullet \mathbf{F}$, et $C' = \langle \mathbf{M}[N/x] \parallel \mathbf{F} \rangle$. Par le lemme d'engendrement (3.7), il existe un type \mathbf{T} tel que $\Gamma \vdash \mathbf{R} : \mathbf{T} ; \Delta$ et $\Gamma ; \mathbf{E} : \mathbf{T} \vdash \Delta$. En utilisant deux fois le lemme d'engendrement (3.7) et la figure de conversion, l'on montre qu'il existe \mathbf{B} avec $\mathbf{T} \equiv \Pi x:A.B$ tel que $\Gamma, x : A \vdash \mathbf{M} : \mathbf{B} ; \Delta$ et $\Gamma ; \mathbf{F} : \mathbf{B}[N/x] \vdash \Delta$ et $\Gamma \vdash \mathbf{N} : \mathbf{A} ; \Delta$. En utilisant le lemme de substitution (3.10) avec le premier et le dernier de ces trois jugements, on obtient que $\Gamma \vdash \mathbf{M}[N/x] : \mathbf{B} ; \Delta[N/x]$. Comme le couple (Γ, Δ) est bien formé et que x n'apparaît pas dans Γ , l'on a $\Gamma \vdash \mathbf{M}[N/x] : \mathbf{B} ; \Delta$, et donc, par une coupure, $C' : (\Gamma \vdash \Delta)$.

(μ) Dans ce cas, $\mathbf{R} = \mu\alpha.C''$ et $C' = C''[E/\alpha]$. Par le lemme d'engendrement (3.7), il existe un type \mathbf{V} tel que $\Gamma \vdash \mathbf{R} : \mathbf{V} ; \Delta$ et $\Gamma ; \mathbf{E} : \mathbf{V} \vdash \Delta$. En utilisant de nouveau le lemme d'engendrement (3.7), on obtient alors que $C'' : (\Gamma \vdash \Delta, \alpha : \mathbf{V})$, d'où, par le lemme de substitution (3.10), $C' : (\Gamma \vdash \Delta)$.

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

Q.E.D.

3.3. Normalisation forte

On donne ici une démonstration de la normalisation forte pour le système correspondant aux types simples, et l'on donne également des outils semblant prometteurs pour une méthode par encodage dans les systèmes de types purs.

Normalisation forte pour les types simples

Dans ce paragraphe, on va montrer que le système de types pur pour $\bar{\lambda}\mu\tilde{\mu}$ correspondant aux types simples est fortement normalisant. On considère donc ce système, qui comporte deux sortes $*$ et \square , un seul axiome $*$: \square , et une seule règle $(*, *, *)$. Pour ce faire, on va le traduire dans le $\bar{\lambda}\mu\tilde{\mu}$ -calcul simplement typé (cf. e.g. [81]), qui est fortement normalisant.

On commence par montrer quelques lemmes spécifiques à ce cadre :

Lemme 3.12 : *Propriété du s.t.p. des types simples pour $\bar{\lambda}\mu\tilde{\mu}$*

Les assertions suivantes sont vraies :

- (i) Si $\Gamma, x : A, \Xi \vdash B : * ; \Delta$ et $\Gamma \vdash A : * ; \Delta$, alors $x \notin \text{fv}_{\mathcal{V}}(B)$.
- (ii) Si $\Gamma \vdash B : * ; \Delta$ et $A \equiv B$, alors B ne contient pas de capsule et $A = B$.
- (iii) Si $\Gamma \vdash \lambda x:A.R : B ; \Delta$, alors $\Gamma \vdash A : * ; \Delta$.

Démonstration.

- (i) *On procède par induction sur B . D'après le lemme 3.9, B est nécessairement une sorte, une variable, ou une Π -quantification. Dans le premier cas, la propriété est évidente. Dans le deuxième cas également, car B ne saurait être x , sans quoi, par le lemme d'engendrement, on aurait $A \equiv *$, donc $A = *$, ce qui est impossible.*

*Reste le troisième cas, i.e. $B = \Pi y:U.V$. On a alors, par le lemme d'engendrement, $\Gamma, x : A, \Xi \vdash U : \rho ; \Delta$ et $\Gamma, x : A, \Xi, y : U \vdash V : \sigma ; \Delta$ avec $(\rho, \sigma, *)$ une règle. Mais alors $\rho = \sigma = *$, et donc on peut appliquer l'hypothèse d'induction, et l'on obtient alors que $x \notin \text{fv}_{\mathcal{V}}(U)$ et $x \notin \text{fv}_{\mathcal{V}}(V)$, ce qui implique que $x \notin \text{fv}_{\mathcal{V}}(B)$.*

- (ii) *On procède par induction sur B . D'après le lemme 3.9, B est nécessairement une sorte, une variable, ou une Π -quantification. Les deux premiers cas sont évidents ; reste le troisième, i.e. $B = \Pi y:U.V$. Alors $A = \Pi y:U'.V'$ avec $U' \equiv U$ et $V' \equiv V$. De plus, comme au point précédent, par le lemme d'engendrement, $\Gamma \vdash U : * ; \Delta$ et $\Gamma, y : U \vdash V : * ; \Delta$. On peut alors appliquer l'hypothèse d'induction à ces deux dérivations avec respectivement U' et V' . On obtient donc que U et V ne contiennent pas de capsule, et que $U = U'$ et $V = V'$. Donc B ne contient pas de capsule, et $A = B$.*

- (iii) *Il suffit d'appliquer deux fois le lemme d'engendrement. Q.E.D.*

On peut noter comme corollaire du point (ii) que tout appelant typable par la sorte $*$ est exclusivement constitué de variables d'appelant et Π -quantifications (et est donc irréductible).

Puis on remarque que, par le théorème de correction des types, si $\Gamma \vdash R : A ; \Delta$ ou $\Gamma ; E : A \vdash \Delta$, alors il existe $\sigma \in \mathcal{S}$ telle que $\Gamma \vdash A : \sigma ; \Delta$ ou $A = \sigma$. Combiné avec le lemme 3.9, ce résultat devient que soit $\Gamma \vdash A : * ; \Delta$, soit $A = *$, soit $A = \square$. En outre, on a déjà vu, par le lemme précédent, que si $\Gamma \vdash R : \square ; \Delta$, alors $R = *$. On montre également qu’il est impossible que $\Gamma \vdash ; E : \square \vdash \Delta$: en effet, ce cas n’est possible que si $E = \tilde{\mu}x.C_1$, ce qui, par le lemme d’engendrement, implique que $C_1 : (\Gamma, x : \square \vdash \Delta)$, ce qui est impossible car les variables ne peuvent être déclarées avec le type \square dans une dérivation valide.

On définit alors deux traductions partielles $\|\cdot\|$ et $|\cdot|$ sur les termes typables :

- $\|x\| = x$;
- $\|\Pi x:A.B\| = \|A\| \rightarrow \|B\|$;
- $|x| = x$;
- $|\lambda x:A.R| = \lambda x.|R|$;
- $|\mu\alpha.C| = \mu\alpha.|C|$;
- $|\alpha| = \alpha$;
- $|R \bullet E| = |R| \bullet |E|$;
- $|\tilde{\mu}x.C| = \tilde{\mu}x.|C|$ si ce terme n’est pas de type $*$;
- $|\tilde{\mu}x.C| = |C|$ si ce terme est de type $*$;
- $|\langle R \parallel E \rangle| = \langle |R| \parallel |E| \rangle$ si le type commun à R et E n’est pas $*$;
- $|\langle R \parallel E \rangle| = |D|(|R|/x)$ si le type commun à R et E est $*$ et $E = \tilde{\mu}x.D$.

La traduction $\|\cdot\|$ traduit les termes sous forme de types du $\tilde{\lambda}\tilde{\mu}\tilde{\mu}$ -calcul simplement typé, tandis que la traduction $|\cdot|$ traduit les termes sous forme de termes du $\tilde{\lambda}\tilde{\mu}\tilde{\mu}$ -calcul simplement typé.

On remarque avec le lemme 3.12 que si $\Gamma \vdash R : * ; \Delta$, alors $\|R\|$ est définie. Cela permet d’étendre la traduction $\|\cdot\|$ aux couples de contextes bien formés. Pour les contextes droits, la traduction est directe : on a $\|\alpha_1 : A_1, \dots, \alpha_k : A_k\| = \alpha_1 : \|A_1\|, \dots, \alpha_k : \|A_k\|$ tandis que pour les contextes gauches, elle s’obtient comme suit : $\|[\]\| = \emptyset$ et $\|\Gamma, x : A\| = \|\Gamma\|, x : \|A\|$ si $A \neq *$ ou $\|\Gamma, x : A\| = \|\Gamma\|$ sinon.

On peut alors traduire les dérivations de typage comme suit :

- si la dérivation est de la forme $\Gamma \vdash * : \square ; \Delta$, la dérivation n’est pas traduite ;
- si la dérivation est de la forme $\Gamma \vdash R : * ; \Delta$, la dérivation se traduit par le fait que $\|R\|$ est définie ;
- si la dérivation est de la forme $\Gamma \vdash R : A ; \Delta$ où A n’est pas une sorte, la dérivation se traduit par $\|\Gamma\| \vdash \|R\| : \|A\| ; \|\Delta\|$;
- si la dérivation est de la forme $\Gamma ; E : A \vdash \Delta$ où A n’est pas une sorte, la dérivation se traduit par $\|\Gamma\| ; |E| : \|A\| \vdash \|\Delta\|$;

- si la dérivation est de la forme $\Gamma; E : * \vdash \Delta$, nécessairement E est de la forme $\tilde{\mu}x.C$, et la dérivation se traduit par $|C| : (|\Gamma| \vdash |\Delta|)$;
- si la dérivation est de la forme $\langle R \parallel E \rangle : (\Gamma \vdash \Delta)$ avec $*$ comme type commun à R et E , nécessairement E est de la forme $\tilde{\mu}x.D$, et la dérivation se traduit par $|E| : (|\Gamma| [|\!|R|\!/x] \vdash |\Delta| [|\!|R|\!/x])$;
- si la dérivation est de la forme $\langle R \parallel E \rangle : (\Gamma \vdash \Delta)$ avec $A \notin \mathcal{S}$ comme type commun à R et E , la dérivation se traduit par $|\langle R \parallel E \rangle| : (|\Gamma| \vdash |\Delta|)$.

Avec les lemmes précédents, on voit que les cas ci-dessus sont les seuls possibles. On vérifie ensuite sur chaque figure d'inférence que cette traduction fonctionne.

Il faut ensuite montrer que si un terme T typable admet une réduction infinie, alors le sujet de la traduction de la dérivation de typage de T dans le $\tilde{\lambda}\tilde{\mu}\tilde{\nu}$ -calcul simple admet une réduction infinie.

Pour ce faire, on va tout d'abord se réduire au cas où le terme ne contient pas de capsules d'une certaine forme.

Définition 3.11 : *Terme principal*

On dit qu'un terme typable est *principal* s'il ne contient aucune capsule de la forme $\langle R \parallel E \rangle$, où le type commun à R et E est $*$.

On remarque qu'un terme principal ne peut se réduire qu'en un terme principal. Ces termes sont ceux qui se traduisent bien en $\tilde{\lambda}\tilde{\mu}\tilde{\nu}$ -calcul simple. On remarque également qu'un terme non principal contient nécessairement une capsule de la forme $\langle R \parallel \tilde{\mu}x.C \rangle$, avec R constitué uniquement de variables d'appelant et de Π -quantifications.

L'on va montrer que l'on peut se restreindre aux termes principaux :

Lemme 3.13 : *Diminution de facteurs de non-principalité*

Soit T un terme non principal admettant une réduction infinie. Nécessairement, $T = \mathcal{K}[\langle R \parallel \tilde{\mu}x.C \rangle]$, avec R constitué uniquement de variables d'appelant et de Π -quantifications. Alors $\mathcal{K}[C[R/x]]$ est typable, et admet une réduction infinie.

Démonstration.

On peut se restreindre au cas où la réduction infinie de T est imbriquée. Deux cas se présentent :

- si, au cours de la réduction, la capsule $\langle R \parallel \tilde{\mu}x.C \rangle$ n'est pas réduite, la propriété est évidente ;
- sinon, cela veut dire que $T \xrightarrow{*} \mathcal{K}'[\langle R \parallel \tilde{\mu}x.C \rangle] \longrightarrow \mathcal{K}'[C[R/x]] \xrightarrow{+} \dots$ est une réduction imbriquée infinie, et donc $\mathcal{K}[C[R/x]] \xrightarrow{*} \mathcal{K}'[\langle R \parallel \tilde{\mu}x.C \rangle] \longrightarrow \mathcal{K}'[C[R/x]] \xrightarrow{+} \dots$ également.

Dans tous les cas, $\mathcal{K}[C[R/x]]$ est typable par réduction du sujet.

Q.E.D.

Corollaire 3.14 : *Restriction aux termes principaux*

Soit T un terme typable admettant une réduction infinie. Alors il existe T' principal admettant une réduction infinie.

Démonstration.

Le lemme précédent permet de diminuer de 1 le nombre de capsules de la forme $\langle R \parallel E \rangle$ où le type commun à R et E est $*$. On peut donc conclure par simple récurrence. Q.E.D.

Reste maintenant à montrer la traduction :

Lemme 3.15 : *Traduction de réduction de terme principal*

Soit T un terme principal (donc typable), et soit T' tel que $T \longrightarrow T'$.

Par la réduction du sujet, T' est également typable, donc les $|T|$ et $|T'|$ sont définies. On a alors $|T| \longrightarrow |T'|$.

Démonstration.

On raisonne par induction sur T en distinguant trois cas :

- $T = R$, et $\Gamma \vdash T : A ; \Delta$. Dans ce cas, T étant réductible, nécessairement $A \neq *$. On distingue alors plusieurs cas suivant la forme de R :
 - a. par le lemme d'engendrement R ne peut pas être une sorte ou une Π -quantification ;
 - b. R ne peut pas être une variable, car il est réductible ;
 - c. si $R = \lambda x : B.S$, par le lemme 3.12, nécessairement B est irréductible, donc $T' = \lambda x : B.S'$; or, par le lemme d'engendrement, il existe V tel que $\Gamma, x : A \vdash S : V ; \Delta$, donc, par hypothèse d'induction, $|S| \longrightarrow |S'|$, et $|T| = \lambda x. |S| \longrightarrow \lambda x. |S'| = |T'|$;
 - d. si $R = \mu \alpha.C$, nécessairement $T' = \mu \alpha.C'$, et on a $C : (\Gamma \vdash \Delta, \alpha : A)$, et par hypothèse d'induction, $|C| \longrightarrow |C'|$, et donc $|T| = \mu \alpha. |C| \longrightarrow \mu \alpha. |C'| = |T'|$.
- $T = E$, et $\Gamma ; T : A \vdash \Delta$. On distingue plusieurs cas suivant la forme de E :
 - a. E ne peut être une variable, car il réductible ;
 - b. si $E = R \bullet F$, par le lemme d'engendrement, A ne peut être $*$, et donc $|T| = |R| \bullet |F|$, et $T' = |R'| \bullet |F|$ ou $T' = |R| \bullet |F'|$. R et F sont principaux, donc, par hypothèse d'induction, selon les cas, $|R| \longrightarrow |R'|$ ou $|F| \longrightarrow |F'|$, et donc $|T| \longrightarrow |T'|$;
 - c. si $E = \tilde{\mu} x.C$, on raisonne comme au cas analogue avec μ .
- $T = \langle R \parallel E \rangle$. Cette capsule étant principale, on a $|T| = \langle |R| \parallel |E| \rangle$. Si $T' = \langle R' \parallel E \rangle$ ou $T' = \langle R \parallel E' \rangle$, on traite le cas facilement par induction. Sinon, on distingue selon l'axiome de réduction utilisé :

- (λ) Dans ce cas, $R = \lambda x:A.M$, $E = N \bullet F$ et $T' = \langle M[N/x] \parallel F \rangle$, et on a $|T| = \langle \lambda x.|M| \parallel |N| \bullet |F| \rangle \longrightarrow \langle |M| [|N|/x] \parallel |F| \rangle = |T'|$.
- (μ) Dans ce cas, $R = \mu \alpha.C_1$, et $T' = C_1[E/\alpha]$, et on a $|T| = \langle \mu \alpha.|C_1| \parallel |E| \rangle \longrightarrow |C_1| [|E|/\alpha] = |T'|$.
- ($\tilde{\mu}$) Ce cas est symétrique du précédent.

Q.E.D.

De tout ceci, on peut déduire le théorème suivant :

Théorème 3.16 : *Normalisation forte des types simples pour $\bar{\lambda}\mu\tilde{\mu}$*

Soit T un terme typable dans le système $(\{*, \square\}, \{* : \square\}, \{(*, *, *)\})$. Alors T n'admet pas de réduction infinie.

Démonstration.

Soit un tel T , on suppose qu'il admet une réduction infinie. Il existe alors T' typable principal admettant une réduction infinie. On a alors que $|T|$ est un terme typable dans le $\bar{\lambda}\mu\tilde{\mu}$ -calcul simple, et admet une réduction infinie, ce qui contredit la forte normalisation du $\bar{\lambda}\mu\tilde{\mu}$ -calcul simple. Q.E.D.

La démonstration présentée ici semble pouvoir s'étendre sans difficulté particulière au système $\lambda F\omega$, en se reposant sur les travaux de Lengrand dans [69, 68], dont on a parlé *supra*.

Outils d'encodage

A priori, il peut sembler illusoire de vouloir encoder un calcul non confluent muni de types classiques dans un calcul confluent muni de types intuitionnistes. Toutefois, la notion de réduction imbriquée, présentée *supra*, semble pouvoir présenter un intérêt dans cette optique en suggérant un moyen de contourner cette difficulté.

On suppose que l'on dispose d'une traduction $\llbracket \cdot \rrbracket$ allant du calcul sous-jacent au système de types pur pour $\bar{\lambda}\mu\tilde{\mu}$ vers le système de types pur (implicite ou explicite) associé, et que cette traduction est compositionnelle et satisfait la propriété suivante :

si $C \xrightarrow{R} C'$, alors il existe un contexte $\mathcal{K}[\]$ tel que $\llbracket C \rrbracket \xrightarrow{+} \mathcal{K}[\llbracket C' \rrbracket]$.

On remarque que cette propriété permet de contourner les difficultés liées à la non-confluence : en effet, si l'on a une divergence avec d'une part $C \longrightarrow C_1$ et d'autre part, $C \longrightarrow C_2$, on peut très bien avoir $\llbracket C \rrbracket \xrightarrow{+} \mathcal{K}_1[\llbracket C_1 \rrbracket]$ et $\llbracket C \rrbracket \xrightarrow{+} \mathcal{K}_2[\llbracket C_2 \rrbracket]$. L'idée d'une telle traduction serait de juxtaposer toutes les réductions possibles

(ici, les deux stratégies possibles : priorité à μ ou à $\bar{\mu}$), la difficulté résidant dans le fait que la traduction doit rester compositionnelle.

On a alors que si C admet une réduction infinie, $\llbracket C \rrbracket$ admet une réduction infinie : soit une suite infinie de capsules $(C_i)_{i \in \mathbb{N}}$ avec $C_0 = C$ telle que pour tout i , $C_i \longrightarrow C_{i+1}$. En appliquant à la chaîne de réduction infinie $(C_i)_{i \in \mathbb{N}}$ la transformation indiquée dans le corollaire 3.2, l'on obtient une réduction infinie imbriquée $\hat{C}_0 \xrightarrow{\mathcal{R}} \mathcal{G}_1[\hat{C}_1] \longrightarrow \mathcal{G}_1[\mathcal{G}_2[\hat{C}_2]] \longrightarrow \dots$ avec $\hat{C}_i \xrightarrow{\mathcal{R}} \mathcal{G}_{i+1}[\hat{C}_{i+1}]$. On a alors, par la propriété précédente, $\llbracket \hat{C}_i \rrbracket \xrightarrow{+} \mathcal{K}_i[\llbracket \hat{C}_{i+1} \rrbracket]$. On pose alors $T_i = \mathcal{K}_0[\mathcal{K}_1[\dots \mathcal{K}_{i-1}[\llbracket \hat{C}_i \rrbracket] \dots]]$, et l'on vérifie que $(T_i)_{i \in \mathbb{N}}$ est une chaîne infinie de réduction commençant au terme $T_0 = \llbracket C_0 \rrbracket$.

Si l'on dispose donc d'une telle traduction, et que l'on vérifie en outre qu'elle transforme un terme typable en terme typable, on peut en déduire que l'extension du système considéré à $\bar{\lambda}\mu\bar{\mu}$ n'altère pas sa normalisation forte. Les outils présentés ici semblent donc être une piste intéressante à suivre pour la production de démonstrations de normalisation forte.

CONCLUSION

Bilan

L'on a étudié ici différentes extensions des systèmes de types purs.

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

L'autre extension, la première présentée ici, et la plus importante, est une extension à un calcul à substitutions explicites, pour laquelle on a délibérément choisi d'adopter un calcul à noms (et non à indices de De Bruijn) à la syntaxe simple, à savoir λx (ou λx^-). L'intérêt de cette extension est double. D'une part, sur le plan du calcul, les calculs à substitutions explicites sont plus réalistes, car la substitution devant être implantée d'une façon ou d'une autre, il est plus raisonnable de la présenter comme une opération de première classe interne au calcul que de lui conserver un statut de méta-opération – on pourra se reporter au chapitre I § 1.3 pour davantage de considérations sur cet aspect. D'autre part, sur le plan de la logique, la substitution correspond à une coupure, qui manque aux systèmes de types purs traditionnels.

Au contraire des travaux effectués précédemment (on pourra en particulier se reporter aux travaux de R. Bloo [9, 12]), l'approche présentée ici a l'intérêt de conserver la propriété de réduction du sujet au moyen de l'introduction d'une nouvelle figure d'inférence d'expansion – cf. chapitre II § 2.1 – ce qui résout la question que pose R. Bloo dans [9] de savoir s'il est possible de conserver cette propriété autrement qu'au moyen de l'introduction de définitions ou de structures similaires dans les contextes de typages – cf. [12], et également [87]. En outre, comme on l'a mentionné, cette figure d'expansion semble présenter un grand intérêt dans le cadre de la synthèse de type – ce point est également évoqué au chapitre II § 2.1, et l'on pourra également regarder, à ce propos, les différents arbres de dérivation donnés dans ce paragraphe.

Par ailleurs, les travaux ici présentés isolent un problème particulier qu'est le problème des sortes terminales. En effet, habituellement, il est nécessaire d'avoir dans un système de types purs des sortes terminales, *i.e.* non typables, afin de maintenir la cohérence du système. Toutefois, comme on l'a déjà mentionné – cf. *e.g.* chapitre II § 2.2 & 2.3 – autant, dans le cas implicite, ces sortes ne posent pas problème, car si un terme a un type équivalent à une telle sorte, il lui est égal

– cf. proposition 2.14 –, autant, dans le cas explicite, le problème se pose avec une acuité particulière. On traite en fait dans les deux cas de deux façons différentes le même problème, à savoir qu’il est impossible d’introduire un type de toutes les sortes. Dans le cas implicite, on traite cela en constatant que ce n’est pas grave, car la proposition 2.14 assure que l’on aura pas besoin d’identifier les sortes en question comme des types corrects. Dans le cas explicite, on introduit une nouvelle sorte, qui sert à repérer les sortes, mais on en fait un prédicat particulier ne pouvant intervenir dans aucune règle du système et induisant une notion restreinte de correction ; un type typable par une sorte est correct pour une introduction dans un contexte ou pour une conversion, tandis qu’un type typable par ce nouveau prédicat particulier est correct uniquement pour une conversion. R. Bloo, dans [9], résolvait ce problème en maintenant quelque peu artificiellement la situation du cas implicite introduisant des substitutions explicites uniquement en sujet de jugement, et en laissant des substitutions implicites en prédicat – il mentionnait d’ailleurs les difficultés que pose l’introduction en prédicat de substitutions explicites. On a en fait dans tous ces cas la même distinction : les sortes terminales sont des types corrects pour un jugement mais ne peuvent pas être introduites comme types dans un contexte. Simplement, les substitutions explicites imposent d’avoir un mécanisme de reconnaissance un peu plus élaboré.

Perspectives

Différentes directions restent à explorer.

D'une part, l'extension ici donnée pour le $\bar{\lambda}\mu\bar{\mu}$ -calcul peut être combinée avec celle des substitutions explicites : en effet, il existe des versions de $\bar{\lambda}\mu\bar{\mu}$ à substitutions explicites, et il pourrait être intéressant de leur étendre les systèmes de types purs en mettant en œuvre les techniques employées ici. En effet, les deux notions de coupure ainsi obtenues (l'une émanant des jugements sous la forme de séquents classiques, l'autre émanant de la substitution dans la syntaxe), si elles peuvent paraître *a priori* redondantes, peuvent se révéler en fait complémentaires, comme on le voit *e.g.* dans [81].

D'autre part, on peut également se pencher davantage sur la distinction à faire entre les deux notions de correction des types que l'on a discutées ici. Et puis, et c'est peut-être là un des axes les plus prometteurs, du fait que, comme on l'a déjà fait remarquer, la règle d'expansion semble présenter un intérêt dans le cadre de la synthèse de type, et répondre à des besoins particuliers des calculs à substitutions explicites en la matière, il pourrait être intéressant de se pencher sur le problème de la synthèse de type dans ce cadre.

Par ailleurs, la notion d'ordre sur les dérivations présentée ici, qui isole les éléments « importants » d'un arbre afin de pouvoir raisonner par induction non sur la dérivation effective mais sur une dérivation plus commode reconstruite à partir de la conclusion peut sans doute être étendue à d'autres systèmes : en effet, dès l'instant où l'on doit remonter de plus d'un niveau dans un arbre de dérivation, ce qui est fréquent, cette technique permet une simplification considérable des démonstrations.

Enfin, on peut trouver dans [68] une approche différente des systèmes de types purs à substitutions explicites reposant sur un calcul de séquents, et vérifiant également la réduction du sujet. Une comparaison entre les deux approches semble intéressante, d'autant que les deux systèmes ne semblent pas avoir les mêmes ensembles de termes typables.

ANNEXE A

Index

Index des figures, tables et dérivations

FIG. 1	– Confluence de β avec α -conversion implicite et explicite . . .	14
FIG. 2	– Le λ -cube de <i>BARENDREGT</i>	26
TAB. 1	– Figures d'inférence valides pour les systèmes de types purs (implicites)	33
FIG. 3	– Le λ -cube de <i>BARENDREGT</i>	37
TAB. 2	– Figures d'inférence de type valides pour les systèmes de types purs explicites de <i>R. BLOO</i>	44
FIG. 4	– Confluence de $\beta\mathbf{x}$ par interprétation	55
TAB. 3	– Figures d'inférence valides pour les systèmes de types purs explicites	61
DÉR. 1	– Typage dans $\lambda \rightarrow \mathbf{x}$ (toutes versions) de $(\lambda x:a.(\lambda z:a.z)x)\langle a:=b \rangle$.	76
DÉR. 2	– Tentative d'inférence de type dans $\lambda \rightarrow \mathbf{x}$ (version de <i>R. BLOO</i>) pour $\lambda x:b.((\lambda z:a.z)x)\langle a:=b \rangle$	76
DÉR. 3	– Typage dans $\lambda \rightarrow \mathbf{x}$ de $(\lambda x:b.(\lambda z:a.z)x)\langle a:=b \rangle$	77
FIG. 5	– « Curryfication » de l'affaiblissement	86
TAB. 4	– Calcul des séquents sur la signature $\{\rightarrow\}$	135
TAB. 5	– Calcul des séquents à formule active sur la signature $\{\rightarrow\}$. . .	136
TAB. 6	– Figures d'inférence de type valides pour $\bar{\lambda}\mu\tilde{\mu}$ simplement typé	137
TAB. 7	– Figures d'inférence de type pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$	144

Index des définitions

1.1	<i>ordre lexicographique, ordre multi-ensemble</i>	21
1.2	<i>ordre récursif de chemin</i>	22
1.3	<i>système de types pur; sorte, axiome, règle</i>	30
1.4	<i>sortes typables/terminales</i>	30
1.5	<i>\mathfrak{T}-expressions</i>	31
1.6	<i>variables libres/liées pour les systèmes de types purs implicites</i>	31
1.7	<i>α-conversion</i>	31
1.8	<i>$\lambda\mathfrak{T}$-calcul</i>	32
1.9	<i>β-réduction, β-conversion</i>	32
1.10	<i>assertion, contexte, jugement (valide) de typage</i>	32
1.11	<i>contexte bien formé</i>	33
1.12	<i>système de types pur élémentaire (complet)</i>	37
1.13	<i>normalisation forte</i>	39
2.1	<i>$\lambda\mathfrak{T}\mathbf{x}$-expressions</i>	45
2.2	<i>sous-terme (propre)</i>	45
2.3	<i>variables libres/liées dans les systèmes de types purs explicites</i>	46
2.4	<i>variables accessibles/inaccessibles</i>	46
2.5	<i>α-conversion</i>	47
2.6	<i>$\lambda\mathfrak{T}\mathbf{x}$-termes</i>	48
2.7	<i>notion de réduction et de conversion \mathbf{x}, $\beta\mathbf{x}$</i>	48
2.8	<i>propagation</i>	49
2.9	<i>assertion, contexte de typage dans les systèmes explicites</i>	60
2.10	<i>relations entre contextes</i>	60
2.11	<i>jugement de typage dans les systèmes explicites</i>	61
2.12	<i>sortes typables/terminales</i>	63
2.13	<i>nœuds principaux et sous-arbres d'une dérivation</i>	65
2.14	<i>contexte bien formé; justification</i>	69
2.15	<i>jugement de sorte</i>	84
2.16	<i>ordre de simplicité sur les dérivations</i>	86
2.17	<i>\mathbf{x}-réduction fortement parallèle, B-réduction parallèle</i>	111

2.18	<i>algèbre de termes \mathcal{T}_Σ</i>	123
2.19	<i>traduction de \mathcal{F} dans \mathcal{T}_Σ</i>	124
3.1	<i>$\bar{\lambda}\mu\tilde{\mu}$-calcul ; appelants, appelés, capsules</i>	132
3.2	<i>variables libres/liées pour $\bar{\lambda}\mu\tilde{\mu}$</i>	133
3.3	<i>réduction pour $\bar{\lambda}\mu\tilde{\mu}$</i>	134
3.4	<i>assertion, contexte, jugement (valide) de typage dans $\bar{\lambda}\mu\tilde{\mu}$</i>	136
3.5	<i>\mathfrak{T}-appelants, \mathfrak{T}-appelés, \mathfrak{T}-capsules</i>	138
3.6	<i>variables libres/liées dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	139
3.7	<i>réduction (à la racine) dans les s.t.p. pour $\bar{\lambda}\mu\tilde{\mu}$</i>	140
3.8	<i>réduction imbriquée dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	142
3.9	<i>assertion, contexte, jugement de typage des s.t.p. pour $\bar{\lambda}\mu\tilde{\mu}$</i>	143
3.10	<i>couple de contextes bien formé</i>	145
3.11	<i>Terme principal</i>	154

Index des énoncés

1.1	<i>Bonne fondation des ordres de suites finies</i>	22
1.2	<i>Bonne fondation de l'ordre récursif de chemin</i>	22
1.3	<i>Initialisation pour les systèmes de types purs implicites</i>	34
1.4	<i>Engendrement pour les systèmes de types purs implicites</i>	34
1.5	<i>Substitution pour les systèmes de types purs implicites</i>	34
1.6	<i>Correction pour les systèmes de types purs implicites</i>	35
1.7	<i>Réduction du sujet pour les systèmes de types purs implicites</i>	36
1.8	<i>Normalisation forte dans les s.t.p. implicites du λ-cube</i>	39
2.1	<i>Conservation des variables accessibles</i>	50
2.2	<i>Normalisation forte de la \mathbf{x}-réduction</i>	51
2.3	<i>Confluence de la \mathbf{x}-réduction</i>	52
2.4	<i>Composition</i>	52
2.5	<i>Propagation et \mathbf{x}-réduction</i>	53
2.6	<i>Réduction implicite</i>	53
2.7	<i>Transposition de réduction</i>	54
2.8	<i>Inclusion des variables accessibles</i>	54
2.9	<i>Confluence de la $\beta\mathbf{x}$-réduction</i>	54
2.10	<i>Non-libération de variables</i>	58
2.11	<i>Dépendances dans les systèmes de types purs explicites</i>	66
2.12	<i>Interversion de variable et γ-constante</i>	69
2.13	<i>Bonne formation</i>	70
2.14	<i>Types convertibles à une sorte terminale</i>	80
2.15	<i>Usage de pseudo-sorte</i>	84
2.16	<i>Correction des types des systèmes de types purs explicites</i>	85
2.17	<i>Bonne fondation de l'ordre de simplicité</i>	86
2.18	<i>Propriétés liées à l'ordre de simplicité</i>	88
2.19	<i>Affaiblissement</i>	89
2.20	<i>Initialisation pour les systèmes de types purs explicites</i>	90
2.21	<i>Engendrement pour les systèmes de types purs explicites</i>	91
2.22	<i>Conversion de contexte</i>	96

2.23	<i>(Gc)-expansion</i>	97
2.24	<i>Substitution explicite</i>	102
2.25	<i>Substitution de variable inaccessible</i>	107
2.26	<i>Réduction du sujet pour $\beta\mathbf{x}$</i>	111
2.27	<i>Semi-commutation avec la propagation</i>	112
2.28	<i>Réduction du sujet pour \mathbf{x}</i>	114
2.29	<i>Réduction du sujet pour B</i>	119
2.30	<i>Transposition de réduction</i>	124
2.31	<i>Préservation de la forte normalisation</i>	125
2.32	<i>Traduction entre systèmes implicites et explicites</i>	126
2.33	<i>Normalisation forte</i>	127
3.1	<i>Interversion de rédex</i>	141
3.2	<i>Réduction par l'extérieur</i>	141
3.3	<i>Bonne formation de couple</i>	145
3.4	<i>Correction des prédicats de contexte</i>	146
3.5	<i>Correction des types dans les s.t.p. pour $\bar{\lambda}\mu\tilde{\mu}$</i>	146
3.6	<i>Initialisation pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	147
3.7	<i>Engendrement pour les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	147
3.8	<i>Dépendances dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	148
3.9	<i>Typage par une sorte pour $\bar{\lambda}\mu\tilde{\mu}$</i>	149
3.10	<i>Substitution dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	150
3.11	<i>Réduction du sujet dans les systèmes de types purs pour $\bar{\lambda}\mu\tilde{\mu}$</i>	151
3.12	<i>Propriété du s.t.p. des types simples pour $\bar{\lambda}\mu\tilde{\mu}$</i>	152
3.13	<i>Diminution de facteurs de non-principalité</i>	154
3.14	<i>Restriction aux termes principaux</i>	155
3.15	<i>Traduction de réduction de terme principal</i>	155
3.16	<i>Normalisation forte des types simples pour $\bar{\lambda}\mu\tilde{\mu}$</i>	156

ANNEXE B

Glossaire

- abstraction 10
 α -conversion 10, 31
 explicitation de l'— 20
 application 11
 β -réduction 11
 β -conversion 13
 confluence (locale) de la — . . . 12
 non-normalisation de la — . . . 13
 confluence
 CHURCH-ROSSER, propriété de . 12
 diamant, propriété du 12
 — locale 12
 constructions, calcul des 26
 coupure 29
 CURRY-HOWARD-(DE BRUIJN)
 correspondance de — 25
 dépendances entre termes et types 27
 DE BRUIJN, indices de 14
 mise à jour des — 14
 η -réduction 13
 expansion, figure d' 72
 extensionnalité 13
 λ -calcul 9
 — à indices voir DE BRUIJN
 — à noms 9, 16
 — typé 22
 λ -cube 26
 $\bar{\lambda}\mu\bar{\nu}$ -calcul 132
 réduction 134
 syntaxe 132
 types simples pour le — 136
 variable libre/liée 133
 λ -terme 9, 10
 — clos 10
 sous-terme 10
 $\lambda\nu$ -calcul 17
 réduction 17
 syntaxe 17
 $\lambda\mathbf{x}$ -calcul 18
 $\lambda\mathbf{x}^-$ -calcul 18
 $\lambda\mathbf{x}\mathbf{c}$ -calcul 20
 $\lambda\mathbf{x}/\mathbf{c}$ -calcul 20
 méta-opération 11, 14
 explicitation de — 16
 normalisation 12
 faible 12
 forte 12
 pseudo-sorte 80
 usage de — 83
 séquents, calcul des 135
 — à formule active 136
 sorte
 jugement de 84
 substitution 11
 composition de — 19
 explicitation de la — 16
 — explicite 16
 système de types pur explicite . . . 44
 α -conversion 47
 $\beta\mathbf{x}$ -réduction 48
 confluence de la — 54
 composition 52
 correction des types 84, 92
 engendrement 91

- expressions d'un — 45
- normalisation forte 123
- propagation 49
- réduction du sujet 95, 111
- substitution explicite 102
- termes 48
- typage dans les — 60
- variable (in)accessible 46
- variable libre/liée 46
- x-réduction 48
 - confluence de la — 52
 - normalisation forte de la — 51
- système de types pur explicite,
 - version de BLOO 43
- système de types pur (implicite) .. 30
 - α -conversion 31
 - axiome 30
 - calcul sous-jacent à un — 32
 - correction des types 35
 - élémentaire 37
 - élémentaire complet 37
 - engendrement 34
 - expressions d'un — 30
 - fortement normalisant 39
 - le λ -cube comme — 37
 - normalisation forte 39
 - réduction du sujet 36
 - règle 30
 - sorte 30
 - typable/terminale 30
 - spécification 30
 - substitution 34
 - typage dans un — 33
 - variable libre/liée 31
- types
 - à la CHURCH 23, 25
 - à la CURRY 23, 24
- jugements de — 24
 - de substitutions explicites . 28
- système de — 23
- variable
 - accessible 46
 - capture de — 12
 - libre/liée 10, 31, 46, 133

ANNEXE C

Bibliographie

-
- [1] ABADI (Martín), CARDELLI (Luca), CURIEN (Pierre-Louis) & LÉVY (Jean-Jacques), « Explicit substitutions ». *In Proc. of 17th Annual ACM Symposium on Principles of Programming Languages*, San Francisco, California, U.S.A., 1990, p. 31–46. Association for Computing Machinery Press.
- [2] ABADI (Martín), CARDELLI (Luca), CURIEN (Pierre-Louis) & LÉVY (Jean-Jacques), « Explicit substitutions ». *Journal of Functional Programming*, 1(4) : p. 375–416, 1991.
- [3] BAADER (Franz) & NIPKOW (Tobias), *Term Rewriting and All That*. Cambridge University Press, Cambridge, U.K., 1998.
- [4] BARBANERA (Franco) & BERARDI (Stefano), « A symmetric λ -calculus for classical program extraction ». *Information and Computation*, 125(2) : p. 103–117, 1996.
- [5] BARENDREGT (Hendrik Pieter), *λ -Calculus, its Syntax and its Semantics*. Coll. « Studies in Logic and the Foundation of Mathematics », Elsevier Science Publishers B.V. (North Holland), Amsterdam, Nederland, 1984.
- [6] BARENDREGT (Hendrik Pieter), « λ -calculi with types ». *In S. ABRAMSKY, D. GABBAY, & T. MAIBAUM, éd., Handbook of Logic in Computer Science*, volume 2 (Background: Computational Structures), p. 117–309. Oxford University Press, Oxford, U.K., 1992.
- [7] BJØRNER (Nikolaj) & MUÑOZ (César), « Absolute explicit unification ». *In Proc. of 11th International Conference on Rewriting Techniques and Applications*, Norwich, U.K., 2000. *Lecture Notes in Computer Science*, volume 1833, p. 31–46. Springer Verlag.
- [8] BLOO (Roel), « Preservation of strong normalisation for explicit substitution ». Rapport technique n° 95-08, Technische Universiteit Eindhoven, 1995.
- [9] BLOO (Roel), *Preservation of Termination for Explicit Substitution*. Proefschrift ter verkrijging van de graad van Doctor, Technische Universiteit Eindhoven, octobre 1997.
- [10] BLOO (Roel), « Pure type systems with explicit substitution ». *Mathematical Structures in Computer Science*, 11(1) : p. 3–19, 2001.
- [11] BLOO (Roel) & GEUVERS (Herman), « On the edge of strong normalization ». *Theoretical Computer Science*, 211 : p. 375–395, 1999.
- [12] BLOO (Roel), KAMAREDDINE (Fairouz) & NEDERPELT (Rob), « The Barendregt cube with definitions and generalised reduction ». *Information and Computation*, 126(2) : p. 123–143, 1996.

- [13] BLOO (Roel), KAMAREDDINE (Fairouz) & NEDERPELT (Rob), « The Barendregt cube with definitions and generalised reduction ». *Information and Computation*, 126(2) : p. 123–143, 1996.
- [14] BLOO (Roel) & ROSE (Kristoffer Høgsbro), « Preservation of strong normalisation in named λ -calculi with explicit substitution and garbage collection ». *In Proc. of Computing Science in the Netherlands*, 1995, p. 62–72.
- [15] BRIAUD (Daniel), « An explicit η rewrite rule ». In M. DEZANNI-CIANCAGLINI, éd., *Proc. of International Conference on Typed Λ -Calculus and Applications*, 1995. *Lecture Notes in Computer Science*, volume 902, p. 94–108. Springer Verlag.
- [16] CHURCH (Alonzo), « A formulation of the theory of simple types ». *Journal of Symbolic Logic*, 5, 1940.
- [17] CHURCH (Alonzo), *The Calculi of Λ -Conversion*. N° 6 in *Annals of Mathematical Studies*. Princeton University Press, Princeton, U.S.A., 1941.
- [18] CHURCH (Alonzo), *Introduction to Mathematical Logic*. Princeton University Press, Princeton, U.S.A., 2^{de} édition, 1956.
- [19] COQUAND (Thierry), *Une théorie des constructions*. Thèse de doctorat, Université Paris VII Denis Diderot, janvier 1985.
- [20] COQUAND (Thierry) & HUET (Gérard), « The calculus of constructions ». *Information and Computation*, 76(2) : p. 95–120, 1988.
- [21] CURIEN (Pierre-Louis) & HERBELIN (Hugo), « The duality of computation ». *In Proc. of International Conference on Functional Programming*, Montréal, Canada, 2000. Association for Computing Machinery Press.
- [22] CURRY (Haskell) & FEYS (Robert), *Combinatory Logics*, volume I. Elsevier Science Publishers B.V. (North Holland), Amsterdam, Nederland, 1958.
- [23] CURRY (Haskell), HINDLEY (J. Roger) & SELDIN (Jonathan), *Combinatory Logics*, volume II. Elsevier Science Publishers B.V. (North Holland), Amsterdam, Nederland, 1972.
- [24] DAVID (René) & GUILLAUME (Bruno), « A λ -calculus with explicit weakening and explicit substitution ». *Mathematical Structures in Computer Science*, 11 : p. 169–206, 2001.
- [25] DAVID (René) & GUILLAUME (Bruno), « Strong normalization of the typed λ_{ws} -calculus ». *In Proc. of 17th International Workshop on Computer Science Logic*, Wien, Österreich, 2003. *Lecture Notes in Computer Science*, volume 2803.

- [26] DE BRUIJN (Nicolaas Govert), « The mathematical language AUTOMATH, its usage and some of its extensions ». In *Proc. of Symposium on Automatic Demonstration*, Versailles, France, 1968. *Lecture Notes in Computer Science*, volume 125, p. 29–61. Springer Verlag.
- [27] DE BRUIJN (Nicolaas Govert), « Λ -calculus notation with nameless dummies, a tool for automatic formula manipulation with application to the Church-Rosser theorem ». In *Koninklijke Nederlandse Akademie van Wetenschappen, Series A, Mathematical Sciences*, 1972, volume 75, p. 381–392.
- [28] DE GROOTE (Philippe), « Strong normalization in a non-deterministic typed λ -calculus ». In A. NERODE & Y. V. MATIYASEVICH, éd., *Proc. of the 3rd International Symposium on Logical Foundations of Computer Science*, Санкт Петербург, Россия, 1994. *Lecture Notes in Computer Science*, volume 813, p. 142–152. Springer Verlag.
- [29] DERSCHOWITZ (Nachum), « A note on simplification orderings ». *Information Processing Letters*, 9(5) : p. 212–215, 1979.
- [30] DEZANI-CIANCAGLINI (Mariangiola), DOUGHERTY (Daniel J.), LENGREND (Stéphane), LESCANNE (Pierre) & VAN BAKEL (Steffen), « Intersection types for explicit substitutions ». *Information and Computation*, 189(1) : p. 17–42, 2004.
- [31] DI COSMO (Roberto) & KESNER (Delia), « Strong normalization of explicit substitutions via cut elimination in proof nets ». In *Proc. of 12th Symposium on Logic in Computer Science*, Warsaw, Polska, 1997, p. 35–46. IEEECS Society Press.
- [32] DI COSMO (Roberto), KESNER (Delia) & POLONOVSKI (Emmanuel), « Proof nets and explicit substitutions ». *Mathematical Structures in Computer Science*, 13(3) : p. 409–450, 2003.
- [33] DOUGHERTY (Daniel J.), GHILEZAN (Silvia) & LESCANNE (Pierre), « Characterizing strong normalization in a language with control operators ». In E. MOGGI & D. S. WARREN, éd., *Proc. of 6th International Conference on Principles and Practice of Declarative Programming*, Verona, Italia, 2004. Association for Computing Machinery Press.
- [34] DOUGHERTY (Daniel J.) & LESCANNE (Pierre), « Reductions, intersection types, and explicit substitutions (extended abstract) ». In S. ABRAMSKY, éd., *Proc. of International Conference on Typed Λ -Calculus and Applications 2001*, mai 2001. *Lecture Notes in Computer Science*, volume 2044, p. 121–135. Springer Verlag.

- [35] DOUGHERTY (Daniel J.) & LESCANNE (Pierre), « Reductions, intersection types, and explicit substitutions ». *Mathematical Structures in Computer Science*, 13(1) : p. 55–85, 2003.
- [36] DOWEK (Gilles), « A complete proof synthesis method for the cube of type systems ». *Journal of Logic and Computation*, 3(3) : p. 287–315, 1993.
- [37] DOWEK (Gilles) & WERNER (Benjamin), « Proof normalisation modulo ». *Journal of Symbolic Logic*, 68(4) : p. 1289–1316, 2003.
- [38] DYCKHOFF (Roy), LENGRAND (Stéphane) & MCKINNA (James), « A sequent calculus for type theory ». In Z. ESIK, éd., *Proc. of 15th International Conference on Computer Science Logic*, Szeged, Magyar, 2006. *Lecture Notes in Computer Science*, volume 4207, p. 441–455. Springer Verlag.
- [39] FERREIRA (Maria), KESNER (Delia) & PUEL (Laurence), « Λ -calculi with explicit substitutions preserving strong normalization ». In *Proc. of 5th International Conference on Algebraic and Logic Programming*, Aachen, Deutschland, 1996. *Lecture Notes in Computer Science*, volume 1139. Springer Verlag.
- [40] FREGE (Gottlob), *Die Grundlagen der Arithmetik*. Verlag von Wilhelm Koebner, Breslau, Deutschland, 1884.
- [41] GENTZEN (Gerhard), « Untersuchungen über das logische Schließen ». *Mathematische Zeitschrift*, 39, 1935.
- [42] GENTZEN (Gerhard), *Collected Papers*, M.E. SZABO, éd. Elsevier Science Publishers B.V. (North Holland), Amsterdam, Nederland, 1969.
- [43] GIANNINI (Paola), HONSELL (Furio) & RONCHI DELLA ROCCA (Simona), « Type inference : some results, some problems ». *Fundamenta Informaticæ*, 19(1-2) : p. 87–126, 1993.
- [44] GIANNINI (Paola) & RONCHI DELLA ROCCA (Simona), « Characterisation of typings in polymorphic type discipline ». In *Proc. of 13th Symposium on Logic in Computer Science*, Indianapolis, Indiana, U.S.A., 1988, p. 61–70. IEEECS Society Press.
- [45] GIRARD (Jean-Yves), *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*. Thèse de doctorat, Université Paris VII Denis Diderot, juin 1972.
- [46] GIRARD (Jean-Yves), « The system F of variable types: fifteen years later ». *Theoretical Computer Science*, 45 : p. 159–192, 1986.

- [47] GIRARD (Jean-Yves), LAFONT (Yves) & TAYLOR (Paul), *Proofs and Types*. Cambridge University Press, Cambridge, U.K., 1989.
- [48] GÖDEL (Kurt), *Collected Works* (in three volumes), S. FEFERMAN, J.W. DAWSON, S.C. KLEENE, G.H. MOORE, R.M. SOLOVAY & J. VAN HEIJENOORT, éd. Oxford University Press, Oxford, U.K., 1986–1995.
- [49] GRIFFIN (Timothy), « A formulae-as-types notion of control ». *In Proc. of 17th ACM Symposium on Principles of Programming Languages*, San Francisco, California, U.S.A., 1990, p. 47–58. Association for Computing Machinery Press.
- [50] GUTIÉRREZ (Francisco) & RUIZ (Blas), « Cut elimination in a class of sequent calculi for pure type systems ». *In R. DE QUEIROZ, E. PIMENTEL, & L. FIGUEIREDO, éd., Electronic Notes in Theoretical Computer Science*, 2003, volume 84. Elsevier Science Publishers B.V. (North Holland).
- [51] HARDIN (Thérèse) & LÉVY (Jean-Jacques), « A confluent calculus of substitutions ». Rapport technique n° Rapport de recherche 90-11, CEDRIC, 1990.
- [52] HARPER (Robert), HONSELL (Furio) & PLOTKIN (Gordon), « A framework for defining logics ». *In Proc. of 2nd Symposium on Logic in Computer Science*, Ithaca, New York, U.S.A., 1987, p. 194–204. IEEE Computer Society Press.
- [53] HERBELIN (Hugo), *Séquents qu'on calcule*. Thèse de doctorat, Université Paris VII Denis Diderot, janvier 1995.
- [54] HERBELIN (Hugo), « Explicit substitutions and reducibility ». *Journal of Logic and Computation*, 11(3) : p. 29–449, 2001.
- [55] HERBELIN (Hugo), *C'est maintenant qu'on calcule*. Mémoire d'habilitation à diriger les recherches, Université Paris XI Sud (Orsay), décembre 2005.
- [56] HILBERT (David), *Gesammelte Abhandlungen* (in drei Bänden). Julius Springer, Berlin, Deutschland, 1932–1935.
- [57] HINDLEY (J. Roger), « Types with intersection, an introduction ». *Formal Aspects of Computing*, 4 : p. 470–486, 1992.
- [58] JOACHIMSKI (Felix) & MATTHES (Ralph), « Short proofs of normalisation ». *Archive for Mathematical Logic*, 42 : p. 59–87, 2003.
- [59] KAMIN (Sam) & LÉVY (Jean-Jacques), « Attempts for generalising the recursive path orderings », 1980.
- [60] KERVARC (Romain), « Substitutions explicites dans le λ -cube ». Mémoire de DÉA, École normale supérieure de Lyon, juillet 2002. LIP Rapport de DÉA N° 2002-04.

- [61] KERVARC (Romain), « Explicit substitution for the λ -cube ». Communication at the 2nd workshop on Coq and Rewriting, Palaiseau, France, 2004.
- [62] KERVARC (Romain) & LESCANNE (Pierre), « Pure type systems, cut, and explicit substitution ». In D. KESNER, F. VAN RAAMSDONK, & J. B. WELLS, éd., *Proc. of the 2nd International Workshop on Higher Order Rewriting (part of the Federated International Conference on Rewriting, Deduction and Programming)*, Aachen, Nordrhein-Westfallen, Deutschland, 2004, p. 72–77.
- [63] KESNER (Delia), « Confluence of extensional and non-extensional λ -calculi with explicit substitutions ». *Theoretical Computer Science*, 238(1-2) : p. 183–220, 2000.
- [64] KESNER (Delia) & LENGAND (Stéphane), « Extending the explicit substitution paradigm ». In *Proc. of 16th International Conference on Rewriting Techniques and Applications*, Nara, Nippon, 2005. *Lecture Notes in Computer Science*, volume 3467, p. 407–422. Springer Verlag.
- [65] KRIVINE (Jean-Louis), *λ -calcul, types et modèles*. Coll. « Études et recherches en informatique », Masson, Paris, France, 1990.
- [66] LANG (Frédéric) & LESCANNE (Pierre), « On strong normalisation of explicit substitution calculi ». *Journal of Functional and Logic Programming*, 2000.
- [67] LEIVANT (Daniel), « Polymorphic type inference ». In *Proc. of 10th ACM Symposium on Principles of Programming Languages*, Charleston, South Carolina, U.S.A., 1983, p. 88–98. Association for Computing Machinery Press.
- [68] LENGAND (Stéphane), *Normalisation and Equivalence in Proof Theory and Type Theory*. Thèse de doctorat, Université Paris VII Denis Diderot & University of St. Andrews, décembre 2006.
- [69] LENGAND (Stéphane) & MIQUEL (Alexandre), « A classical version of $F\omega$ ». In S. VAN BAKEL & S. BERARDI, éd., *1st Workshop on Classical logic and Computation*, juillet 2006.
- [70] LESCANNE (Pierre), « From $\lambda\sigma$ to $\lambda\nu$: a journey through calculi of explicit substitutions ». In *Proc. of 21st ACM Symposium on Principles of Programming Languages*, Portland, Oregon, U.S.A., 1994, p. 60–69. Association for Computing Machinery Press.
- [71] LONGO (Giuseppe) & MOGGI (Eugenio), « Constructive natural deduction and its ω -set interpretation ». *Mathematical Structures in Computer Science*, 1 : p. 215–253, 1991.

- [72] MAGNUSSON (Lena), *The Implementation of ALF - A Proof Editor Based on Martin-Löf's Monomorphic Type Theory with Explicit Substitution*. Ph.D. Thesis, Chalmers Tekniska Högskola & Göteborgs Universitet, janvier 1995.
- [73] MARTIN-LÖF (Per), *Intuitionistic Type Theory*. Bibliopolis, Napoli, 1984.
- [74] MATTHES (Ralph) & UUSTALU (Tarmo), « Substitution in non-wellfounded syntax with variable binding ». *Theoretical Computer Science*, 327(1-2) : p. 155–174, 2004.
- [75] MELLIÈS (Paul-André), « Typed λ -calculi with explicit substitution may not terminate ». In M. DEZANI-CIANCAGLINI & G. PLOTKIN, éd., *Proc. of International Conference on Typed λ -Calculus and Applications*, Edinburgh, Scotland, U.K., 1995. *Lecture Notes in Computer Science*, volume 902, p. 328–334. Springer Verlag.
- [76] MIQUEL (Alexandre), *Le Calcul des constructions implicite : syntaxe et sémantique*. Thèse de doctorat, Université Paris VII Denis Diderot, décembre 2001.
- [77] MUÑOZ (César), « Dependent types and explicit substitutions ». *Mathematical Structures in Computer Science*, 11(1) : p. 91–129, 2001.
- [78] MUÑOZ (César), « Proof-term synthesis on dependent-type systems via explicit substitutions ». *Theoretical Computer Science*, 266(1-2) : p. 407–440, 2001.
- [79] PARIGOT (Michel), « $\lambda\mu$ -calculus: an algorithmic interpretation of classical natural deduction ». In A. VORONKOV, éd., *Proc. of International Conference on Logic Programming and Automated Reasoning*, Санкт Петербург, Россия, 1992. *Lecture Notes in Computer Science*, volume 624, p. 190–201. Springer Verlag.
- [80] POLONOVSKI (Emmanuel), « PSN implies SN ». In D. KESNER, F. VAN RAAMSDONK, & J. B. WELLS, éd., *Proc. of the 2nd International Workshop on Higher Order Rewriting (part of the Federated International Conference on Rewriting, Deduction and Programming)*, Aachen, Nordrhein-Westfallen, Deutschland, 2004, p. 78–83.
- [81] POLONOVSKI (Emmanuel), « Strong normalization of $\lambda\mu\tilde{\mu}$ -calculus with explicit substitution ». In *Proc. of Foundations of Software Science and Computation Structures*, Barcelona, España, 2004. *Lecture Notes in Computer Science*, volume 2987, p. 423–437. Springer Verlag.

- [82] RENARDEL DE LAVALETTE (Gérard R.), « Strictness analysis via abstract interpretation for recursively defined types ». *Information and Computation*, 99(2) : p. 154–177, 1992.
- [83] ROSE (Kristoffer Høgsbro), *Operational Reduction Models for Functional Programming Languages*. Ph.D. Thesis, Københavns Universitet (Naturvidenskabelige Fakultet), février 1996.
- [84] RUSSELL (Hon. Bertrand), *The Principles of Mathematics*. Cambridge University Press, Cambridge, U.K., 1903.
- [85] RUSSELL (Hon. Bertrand) & WHITEHEAD (Alfred North), *Principia Mathematica* (in three volumes). Cambridge University Press, Cambridge, U.K., 1910.
- [86] SALLÉ (Patrick), « Une extension de la théorie des types en λ -calcul ». In *Proc. of 5th Colloquium on Automata, Languages and Programming*, 1978. *Lecture Notes in Computer Science*, volume 62, p. 398–410.
- [87] SEVERI (Paula) & POLL (Erik), « Pure type systems with definitions ». In A. NERODE & Y. V. MATIYASEVICH, éd., *Proc. of the 3rd International Symposium on Logical Foundations of Computer Science*, Санкт Петербург, Россия, 1994. *Lecture Notes in Computer Science*, volume 813, p. 316–328. Springer Verlag.
- [88] SKOLEM (Thoralf), *Selected Works in Logic*, J.E. FENSTAD, éd. Scandinavian University Books, Oslo, Norge, 1970.
- [89] STRECKER (Martin), *Construction and Deduction in Type Theories*. Dissertation zur Verleihung des akademischen Grades Doktor der Naturwissenschaft, Universität Ulm, 1999.
- [90] TAKAHASHI (Masako), « Parallel reduction in λ -calculus ». *Information and Computation*, 118 : p. 120–127, 1995.
- [91] TERESE, *Term Rewriting Systems*, volume 55. Coll. « Cambridge Tracts in Theoretical Computer Science », Cambridge University Press, Cambridge, U.K., 2003.
- [92] URBAN (Christian), *Classical Logic and Computation*. Ph.D. Thesis, University of Cambridge, octobre 2000.
- [93] VAN BAKEL (Steffen) & DEZANI-CIANCAGLINI (Mariangiola), « Characterizing strong normalization for explicit substitutions ». In S. RAJSBAUM, éd., *Proc. of Latin American Theoretical Informatics Symposium*, Cancún, México, 2002. *Lecture Notes in Computer Science*, volume 2286, p. 356–370. Springer Verlag.

-
- [94] VAN DALEN (Dirk), *Logic and Structure*. Springer Verlag, Berlin – Heidelberg, Deutschland, 1994.
- [95] VAN RAAMSDONK (Femke), *Confluence and Normalisation for Higher-Order Rewriting*. Proefschrift ter verkrijging van de graad van Doctor, Vrije Universiteit Amsterdam, 1996.
- [96] VESTERGAARD (René) & WELLS (Joe B.), « Cut rules and explicit substitutions ». *Mathematical Structures in Computer Science*, 11(1) : p. 131–168, 2001.
- [97] ZIMMER (Pascal), *Récursion généralisée et inférence de types avec intersection*. Thèse de doctorat, Université de Nice – Sophia Antipolis, avril 2004.

*Alle Sätze der Logik sagen aber dasselbe.
Nämlich nichts.*

Ludwig WITTGENSTEIN,
Tractatus logico-philosophicus, 5.43 (1921)

*Mais toutes les propositions de la logique disent
la même chose. À savoir : rien.*

(traduction de Gilles-Gaston GRANGER)

