UNIVERSITÉ PARIS VII

THÈSE DE DOCTORAT

SPÉCIALITÉ : INFORMATIQUE

Sémantique du lambda calcul avec ressources

présentée par

Carolina Lavatelli

soutenue le 22 janvier 1996 devant le jury :

M.	Guy Cousineau	Président
M.	Jean-Jacques Lévy	Rapporteur
Mme.	Simona Ronchi della Rocca	Rapporteur
M.	Pierre-Louis Curien	Directeur
M.	Gérard Boudol	Codirecteur

À mes parents

Remerciements

J'ai eu, pendant les années de recherche qui ont abouti à cette thèse, deux directeurs, Pierre-Louis Curien et Gérard Boudol, à qui je veux dire ici toute ma gratitude. Ils ont fait preuve d'une énorme générosité intellectuelle et d'une infinie patience. Autant dire que je leur dois en grande partie le bonheur d'avoir conclu ce travail.

Je remercie vivement Guy Cousineau, Jean-Jacques Lévy et Simona Ronchi della Rocca de leur participation au jury, de leurs suggestions et d'avoir contribué à faire de ma soutenance un moment heureux.

Pendant ces quatre années de thèse j'ai fait partie de l'équipe Lambda Calcul Typé et Programmation Fonctionnelle du Laboratoire d'Informatique de l'École Normale Supérieure où Pierre-Louis Curien m'a accueillie. Je le remercie de sa confiance et des moyens qu'il a mis à ma disposition. Une reconnaissance particulière revient à tous ceux qui font ou ont fait partie du laboratoire et avec qui j'ai passé des très bons moments : Roberto Bellucci, François Bouladoux, Antonio Bucciarelli, Giuseppe Castagna, Roberto Dicosmo, Maribel Fernández, Giuseppe Longo, Femke van Raamsdonk, Alejandro Ríos.

J'ai aussi fréquenté le Centre de Mathématiques Appliquées à Sophia-Antipolis, où j'ai travaillé avec Gérard Boudol. Je remercie son hospitalité et sa cordialité. Je remercie aussi les autres membres du CMA, spécialement Roberto Amadio, Gérard Berry, Ilaria Castellani, Cosimo Laneve, Davide Sangiorgi qui ont facilité mes séjours et contribué à les rendre très agréables.

Parmi mes amis je remercie tout particulièrement Juan Echagüe qui a voulu m'écouter balbutier devant mes transparents. Merci aussi à Delia Kesner, Maribel Fernández, Roberto Dicosmo et Sandra Gayol, qui m'ont aidé les derniers jours avant la soutenance.

La compagnie et le soutien que José m'a apporté pendant ces années ont été essentiels pour moi et pour le bon déroulement de ce travail; je l'en remercierai toujours. Un grand merci à Tomas, notre enfant, pour sa patience et pour avoir compris que pour rédiger ma thèse j'avais besoin de beaucoup de mon temps.

Je remercie enfin mes parents, mes sœurs, mes grands-parents de m'avoir toujours encouragée.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Le lambda calcul avec ressources λ_r, défini par Boudol, est un raffinement non-déterministe du lambda calcul faible qui permet de contrôler la disponibilité des arguments par la syntaxe. Les ressources ont été introduites afin de remédier à certaines faiblesses du calcul pur, révélées par le codage du lambda calcul faible dans le π-calcul de Milner. Nous abordons dans cette thèse la relation entre des lambda calculs étendus et le π-calcul, l'étude syntaxique du calcul λ_r, et l'étude de ses modèles.

1.1 Fonctionnalité, non-déterminisme et π-calcul

L'apparition de nouveaux paradigmes de calcul parallèle, pour lesquels la communication entre processus n'est pas une simple échange de signaux de synchronisation - le π-calcul de Milner, Parrow et Walker [53], où la communication de noms de canaux implique une re-configuration dynamique des processus, CHOCS de Thomsen [79, 80], où l'on communique des processus - et la nécessité de comprendre leur expressivité, a ouvert des voies très prometteuses pour ce qui est de la relation entre langages parallèles ou distribués et les langages fonctionnels, et de la façon dont ils peuvent être combinés.

Un des premiers résultats sur la relation entre le π-calcul et le λ-calcul faible (base conceptuelle de la programmation fonctionnelle), établi par Milner à l'aide d'une codification de λ dans π, est que ce dernier est strictement plus puissant que le premier, en ce qui concerne les λ-expressions (dans le sens où les π-contextes permettent de distinguer plus d'expressions que les λ-contextes). Les raisons sont multiples; le π-calcul permet de simuler le test de convergence, ajoute le non-déterminisme. Plus important encore, la codification utilisée modélise le processus de substitution du λ-calcul par une consommation de ressources. Le π-calcul incorpore donc la possibilité de bloquer l'évaluation par manque de ressources. En effet, la différence entre π et λ provient du fait que la (β)-conversion - c'est-à-dire, le processus de substitution du λ-calcul - est simulée dans π par un mécanisme non-atomique où chaque remplacement d'une variable par un argument correspond à une synchronisation entre le processus qui contient la variable et le processus qui contient l'argument. Ces synchronisations ont lieu, en gros, chaque fois que le remplacement est nécessaire pour continuer l'évaluation, i.e. lorsque la variable à être substituée apparaît en tête. On peut donc voir la simulation de la β-conversion comme la consommation de la ressource représentée par l'argument. Sans entrer dans les détails du codage, le processus qui contient un argument du λ-calcul dispose de lui de façon illimitée. Ce qui correspond à l'idée que les arguments du λ-calcul sont des ressources inépuisables du processus de substitution.

Mais, à différence du λ-calcul, on peut construire dans π des processus qui se comportent comme des arguments épuisables, ce qui amène à des situations de blocage par manque de ressources. Ce phénomène est à l'origine de la définition du lambda calcul avec ressources par Boudol, dont il est question dans une grande partie de ce travail. Non seulement π est plus expressif que λ mais a aussi un pouvoir de séparation strictement plus grand. L'écart entre ces calculs est en fait très important et ne peut pas être comblé par les extensions classiques du lambda calcul faible, motivés par la théorie de domaines à la Scott - avec appel strict, appel par valeur ou test de convergence (parallèle), choix non-déterministe, composition parallèle - nécessaires et/ou suffisantes pour compléter les modèles.

En utilisant des fonctions de codage compositionnelles qui préservent la notion d'observation, dans le cadre de la théorie de l'expressivité des langages de programmation proposée par Mitchell [56], nous avons étudié le pouvoir de séparation de quelques extensions du lambda calcul faible et la façon dont on peut les combiner afin de rapprocher le calcul λ du calcul π (cf. sections 2.2.1 et 2.2.2). Ajoutés au calcul faible, le test de convergence séquentiel, l'appel par valeur, et aussi l'appel strict sont e'quivalents; le même genre de résultat est vérifié si l'on considère le choix non-déterministe et les fonctions parallèles. Néanmoins, la combinaison de ces deux classes de langages n'est pas toujours admissible en ce qui concerne l'équivalence engendrée entre programmes.

En conclusion, le raffinement du lambda calcul faible avec test de convergence et choix non-déterministe, appelé λ_j [14], constitue un calcul approprié pour être codé dans le π-calcul. Dans la section 2.2.5, nous présentons une classification des extensions du lambda calcul par rapport aux fonctionnalités ajoutées[1] afin de mieux situer les calculs étudiés dans le chapitre 2.

Le codage de λ_j dans π est adéquat mais pas complètement, principalement à cause de la η-conversion. En effet, il permet de distinguer deux utilisations successives de la variable x dans xx par exemple, et de rendre cette expression différente de $x(\lambda y.xy)$. La distinction entre ces deux termes est irréalisable dans toutes les extensions du lambda calcul motivées par la théorie de domaines (cf.

---

[1] Pour le lambda calcul faible cf. [2, 14, 61, 62, 15, 4, 29, 72, 30]; pour la lambda calcul classique cf. [28, 49, 31]; pour PCF cf. [66, 6, 39, 13, 76, 77].

sections 2.2.4 et 2.3).

1.2 Ressources dans le lambda calcul

Le π-calcul n'ajoute au λ-calcul pas seulement la possibilité de tester la convergence d'un programme et le parallélisme, mais aussi la possibilité de bloquer l'évaluation d'un terme qui n'est pas en forme normale de tête faible. Pour rendre compte de cette facilité dans le lambda calcul, Boudol [17] a introduit les ressources. La syntaxe des termes du calcul λ_r est définie par :

$$M ::= x \mid \lambda x.M \mid (MP) \mid M\langle P/x\rangle$$

$$P ::= \mathbf{1} \mid M \mid (P \mid P) \mid M^\infty$$

Le lambda calcul faible gère ses arguments comme des *ressources permanentes*: la β-conversion de $(\lambda x.M)N$ donne comme résultat $M[N/x]$, où l'argument N a remplacé x *autant de fois* que la variable apparaissait libre dans M. Autrement dit, elle transforme l'argument N en une ressource *inépuisable* du processus chargé de réaliser la substitution $[N/x]$. Le lambda calcul avec ressources λ_r permet de modéliser des vraies ressources, i.e. *épuisables*.

Dans λ_r, les arguments infiniment disponibles viennent accompagnés d'une *multiplicité infinie*; l'application standard s'écrit MN^∞. La disponibilité finie ou limitée des arguments est conséquence de l'introduction de *paquets finis de termes* comme arguments. Ces paquets se comportent comme des multi-ensembles où chaque élément (copie d'un terme) est utilisé une fois au plus. Le paquet **1** représente l'argument vide de ressources, et les paquets finis ont la forme $(M_1 \mid \ldots \mid M_n)$. La syntaxe des paquets rappelle la définition des processus dans les calculs parallèles; en effet, l'opérateur $\mid$ est commutatif, associatif et **1** est son élément neutre.

Un radical $(\lambda x.M)P$ se réduit vers le terme $M\langle P/x\rangle$ où $\langle P/x\rangle$ est une substitution explicite dans le sens de Abadi, Cardelli, Curien et Lévy [1]. La présence explicite des substitutions sert à définir une stratégie normalisante d'évaluation qui ne perd pas de ressources. La substitution effective de x par P obéit à deux lois : (1) il y a au plus autant de remplacements que de ressources dans P et (2) ces remplacements sont réalisés par "nécessité", c'est-à-dire, lorsque x apparaît comme terme de tête. L'exemple suivant d'évaluation dans λ_r montre le rôle des substitutions explicites du calcul dans la définition du mécanisme de substitution retardée : si P est un paquet composé du terme N et du paquet Q, alors

$$xR_1 \ldots R_i\langle P/x\rangle R_{i+1} \ldots R_n \rightarrow_r NR_1 \ldots R_i\langle Q/x\rangle R_{i+1} \ldots R_n$$

pourvu que les variables libres de N ne soient pas capturées. Remarquons que λ_r est non-déterministe puisque N est une composante quelconque de P; l'opération de saisie ne suit aucun critère dans la consommation de ressources. Le choix non-déterministe est codé par $M \oplus N = x\langle M \mid N/x\rangle$.

Il faut souligner que le lambda calcul faible est un sous-calcul de λ_r. En effet, les termes ne contenant que des paquets de la forme M^∞ représentent les termes du lambda calcul faible (cf. [17]). Suivant la convention usuelle, on note $\mathbf{I}$ la fonction identité et Ω le terme $(\lambda x.xx^\infty)(\lambda x.xx^\infty)^\infty$.

En outre, la version déterministe de λ_r, appelée lambda calcul avec multiplicités et notée λ_m, s'est révélée très appropriée dans l'étude du codage du λ-calcul dans le π-calcul [19, 20]. Dans le calcul avec multiplicités, les arguments sont de la forme M^n, où n est soit ∞ soit un entier positif, et, dans ce cas, M^n désigne le paquet homogène $\underbrace{(M \mid \ldots \mid M)}_{n}$. La saisie de ressources dans λ_m s'écrit

$$xR_1 \ldots R_i\langle N^{m+1}/x\rangle R_{i+1} \ldots R_n \rightarrow_r NR_1 \ldots R_i\langle N^m/x\rangle R_{i+1} \ldots R_n$$

Les formes normales de λ_r, les termes irréductibles clos, sont de deux types : abstractions $\lambda x.M$ ou termes bloqués comme $xP_1 \ldots P_n\langle \mathbf{1}/x\rangle$ où il n'y pas de ressource disponible pour la variable de tête x (ce qui empêche de continuer avec l'évaluation). Néanmoins, les formes normales que nous retiendrons comme valeurs, résultats observables par des agents externes, sont les abstractions. Ceci suppose que les substitutions explicites appliquées à une abstraction puissent être déplacées à l'intérieur de l'abstraction; c'est-à-dire, $(\lambda x.M)\langle P/y\rangle \rightarrow_r \lambda x.(M\langle P/y\rangle)$ pourvu que $x \neq y$ et que les variables libres de P ne soient pas capturées. La notion de convergence - i.e. de réduction sur une valeur - que nous utiliserons, définie sur les "programmes" du langage (ses termes clos), ne permet pas d'observer les termes bloqués et les identifie donc aux termes divergents (ceux dont on ne peut tirer aucune information). Étant donné le caractère non-déterministe du calcul, il est important de souligner que notre prédicat de convergence $M{\Downarrow_r}$ est du type "may testing" :

$$M{\Downarrow_r} \text{ ssi } M \text{ peut se réduire sur une valeur}$$

Il en découle que, par exemple, $x\langle(\mathbf{I} \mid \Omega)/x\rangle{\Downarrow_r}$ même en ayant une évaluation divergente[2], car le terme se réduit sur la valeur $\mathbf{I}$. La notation $M \Uparrow_r$ est utilisée pour $\neg(M{\Downarrow_r})$. Le préordre observationnel entre termes de λ_r est défini selon le schéma général de Morris par :

$$M \sqsubseteq_r N \;\Leftrightarrow\; \forall\ \lambda_r\text{-contexte } C \;\; C[M]{\Downarrow_r} \;\Rightarrow\; C[N]{\Downarrow_r}$$

En plus du lambda calcul avec ressources, il est question dans ce travail de son extension par un opérateur de test de convergence, noté λ_r^c, dont l'intérêt sera mis

---

[2] En général, les langages non-déterministes admettent une définition de convergence de type "must testing" qui dit qu'un terme M converge ssi toutes les réductions de M terminent sur une valeur. L'exemple, convergent dans le cas du "may testing", ne l'est pas si l'on considère un "must testing".

en évidence dans les sections suivantes de l'introduction. Le préordre observationnel correspondant sera noté $\sqsubseteq_{rc}$. Rappelons la définition du test de convergence $\mathbf{c}$:

$$\begin{cases} \mathbf{c}M \to \mathbf{I} & \text{si } M \text{ réduit sur une valeur} \\ \mathbf{c}M \text{ diverge} & \text{sinon} \end{cases}$$

1.2.1 Résultats syntaxiques

Dans quelle mesure le calcul avec ressources est-il différent du lambda calcul faible étudié par Abramsky et Ong [2, 3] et de ses extensions classiques ? Quelques exemples serviront à illustrer les résultats syntaxiques de Boudol et Laneve [18] sur le calcul avec multiplicités. Nous énoncerons ensuite nos contributions. Dans les trois exemples suivants, M et N sont des termes du lambda calcul pur, égaux dans la théorie observationnelle engendrée par la bisimulation applicative [2] mais séparables par des contextes de λ_r (qui sont en fait des contextes du calcul avec multiplicités), i.e. $\neg(N \sqsubseteq_r M)$ est vérifié.

Exemple 1.2.1 Soit $M = xx^\infty$, $N = x(\lambda y.xy^\infty)^\infty$ et $C = [\,]\langle \mathbf{I}/x \rangle$. Observons que $\mathbf{I}$ ne peut être utilisé qu'une fois au cours de l'évaluation. Alors, si $\mathbf{I} = \lambda z.z$, on a

$$C[N] \to_r \mathbf{I}(\lambda y.xy^\infty)^\infty \langle \mathbf{1}/x \rangle \to_r z\langle (\lambda y.xy^\infty)^\infty/z \rangle \langle \mathbf{1}/x \rangle \overset{\star}{\to}_r \lambda y.(xy^\infty \langle \mathbf{1}/x \rangle) \Downarrow_r$$

$$C[M] \to_r \mathbf{I}x^\infty \langle \mathbf{1}/x \rangle \to_r z\langle x^\infty/z \rangle \langle \mathbf{1}/x \rangle \to_r x\langle \mathbf{1}/x \rangle \ \Uparrow_r$$

où la dernière évaluation diverge car $x\langle \mathbf{1}/x \rangle$ est un terme bloqué. □

Cet exemple montre que le calcul avec multiplicités (donc le calcul avec ressources) est strictement plus discriminant que le lambda calcul faible. On observera que la capacité de bloquer l'évaluation par manque de ressources suffit; le non-déterminisme de λ_r n'est pas nécessaire. Ajoutons enfin que le type de contexte utilisé pour distinguer M et N dans l'exemple 1.2.1 peut être construit dans le π-calcul pour séparer les codages de M et de N (voir [17][3]).

Le résultat de [18] qui nous intéresse à propos de la relation entre le calcul avec multiplicités et les extensions du lambda calcul faible avec test de convergence (parallèle) - sur les termes du lambda calcul pur, est le suivant : *les λ-contextes avec multiplicités séparent strictement plus de λ-termes que les λ-contextes avec test de convergence parallèle* $\mathbf{p}$*, donc avec test de convergence séquentiel* $\mathbf{c}$. Rappelons la définition de $\mathbf{p}$:

$$\begin{cases} \mathbf{p}MN \to \mathbf{I} & \text{si } M \text{ ou } N \text{ se réduit sur une valeur} \\ \mathbf{p}MN \text{ diverge} & \text{sinon} \end{cases}$$

---

[3] Le π-contexte $(\nu x)([\,] \mid x(w).[\![\mathbf{I}]\!]w)$, où le processus qui contient l'argument $\mathbf{I}$ n'est pas précédé d'une réplication !, est analogue au λ_r-contexte C défini dans l'exemple 1.2.1. Une extension du codage mentionné est présentée dans le chapitre 2.

Remarquons que les termes de l'exemple 1.2.1 ne sont pas séparables même en ajoutant les combinateurs **c** ou **p** au lambda calcul.

Les deux exemples suivants, que nous développerons dans la section 3.8.1, montrent comment des arguments avec multiplicités finies permettent de simuler le rôle du test de convergence (parallèle) dans la séparation de λ-termes.

Exemple 1.2.2 Supposons $B = x(\lambda y.\Omega)\Omega$, et soit

$$M = \lambda x.xB(\lambda y.\Omega) \qquad \text{et} \qquad N = \lambda x.x(\lambda z.Bz)(\lambda y.\Omega)$$

Ces termes sont dus à Abramsky et Ong [3]. Ils ont été construits dans le cadre du problème d'adéquation complète pour le lambda calcul faible, et servent à montrer que les contextes avec test de convergence permettent de séparer plus de termes que les λ-contextes purs. En effet, le contexte $A = []\mathbf{c}$ fait la différence : $A[N]$ converge tandis que $A[M]$ diverge. Par ailleurs, le λ_r-contexte $C = []U$, où $U = \lambda vw.v$, sépare aussi M et N : c'est le fait que U ne peut être utilisé qu'une seule fois pendant l'évaluation qui garantit $C[M] \Uparrow_r$. $\Box$

Exemple 1.2.3 Supposons $B = x\Omega\Omega$, et soit

$$M = \lambda x.xB(xB\Omega) \qquad \text{et} \qquad N = \lambda x.xB(x(\lambda z.Bz)\Omega)$$

Les termes M et N ont été construits par Boudol et Laneve [18] pour montrer le pouvoir de discrimination des contextes avec **p** sur les λ-termes. Ils montrent que M et N ne sont pas séparables par des contextes avec le test de convergence séquentiel **c**. Il est clair que le contexte $A = []\mathbf{p}$ sépare M et N : on a que $A[N]$ converge tandis que $A[M]$ diverge. Dans le cadre des ressources, le contexte C suivant, construit en utilisant la technique de séparation de Böhm [8, 46], distingue aussi M et N :

$$C = [](P \mid P)FK \quad \text{où} \quad P = \lambda x_1x_2x_3.x_3x_1x_2 \ , \ K = \lambda x_1x_2.x_1 \ , \ F = \lambda x_1x_2.x_2$$

Remarquons que P a une multiplicité 2 (i.e. peut être utilisé au plus deux fois pendant l'évaluation) et que celle de K et de F est 1. L'évaluation de $C[N]$ converge avec deux utilisations de P, alors que $C[M]$ en aurait besoin de trois. $\Box$

La disparité entre le pouvoir de séparation des multiplicités et celui de **c** est d'autant plus surprenante que ce combinateur n'est pas définissable dans le calcul avec multiplicités. Ce qui entraîne la non-définissabilité de **p**, car $\mathbf{c} = \lambda x.\mathbf{p}xx$.

Nous résumons nos résultats de définissabilité et de pouvoir de séparation :

Le pouvoir de séparation des ressources, sur les λ-termes, est exactement celui des multiplicités. Le non-déterminisme de λ_r n'est pas significatif. Comme le montre l'exemple 1.2.3, les multiplicités finies fournissent une forme de non-déterminisme suffisante pour simuler le comportement de **p** sur les λ-termes. La preuve, donnée dans la section 5.7, est une extension du Crux Lemma de [18] (voir aussi la section 3.8.1 où la propriété est énoncée).

Simplification d'arguments. La propriété de simplification sur λ_r est motivée par l'observation que les ressources non-résolubles d'un terme, de type $\lambda\vec{x}.\Omega$, ne sont utiles qu'au plus une fois dans les dérivations convergentes du terme. Nous montrons que toutes sauf la ressource de plus haut degré de non-résolution peuvent être simplifiées (i.e. éliminées) sans changer le contenu calculatoire du terme. Cette propriété, montrée dans la section 3.6, est utilisée pour prouver, syntaxiquement, que λ_r^c est strictement plus discriminant que λ_r.

Le test de convergence c n'est pas définissable dans λ_r. Donc **p** non plus, même si λ_r permet de coder le choix non-déterministe. Nous présentons dans la section 3.8.2 une preuve syntaxique simple, dans le style de [3] pour le lambda calcul, et aussi de [18] pour le calcul avec multiplicités. De plus, l'exemple 1.2.2 semble indiquer que le test de convergence **c** n'est pas définissable dans λ_r, même si on ne cherche à simuler son effet que sur les λ-termes : la multiplicité de l'argument $U = \lambda vw.v$ dépend du nombre d'occurrences libres de la variables x.

Le lambda calcul avec ressources et test de convergence λ_r^c est strictement plus discriminant que λ_r. La non-définissabilité de **c** dans λ_r est conséquence directe de ce résultat[4], qui, par ailleurs, (nous) était insoupçonnable[5]. Les exemples 1.2.2 et 1.2.3 mettent en évidence que les multiplicités finies remplissent correctement le rôle du test de convergence (parallèle) lorsqu'il s'agit de λ-termes. Mais qu'en est-il pour des termes de λ_r ? Il faut chercher dans la présence de paquets finis de termes (arguments épuisables) la source du pouvoir supplémentaire du test de convergence explicite. En effet, le test de convergence permet, en gros, de compter le nombre de termes non-divergents d'un paquet. Ce que λ_r n'est pas toujours capable de faire, par exemple lorsqu'il s'agit de termes non-résolubles (c'est la propriété que nous avons appelée de simplification d'arguments.) Nous construisons un exemple, qui sera repris dans la section 3.7 :

Exemple 1.2.4 Supposons $B = \lambda zx.xz^\infty$; soit

$$M = B(\lambda y.\Omega \mid \lambda y.\Omega) \qquad \text{et} \qquad N = B(\lambda y.\Omega)$$

Les termes M et N ne sont pas séparables par des λ_r-contextes : au plus un terme non-résoluble (comme $\lambda y.\Omega$) peut être utilisé dans une évaluation convergente. En effet, lorsque $\lambda y.\Omega$ devient le terme de tête, soit le terme obtenu converge soit il y a encore des arguments à consommer et alors diverge. Par conséquent, la seule différence entre M et N, qui est d'avoir un sous-terme avec un nombre différent de ressources non-résolubles, n'est pas observable dans λ_r. Par contre,

---

[4] Si l'introduction d'un opérateur change la théorie, cet opérateur n'est pas définissable dans le langage réduit (cf. [33]).

[5] Nous avons trouvé ce résultat indépendamment de la preuve de non-définissabilité de **c** dans λ_m [18].

le λ_r^c-contexte $C = [](\lambda v.(cv)(cv)$ donne $C[M]\Downarrow_r$ alors que $C[N] \Uparrow_r$ par manque de ressources. □

Le langage des multiplicités est déterministe. Le lambda calcul avec multiplicités est un calcul déterministe par définition de ses règles. Nous montrons dans la section 3.8.2 que, même en libéralisant les règles d'évaluation, le choix non-déterministe n'est pas représentable dans le langage des multiplicités.

D'autre part, nous nous proposons de constituer un fondement algébrique de λ_r tel qu'il existe pour le lambda calcul faible; la difficulté technique consiste à surmonter la non-confluence et la notion de substitution partielle, intrinsèque au calcul. Les points développés sont les suivants :

Caractérisation de la théorie $\simeq_r$. Dans la section 3.6 nous étudions la théorie engendrée par la congruence observationnelle $\simeq_r$. Le but est de caractériser la classe des termes non-résolubles que les contextes de λ_r ne peuvent pas distinguer. Pour le lambda calcul pur, la théorie sensible [8] identifie tous les termes non-résolubles, en particulier $\lambda x.\bot = \bot$, ce qui n'est pas vérifié dans la théorie faible [3]. Dans le cadre de λ_r, la non-confluence de la relation d'évaluation empêche une classification des termes en résolubles ou non-résolubles, comme il existe par exemple pour le calcul pur et avec multiplicités [18], déterministes. Nous introduisons les *degrés de fonctionnalité* et de *non-résolution* d'un terme M de λ_r, qui mesurent les quantités maximales d'abstractions imbriquées, soit de M, soit des termes auxquels M peut être converti (cf. [3] pour ces notions dans le lambda calcul faible). En raison du type de non-déterminisme du calcul avec ressources, les termes peuvent avoir plusieurs dégrés de fonctionalité et de non-résolution à la fois. Par exemple, si $P = (\lambda y.y \mid \lambda yz.y \mid \lambda y.\Omega \mid \Omega)$, alors $x\langle P/x\rangle$ a les degrés de fonctionnalité 1 et 2 et les degrés de non-résolution 0 et 1. Nous montrons que la théorie $\simeq_r$ est "fully lazy" (dans la terminologie d'Abramsky et Ong [3]). Sommairement, deux termes *strictement non-résolubles* sont égaux dans la théorie ssi leur degré maximal de non-résolution est le même. De plus, nous montrons que la théorie est maximale, i.e. elle ne peut pas être élargie sans perdre la propriété de "full laziness" ou devenir incohérente. Ce type de résultat a été prouvé par Boudol et Laneve [18] dans le cadre du calcul avec multiplicités.

Élimination des substitutions explicites - λ-calcul avec paquets. La formalisation de la notion d'évaluation dans le calcul avec ressources par le moyen des substitutions explicites répond au souci de définir d'emblée une stratégie normalisante, proche en réalité d'une machine abstraite. Nous montrons que l'on peut éliminer les substitutions explicites du langage de ressources sans changer pour autant la théorie observationnelle.

Un nouveau calcul λ_d, sans substitutions explicites, appelé lambda calcul avec paquets, est défini dans la section 3.3. La β-réduction est basée sur un processus de substitution non-déterministe qui ne privilégie aucune stratégie de distribution de ressources. Par exemple, si $M = (\lambda x.x(x \mid x))(L \mid N)$, alors $M \rightarrow L(N \mid \Omega)$, et aussi $M \rightarrow \Omega(L \mid N)$, ou encore $M \rightarrow L(\Omega \mid \Omega)$ (entre autres). C'est-à-dire, réaliser une substitution $\langle P/x \rangle$ sur un terme M signifie engendrer l'ensemble des termes obtenus par le processus suivant : diviser le paquet P en autant de sous-paquets $P_0, \ldots, P_n$ qu'il y a d'occurrences libres $u_0, \ldots, u_n$ de la variable x, associer chaque P_i à chaque u_i, et, finalement, substituer à chaque u_i un des termes qui composent P_i. Le résultat de la substitution est bien un ensemble : il y a en général plusieurs manières de partager la substitution d'origine, d'attribuer les sous-paquets obtenus aux occurrences libres de la variable en question, et, finalement, de choisir, pour chaque occurrence, le terme qui doit lui être substitué.

Il est assez simple de transformer les termes de λ_r en termes de λ_d : l'idée est que les substitutions explicites sont calculées en utilisant le processus de substitution de λ_d, défini auparavant. À partir d'un terme dans λ_r on obtient un ensemble de termes dans λ_d. Le résultat principal est que la notion de convergence dans λ_r peut être simulée par la convergence du calcul avec paquets, et inversement. Il s'ensuit que le sous-terme de tête M d'un terme convergent de λ_r peut être remplacé par un des termes de λ_d associés à M sans perturber la convergence. Autrement dit, il est toujours possible de distribuer les substitutions avant de procéder à l'évaluation.

Notre résultat est comparable dans l'esprit à celui d'Abadi et al. [1, 26] qui relie le lambda calcul avec substitutions explicites $\lambda\sigma$ au lambda calcul pur. Les résultats ne sont pourtant pas techniquement proches: les preuves de [1, 26] reposent sur la normalisation forte et la confluence du calcul des substitutions explicites; or, la substitution du calcul avec paquets est essentiellement non-confluente.

Nos motivations concrètes pour l'introduction du calcul avec paquets sont de deux ordres : premièrement, ce calcul supporte la définition d'une notion de réduction forte classique, propriété essentielle pour la construction des approximants des termes en général (cf. paragraphe suivant); deuxièmement, le traitement des environnements dans les modèles de λ_r que nous construisons dans le chapitre 4 correspond à la notion de substitution du calcul avec paquets.

Interprétation algébrique de λ_r par approximants. Les formes normales approchées, ou approximants, des expressions du λ-calcul et leur utilité pour l'étude des sémantiques dénotationnelle et opérationnelle remontent aux travaux de Wadsworth [81, 82], Hyland [45] et Lévy [47], qui reposent sur la propriété essentielle suivante : *le comportement calculatoire des termes est déterminé par leurs formes normales approchées.*

Pour la construction des approximants, le langage du λ-calcul est étendu avec une constante $\perp$, forme normale (car irréductible) dont le contenu calculatoire est nul. Les travaux mentionnés auparavant concernent deux types de théories, sensible [8], qui correspond à l'interprétation du λ-calcul dans les modèles de Scott, et faible [3], qui correspond au modèle élevé[6] d'Abramsky :

$$\text{théorie sensible}\left\{\begin{array}{l} \bot M = \bot\} \quad \text{théorie faible} \\ \lambda x.\bot = \bot \end{array}\right.$$

La procédure à suivre pour le calcul des approximants d'un terme M consiste à :

- réduire M (il s'agit de la réduction pleine, une stratégie particulière n'est pas suffisante en général)
- remplacer les radicaux restants par $\bot$

Prenons comme exemple

$$Y = \lambda f.(XX) \qquad \text{où} \qquad X = \lambda x.f(xx)$$

Pour tout $n = 0, 1, \ldots$

$$Y \overset{n}{\to} \lambda f.f^{n+1}(XX) \to \ldots$$

On obtient donc les approximants de Y en remplaçant le radical (XX) par $\bot$, d'où l'ensemble infini d'approximants

$$\lambda f.\bot \quad \lambda f.f(\bot) \quad \ldots \quad \lambda f.f^{n+1}(\bot) \quad \ldots$$

On remarquera que dans la théorie sensible $\lambda f.\bot$ est en réalité $\bot$. La notion d'approximant a servi à Wadsworth et Hyland pour montrer l'adéquation du modèle de Scott et du modèle des graphes, respectivement, du point de vue calculatoire. En effet, chaque expression est égale à la limite des ses approximants dans ces modèles. L'approche de Lévy est différente, elle consiste à montrer que l'interprétation des λ-termes sur le domaine des approximants engendre une precongruence, d'où l'on obtient une sémantique algébrique du λ-calcul. La particularité de son interprétation est qu'elle distingue les termes $\lambda x.\Omega$ et Ω; c'est-à-dire, les termes observables sont les formes normales de tête faibles [8]. Le résultat de continuité syntaxique qui est à la base de la construction de Lévy est le suivant : étant donnés un terme M et un contexte C, l'ensemble des approximants de $C[M]$ est égal à l'union, sur les approximants A de M, des ensembles d'approximants de $C[A]$. Comme corollaire : pour tout terme M et contexte C qui ferme M, $C[M]$ converge ssi il existe un approximant A de M tel que $C[A]$ converge. Autrement dit, un *unique* approximant est nécessaire et suffisant pour reproduire le contenu calculatoire d'un terme dans un contexte. L'unicité est conséquence de la confluence du λ-calcul car elle implique que l'ensemble des approximants d'un

---

[6] lifted

terme peut être complété de façon à constituer un ensemble *dirigé*. Ainsi, lorsque deux approximants A_1 et A_2 sont nécessaires pour reproduire le comportement de deux occurrences différentes de M dans une évaluation convergente de $C[M]$, il est toujours possible de prendre la borne supérieure de A_1 et A_2.

La non-confluence du calcul avec ressources rend intéressant le développement d'une notion d'approximant, qui constitue le thème de la section 3.7, dans le but d'obtenir une sémantique algébrique adéquate par rapport au préordre observationnel $\sqsubseteq_r$. L'interprétation des termes coïncide avec l'ensemble des ses approximants (que l'on doit encore définir) et la sémantique algébrique est l'inclusion des interprétations. L'adéquation repose sur une propriété de continuité syntaxique analogue à celle de Lévy. Or la non-confluence de λ_r empêche la construction d'ensembles dirigés d'approximants. La conséquence immédiate est la perte de l'unicité. En effet, notre résultat principal, appelé "lemme d'approximation", dit qu'un *nombre fini* d'approximants peut être nécessaire pour reproduire le comportement d'un terme dans un contexte donné. Formellement,

$$\forall M \in \Lambda_r \;\; C[M]\Downarrow_r \Leftrightarrow \exists A_1, \ldots, A_n \text{ approximants de } M \quad C[A_1 \oplus \ldots \oplus A_n]\Downarrow_r$$

En ce qui concerne des travaux dans ce domaine, nous pouvons mentionner une forme limitée du lemme d'approximation, montrée dans [18], où C est un contexte du calcul avec multiplicités et M est un λ-terme pur pour laquelle la notion d'approximant utilisée est celle de Lévy; et aussi la définition des arbres de Böhm non-déterministes par de Liguoro et Piperno [49], pour l'étude des modèles d'un lambda calcul classique non-déterministe avec un préordre observationnel de type "must testing", dont nous avons appris l'existence après avoir rédigé l'essentiel cette thèse.

Techniquement, le point de départ pour la construction des approximants de λ_r est basé sur deux observations qui se chevauchent :

- la relation de réduction généralisée nécessaire doit permettre la distribution de ressources non seulement sur la variable de tête mais sur toutes les occurrences des variables; la stratégie faible n'est pas complète, comme elle ne l'était dans le cas du λ-calcul;

- les substitutions explicites doivent être éliminées à un moment ou à un autre dans la construction des approximants. En effet, les saisies potentielles doivent disparaître au même titre que les β-radicaux.

Notre approche consiste à définir les approximants de λ_r sur le langage du calcul avec paquets augmenté de la constante $\perp$. Ceci à l'avantage de permettre une construction des approximants par récurrence, car il n'y a pas de substitutions explicites. De plus, λ_d admet une réduction pleine classique; il suffit d'autoriser l'application de la β-réduction non-déterministe à tout niveau du terme.

Un approximant de λ_d est soit $\perp$, précédé d'un nombre fini d'abstractions; soit une forme normale de tête où les paquets sont des compositions parallèles finies d'approximants. Étant donné un terme dans λ_d, son meilleur approximant (ou approximant directe) se construit

- en remplaçant les radicaux par $\perp$;
- en calculant ensuite les approximants directes des paquets présents dans le terme, lorsque le sous-terme de tête est une variable;
- pour les paquets, on calcule un approximant pour chaque terme de la composition parallèle; les termes avec multiplicité infinie sont remplacés par des paquets finis.

Un approximant d'un terme dans λ_d est l'approximant directe d'un deuxième terme sur lequel le premier se réduit. L'interprétation algébrique d'un terme M dans λ_r comme l'ensemble des approximants de l'image de M dans λ_d (un ensemble de termes, cf. paragraphe sur le calcul avec paquets). Par exemple, parmi les approximants de $x(xx)\langle K \mid F/x\rangle$, où $K = \lambda zy.z$ et $F = \lambda zy.y$, on a $\perp$, $\lambda y.\perp$ et $\lambda y.\mathbf{I}$.

Le reste du chapitre 3 est consacré à des propriétés de la sémantique observationnelle du calcul avec ressources et test de convergence, nécessaires pour les preuves d'adéquation (complète) des modèles que nous donnons dans les chapitres 5 et 6. Parmi ces propriétés on trouve :

- le *lemme des contextes pour* λ_r^c dans la section 3.5.2 : le préordre observationnel sur λ_r a une caractérisation en fonction de contextes applicatifs (cf. [17]); nous étendons ce résultat au calcul avec ressources augmenté avec test de convergence;
- la *validité de la* η*-expansion faible* dans la section 3.5.3 : le calcul avec ressources et test de convergence vérifie une forme faible de η-expansion, à savoir : si M est un terme convergent de λ_r^c, alors M et $\lambda y.My^\infty$ sont égaux dans la théorie engendrée par $\simeq_{rc}$. Une propriété similaire est prouvée dans [17] pour le calcul λ_r.

1.2.2 Équations aux domaines. Modèles à environnements

Quel genre d'équation aux domaines peut-on associer au calcul avec ressources? Qu'est-ce qu'un modèle de ce calcul? Ces deux questions délimitent le sujet du chapitre 4 de la thèse.

Dans l'approche de Scott, les expressions du lambda calcul sont en même temps des objets et des fonctions sur ces objets, ce qui amène a considérer des domaines isomorphes à (certains) espaces de fonctions continues. C'est ainsi pour le $\lambda\beta K$-calcul, pour le calcul faible, pour des raffinements de celui-ci - pas seulement avec test de convergence, mais aussi avec appel par valeur, avec fonctions parallèles, avec choix non-déterministe[7]. Les espaces des fonctions utilisés varient d'un calcul mais les équations aux domaines sont essentiellement de la forme $D = D \to D$ et les modèles à environnements engendrés vérifient des postulats bien connus qui garantissent une représentation correcte du processus de substitution du lambda calcul (cf. par exemple Scott [74, 75], Hindley et Longo [41], Abramsky et Ong [2, 3, 59, 60], Boudol [14, 17], et, pour les modèles de PCF Milner [51], Plotkin [66], et aussi Berry, Curien et Lévy [12]). Encore plus, les solutions canoniques des équations aux domaines ont des présentations comme modèles de filtres, sur des logiques définies autour de la flèche et de la conjonction [22, 23, 9].

Pour ce qui est des calculs faibles, l'équation aux domaines devient $D = (D \to D)_\perp$ afin de distinguer l'élément divergent $\perp$ de l'élément $\lambda x.\perp$ (sémantique par approximants de Lévy). La solution canonique sur les ordres partiels complets est un modèle adéquat, surabondant dans le sens où il contient des fonctions - possédant un certain degré de parallélisme - qui ne sont pas exprimables dans le calcul : le test de convergence $\mathbf{p}$ complète le lambda calcul faible par rapport à ce modèle, comme le "parallel or" complète PCF. D'où, en appliquant le résultat de Boudol et Laneve, i.e. que le calcul de ressources est strictement plus discriminant que le lambda calcul avec $\mathbf{p}$ (sur les lambda termes purs), on peut tirer une première conclusion, à savoir, que le modèle canonique du lambda calcul faible ne peut pas définir un modèle adéquat de λ_r. En effet, si $\sqsubseteq$ désigne le préordre observationnel dans $\lambda + \mathbf{p}$, et $\sqsubseteq_D$ est le préordre dans le domaine D sur les interprétations des λ-termes, on a,

$$\left\{ \begin{array}{c} M \sqsubseteq_r N \;\Rightarrow\; M \sqsubseteq N \Leftrightarrow M \sqsubseteq_D N \\ \text{et} \\ M \sqsubseteq_r N \;\not\Leftarrow\; M \sqsubseteq N \end{array} \right\} \quad \text{implique} \quad M \sqsubseteq_D N \;\not\Rightarrow\; M \sqsubseteq_r N$$

Ce raisonnement ne nous donne cependant pas d'information sur ce qu'est un modèle de λ_r. Le lambda calcul avec ressources a une forme de parallélisme (ou de non-déterminisme) plus "explicite" que celle exprimable par le combinateur $\mathbf{p}$ - rappelons que $M \oplus N = x\langle M \mid N/x\rangle$ - dont la contrepartie mathématique est le "join" dans le treillis des dénotations selon l'approche de Boudol [14][8]. Les modèles de λ_r que nous construisons sont aussi des treillis complets (avec des bornes supérieures arbitraires). Cependant, le "join", en étant idempotent, ne permet pas d'exprimer la différence entre ressources infiniment disponibles et ressources épuisables. En termes logiques, la notion de conjonction dans les

---

[7] On fait référence ici à des calculs où le non-déterminisme reste fonctionnel, et non pas par exemple à celui de Hennessy et Plotkin [40], dont l'équation aux domaines associée fait apparaître un domaine des parties comme domaine des résultats.

[8] La solution de l'équation aux domaines dans les treillis complets est un modèle adéquat du lambda calcul faible et complètement adéquat de $\lambda + \mathbf{c} + \oplus$.

modèles de filtres usuels, ou dans les systèmes de types avec intersection associés [25, 70, 44], n'est pas appropriée pour définir une théorie de la fonctionnalité pour λ_r. Boudol [17] propose une modification de la conjonction (qui perd le caractère idempotent), et une notion correspondante d'affectation de types qui engendre une sémantique adéquate par rapport au préordre observationnel dans λ_r. L'opération de conjonction dans ce système est appelée “produit” et notée ×. Nous nous occuperons de ce système dans la section suivante de cette introduction.

L'équation $D = (D \rightarrow D)_\perp$ n'était pas assez expressive pour permettre l'interprétation des arguments de λ_r; nous la remplaçons par une autre équation, de la forme $D = (\mathcal{M}(D) \rightarrow D)_\perp$, où D est un treillis complet p-algébrique[9] et $\mathcal{M}(D)$ est, en gros, un domaine de multi-ensembles de termes. Nous définissons deux équations de ce type, qui se distinguent par la construction $\mathcal{M}(D)$ utilisée.

Pour les deux équations, on observera que la nécessité de travailler sur des treillis complets ne provient pas seulement de l'interprétation du non-déterminisme, mais aussi de la modélisation du processus de substitution, qui correspond à celui du calcul avec paquets. Pour calculer une dénotation, on partage les environnements comme on partage les substitutions dans le calcul avec paquets, on distribue les sous-environnements obtenus, et, finalement, pour chaque occurrence de variable, on calcule sa valeur dans le sous-environnement correspondant. On obtient ainsi un ensemble de dénotations, correspondant, chacune, à une approximation du terme d'origine dans l'environnement d'origine. Cet ensemble n'est pas dirigé (cf. discussion sur les approximants); on prend donc la borne supérieure de toutes ces dénotations (c'est ici que la nécessité de travailler sur les treillis complets apparaît).

Les paragraphes suivants décrivent sommairement les équations aux domaines pour λ_r ainsi que leurs solutions comme domaines logiques, dans le style des domaines de filtres [22, 23]. Pour chacun d'eux, la dénotation d'une expression dans un environnement coïncide avec l'ensemble des types qu'elle peut recevoir dans un système d'affectation de types particulier, basé sur celui défini dans [17]. C'est sur ces systèmes que nous prouverons les résultats d'adéquation (complète) des modèles pour λ_r (resp. λ_r^c); nous reviendrons sur ce point dans la section suivante de cette introduction.

Première équation. Nous définissons un domaine $\mathcal{M}(D)$ qui est la complétion par idéaux de l'ensemble ordonné de multi-ensembles d'éléments compacts et premiers de D. La construction, qui donne un ordre partiel complet ω-algébrique, rappelle celle du domaine de parties de Hoare [38]. Puisque $\mathcal{M}(D)$ n'a pas des bornes supérieures arbitraires, il n'est pas le domaine des arguments à proprement parler, mais permet d'interpréter des approximations des paquets de termes. Le choix non-déterministe est toujours dénoté par le “join” dans le domaine D.

---

[9] Tout élément de D est égal à la borne supérieure des éléments à la fois compacts et premiers qu'il domine.

Le domaine logique qui résout cette équation est construit sur la théorie de types dont le langage est

$$(\texttt{Ft}) \quad \phi \ ::= \ \omega \,|\, \pi \rightarrow \phi$$

$$(\texttt{Fb}) \quad \pi \ ::= \ \phi \,|\, \pi \times \pi$$

La relation de conséquence entre types, $\tau \leq \tau'$, est définie, pour ce qui est de la flèche, par les lois standards dans les théories faibles. Pour ce qui est du *produit* $\times$, les lois garantissent que ce constructeur est associatif, commutatif, a ω comme élément neutre, et qu'il est covariant en ses deux arguments. Cependant, $\times$ n'est pas idempotent. On remarquera que ces propriétés sont celles des multi-ensembles, et aussi celles de la composition parallèle.

La solution canonique $\mathcal{C}$ de l'équation est l'ensemble des *cônes supérieures non-vides*[10] sur la théorie ($\texttt{Ft}$, $\leq$). La preuve (à isomorphisme près) de

$$\mathcal{C} = (\mathcal{M}(\mathcal{C}) \rightarrow \mathcal{C})_{\perp}$$

repose sur la caractérisation de $\mathcal{M}(\mathcal{C})$ comme l'ensemble des filtres sur la théorie ($\texttt{Fb}$, $\leq$): les filtres compactes $\uparrow\pi$ sont en correspondance isomorphe avec les idéaux engendrés par un unique multi-ensemble d'éléments de $\mathcal{C}$.

L'intérêt de cette première équation est qu'elle est la contrepartie, dans la théorie des domaines, de la théorie de la fonctionnalité pour λ_r proposée par Boudol [17][11], et que sa solution $\mathcal{C}$ constitue un modèle complètement adéquat de λ_r^c. L'équation, l'interprétation des termes de λ_r et la construction de la solution sont présentés dans la section 4.2.

Deuxième équation. Le domaine $\mathcal{M}(D)$ de la deuxième équation est construit pour permettre l'interprétation des paquets de termes. L'ensemble de base est celui des multi-ensembles d'éléments de D (compacts et premiers, comme pour la première équation). Mais nous faisons la complétion par idéaux sur l'ensemble des parties finies de multi-ensembles, ce qui donne un treillis complet avec un produit $\bullet$ - commutatif, associatif et avec le multi-ensemble vide comme élément neutre - approprié pour interpréter la composition parallèle. Dans le cadre de cette équation, la dénotation de l'application et des paquets des termes est:

$$\begin{aligned} [\![P \mid Q]\!]_{\rho} &= \bigsqcup [\![P]\!]_{\rho_0} \bullet [\![Q]\!]_{\rho_1} \\ [\![MP]\!]_{\rho} &= \bigsqcup [\![M]\!]_{\rho_0}([\![P]\!]_{\rho_1}) \end{aligned}$$

où ρ_0, ρ_1 constituent un partage de l'environnement ρ.

---

[10] On entend par cône supérieure un ensemble de types fermé par le haut par rapport à la relation $\leq$.

[11] Le langage de types est celui de [17] et la relation $\leq$ traduit la manipulation des hypothèses dans le système de types.

Nous raffinons le langage de types du domaine $\mathcal{C}$ par l'ajout d'une conjonction idempotente $\wedge$:

$$(\mathrm{Ft}\ )\quad \phi\ \ ::=\ \ \omega \,|\, \pi \to \phi \,|\, \phi\wedge\phi$$

$$(\mathrm{Fb}\ )\quad \pi\ \ ::=\ \ \phi \,|\, \pi \times \pi \,|\, \pi\wedge\pi$$

La relation de conséquence entre types $\leq$ est définie comme dans le cas du modèle $\mathcal{C}$ pour ce qui est de la flèche et du produit. Pour la conjonction on ajoute les lois standards, et les constructeurs $\times$ et $\wedge$ sont reliés par la loi de distribution suivante :

$$(\pi_0 \times \pi)\wedge(\pi_1 \times \pi)\ \leq\ (\pi_0\wedge\pi_1) \times \pi$$

qui permet d'écrire les formules π comme des conjonctions de produits.

On définit $\mathcal{F}$ et $\mathcal{F}_{\mathrm{b}}$ comme les ensembles des *filtres* sur les théories (Ft , $\leq$) et (Fb , $\leq$), respectivement. On montre que $\mathcal{F}$ est solution de l'équation, à isomorphisme près, i.e.

$$\mathcal{F} = (\mathcal{M}(\mathcal{F}) \to \mathcal{F})_\perp$$

La partie essentielle de la preuve est l'isomorphisme entre $\mathcal{M}(\mathcal{F})$ et $\mathcal{F}_{\mathrm{b}}$, à travers l'isomorphisme entre éléments compacts. C'est ici que la présence de la conjonction est cruciale, car les éléments compacts de $\mathcal{M}(\mathcal{F})$ sont des unions finies. La relation entre les opérations dans le domaine et les constructeurs des types dans la logique est donc la suivante : les unions finies sont les conjonctions et les produits $\bullet$ sont les produits $\times$.

L'intérêt de cette équation est que l'interprétation associée est définie de façon compositionnelle tant au niveau des termes que des paquets et que la construction $\mathcal{M}$ est montrée universelle dans la catégorie des treillis complets p-algébriques (dans la section 4.3.2). L'équation, la fonction d'interprétation associée et la construction de la solution sont présentés dans la section 4.3.

1.2.3 Modèles adéquats de λ_r. Complétude

La dénotation d'une expression dans un environnement, dans les domaines $\mathcal{C}$ et $\mathcal{F}$, coïncide avec l'ensemble des types que l'expression peut recevoir dans deux systèmes d'affectation de types basés sur celui défini dans [17], respectivement. Nous donnons les preuves d'adéquation et de complétude dans ce formalisme; elles sont basées sur la caractérisation de la convergence par les systèmes de types : un terme est convergent ssi il a un type différent de ω.

On peut reformuler en termes de types avec intersection, l'idée énoncée plus haut sur l'impossibilité d'exprimer la différence entre ressources infiniment disponibles et ressources épuisables dans le domaine usuel.

La propriété fondamentale de la discipline de types avec intersection [70, 21, 24, 25], en comparaison avec la théorie de base de la fonctionnalité pour le lambda calcul due à Curry, est que les types sont préservés par expansion (et non seulement par réduction). Lorsque le λ-terme $M[N/x]$ a un type ϕ, on peut construire un typage pour M en supposant que x a le type $\sigma_1 \wedge \ldots \wedge \sigma_n$, où les σ_i sont les types donnés à N (i.e. à chaque occurrence de N dans $M[N/x]$) pour obtenir ϕ. Par exemple, l'auto-application $\lambda x.xx$ a le type $((\phi \to \sigma) \wedge \phi) \to \sigma$, tandis qu'elle n'a pas de type significatif dans la théorie de base - ce qui permet à Curry d'éviter certains paradoxes.

Supposons qu'un terme M a un type $\sigma_1 \wedge \ldots \wedge \sigma_n \to \phi$. Pour déduire que (MN) a le type ϕ, il faut être capable de prouver que, pour chaque σ_i, il existe une dérivation qui donne ce type à N. Cette condition d'existence est étroitement liée au caractère permanent ou inépuisable des arguments dans le lambda calcul et implique l'idempotence de la conjonction, i.e. $\sigma \wedge \sigma = \sigma$. Du point de vue de la notion de convergence dans le calcul avec ressources, cette façon de procéder est évidemment correcte lorsque (tous) les arguments ont une multiplicité infinie. Mais la caractérisation est impossible si l'on considère des termes avec multiplicité finie, comme le montre l'exemple suivant :

Exemple 1.2.5 Soit $M = \lambda x.x(x\mathbf{I})$ un λ-terme, et $\delta = \phi \to \phi$. Dans la théorie de la fonctionnalité de Curry, on dérive le type $(\delta \to \delta) \to \delta$ pour M :

$$\dfrac{\dfrac{x : \delta \to \delta \vdash x : \delta \to \delta \qquad \dfrac{x : \delta \to \delta \vdash x : \delta \to \delta \quad \vdash \mathbf{I} : \delta}{x : \delta \to \delta \vdash x\mathbf{I} : \delta}}{x : \delta \to \delta \vdash x(x\mathbf{I}) : \delta}}{\vdash \lambda x.x(x\mathbf{I}) : (\delta \to \delta) \to \delta}$$

Cette dérivation est aussi une dérivation dans le système de types avec intersection, constructeur qui n'est pas nécessaire car on donne le même type aux deux occurrences de x. En admettant que le système de types pour λ_r puisse être le système classique avec intersection, le terme $\hat{M} = \lambda x.x(x\mathbf{I}^\infty)^\infty$ de λ_r, obtenu en ajoutant la multiplicité infinie aux arguments présents dans M, aurait le type $(\delta \to \delta) \to \delta$. De plus, $\hat{M}\mathbf{I}^\infty$ aurait le type δ, ce qui est intuitivement correct du point de vue de la convergence, car le terme se réduit sur la valeur $\mathbf{I}$:

$$\hat{M}\mathbf{I}^\infty \to_r x(x\mathbf{I}^\infty)^\infty \langle \mathbf{I}^\infty / x \rangle \to_r \mathbf{I}(x\mathbf{I}^\infty)^\infty \langle \mathbf{I}^\infty / x \rangle \overset{\star}{\to}_r x\mathbf{I}^\infty \langle \mathbf{I}^\infty / x \rangle \overset{\star}{\to}_r \mathbf{I}$$

Mais, le terme $\hat{M}\mathbf{I}$ (où $\mathbf{I}$ a multiplicité 1) aurait lui aussi le type δ, tout en étant divergent, alors le blocage deviendrait observable (quand le seul type que l'on peut accepter pour $\hat{M}\mathbf{I}$ est ω) :

$$\hat{M}\mathbf{I} \overset{\star}{\to}_r \mathbf{I}(x\mathbf{I})\langle \mathbf{1}/x \rangle \to x\mathbf{I}\langle \mathbf{1}/x \rangle \ \Uparrow_r$$

Par conséquent, les systèmes de types avec intersection ne sont pas appropriés pour caractériser la convergence dans λ_r. $\square$

Boudol propose dans [17] une modification dans le traitement de la conjonction basée sur l'élimination de la contraction d'hypothèses (i.e. l'élimination de l'idempotence dans la théorie des types), qui permet de caractériser la convergence dans λ_r. Le langage du nouveau système, appelé $\mathcal{P}$, est celui du modèle $\mathcal{C}$. Les types $\phi \in \texttt{Ft}$ sont les types des termes, tandis que les types $\pi \in \texttt{Fb}$ sont ceux des paquets. Les règles manipulent les hypothèses de façon multiplicative; dans ce système, le terme $\hat{M}$ de l'exemple 1.2.5 a le type suivant :

$$\dfrac{\dfrac{x : \delta \to \delta \vdash x : \delta \to \delta \quad x : \delta \to \delta \vdash (x\mathbf{I}^\infty)^\infty : \delta}{x : (\delta \to \delta) \times (\delta \to \delta) \vdash x(x\mathbf{I}^\infty)^\infty : \delta}}{\vdash \lambda x.x(x\mathbf{I}^\infty)^\infty : (\delta \to \delta) \times (\delta \to \delta) \to \delta}$$

Nous avons utilisé ici, implicitement, le fait que parmi les types de N^∞ on a les types de N, quelque soit N. Par ailleurs, le fait d'avoir deux facteurs dans le produit $(\delta \to \delta) \times (\delta \to \delta)$ du type de $\hat{M}$ indique que le nombre d'occurrences “significatives” de la variable x dans le corps de $\hat{M}$, en ce qui concerne la convergence, est deux. Une application $\hat{M}P$ a un type différent de ω si P peut être typé par $(\delta \to \delta) \times (\delta \to \delta)$; c'est-à-dire, si P contient au moins deux ressources (possiblement différentes) de type $\delta \to \delta$. Par exemple, $\vdash (\mathbf{I} \mid \mathbf{I}) : (\delta \to \delta) \times (\delta \to \delta)$. Donc $\vdash \hat{M}(\mathbf{I} \mid \mathbf{I}) : \delta$. Mais $\hat{M}\mathbf{I}$ ne peut pas avoir ce type.

Le système $\mathcal{P}$ sera utilisé pour décrire les dénotations des termes avec ressources dans le domaine $\mathcal{C}$ de cônes. Quant au domaine de filtres $\mathcal{F}$, le système de types $\mathcal{P}_\wedge$ qui lui est naturellement associé est l'extension de $\mathcal{P}$ par les règles de typage de la conjonction, avec une manipulation standard des hypothèses. C'est-à-dire,

$$\frac{\Gamma \vdash T : \tau_0 \quad \Gamma \vdash T : \tau_1}{\Gamma \vdash T : \tau_0 \wedge \tau_1}$$

plus la règle d'élimination

$$\frac{\Gamma \vdash T : \tau_0 \wedge \tau_1}{\Gamma \vdash T : \tau_i} \qquad i = 0, 1$$

La sémantique abstraite sur les λ_r-termes engendrée par la notion de typage dans $\mathcal{P}$ est prouvée adéquate dans [17]. Cependant, nous montrons qu'elle n'est pas complètement adéquate.

Exemple 1.2.6 Soit $M = x(\lambda yz.\Omega \mid \lambda z.\Omega)$ et $N = x(\lambda yz.\Omega)$. Le terme N est clairement une simplification de M; donc $M \simeq_r N$ (voir aussi l'exemple 1.2.4). Pourtant, M a des types que N n'a pas.

On définit $\gamma_1 = \omega \to \omega$ et $\gamma_2 = \omega \to (\omega \to \omega)$. Il est clair que

$$\vdash \lambda z.\Omega : \gamma_1 \quad \text{et} \quad \vdash \lambda yz.\Omega : \gamma_2$$

sont prouvables dans $\mathcal{P}$. Alors,

$$\frac{x:\gamma_2\times\gamma_1\rightarrow\phi\vdash x:\gamma_2\times\gamma_1\rightarrow\phi \qquad \vdash(\lambda yz.\Omega\mid\lambda z.\Omega):\gamma_2\times\gamma_1}{x:\gamma_2\times\gamma_1\rightarrow\phi\vdash M:\phi}$$

Mais, pour tout $\phi\neq\omega$, on a $\neg(x:\gamma_2\times\gamma_1\rightarrow\phi\vdash N:\phi)$, essentiellement parce que l'argument $(\lambda yz.\Omega)$ dans N ne peut pas avoir un type produit. □

Comme nous l'avons déjà signalé, la raison de l'incomplétude est que le système de types permet de compter le nombre de ressources non-résolubles d'un paquet, mais pas λ_r. Des propriétés similaires sont vérifiées dans le cas de $\mathcal{P}_\bigwedge$; c'est-à-dire, la sémantique abstraite engendrée par la notion de typage est adéquate; le même exemple sert à montrer un résultat d'incomplétude.

Les points principaux que nous étudions sont :

Correspondance entre le domaine $\mathcal{C}$ et le système de types $\mathcal{P}$, et entre $\mathcal{F}$ et $\mathcal{P}_\bigwedge$. Les fonctions d'interprétation des λ_r-termes sont étendues à λ_r^c : $[\![\mathbf{c}M]\!]_\rho$ est $[\![\mathbf{I}]\!]_\rho$ si $[\![M]\!]_\rho\neq\bot$, et $\bot$ sinon. Nous étendons aussi les systèmes de types, avec deux règles, l'une pour typer les termes construits avec le test de convergence, l'autre, appelée $\leq$-règle, pour garantir que l'ensemble de types qu'on peut affecter à un terme est un cône supérieur dans le cas de $\mathcal{P}$, et un filtre dans le cas de $\mathcal{P}_\bigwedge$:

$$\frac{\Gamma\vdash M:\omega\rightarrow\omega}{\Gamma\vdash\mathbf{c}M:\phi\rightarrow\phi} \qquad \frac{\Gamma\vdash T:\tau}{\Gamma\vdash T:\tau'}\,\tau\;\leq\;\tau'$$

En fait, $\omega\rightarrow\omega$ est le plus grand type des termes convergents et $\phi\rightarrow\phi$ celui de l'identité. Par ailleurs, T est soit un terme soit un paquet de termes. Ces extensions sont définies dans les sections 5.1 et 5.2 pour le domaine $\mathcal{C}$, et dans la section 6.1 pour ce qui est de $\mathcal{F}$.

Les systèmes de types $\mathcal{P}$ et $\mathcal{P}_\bigwedge$ vérifient quelques propriétés importantes, comme la *stabilité par expansion* et une *propriété d'extensionnalité* qui sont montrées dans les sections 5.3 et 6.1 respectivement.

La sémantique abstraite engendrée par le système de types $\mathcal{P}$ est :

$$M\sqsubseteq_{\mathcal{P}} N \overset{def}{\Leftrightarrow} \forall\Gamma\,\forall\phi\;\;(\Gamma\vdash M:\phi\;\Rightarrow\;\Gamma\vdash N:\phi)$$

et similairement pour $\mathcal{P}_\bigwedge$, dont le préordre est noté $\sqsubseteq_{\mathcal{P}_\bigwedge}$. Nous terminons par les preuves de la caractérisation du préordre dans les domaines $\mathcal{C}$ et $\mathcal{F}$ par ceux dans les systèmes de types. Dans les section 5.4 et 6.1 nous montrons, respectivement,

$$\forall\,\lambda_r^c\text{-termes } M,N\;\;(M\sqsubseteq_{\mathcal{C}} N\Leftrightarrow M\sqsubseteq_{\mathcal{P}} N)\;\;\&\;\;(M\sqsubseteq_{\mathcal{F}} N\Leftrightarrow M\sqsubseteq_{\mathcal{P}_\bigwedge} N)$$

Adéquation des modèles $\mathcal{C}$ et $\mathcal{F}$ pour λ_r^c. En utilisant les caractérisations du point précédent, les résultats d'adéquation sont montrés sur les systèmes de types, i.e. :

$$\forall\, \lambda_r^c\text{-terme } M, N \ \ (M \sqsubseteq_{\mathcal{P}} N \ \Rightarrow \ M {\sqsubseteq}_{rc} N) \ \ \& \ \ (M \sqsubseteq_{\mathcal{P}_\wedge} N \ \Rightarrow \ M {\sqsubseteq}_{rc} N)$$

Les preuves sont conséquence de la caractérisation, dans le système de types, de la notion d'observation :

$$\forall \lambda_r^c\text{-terme } M \ \ (M{\Downarrow}_{rc} \Leftrightarrow \ \vdash M : \omega \to \omega)$$

La partie $\Rightarrow$ est conséquence de la stabilité du typage par expansion, car les abstractions ont toutes le type $\omega \to \omega$. La partie $\Leftarrow$ est prouvée par la technique de *réalisabilité* : on interprète les types par des ensembles de termes, et on montre que l'interprétation est close par rapport au préordre observationnel (ici, par exemple, on utilise le lemme des contextes pour λ_r^c afin de simplifier les preuves), et correcte par rapport au typage dans le système correspondant. Ceci est fait dans la section 5.5 pour le système de types $\mathcal{P}$; la preuve est une extension de celle de [17] (qui est pour λ_r et dont le système ne contient pas $\leq$-règle, qui n'est en fait nécessaire que si l'on s'intéresse à la complétude de la sémantique). En ce qui concerne $\mathcal{P}_\wedge$, la preuve se trouve dans la section 6.1. La construction de l'interprétation des types est assez délicate dans ce cas, car il faut éviter de confondre un paquet unitaire, i.e. composé d'une unique ressource M, avec le terme M proprement dit.

Complétude du modèle $\mathcal{C}$ pour λ_r^c. Conjecture sur $\mathcal{F}$. Nous montrons dans la section 5.6 que le modèle de cônes est complètement adéquat pour le calcul de ressources avec test de convergence, par la *technique de définissabilité des points compacts* d'Abramsky [2, 3]. L'idée est de construire des *termes ou paquets caractéristiques* pour chaque point compact du domaine (les analogues des filtres principaux dans les théories usuelles), qui sont utilisés dans la preuve de *complétude de l'interprétation des types par rapport à la notion de typage.* Une autre propriété importante utilisée est l'extensionnalité du typage.

Nous ne détaillerons pas ici la construction des termes caractéristiques (cf. 5.6.1), qui est assez délicate, mais il faut souligner que le test de convergence y joue un rôle majeur. Rappelons cependant que les exemples 1.2.1 et 1.2.4 montrent que les raisons de l'incomplétude dans le calcul avec ressources ne sont pas les mêmes que dans le lambda calcul faible.

En ce qui concerne le modèle de filtres, la technique de définissabilité n'est pas appropriée : le calcul de ressources n'a pas la capacité de tester si un paquet d'arguments a un type intersection (conjonction) car il faudrait pouvoir dupliquer les ressources du paquet. Nous discuterons d'une extension possible du calcul qui incorpore un opérateur de duplication (appelé aussi produit synchrone). Néanmoins, notre conjecture est que le modèle de filtres est aussi complètement adéquat pour λ_r^c. La raison essentielle serait que la manipulation des hypothèses dans le système avec conjonction n'autorise pas de contractions.

Plan de la thèse

Le **chapitre 2** introduit les calculs de base, λ et π, et des extensions standards de λ qui trouvent leur motivation dans la théorie des domaines : avec test de convergence, choix non-déterministe, appel par valeur. On étudie les relations entre ces calculs décrites dans la section 1.1. Le **chapitre 3** est consacré à l'étude syntaxique des calculs avec ressources. On définit λ_r, λ_m, le calcul avec paquets et λ_r^c. Plusieurs propriétés de la sémantique observationnelle standard sont données (pour λ_r^c). On caractérise la théorie associée à λ_r, on construit sa sémantique algébrique et on étudie son pouvoir de discrimination et son expressivité. Le **chapitre 4** aborde l'étude des modèles de λ_r. On introduit les définitions et notations de base de la théorie des domaines utilisée. On définit deux équations avec leurs fonctions d'interprétation et on construit leurs solutions canoniques $\mathcal{C}$ et $\mathcal{F}$. Le **chapitre 5** est consacré à la preuve d'adéquation complète de $\mathcal{C}$ par rapport à λ_r^c. On définit le système de types $\mathcal{P}$. On montre aussi que les ressources ne discriminent pas plus que les multiplicités si l'on considère les λ-termes. Le **chapitre 6** concerne l'adéquation de $\mathcal{F}$ par rapport à λ_r^c. On définit le système de types $\mathcal{P}_\wedge$. Le **chapitre 7** contient quelques remarques finales et problèmes ouverts.

Chapitre 2

Lambda calculs non-déterministes et π-calcul

L'étude de la relation entre le lambda calcul faible d'Abramsky et le calcul de processus π de Milner, Parrow et Walker [53] débute par le codage de λ dans π proposé par Milner [54]. Dans le cadre de la bisimulation applicative pour le lambda calcul et du préordre contextuel pour le π-calcul - que nous définirons dans la section 2.1, Milner démontre que le codage est adéquat. Plus précisément, le π calcul est strictement plus discriminant que le lambda calcul[1]. D'une part, il existe des processus du π-calcul qui se comportent, sur le codage, comme les combinateurs de test de convergence (parallèle), qui ne sont pas définissables dans le lambda calcul faible. D'autre part, comme le souligne Boudol [17], le type de codage utilisé ne valide pas l'égalité $xx = x(\lambda y.xy)$, pourtant vérifiée dans les modèles du lambda calcul faiblement extensionnels (où $M \neq \Omega \Rightarrow M = \lambda x.(Mx)$, pourvu que x ne soit pas une variable libre de M). Ceci indique que les extensions classiques du lambda calcul faible - avec fonctions parallèles, choix non-déterministe, test de convergence (parallèle), motivées principalement par le problème d'adéquation complète du modèle obtenu à partir de l'équation aux domaines $D = (D \to D)_\perp$ [2, 3, 14, 15], ne peuvent pas être codées de façon complètement adéquate dans π. Cette affirmation peut sembler contradictoire avec le résultat de Sangiorgi [72] sur la caractérisation de l'égalité engendrée par le codage de Milner : Sangiorgi montre que ce qui manque au lambda calcul est une forme de non-déterminisme. Mais ce résultat est basé sur une notion de bisimulation applicative étendue, suffisamment intensionnelle pour distinguer $\lambda y.xy$ de x. Nous reviendrons sur ce point dans les sections 2.4 et 3.8.1.

Le but de ce chapitre est d'identifier des extensions du lambda calcul faible susceptibles d'être codées de façon adéquate dans le π-calcul.

La section 2.1 est un récapitulatif des notions de base du λ-calcul et du π-

---

[1] Un résultat similaire est montré pour le calcul avec appel par valeur de Plotkin [65]. Thomsen [79, 80] définit des codages similaires dans son calcul de processus CHOCS, où les processus sont des objets de première classe qui peuvent être passés comme arguments.

calcul où l'on introduit notamment les conventions de notation qui seront utilisées par la suite.

Dans la section 2.2 on fait une étude comparative de divers raffinements du lambda calcul faible, selon l'approche de Boudol [14, 15] qui consiste à séparer le test de convergence (ou l'appel par valeur) des fonctions parallèles (ou non-déterministes). Nous menons cette étude dans le cadre de la théorie de l'expressivité des langages de programmation proposée par Mitchell [56]. On montre en particulier que les fonctions parallèles et le choix non-déterministe sont mutuellement transposables, ainsi que le test de convergence et l'appel par valeur, considérés comme extensions du calcul faible. L'ajout de nouveaux constructeurs à un langage de programmation peut modifier la théorie équationnelle engendrée par la notion d'observation. C'est ce qui arrive lorsque l'on incorpore soit les fonctions parallèles soit le choix non-déterministe au calcul faible avec test de convergence ou avec appel par valeur. Malgré les simulations entre constructeurs dont nous avons parlé, leurs combinaisons ne sont pas toutes transposables. En effet, la combinaison de l'appel par valeur et du choix non-déterministe ne définit pas un calcul intéressant (dans le cadre du lambda calcul faible, avec la sémantique contextuelle à la Morris); cependant, le calcul faible avec test de convergence et choix non-déterministe (noté λ_j) a un modèle complètement adéquat [14]. Nous concluons que ce calcul est bien adapté pour être codé dans π (les calculs avec fonctions parallèles sont écartés en raison des propriétés du codage utilisé). Il faut souligner que la théorie de Mitchell, où l'on peut composer des simulations, ne dit rien sur la relation entre deux langages mutuellement transposables lorsque ces langages sont augmentés par des nouveaux constructeurs. Il serait certainement intéressant de chercher à développer une théorie de l'expressivité qui considère l'extension des langages. Pour compléter le panorama du non-determinisme dans le lambda calcul, nous présentons, à la fin de la section, et sous une forme schématique, d'autres extensions, qui repose sur les versions classique, faible et typée du calcul.

La section 2.3 est consacrée au codage du calcul λ_j dans le π-calcul, qui est montré adéquat. En fait, le codage repose sur une présentation de λ_j avec des substitutions semi-explicites; i.e. les abstractions et les variables sont des clôtures.

Dans la section 2.4 on revient sur les raisons de l'incomplétude des codages des lambda calculs.

2.1 Préliminaires

2.1.1 Le λ-calcul faible

Nous traiterons dans cette thèse plusieurs extensions du lambda calcul faible défini par Abramsky [2], noté ici λ. Nous commençons par fixer les notions de base et la notation utilisée.

Soit Var un ensemble infini dénombrable de *variables* $x, y, z, \ldots$. Les termes du λ calcul sont de trois types, à savoir : variables, abstractions et applications. La syntaxe des termes est donnée par

$$(\Lambda) \qquad M \ ::= \ x \,|\, (\lambda x.M) \,|\, (MN)$$

On note $L, M, N, \ldots$ les termes du langage et on utilise $\tilde{N}$ pour désigner la séquence $N_1 \ldots N_n$ lorsqu'il n'y a pas d'ambiguïté sur la valeur de n; on note $|\tilde{N}|$ la taille n de la séquence. Nous suivons la convention usuelle concernant l'écriture des λ-termes pour éviter l'utilisation systématique de parenthèses.

- L'application (MN) peut s'écrire MN quand il n'y a pas d'ambiguïté. L'expression $MN_1 \ldots N_k$ est une abréviation de $(\ldots((MN_1)N_2)\ldots N_k)$. C'est ce qu'on appelle une association vers la gauche.
- Pour les abstractions, nous adoptons l'association vers la droite : l'expression $\lambda x_1 \ldots x_n.M$ est une notation pour le terme $(\lambda x_1(\lambda x_2(\ldots(\lambda x_n.M)\ldots)))$.

Des termes (combinateurs) utilisés souvent sont

$$\mathbf{I} = \lambda x.x \quad K = \lambda xy.x \quad F = \lambda xy.y$$

$$\Delta = \lambda x.xx \quad \Omega = \Delta\Delta$$

L'ensemble des *variables libres* d'un terme M, $fv(M)$, est défini par

$$fv(x) = \{x\} \quad fv(\lambda x.M) = fv(M) - \{x\} \quad fv(MN) = fv(M) \cup fv(N)$$

Les *variables liées* de M, $bv(M)$, sont

$$bv(x) = \emptyset \quad bv(\lambda x.M) = bv(M) \cup \{x\} \quad bv(MN) = bv(M) \cup bv(N)$$

On étend la notation aux ensembles de termes : $fv(M_1, \ldots, M_k)$ dénote les variables libres de l'ensemble $\{M_1, \ldots, M_k\}$, $bv(M_1, \ldots, M_k)$ ses variables liées, et $var(M_1, \ldots, M_k)$ représente l'union $fv(M_1, \ldots, M_k) \cup bv(M_1, \ldots, M_k)$. Un terme M est *clos* si $fv(M) = \emptyset$; M est *ouvert* sinon. On note Λ^o l'ensemble des termes clos de Λ.

Nous utiliserons la notion usuelle de *contexte* basée sur les constructeurs des termes plus une constante []. La syntaxe des λ-contextes est donnée par

$$C \ ::= \ [\,] \,|\, x \,|\, (\lambda x.C) \,|\, (CC)$$

On note $C[M]$ le terme qui résulte du remplacement des occurrences de [] dans C par M. Remarquons que dans le terme $C[M]$, des variables libres de M peuvent être liées par C. Par exemple, si $C = \lambda x.[\,]$ et $M = x$. On dira que le contexte C *ferme* le terme M si $C[M]$ est un terme clos. Les lettres $A, B, C, D, E, \ldots$

désigneront le plus souvent des contextes. Les *contextes à plusieurs trous* sont construits avec des constantes $[]_i$. On adopte la convention que pour tout contexte C à plusieurs trous il existe n tel que $[]_1, \ldots, []_n$ sont exactement les constantes de C. On note $C[M_1, \ldots, M_n]$ le terme obtenu par remplacement de $[]_i$ par M_i dans C.

La substitution de x par N dans M, notée $M[N/x]$, est définie par :

$$\begin{aligned}
y[N/x] &= \begin{cases} N & \text{si } y = x \\ y & \text{sinon} \end{cases} \\
(\lambda y.M)[N/x] &= \begin{cases} \lambda y.M & \text{si } y = x \\ \lambda z.(M[z/y])[N/x] & \text{sinon, où } z \notin fv(M,N) \end{cases} \\
(MM')[N/x] &= (M[N/x])(M'[N/x])
\end{aligned}$$

On dit que deux termes M, N sont égaux à α-conversion près si $M{=_\alpha}N$, où $=_\alpha$ est la congruence engendrée par

$$\lambda x.M = \lambda z.M[z/x] \quad \text{avec } z \notin var(M)$$

La β-conversion de termes s'écrit

$$(\lambda x.M)N \rightarrow M[N/x]$$

Un sous-terme de L de la forme $(\lambda x.M)N$ est un *radical*[2] de L. Par ailleurs, nous supposons que les notions de *forme normale (fn)*, *forme normale de tête (fnt)* et *forme normale de tête faible (fntf)* et aussi de *réduction normale (réduction du radical le plus à gauche)* sont connues du lecteur[3] (cf. [8]). Les fntf sont les abstractions du langage. Sans entrer dans les détails des théories du lambda calcul, on écrit $M{=_\beta}N$ si M et N sont β-convertibles. Nous rappelons qu'un terme M est *résoluble* ssi $\exists \tilde{N}\ M\tilde{N}{=_\beta}\mathbf{I}$.

Fait 2.1.1 *(Caractérisation de Wadsworth, voir [8])*
Pour tout $M \in \Lambda$, M est résoluble ssi M a une fnt.

Par contraposition, les termes sans fnt sont les termes *non-résolubles*. Dans les théories *raisonnables*[4] du lambda calcul [8], tous les termes non-résolubles sont identifiés et donc en particulier $\lambda x.\Omega = \Omega$. Du point de vue calculatoire, la théorie raisonnable correspond à la stratégie de réduction normale, qui arrête une évaluation face à une fnt; ces termes constituent les *valeurs* ou *termes observables* du calcul. Cependant, beaucoup des langages de programmation basés sur le lambda

---

[2] redex

[3] normal form, head normal form, weak normal form, leftmost-outermost reduction

[4] sensible

calcul, n'utilisent pas cette notion de résultat : les termes $\lambda x.\Omega$ et Ω constituent des programmes essentiellement différents. L'évaluation dans ces langages est *faible*; c'est-à-dire, correspond à une stratégie de réduction normale qui s'arrête face à une forme normale de tête faible. La relation d'évaluation correspondante, $\rightarrow_\lambda$, est définie par les règles de la figure 2.1. Remarquons que ni les corps des abstractions ni les arguments des applications ne sont évalués. On utilisera $\overset{*}{\rightarrow}_\lambda$ pour la clôture reflexive et transitive de $\rightarrow_\lambda$ et $\overset{+}{\rightarrow}_\lambda$ pour sa clôture transitive. De plus, on écrira $M \not\rightarrow_\lambda$ lorsqu'il n'existe pas N tel que $M \rightarrow_\lambda N$.

$$(\beta)\ (\lambda x.M)N \rightarrow_\lambda M[N/x] \qquad \frac{N \rightarrow_\lambda N' \quad M =_\alpha N}{M \rightarrow_\lambda N'} \qquad \frac{M \rightarrow_\lambda M'}{MN \rightarrow_\lambda M'N}$$

Fig. 2.1 - *Évaluation faible du λ calcul*

La description alternative de la stratégie d'évaluation dans le style de la déduction naturelle [65, 2], présentée dans la figure 2.2, définit le *prédicat de convergence* $\Downarrow_\lambda$ sur les termes clos.

$$\frac{}{\lambda x.M \Downarrow_\lambda \lambda x.M} \qquad \frac{M \Downarrow_\lambda \lambda x.M' \quad M'[N/x] \Downarrow_\lambda L}{MN \Downarrow_\lambda L}$$

Fig. 2.2 - *Prédicat de convergence $\Downarrow_\lambda$*

Comme nous l'avons souligné, les *valeurs* du calcul faible, notées $V, W \ldots$ sont

$$\mathbb{V}_\lambda = \{\lambda x.M \,/\, M \in \Lambda\}$$

Soit $M \in \Lambda^o$, on dit que M *converge*, noté $M \Downarrow_\lambda$, s'il existe $V \in \mathbb{V}_\lambda$ t.q. $M \Downarrow_\lambda V$. On dit que M *diverge*, noté $M \Uparrow_\lambda$, si $\neg(M \Downarrow_\lambda)$. Dans le cadre du lambda calcul faible, pour tout terme M clos, $M \not\rightarrow_\lambda$ ssi $M = \lambda x.N$; c'est-à-dire, l'ensemble des termes *irréductibles* coïncide avec l'ensemble des valeurs. Par conséquent, un terme M clos est divergent si l'évaluation de M est infinie.

Le critère de convergence est trop superficiel pour engendrer à lui seul une théorie contextuelle, autrement dit, l'égalité définie par $M = N \overset{def}{\Leftrightarrow} (M \Downarrow_\lambda \Leftrightarrow N \Downarrow_\lambda)$ n'est pas une congruence. Par exemple, $K = F$ mais $K\mathbf{I}\Omega \neq F\mathbf{I}\Omega$.

Néanmoins, le prédicat $\Downarrow_\lambda$ est en rapport direct avec la classe $\mathcal{F}$ de termes *observables*, fermée par α- et β-conversion et définie par

$$\mathcal{F} = \{M \in \Lambda^o \,/\, \exists N\ M =_\beta \lambda x.N\}$$

Fait 2.1.2 *(Lemmes 2.1.1.2 et 2.1.2.6 [59])*
Pour tout $M \in \Lambda^o$, $\exists V\ M {\Downarrow_\lambda} V \Leftrightarrow \exists V\ M \xrightarrow{*}_\lambda V \Leftrightarrow M \in \mathcal{F}$.

Le schéma de Morris [57] de *préordre observationnel* sur les λ-termes, spécialisé à $\mathcal{F}$, est

$$M \sqsubseteq_\lambda N \overset{def}{\Leftrightarrow} \forall C\ \ C[M]{\Downarrow_\lambda} \ \Rightarrow\ C[N]{\Downarrow_\lambda}$$

où l'on sous-entend que les termes $C[M]$ et $C[N]$ sont clos. Les résultats généraux sur ce type de préordre sont directement applicables (cf. Chapitre 4 [59]) : $\sqsubseteq_\lambda$ est une pre-congruence donc l'égalité observationnelle associée $M \simeq_\lambda N$, définie par $M \sqsubseteq_\lambda N$ & $N \sqsubseteq_\lambda M$, est une congruence. Par conséquent, $\{M = N \,/\, M, N \in \Lambda^o$ & $M \simeq_\lambda N\}$ est une λ-théorie (qui est contextuelle par définition.)

Il n'est pas surprenant que le calcul faible vérifie un *lemme des contextes* dans la ligne des résultats de Milner [51], Lévy [48] et Berry [10], qui exprime le fait que le comportement d'un terme est caractérisé par son comportement comme fonction. En effet, les *contextes applicatifs* définis par la grammaire

$$A ::= [\,] \mid \lambda x.A \mid (AM)$$

sont assez puissants pour distinguer deux termes M, N t.q. $\neg(M \simeq_\lambda N)$. Le préordre $\sqsubseteq_{\mathcal{A}}$ est défini par

$$M \sqsubseteq_{\mathcal{A}} N \overset{def}{\Leftrightarrow} \forall \text{ contexte applicatif } A\ \ A[M]{\Downarrow_\lambda} \ \Rightarrow\ A[N]{\Downarrow_\lambda}$$

Sur les termes clos, le préordre applicatif coïncide avec la *bisimulation applicative* $\sqsubseteq^B$ proposée par Abramsky, dont la dénomination évoque les travaux de Milner et Park dans le domaine de la théorie de la concurrence. Soit $M, N \in \Lambda^o$,

$$M \sqsubseteq^B N \ \overset{def}{\Leftrightarrow}\ \forall \tilde{P} \in \Lambda^o\ M\tilde{P}{\Downarrow_\lambda} \ \Rightarrow\ N\tilde{P}{\Downarrow_\lambda}$$

Lemme 2.1.3 *(Lemme des contextes [3])*
Pour tout $M, N \in \Lambda^o$, $M \sqsubseteq_{\mathcal{A}} N \Leftrightarrow M \sqsubseteq^B N \Leftrightarrow M \sqsubseteq_\lambda N$.

La caractérisation de la λ-théorie engendrée par $\simeq_\lambda$ [59, 3] repose sur la classification des termes selon leur *degré de fonctionnalité* ou de *non-résolution*[5] [50, 59, 3] :

Définition 2.1.4 *Soit* $M \in \Lambda$.

- *Le degré de fonctionnalité de* M *est* 0, *noté* $M \in O_0$, *si* $\neg(\exists N\ M =_\beta \lambda x.N)$.
- *Le degré de fonctionnalité de* M *est* n, *noté* $M \in O_n$, *si* n *est le plus grand* i *tel que* $\exists N\ M =_\beta \lambda x_1 \ldots x_i.N$.

---

[5] (functionality) order, proper order

- *Le degré de fonctionnalité de* M *est* ∞, *noté* $M \in O_\infty$, *si* $\forall n \in \mathbb{N}\ M \notin O_n$.

- M *est non-résoluble de degré [maximum]* 0, *noté* $M \in PO_0$, *si* $M \in O_0$ *et* $\neg(\exists x\ \exists \tilde{N}\ M = x\tilde{N})$. *Dans ce cas on dit que* M *est* fortement non-résoluble[6].

- M *est non-résoluble de degré [maximum]* n, *noté* $M \in PO_n$, *si* n *est le plus grand* i *tel que* $\exists N \in PO_0\ M =_\beta \lambda x_1 \ldots x_i.N$.

- M *est non-résoluble de degré* ∞, *noté* $M \in PO_\infty$, *si* $M \in O_\infty$.

Par exemple $x\tilde{N} \in \mathrm{O}_0$ et $(\lambda xy.xx)(\lambda xy.xx) \in \mathrm{O}_\infty$. L'ensemble $\mathcal{F}$ des termes observables est donc $\{M \,/\, \exists n > 0\ M \in \mathrm{O}_n\}$. Abramsky et Ong [59, 3] montrent qu'un λ-terme M est non-résoluble ssi $\exists n\ M \in \mathrm{PO}_n$. De plus, $M \in \mathrm{PO}_0$ ssi $\neg(\exists V \in \mathbb{V}_\lambda\ M \overset{*}{\rightarrow}_\lambda V)$. Dans le cas où M est clos, $M \in \mathrm{PO}_0$ ssi $M \Uparrow_\lambda$. Par ailleurs, la théorie engendrée par $\simeq_\lambda$ est "fully lazy" maximale, c'est-à-dire, identifie les termes non-résolubles de même degré maximum : pour $m, n \in \mathbb{N} \cup \infty$,

$$\forall M \in \mathrm{PO}_m\ \forall N \in \mathrm{PO}_n\ \ (M \simeq_\lambda N \Leftrightarrow m = n)$$

Nous terminons la présentation du λ calcul faible avec l'interprétation intensionnelle de Lévy [47], basée sur une spécialisation de la notion d'approximant de Wadsworth [81, 82] et Hyland [45], qui distingue $\lambda x.\Omega$ de Ω. La définition a été prise de [18] :

Définition 2.1.5 *L'ensemble* $\mathcal{L}$ *d'*approximants, *désignés par* $A, B, \ldots$, *est le plus petit sous-ensemble de* Λ *qui contient* $\lambda x_1 \ldots x_n.\Omega$ *et* $\lambda x_1 \ldots x_n.xA_1 \ldots A_m$ *lorsque* $A_i \in \mathcal{L}$. *L'*approximant direct *de* $M \in \Lambda$ *est le terme* $\varpi(M) \in \mathcal{L}$ *défini par récurrence comme suit :*

$$\begin{array}{rcl} \varpi(\lambda\tilde{x}.(\lambda y.M)NM_1 \ldots M_m)) & = & \lambda\tilde{x}.\Omega \\ \varpi(\lambda\tilde{x}.yM_1 \ldots M_m) & = & \lambda\tilde{x}.y\varpi(M_1) \ldots \varpi(M_m) \end{array}$$

L'*interprétation algébrique* de $M \in \Lambda$, appelée aussi interprétation intensionnelle, est $\mathcal{A}(M) = \{\varpi(N) \,/\, M =_\beta N\}$. Le préordre engendré, noté $M \leq_\mathcal{L} N$, est l'inclusion des ensembles des approximants $\mathcal{A}(M) \subseteq \mathcal{A}(N)$. L'égalité $M =_\mathcal{L} N$ est $\mathcal{A}(M) = \mathcal{A}(N)$. Lévy prouve un théorème de continuité syntaxique : $C[M] = \bigcup\{C[A] \,/\, A \in \mathcal{A}(M)\}$. D'où l'interprétation algébrique est adéquate : $M \leq_\mathcal{L} N \Rightarrow M \sqsubseteq_\lambda N$. Cependant, l'implication inverse n'est pas vérifiée. Une présentation alternative de $\leq_\mathcal{L}$ en termes d'arbres de Böhm adaptés pour le calcul faible est définie par Longo dans [50].

Notation 2.1.6 *Les notations que nous avons introduites dans cette section seront reprises dans les définitions des lambda calculs étendus que nous allons étu-*

---

[6] strongly unsolvable

dier, décorées convenablement. En récapitulant, dans le cadre d'un lambda calcul hypothétique λ_h, on aura :

symbole	***signification***
λ_h	*extension du λ calcul faible*
Λ_h	*ensemble des termes*
Λ_h^o	*ensemble des termes clos*
$\mathbb{V}_h$	*ensemble des valeurs*
$\longrightarrow_h$	*évaluation*
$\not\rightarrow_h$	*impossibilité d'évaluer*
$\Downarrow_h$	*prédicat de convergence*
$\Uparrow_h$	*prédicat de divergence*
$\sqsubseteq_h$	*préordre observationnel*
$\simeq_h$	*équivalence observationnelle*

L'ensemble des variables sera toujours désigné par Var; les ensembles des variables libres, des variables liées et de toutes les variables d'un terme seront notés à l'aide des fonctions $fv, bv, var : \Lambda_h \to Var$ comme pour le calcul λ. Nous donnerons rarement la définition de contexte pour chaque langage; on sous-entend qu'ils sont construits selon la syntaxe des termes plus la constante $[]$ (ou les constantes $[]_i$). Le préordre observationnel $\sqsubseteq_h$ et la congruence $\simeq_h$ ne seront pas explicités non plus. Par ailleurs, les notations $\sqsubseteq_{\mathcal{A}}$ et $\simeq_{\mathcal{A}}$ utilisées dans le cadre d'un lambda calcul λ_h désigneront toujours le préordre et l'équivalence applicatifs associés à $\sqsubseteq_h$ et $\simeq_h$ respectivement.

2.1.2 Le π-calcul

Le calcul π de Milner, Parrow et Walker [53] est une extension de CCS [52] basée sur la communication de canaux et la création de canaux privés. Nous nous limitons ici a présenter le mini π-calcul [54] (cf. [16] pour une version asynchrone du calcul, où l'on peut aussi coder le lambda calcul); les processus du langage sont définis par la grammaire

$$(\Pi) \qquad P ::= \mathbf{0} \,|\, \overline{x}z.P \,|\, x(y).P \,|\, (P|P) \,|\, !P \,|\, (\nu y)P$$

où $x, y, z \in \mathcal{N}$, un ensemble dénombrable de noms de canaux. La définition complète de π inclut en plus la somme $P + Q$, ou choix non-déterministe, et le "matching " $[x = v]P$. Il existe aussi des versions "polyadiques" du calcul [55], où les *gardes d'entrée* et de *sortie* sont de la forme $x(y_1, \ldots y_n)$ et $\overline{x}y_1 \ldots y_n$ respectivement.

La syntaxe des **π-contextes** est donnée par :

$$C ::= [] \,|\, \mathbf{0} \,|\, \overline{x}z.C \,|\, x(y).C \,|\, (C|C) \,|\, !C \,|\, (\nu y)C$$

On note $\overline{\mathcal{N}}$ l'ensemble de noms de canaux surlignés et on utilise n pour dénoter des éléments de $\mathcal{N} \cup \overline{\mathcal{N}}$. L'opération de surlignement est idempotente, c'est-à-dire, $\overline{\overline{n}} = n$. On dit que x est le *nom de sortie* du processus $\overline{x}z.P$ et que $x(y)P$ a x comme *nom d'entrée*. Dans ces cas P constitue le *corps* du processus et $\overline{x}z$ (resp. $x(y)$) est sa *garde*. Les constructions restantes, à savoir $|$, $!$ et (νy), sont la composition parallèle, la réplication et la restriction sur y. Le préfixe d'entrée $x(y)$ et la restriction (νy) lient y. On note $bn(P)$ et $fn(P)$ les ensembles des noms liées et des noms libres de P respectivement, définis comme d'habitude. L'ensemble de tous les noms qui apparaissent dans P est noté $n(P)$. Dans la plupart des cas, les processus $\overline{x}z.\mathbf{0}$ et $x(y).\mathbf{0}$ sont notés $\overline{x}z$ et $x(y)$ respectivement. Désormais, on emploie les mots "processus" et "terme" indistinctement.

Nous présentons le système de réduction du π calcul en suivant l'approche de la *machine chimique* (CHAM) [7] [11], dans la figure 2.3. Les processus sont interprétés par des *solutions* (multi-ensembles) qui contiennent des *molécules* (composantes du processus). La notation pour les solutions est $\{\!\mid m_1, \ldots, m_k \mid\!\}$. La machine est composée de quelques lois générales de calcul et de deux types de règles entre solutions: irréversibles et structurelles. Il y a une unique règle irréversible, $\mapsto$ - appelée aussi *règle de réaction* - qui décrit la communication entre processus, et des règles structurelles, notées $\rightleftharpoons$, qui servent à définir la signification des opérateurs. Pour plus de clarté dans la définition, nous omettons les symboles externes $\{\!\mid$ et $\mid\!\}$.

$x(y).P, \overline{x}z.Q$	$\mapsto$	$P[z/y] \mid Q$	$((\nu x)P \mid Q)$	$\rightleftharpoons$	$(\nu x)(P \mid Q)$	si $x \notin fn(Q)$
$P \mid Q$	$\rightleftharpoons$	P, Q	$(\nu x)P$	$\rightleftharpoons$	$(\nu y)P[y/x]$	si $y \notin fn(P)$
$!P$	$\rightleftharpoons$	$P, !P$	$(\nu x)P$	$\rightleftharpoons$	P	si $x \notin fn(P)$
$(\nu x)P$	$\rightleftharpoons$	$(\nu x)\{\!\mid P \mid\!\}$	$(\nu x)(\nu y)P$	$\rightleftharpoons$	$(\nu y)(\nu x)P$	
$\mathbf{0}$	$\rightleftharpoons$	$\emptyset$				

FIG. 2.3 - *Réduction dans le π-calcul*

Les lois générales autorisent des calculs dans les sous-solutions d'une solution (sous les restrictions dans notre cas) ainsi que dans les solutions composantes S_i d'une solution $S_1 \uplus S_2$, où $\uplus$ est l'union de multi-ensembles. Les notions d'α-conversion et de substitution sont standards. La clôture reflexive et transitive de $\rightleftharpoons$, $\overset{\star}{\rightleftharpoons}$, sera notée $\equiv$. La clôture transitive est $\overset{+}{\rightleftharpoons}$.

---

[7] chemical abstract machine

Par exemple, soit P le processus $x(z)\overline{z}b \,|\, (\nu a)(\overline{x}a \,|\, x(y)\overline{y}r)$. Alors

$$\begin{array}{rcl}
\{\!| P |\!\} & \equiv & \{\!| x(z)\overline{z}b, (\nu a)\{\!| \overline{x}a, x(y)\overline{y}r |\!\} |\!\} \\
 & \mapsto & \{\!| x(z)\overline{z}b, (\nu a)\{\!| \overline{a}r |\!\} |\!\} \\
 & \equiv & \{\!| x(z)\overline{z}b \,|\, (\nu a)\overline{a}r |\!\}
\end{array}$$

Par ailleurs,

$$\begin{array}{rcl}
\{\!| P |\!\} & \equiv & \{\!| (\nu a)\{\!| x(z)\overline{z}b, \overline{x}a, x(y)\overline{y}r |\!\} |\!\} \\
 & \mapsto & \{\!| (\nu a)\{\!| \overline{a}b, x(y)\overline{y}r |\!\} |\!\} \\
 & \equiv & \{\!| (\nu a)\overline{a}b \,|\, x(y)\overline{y}r |\!\}
\end{array}$$

Remarquons que tout processus a une solution associée, ce qui justifie l'abus de notation qui consiste à utiliser $\mapsto$ et $\rightleftharpoons$ sur des processus.

Définition 2.1.7 *La relation de* réduction *entre processus, $P \rhd Q$, est définie par :*

$$P \rhd Q \overset{def}{\Leftrightarrow} P \overset{\star}{\rightleftharpoons}\mapsto\overset{\star}{\rightleftharpoons} Q$$

Le critère de convergence est la capacité de communiquer avec l'environnement à travers une entrée ou une sortie, possiblement après quelques étapes de réduction.

Définition 2.1.8 *La convergence immédiate $\downarrow n$ où $n \in \mathcal{N} \cup \overline{\mathcal{N}}$ est définie comme suit :*

$$\begin{array}{r} x(y).P \downarrow x \\ \overline{x}z.P \downarrow \overline{x} \end{array} \qquad P \downarrow n \;\Rightarrow\; \left\{ \begin{array}{r} (P|Q) \downarrow n \\ (Q|P) \downarrow n \\ !P \downarrow n \end{array} \right. \qquad P \downarrow n \;\&\; n \neq y \;\Rightarrow\; (\nu y)P \downarrow n$$

Nous écrivons $P {\Downarrow}(Q, n)$ lorsque $P \overset{\star}{\rhd} Q \;\&\; Q \downarrow n$. Le prédicat de convergence $\Downarrow_\pi$ et le préordre observationnel $\sqsubseteq_\pi$ sont définis par :

$$P{\Downarrow_\pi} \overset{def}{\Leftrightarrow} \exists Q \; \exists n \in \overline{\mathcal{N}} \cup \mathcal{N} \;.\; P{\Downarrow}(Q, n)$$

$$P \sqsubseteq_\pi Q \overset{def}{\Leftrightarrow} \forall \; \pi\text{-}contexte \; C \,.\, C[P]{\Downarrow_\pi} \;\Rightarrow\; C[Q]{\Downarrow_\pi}$$

La congruence engendrée par $\sqsubseteq_\pi$ est $\simeq_\pi$.

En particulier on a $P \simeq_\pi Q \;\&\; Q{\Downarrow_\pi} \;\Rightarrow\; P{\Downarrow_\pi}$, et aussi $P \rhd Q \;\&\; Q{\Downarrow_\pi} \;\Rightarrow\; P{\Downarrow_\pi}$.

Le système des transitions étiquetées du π-calcul est présenté ici comme une relation entre solutions. Les étiquettes α du système sont les entrées $x(y)$, les sorties libres ou liées $\overline{x}z$ et $\overline{x}(w)$, ou encore la transition muette τ. On utilise $n(\alpha)$, $fn(\alpha)$ et $bn(\alpha)$, de même que sur les processus.

$$\alpha = x(y) \text{ et } \left\{ \begin{array}{l} S \equiv (\nu\vec{u})\{\!\mid x(y)P, \ldots \mid\!\} \\ S' \equiv (\nu\vec{u})\{\!\mid P, \ldots \mid\!\} \end{array} \right. \qquad \text{si } x \notin \vec{u} \text{ et } y \text{ dans } P \text{ seulement}$$

$$\alpha = \overline{x}z \text{ et } \left\{ \begin{array}{l} S \equiv (\nu\vec{u})\{\!\mid \overline{x}z.P, \ldots \mid\!\} \\ S' \equiv (\nu\vec{u})\{\!\mid P, \ldots \mid\!\} \end{array} \right. \qquad \text{si } x, z \notin \vec{u}$$

$$\alpha = \overline{x}(z) \text{ et } \left\{ \begin{array}{l} S \equiv (\nu\vec{u})\{\!\mid \overline{x}z.P, \ldots \mid\!\} \\ S' \equiv (\nu\vec{u'})\{\!\mid P, \ldots \mid\!\} \end{array} \right. \qquad \text{si } x \notin \vec{u},\ z \in \vec{u} \text{ et } \vec{u'} = \vec{u} - z$$

$$\alpha = \tau \text{ si } S \rhd S'$$

FIG. 2.4 - *Transitions étiquetées*

Soit S, S' deux solutions; nous écrivons $S \xrightarrow{\alpha} S'$ si et seulement si un des cas définis dans la figure 2.4 est vérifié.

Définition 2.1.9 *Une relation $\mathcal{R}$ entre solutions est une* bisimulation forte[8] *si $(S, T) \in \mathcal{R}$ implique*

- *si $S \xrightarrow{\alpha} S'$ et α est τ, $\overline{x}z$ ou $\overline{x}(y)$, alors il existe T' t.q. $T \xrightarrow{\alpha} T'$ et $(S', T') \in \mathcal{R}$*
- *si $S \xrightarrow{x(y)} S'$, alors il existe T' t.q. $T \xrightarrow{x(y)} T'$ et, $(S'[w/y], T'[w/y]) \in \mathcal{R}$ pour tout w.*

Deux solutions S et T sont fortement bisimilaires ou équivalentes par bisimulation, noté $S \sim_\pi T$, lorsqu'il existe une bisimulation forte $\mathcal{R}$ qui vérifie $(S, T) \in \mathcal{R}$.

Nous utiliserons aussi la notion de *bisimulation forte à $\sim_\pi$ près*, suffisante pour prouver l'équivalence de deux solutions, dont la définition demande seulement $(S', T') \in \sim_\pi \mathcal{R} \sim_\pi$. En effet, si (S, T) est dans une bisimulation forte à $\sim_\pi$ près $\mathcal{R}$, alors $S \sim_\pi T$. Pour une description exhaustive de ces notions cf. [52].

La correspondance directe entre solutions et termes du calcul nous permet de parler de transitions étiquetées d'un terme et d'équivalences par bisimulation entre termes. La relation entre $\sim_\pi$ et $\simeq_\pi$ est la suivante:

Proposition 2.1.10 *Pour toute paire de processus $P, Q \in \Pi$,*

1. $\{\!\mid P \mid\!\} \sim_\pi \{\!\mid Q \mid\!\} \;\Rightarrow\; P \simeq_\pi Q$

---

[8] strong bisimulation

2. $P \simeq_\pi Q \not\Rightarrow \{\!| P |\!\} \sim_\pi \{\!| Q |\!\}$

Preuve.

1. Soit $\{\!| P |\!\} \sim_\pi \{\!| Q |\!\}$ et $C[P]\Downarrow_\pi$. On montre $C[Q]\Downarrow_\pi$ par récurrence sur C.

   - $C = []$: Rappelons que $P\Downarrow_\pi \Leftrightarrow \exists P', n \,.\, P \overset{\star}{\rhd} P' \downarrow n$ et aussi que $P \overset{\star}{\rhd} P' \Leftrightarrow \{\!| P |\!\} \overset{\star}{\rhd} \{\!| P' |\!\}$. Puisque $\{\!| P |\!\} \sim_\pi \{\!| Q |\!\}$, il existe Q' t.q. $\{\!| Q |\!\} \overset{\star}{\rhd} \{\!| Q' |\!\}$ et $\{\!| P' |\!\} \sim_\pi \{\!| Q' |\!\}$. Dans le cas où $n \in \mathcal{N}$, on a $P' \equiv (\nu\vec{u})(n(y)P_0|P_1)$ (ou simplement $P' \equiv (\nu\vec{u})n(y).P_0$) avec $n \notin \vec{u}$. Donc $\{\!| P' |\!\} \xrightarrow{n(y)}$. Si $n \in \overline{\mathcal{N}}$, alors $P' \equiv (\nu\vec{u})(\overline{n}y.P_0|P_1)$ (ou $P' \equiv (\nu\vec{u})\overline{n}y.P_0$). Par conséquent, on a soit $\{\!| P' |\!\} \xrightarrow{\overline{n}(y)}$ soit $\{\!| P' |\!\} \xrightarrow{\overline{n}y}$, selon que $y \in \vec{u}$ ou pas.
   - Le cas de récurrence s'ensuit directement de l'hypothèse de récurrence et de la propriété de congruence de $\simeq_\pi$.

2. L'exemple suivant montre deux processus P et Q qui diffèrent dans le nombre de transitions muettes dont ils sont capables :

$$P = (\nu x)(x(y).x(z).u(t).\mathbf{0} \mid (\nu y)(\nu z)\overline{x}y.\overline{x}z.\mathbf{0})$$

$$Q = (\nu x)(x(y).u(t).\mathbf{0} \mid (\nu y)\overline{x}y.\mathbf{0})$$

□

On définit Com(P) comme l'ensemble de canaux à travers lesquels P peut communiquer avec l'environnement. C'est-à-dire,

$$\mathrm{Com}(P) = \{n \in \mathcal{N} \cup \overline{\mathcal{N}} \;/\; P \downarrow n\}$$

Un processus P est *stable* ssi une des conditions suivantes est vérifiée :

- $P = \overline{u}z.Q$ ou $P = u(x).Q$, ou
- $P = (\nu z)Q$ ou $P = !Q$ avec Q stable, ou
- $P = (Q|R)$ avec Q et R stables et $\mathrm{Com}(Q) \cap \overline{\mathrm{Com}(R)} = \emptyset$

Proposition 2.1.11 *Soit P un processus stable.*

1. $\neg(\exists n.\, \{n, \overline{n}\} \subseteq Com(P))$

2. $\neg(\exists P'.\, P \rhd P')$

Preuve. Le premier énoncé est évident. On prouve le deuxième par récurrence sur P. Lorsque P est un processus avec une seule composante (d'entrée ou de sortie) ou un processus gardé par une restriction, l'énoncé est immédiat. Dans le cas où P est une composition parallèle $(P_1 \mid P_2)$, par définition les canaux visibles de P_1 et P_2 ont une intersection vide, ce qui empêche toute communication. Pour $P = !Q$ avec Q stable, on suppose qu'il existe P' t.q. $P \triangleright P'$. C'est-à-dire, il existe Q' t.q.

- ou bien $Q \triangleright Q'$ et $!Q \rightleftharpoons (Q|!Q) \triangleright (Q'|!Q)$

- ou bien $(Q|Q) \triangleright Q'$, où Q' est le résultat d'une communication entre deux copies de Q, et $!Q \rightleftharpoons (Q|Q|!Q) \triangleright (Q'|!Q)$

La première possibilité contredit l'hypothèse de récurrence puisque Q est stable. La seconde suppose l'existence de $n, \overline{n}$ dans Com(Q), en contradiction avec le point (1) précédent. $\square$

La proposition suivante établit des conditions syntaxiques suffisantes pour une simplification[9] des solutions (des processus), que nous intégrons par la suite dans l'équivalence structurelle $\equiv$.

Proposition 2.1.12 *Soit P stable et Com$(P) = \emptyset$. Alors $(P \mid Q) \sim_\pi Q$ pour tout Q.*

Preuve. La relation $\mathcal{R}$ définie ci-après est une bisimulation forte :

$$\mathcal{R} = \{(S,T) \,/\, S = (P|Q) \,\&\, T = Q \,\&\, P \text{ est stable } \,\&\, \mathrm{Com}(P) = \emptyset\}$$

Soit $(S,T) \in \mathcal{R}$ et $S \xrightarrow{\alpha} S'$. Par définition, $S = (P|Q)$ et $T = Q$. Si la transition α est $\overline{x}y$, $\overline{x}(y)$ ou τ, alors elle provient de Q puisque Com(P) est vide et que des actions muettes ne sont pas possibles par la proposition 2.1.11(2). Par conséquent :

$$\exists Q'.\ Q \xrightarrow{\alpha} Q' \ \&\ S' = (P|Q') \ \&\ T' = Q'$$

On a alors $(S', T') \in \mathcal{R}$.

Lorsque $\alpha = x(y)$, $\forall w.\ (S'[w/y], T'[w/y]) \in \mathcal{R}$ est vérifié grâce à la condition de bord des transitions de ce type, i.e. $y \notin n(P)$. Donc $S'[w/y] = (P|Q'[w/y])$ et $T' = Q'[w/y]$. $\square$

2.2 Extensions du λ-calcul

Nous étudions ici diverses extensions du lambda calcul faible et leurs interrelations sur la base de la théorie de l'expressivité développée par Mitchell [56], où un

---

[9] Cette procédure de simplification sera très utile dans les preuves de préservation des règles de réduction de λ_j par le codage dans π.

langage de programmation $\mathcal{L}_1$ n'est pas plus expressif qu'un langage $\mathcal{L}_2$ s'il existe une fonction θ qui réduit chaque constructeur de $\mathcal{L}_1$ en un constructeur de $\mathcal{L}_2$ essentiellement équivalent. Ces fonctions de traduction θ - "abstraction-preserving reductions" dans la terminologie de [56] - doivent vérifier :

$$\theta : \Lambda_1 \rightarrow \Lambda_2$$

$$(1) \quad \theta C[M] = (\theta C)[\theta M]$$

$$(2) \quad M \simeq_1 N \Leftrightarrow \theta M \simeq_2 \theta N$$

C'est-à-dire, les traductions que l'on veut considérer sont définies sur la syntaxe des termes du langage (afin de vérifier l'équation 1), dans le but d'écarter des traductions via des énumérations de Gödel, qui existent dans la plupart des cas et satisfont l'équation 2. La question de savoir si l'équation 1 permet effectivement d'éliminer les traductions à la Gödel est laissée comme problème ouvert dans [56] [10].

On introduit d'abord les fonctions parallèles, par le moyen de l'opérateur de composition parallèle $\|$, et le choix non-déterministe $\oplus$. On montre qu'il existe une traduction entre $\lambda_{\|}$ et $\lambda_{\oplus}$ qui vérifie 1 et 2. On présente ensuite les lambda calculs faibles avec test de convergence, λ_c, et avec appel par valeur λ_v pour lesquels on prouve la simulation mutuelle des notions de convergence. Les traductions présentées vérifient l'équation 1, et on indique pourquoi l'équation 2 devrait être vérifiée.

2.2.1 Fonctions parallèles

Nous examinons deux formes d'introduction des fonctions parallèles dans le λ-calcul, selon l'approche de Boudol [14, 17]. La première consiste à intégrer au langage des termes un opérateur de composition parallèle $\|$ asynchrone [11] avec une sémantique opérationnelle par entrelacement [12]. C'est-à-dire, l'évaluation de $(M \parallel N)$ entame deux sous-évaluations en parallèle, une pour chaque composante et le terme converge lorsque au moins une de ces évaluations termine sur une valeur.

Le lambda calcul étendu avec $\|$ est noté $\lambda_{\|}$. La syntaxe des termes et des valeurs est donnée par la grammaire

$$\begin{array}{llcl} (\Lambda_{\|}) & M & ::= & x \mid \lambda x.M \mid (MM) \mid (M \parallel M) \\ (\mathbb{V}_{\|}) & V & ::= & \lambda x.M \mid (V \parallel M) \mid (M \parallel V) \end{array}$$

---

[10] Dans son travail sur PCF, Riecke [68] propose des conditions sémantiques, basées sur la structure des types, pour obtenir des traductions complètement adéquates (équation 2), appelées "functional translations".

[11] On parlera de la distinction synchrone-asynchrone dans la section 2.2.5.

[12] interleaving

Le prédicat de convergence sur $\Lambda_{\|}{}^{o}$, $\Downarrow_{\|}$, est défini en ajoutant les règles de la figure 2.5 a‘ celles pour le λ-calcul de la figure 2.2 (en supposant que $\Downarrow_{\|}$ remplace $\Downarrow_{\lambda}$).

$$\begin{array}{cc}
\dfrac{M\Downarrow_{\|}V}{(M \parallel N)\Downarrow_{\|}(V \parallel N)} & \dfrac{N\Downarrow_{\|}V}{(M \parallel N)\Downarrow_{\|}(M \parallel V)} \\[2ex]
\dfrac{M\Downarrow_{\|}V \quad N\Downarrow_{\|}W}{(M \parallel N)\Downarrow_{\|}(V \parallel W)} & \dfrac{M\Downarrow_{\|}(M_0 \parallel M_1) \quad (M_0N \parallel M_1N)\Downarrow_{\|}V}{MN\Downarrow_{\|}V}
\end{array}$$

FIG. 2.5 - *Prédicat de convergence* $\Downarrow_{\|}$

La deuxième façon d'introduire le parallélisme est par l'ajout de l'opérateur non-déterministe $\oplus$. Une évaluation de $M \oplus N$ est soit une évaluation de M soit une évaluation de N. On note $\lambda_{\oplus}$ le lambda calcul étendu avec $\oplus$ dont les termes et les valeurs sont définis par

$$\begin{array}{lrcl}
(\Lambda_{\oplus}) & M & ::= & x \,|\, \lambda x.M \,|\, (MM) \,|\, (M \oplus M) \\
(\mathbb{V}_{\oplus}) & V & ::= & \lambda x.M
\end{array}$$

Le prédicat de convergence sur $\Lambda_{\oplus}{}^{o}$, $\Downarrow_{\oplus}$, est défini en ajoutant les règles de la figure 2.6 à celles de la figure 2.2.

$$\dfrac{M\Downarrow_{\oplus}V}{(M \oplus N)\Downarrow_{\oplus}V} \qquad \dfrac{N\Downarrow_{\oplus}V}{(M \oplus N)\Downarrow_{\oplus}V}$$

FIG. 2.6 - *Prédicat de convergence* $\Downarrow_{\oplus}$

L'introduction de $\|$ ou de $\oplus$ dans le λ-calcul change radicalement la notion de convergence. Dans le cadre de $\lambda_{\|}$, une valeur ne coïncide pas nécessairement avec une forme normale; si $N\Downarrow_{\|}$, alors $(M \parallel N)\Downarrow_{\|}$ même lorsque M a des évaluations de longueur infinie. De plus, un terme peut avoir plusieurs valeurs différentes, car la convergence n'arrête pas l'évaluation. Cependant, le calcul $\lambda_{\|}$ est Church-Rosser : il suffit de poursuivre l'évaluation de chacune des branches convergentes. Le calcul $\lambda_{\oplus}$ ne vérifie pas cette propriété, mais les valeurs y sont toujours des formes normales. Il est clair qu'un terme peut avoir plusieurs résultats en raison du caractère non-déterministe de $\oplus$: $(M \oplus N)$ a toutes les valeurs de M et de N.

Quel rapport peut-on établir entre $\lambda_{||}$ et $\lambda_{\oplus}$? Intuitivement, les évaluations convergentes d'un terme dans $\lambda_{\oplus}$ offrent des résultats partiels qui, mis ensemble, constituent la sémantique du terme. C'est précisément cette réunion de résultats que l'opérateur de composition parallèle permet de décrire à un niveau syntaxique[13]. Supposons

$$(\Lambda_\star) \qquad M \ ::= \ x \,|\, \lambda x.M \,|\, (MM) \,|\, (M \star N)$$

et définissons $\Downarrow^\star_{||}$ (resp. $\Downarrow^\star_{\oplus}$) sur $\Lambda_\star$ comme $\Downarrow_{||}$ (resp. $\Downarrow_{\oplus}$), en remplaçant $||$ (resp. $\oplus$) par $\star$. C'est-à-dire, on considère la traduction identité, qui vérifie immédiatement l'équation 1 de Mitchell.

Proposition 2.2.1 *Pour tout* $M \in \Lambda_\star$, $M \Downarrow^\star_{||} \Leftrightarrow M \Downarrow^\star_{\oplus}$

Preuve. La preuve est par récurrence sur les dérivations de $M \Downarrow^\star_{||}$ et de $M \Downarrow^\star_{\oplus}$. Pour l'implication $\Rightarrow$, il suffit de choisir une branche convergente comme évaluation de M. Dans le sens inverse, si $M \Downarrow^\star_{\oplus} \lambda x.N$, alors $M \Downarrow^\star_{||} (\ldots \star \lambda x.N \star \ldots)$. $\square$

On montre facilement que l'équation 2 est aussi vérifiée.

Corollaire 2.2.2 *Pour tout* $M, N \in \Lambda_\star$, $M \simeq^\star_{||} N \Leftrightarrow M \simeq^\star_{\oplus} N$

Preuve. Soit $M \simeq^\star_{||} N$ et $C[M] \Downarrow^\star_{\oplus}$. Par la proposition 2.2.1, $C[M] \Downarrow^\star_{||}$. Donc $C[N] \Downarrow^\star_{||}$ par hypothèse. Une nouvelle application de la proposition 2.2.1 donne $C[N] \Downarrow^\star_{\oplus}$. $\square$

2.2.2 Test de convergence et appel par valeur : λ_c et λ_v

Nous présentons brièvement les lambda calculs λ_c et λ_p, définis dans [3], avec test de convergence **c** et test de convergence parallèle **p**), respectivement, et étudions deux mécanismes de contrôle étroitement liés à **c**, l'appel par valeur et l'appel strict (rappelons que les combinateurs **c**, et **p** ont été introduits pour rapprocher le lambda calcul faible des modèles adéquats sur l'équation aux domaines $D = (D \to D)_\perp$.) Appliqués à des arguments, leur signification est la suivante : $\mathbf{c}ML$ veut dire L à condition que M converge, tandis que $\mathbf{p}MNL$ veut dire L à condition que M converge ou que N converge. Les ensembles des termes et des valeurs de λ_c et de λ_p sont définis par les grammaires suivantes :

$$\begin{array}{ll} \Lambda_c & M ::= x \,|\, \lambda x.M \,|\, (MM) \,|\, \mathbf{c} \\ \mathbb{V}_c & V ::= \lambda x.M \,|\, \mathbf{c} \end{array}$$

---

[13] En effet, $\oplus$ et $||$ ont la même interprétation dans le domaine complètement adéquat du lambda calcul faible (treillis complet algébrique) (cf. [14, 17]): la dénotation de $M \oplus N$ et de $M \,||\, N$ est la borne supérieure des dénotations de M et N.

$$\begin{array}{ll} \Lambda_p & M ::= x \mid \lambda x.M \mid (MM) \mid \mathbf{p} \\ \mathbb{V}_p & V ::= \lambda x.M \mid \mathbf{p} \end{array}$$

L'évaluation des termes clos dans λ_c (resp. λ_p) est définie par les règles de la figure 2.7 (resp. figure 2.8), qui s'ajoutent à celles pour λ, dans la figure 2.2.

$$\boxed{\mathbf{c}{\Downarrow_c}\mathbf{c} \qquad \frac{M{\Downarrow_c}\mathbf{c} \quad N{\Downarrow_c}}{(MN){\Downarrow_c}\mathbf{I}}}$$

FIG. 2.7 - *Prédicat de convergence* $\Downarrow_c$

$$\boxed{\mathbf{p}{\Downarrow_p}\mathbf{p} \qquad \mathbf{p}M{\Downarrow_p}\mathbf{p}M \qquad \frac{M{\Downarrow_p}\mathbf{p} \quad N{\Downarrow_p}}{(MNL){\Downarrow_p}\mathbf{I}} \qquad \frac{M{\Downarrow_p}\mathbf{p} \quad L{\Downarrow_p}}{(MNL){\Downarrow_p}\mathbf{I}}}$$

FIG. 2.8 - *Prédicat de convergence* $\Downarrow_p$

Il est clair que λ_p contient le calcul λ_c : il suffit de poser $\mathbf{c} = \lambda y.\mathbf{p}yy$. De plus, la combinaison de λ_c et $\lambda_\oplus$ (ou $\lambda_{||}$) permet d'exprimer le test de convergence parallèle : on pose $\mathbf{p} = \lambda xy.(\mathbf{c}x) \oplus (\mathbf{c}y)$. En fait ces calculs ont le même modèle complètement adéquat [2, 14].

Quel type de mécanisme de contrôle doit-on ajouter au λ calcul pour permettre une représentation de $\mathbf{c}$? La réponse est basée sur l'observation que le comportement de $\mathbf{c}M$ est celui d'une application stricte (i.e. $\mathbf{c}M$ demande à l'argument d'être convergent avant de rendre un résultat). Le lambda calcul avec appel par valeur (sans appel par nom, cf. Plotkin [65]), peut être simulé dans λ_c comme le montre Ong [59]. Par ailleurs, dans ce cadre $\mathbf{c}$ est simplement $\lambda x.\mathbf{I}$, une abstraction indépendante de son argument qui se comporte comme l'identité une fois l'argument appliqué.

Or l'appel par valeur ne permet pas de simuler l'appel par nom (ou application paresseuse), donc le calcul λ_c, ce qui nous mène à considérer un calcul λ_v, avec deux constructeurs d'abstraction: λ et λ^v. L'application paresseuse dans λ_v est de la forme $(\lambda x.M)N$ et l'évaluation de ce terme donne $M[N/x]$ quelque soit N. La forme d'une application par valeur est $(\lambda^v x.M)N$; l'évaluation de ce terme donne $M[V/x]$, pourvu que V soit une valeur de N (N doit être convergent). On définit Λ_v et $\mathbb{V}_v$ par les grammaires suivantes et le prédicat de convergence $\Downarrow_v$ dans la figure 2.9 (comme d'habitude, il faut y ajouter les règles de la figure 2.2) :

$$\begin{array}{ll} (\Lambda_v) & M \ ::= \ x \mid \lambda x.M \mid \lambda^v x.M \mid (MM) \\ (\mathbb{V}_v) & V \ ::= \ \lambda x.M \mid \lambda^v x.M \end{array}$$

$$\boxed{\frac{}{\lambda^v x.M \Downarrow_v \lambda^v x.M} \qquad \frac{M \Downarrow_v \lambda^v x.M' \quad N \Downarrow_v N' \quad M'[N'/x] \Downarrow_v V}{MN \Downarrow_v V}}$$

FIG. 2.9 - *Prédicat de convergence* $\Downarrow_v$

Les calculs λ_c et λ_v sont susceptibles d'être codés l'un dans l'autre, tout en préservant la convergence[14]. Mais leur capacité d'incorporer des constructions nouvelles diffère. Nous examinons ici la traduction de λ_c dans λ_v; la traduction opposée sera traitée dans la section 2.2.3, où l'on indiquera pourquoi ces traductions devraient vérifier l'équation (2) de Mitchell.

Définition 2.2.3 *La fonction de codage de* λ_c *dans* λ_v, $(\,)_c^v : \Lambda_c \rightarrow \Lambda_v$, *est définie comme suit :*

$$\begin{array}{rcl} (x)_c^v & = & x \\ (\lambda x.M)_c^v & = & \lambda x.(M)_c^v \\ (MN)_c^v & = & (M)_c^v\,(N)_c^v \\ (\boldsymbol{c})_c^v & = & \lambda^v x.\boldsymbol{I} \end{array}$$

Théorème 2.2.4 ***(Simulation de*** λ_c ***dans*** λ_v***)***
Pour tout $M \in \Lambda_c$, $M \Downarrow_c \Leftrightarrow (M)_c^v \Downarrow_v$

Preuve. On procède par récurrence sur la longueur des dérivations $M \Downarrow_c$ et $(M)_c^v \Downarrow_v$, en sachant que $(M[N/x])_c^v = (M)_c^v[(N)_c^v/x]$. Nous nous limitons à donner un aperçu de la preuve de l'implication $(M)_c^v \Downarrow_v \Rightarrow M \Downarrow_c$. Plus particulièrement, nous montrons

(a) $(M)_c^v \Downarrow_v \lambda x.M' \Rightarrow M \Downarrow_c \lambda x.M''$ et $(M'')_c^v = M'$

(b) $(M)_c^v \Downarrow_v \lambda^v x.M' \Rightarrow M \Downarrow_c \mathbf{c}$ et $M' = \mathbf{I}$

---

[14] Le calcul λ_v devrait s'appeler λ_v^n pour garder la notation originale [14] où n indique l'appel par nom et v l'appel par valeur. On préfère ne pas mentionner n car on a dit que tous les calculs seraient des extensions du calcul faible ou avec appel par nom. Dezani-Ciancaglini, de Liguoro et Piperno [29] emploient des variables différentes pour modéliser l'appel par nom et l'appel par valeur dans le même calcul.

Le cas de base est immédiat. Soit $(M)^v_c \Downarrow_v V$ en $k > 0$ étapes de dérivation. Alors il existe des termes M_1 et M_2 t.q. $(M)^v_c = (M_1 M_2)^v_c$. Deux situations sont possibles:

(appel par nom) Ici $(M_1)^v_c \Downarrow_v \lambda x.M_1'$ et $M_1'[(M_2)^v_c/x] \Downarrow_v V$, toutes deux déductions de longueur $< k$.

Par récurrence, $M_1 \Downarrow_c \lambda x.N_1'$ et $(N_1')^v_c = M_1'$. D'autre part,

$$(N_1'[M_2/x])^v_c = (N_1')^v_c[(M_2)^v_c/x] = M_1'[(M_2)^v_c/x]$$

Par conséquent, si $V = \lambda x.M'$, alors **(a)** est vérifié sur $N_1'[M_2/x]$ par hypothèse de récurrence. C'est-à-dire, $N_1'[M_2/x] \Downarrow_c \lambda x.M''$ et $(M'')^v_c = M'$. Lorsque $V = \lambda^v x.M'$, on a $N_1'[M_2/x]$ par **(b)**. Donc $M \Downarrow_c$.

(appel par valeur) Ici $(M_1)^v_c \Downarrow_v \lambda^v x.M_1'$, $(M_2)^v_c \Downarrow_v M_2'$ et $M_1'[M_2'/x] \Downarrow_v V$. Par récurrence, $M_1 \Downarrow_c \mathbf{c}$, $M_2 \Downarrow_c$ et $M_1' = \mathbf{I}$. Puisque $M_1'[M_2'/x]$ ne peut se réduire que vers l'identité $\mathbf{I}$, alors $V = \mathbf{I}$. De plus, $M = M_1 M_2 \Downarrow_c \mathbf{I}$. Ceci implique **(a)**, avec $M' = M'' = x$.

□

Même si le codage de λ_c dans λ_v préserve la convergence, les calculs λ_c et λ_v ne sont pas tout à fait transposables. D'après le théorème 2.2.4, à partir d'une preuve de $(\lambda x.(\mathbf{c}x)M)N \Downarrow_c$, qui repose sur $N \Downarrow_\lambda$ et $M[N/x] \Downarrow_\lambda$ (en supposant $M, N \in \Lambda$ par simplicité), on doit pouvoir construire une preuve de $(\lambda^v x.M)N \Downarrow_v$. C'est-à-dire, une preuve de $M[N'/x] \Downarrow_\lambda$ où $N \Downarrow_\lambda N'$. Pourquoi l'implication

$$M[N/x] \Downarrow_\lambda \text{ et } N \Downarrow_\lambda N' \Rightarrow (M[N'/x]) \Downarrow_\lambda$$

est-elle vérifiée? Parce que tout terme converge (dans le sens de $\Downarrow_\lambda$, $\Downarrow_c$ et $\Downarrow_v$) vers une unique valeur (i.e. les calculs sont Church-Rosser). Lorsque la preuve de $(M[N/x]) \Downarrow_\lambda$ utilise la convergence de N, par exemple si $M = x\mathbf{I}$, c'est N' qui est utilisé.

Il s'ensuit que si des nouveaux opérateurs sont incorporés à λ_v et à λ_c, la simulation n'est pas nécessairement préservée. En particulier, si l'on ajoute l'opérateur non-déterministe $\oplus$ aux calculs. La fragilité de λ_v pour ce qui regarde l'introduction de nouveaux constructeurs sera mise en évidence dans la section 2.2.4, où l'on montre que λ_v ne peut pas incorporer des éléments non-déterministes tout en gardant une interprétation fonctionnelle[15].

D'autre part, on constate que pour simuler la preuve de $\mathbf{c}N \Downarrow_c$, où $N \in \Lambda$, on doit montrer $(\lambda^v x.\mathbf{I})N \Downarrow_v$. Or la prémisse $N \Downarrow_v V$ de la règle de l'appel par valeur

---

[15] La solution apportée dans [29] consiste en l'incorporation des opérateurs $||$ et $\oplus$ au calcul λ_v, ce qui permet de garder, en parallèle, toutes les évaluations possibles d'une application par valeur. Il faut noter que le scénario observationnel considéré est de type "must testing" pour le choix non-déterministe et de type "may testing" pour la composition parallèle.

est excessivement informative : la valeur de N n'est pas importante pourvu qu'il en ait une, car pour tout N' on a $\mathbf{I}[N'/x]\Downarrow_v$. Ceci suggère la définition d'un calcul λ_s, avec application stricte, qui admet l'incorporation d'opérateurs non-déterministes. Comme dans λ_v, les abstractions de λ_s seraient de deux types, $\lambda x.M$ correspondant à l'appel par nom et et $\lambda^s x.M$ à l'appel strict défini par

$$\frac{M\Downarrow_s \lambda^s x.M' \quad N\Downarrow_s \quad M'[N/x]\Downarrow_s V}{MN\Downarrow_s V}$$

Tout comme pour les calculs λ_v et λ_c, on peut définir des codages de λ_c dans λ_s et vice versa qui préservent la convergence où le test de convergence $\mathbf{c}$ correspond à $\lambda^s x.\mathbf{I}$ dans λ_s, et $(\lambda^s x.M)$ correspond à $\lambda x.(\mathbf{c}x)(M)$ dans λ_c.

2.2.3 Simulation de λ_v dans λ_c

La simulation de λ_v dans λ_c est en rapport directe avec la simulation de l'appel par valeur dans λ_c proposée par Ong [60], qui code l'application par $\overline{MN} = \mathbf{c}\overline{N}(\overline{MN})$. La modification que nous y avons apportée est justifiée par la présence de l'appel par nom dans λ_v. Le résultat de Ong est donc corollaire du théorème 2.2.6 ci-dessous.

Définition 2.2.5 *La fonction de codage de λ_v dans λ_c, $(\,)^c_v : \Lambda_v \rightarrow \Lambda_c$, vérifie*

$$\begin{array}{rcl} (x)^c_v & = & x \\ (\lambda x.M)^c_v & = & \lambda x.(M)^c_v \\ (MN)^c_v & = & (M)^c_v\,(N)^c_v \\ (\lambda^v x.M)^c_v & = & \lambda x.(\mathbf{c}x)(M)^c_v \end{array}$$

Ce codage des abstractions par valeur $\lambda^v x.M$ rappelle la première traduction de "call-by-value PCF" (sans appel par nom) dans "lazy PCF " (avec appel par nom et test de convergence) de Riecke [68]. La version de PCF utilisée contient, en plus, un opérateur parallèle, nécessaire pour la adéquation complète du modèle standard [66]. Par un exemple très simple, Riecke montre que le codage n'est pas complètement adéquat (il est adéquat; une seule direction de l'équation (2) de Mitchell est vérifiée) et propose une traduction basée sur les types pour laquelle il prouve l'adéquation complète; mais cette traduction n'est pas compositionnelle. L'exemple mentionné est le suivant : soit $M_1 = \lambda^v x.x$ et $M_2 = \lambda^v x.\lambda^v y.(xy)$; dans le cadre du "call-by-value PCF" on a $M_1 \simeq M_2$. Mais leurs traductions dans "lazy PCF" ne sont pas observationnellement équivalentes. Avec notre notation, $(M_1)^c_v = \lambda x.(\mathbf{c}x)x$ et $(M_2)^c_v = \lambda x.(\mathbf{c}x)(\lambda y.\mathbf{c}y(xy))$. En effet, le contexte $C = [](\lambda z.3)\Omega$ sépare ces deux traductions :

$$\begin{array}{ll} C[(M_1)^c_v] & \rightarrow \mathbf{c}(\lambda z.3)\Omega \rightarrow (\lambda z.3)\Omega \rightarrow 3 \quad \text{converge} \\ & \\ C[(M_1)^c_v] & \rightarrow \mathbf{c}(\lambda z.3)(\lambda y.\mathbf{c}y((\lambda z.3)y))\Omega \rightarrow \\ & (\lambda y.\mathbf{c}y((\lambda z.3)y))\Omega \rightarrow (\mathbf{c}\Omega)((\lambda z.3)\Omega) \quad \text{diverge} \end{array}$$

La question qui se pose naturellement est si l'exemple peut être reformulé dans notre cadre, où le calcul λ_v contient aussi l'appel par nom. La réponse est non, car le contexte $[](\lambda z.\mathbf{I})\Omega$ sépare M_1 et M_2. I.e. $C[M_1] \overset{\star}{\to}_v \mathbf{I}$ mais $C[M_2] \Uparrow_v$. Le théorème suivant 2.2.6 établit la simulation de la convergence dans λ_v par la convergence dans λ_c. Il semble que les traductions $(-)^c_v$ et $(-)^v_c$ sont complètement adéquates : $\lambda_{\|}$ et $\lambda_{\oplus}$ ont une traduction complètement adéquate et, de plus, les calculs $\lambda_c+ \parallel$ et $\lambda_v + \oplus$ ont essentiellement le même modèle complètement adéquat [14, 15]. A première vue, une preuve syntaxique reposerait sur deux propriétés de simplification des traductions : $((M)^c_v)^v_c \simeq_c M$ et $((M)^v_c)^c_v \simeq_v M$. Il est difficile d'imaginer comment une des traductions pourrait satisfaire l'équation (2) mais pas l'autre. En fait, la théorie de l'expressivité énoncée par Mitchell ne fournit aucune indication pour la preuve de l'équation 2. Dans le cas de la traduction identité, comme celle qui relie $\lambda_{\|}$ a $\lambda_{\oplus}$, nous avons vu que l'équation est conséquence directe de la simulation mutuelle de la convergence. Dans un cas plus général comme celui de λ_v et λ_c il semble que deux traductions soient nécessaires.

Théorème 2.2.6 ***(Simulation de λ_v dans λ_c)***
Pour tout $M \in \Lambda_v$, $M\Downarrow_v \;\Leftrightarrow\; (M)^c_v\Downarrow_c$.

La preuve de ce théorème est une adaptation de celle de Ong [60]. Nous introduisons d'abord quelques notations et énonçons quelques résultats. Les preuves qui ne sont pas données ici se trouvent dans [60]; on indiquera les références exactes.

On définit deux relations de réduction, $\to_v$ sur Λ_v et $\to_c$ sur Λ_c, comme suit :

(β)	$(\lambda x.M)N \to_v M[N/x]$
(β^v)	Si $V \in \mathbb{V}_v$ alors $(\lambda^v x.M)V \to_v M[V/x]$
(Arg)	Si $N \to_v N'$ alors $(\lambda^v x.M)N \to_v (\lambda^v x.M)N'$
(App)	Si $M \to_v M'$ alors $(MN) \to_v (M'N)$

(β)	$(\lambda x.M)N \to_c M[N/x]$
$(C1)$	$\mathbf{cc} \to_c I$
$(C2)$	$\mathbf{c}(\lambda x.M) \to_c I$
$(C3)$	Si $M \to_c M'$ alors $\mathbf{c}M \to_c \mathbf{c}M'$
(App)	Si $M \to_c M'$ alors $(MN) \to_c (M'N)$

Les énoncés suivants ont des preuves par récurrence évidentes :

$$\text{Pour tout } M \in \Lambda_v,\; M\Downarrow_v V \;\Leftrightarrow\; M \overset{\star}{\to}_v V$$

$$\text{Pour tout } M \in \Lambda_c,\; M\Downarrow_c V \;\Leftrightarrow\; M \overset{\star}{\to}_c V$$

La troisième relation de réduction dont nous avons besoin, $\rightarrow_{\beta c} : \Lambda_c \rightarrow \Lambda_c$, est la réduction forte dans λ_c, qui autorise l'application de (β), $(C1)$ et $(C2)$ dans tout contexte, i.e. en position de fonction, en position d'argument et aussi sous les abstractions. Pour toute séquence de réductions $\rightarrow_{\beta c}$, il existe une *réduction standard* composée d'une séquence de réductions $\rightarrow_c$, suivie d'une séquence de réductions $\rightarrow_{\beta c}$ qui réduit toujours le radical le plus à gauche (cf. théorème 4.4.3.4 [60].)

L'étape suivante consiste à introduire une nouvelle relation de réduction sur Λ_c, notée $\rightarrow_o$, pour simuler $\rightarrow_v$ pas à pas.

(1) Si $M \neq (\mathbf{c}x)M'$ alors $(\lambda x.M)N \rightarrow_o M[N/x]$
(2) Si $N \in \mathbb{V}_c$ alors $(\lambda x.(\mathbf{c}x)M)N \rightarrow_o M[N/x]$
(3) Si $N \rightarrow_o N'$ alors $(\lambda x.(\mathbf{c}x)M)N \rightarrow_o (\lambda x.(\mathbf{c}x)M)N'$
(4) Si $M \rightarrow_o M'$ alors $(MN) \rightarrow_o (M'N)$

Proposition 2.2.7 *Pour tout* $M, N \in \Lambda_c$*,* $M \rightarrow_o N \Rightarrow M \overset{*}{\rightarrow}_{\beta c} N$*.*

Preuve. L'énoncé est prouvé par récurrence sur la longueur l de dérivation de $M \rightarrow_o N$. Quand $l = 1$, si la règle appliquée est (1), alors $M \rightarrow_{\beta c} N$ par la règle (β). Si c'est (2) qu'on utilise, alors

$$M = (\lambda x.(\mathbf{c}x)M)N \rightarrow_{\beta c} (\mathbf{c}N)(M[N/x]) \rightarrow_{\beta c} I(M[N/x]) \rightarrow_{\beta c} M[N/x] = N$$

Dans le cas où $l > 1$, l'énoncé découle de l'hypothèse de récurrence et du fait que $\rightarrow_{\beta c}$ est la réduction forte. □

Par récurrence sur M on peut montrer

$$(M)^c_v[(N)^c_v/x] = (M[N/x])^c_v$$

Proposition 2.2.8 *Pour tout* $M, N \in \Lambda_v$*,* $M \rightarrow_v N \Leftrightarrow (M)^c_v \rightarrow_o (N)^c_v$*.*

Preuve. Par récurrence sur la preuve de $M \rightarrow_v N$, en utilisant l'énoncé précédent. □

Le déterminisme de la relation $\rightarrow_o$ est un corollaire de la proposition 2.2.8.

Preuve du théorème 2.2.6, partie $\Rightarrow$: soit M un terme clos de Λ_v et supposons $M \Downarrow_v V$. Alors $M \overset{*}{\rightarrow}_v V$ et, par la proposition 2.2.8, on a $(M)^c_v \overset{*}{\rightarrow}_o (V)^c_v$. D'où $(M)^c_v \overset{*}{\rightarrow}_{\beta c} (V)^c_v$, en appliquant 2.2.7. Donc $(M)^c_v \Downarrow_c$ par le théorème de standardisation pour $\rightarrow_{\beta c}$ [60]. □

La preuve de la seconde moitié du théorème 2.2.6 nécessite quelques définitions et propriétés supplémentaires.

Définition 2.2.9 *On appelle* ***contextes faibles*** *les contextes définis par:*

$$Q ::= (\boldsymbol{c}Q)N|(QN)|[]$$

Définition 2.2.10 *(4.4.3.13 dans [60])*
Soit $M \in \Lambda_c$. Une $\rightarrow_{\beta c}$-réduction infinie

$$M = M_0 \rightarrow_{\beta c} M_1 \rightarrow_{\beta c} \ldots \rightarrow_{\beta c} M_i \rightarrow_{\beta c} \ldots$$

est ***quasi-faible*** *s'il existe une collection infinie d'indices $n_1, \ldots, n_i, \ldots$ t.q. $n_i < n_{i+1}$ et $M_{n_i - 1} \rightarrow_c M_{n_i}$ pour tout i.*

Remarque 2.2.11 *Soit $M \in \Lambda_c$. Si M a une réduction infinie quasi-faible, alors $Q[M]$ a une réduction quasi-faible infinie quelque soit le contexte faible Q.*

Proposition 2.2.12 *(4.4.3.18 dans [60])*
Soit $M \in \Lambda_c$. M a une réduction infinie quasi-faible ssi $M \Uparrow_c$.

Notation 2.2.13 *On note Λ_o l'ensemble des termes de Λ_c qui correspondent à des traductions de termes dans Λ_v. C'est-à-dire, $L \in \Lambda_o$ ssi $\exists M \in \Lambda_v$ $(M)^c_v = L \in \Lambda_c$*
Étant donné $M \in \Lambda_o$, on écrit $M \downarrow_o$ lorsque M a une réduction $\rightarrow_o$ finie qui termine sur un terme irréductible. Sinon, on écrit $M \uparrow_o$. La notation $M \overset{\Delta}{\rightarrow}_o M'$ indique que le radical réduit est Δ.

Remarquons que $M \downarrow_o N$ implique que N est une abstraction; le combinateur **c** de λ_c est uniquement utilisée appliquée à des arguments. Le lemme suivant est fondamental dans la preuve du théorème 2.2.6.

Lemme 2.2.14 *Soit $M \in \Lambda_o$. Si $M \uparrow_o$, alors M a une réduction quasi-faible infinie.*

Preuve. Soit $M \in \Lambda_o$ et $M \uparrow_o$, i.e.

$$M = M_0 \overset{\Delta_1}{\rightarrow}_o M_1 \overset{\Delta_2}{\rightarrow}_o M_2 \ldots \overset{\Delta_n}{\rightarrow}_o M_n \overset{\Delta_{n+1}}{\rightarrow}_o \ldots$$

Un des cas suivants est vérifié:

A. Un nombre non-borné de réductions Δ_i utilise les règles (1) ou (2).

B. $\exists N \geq 1 \;\; \forall i \geq N \;\; \Delta_i$ utilise (3) comme dernière règle.

C. $\exists N \geq 1 \;\; \forall i \geq N \;\; \Delta_i$ utilise (4) comme dernière règle.

Supposons que le cas **A** n'est pas vérifié et que N est le plus petit n t.q. $\forall i \geq n$. Δ_i utilise (3) ou (4) comme dernière règle.

- Si Δ_N utilise (3) alors $M_N = (\lambda x.(\mathbf{c}x)M')L$ où L n'est pas une abstraction; sinon Δ_{N+1} utiliserait (2), en contradiction avec l'hypothèse. Par conséquent, (3) est le seul candidat pour Δ_{N+1}, avec prémisse $L \to_o L'$. Par la même raison, L' ne peut pas être une abstraction. En réitérant cet argument on obtient le cas **B** (L ne converge pas).

- Supposons que Δ_N utilise (4). Alors $M_N = (M_1 M_2)$. Si $M_1 \uparrow_o$, il est clair que le cas **C** est vérifié. Sinon, $M_1 \downarrow_o \lambda x.M_1'$ avec $M_1' = (\mathbf{c}x)M_1''$. On ne peut pas avoir $M_1' \neq (\mathbf{c}x)M_1''$ puisque la règle (1) n'est pas applicable. De plus, M_2 n'est pas une abstraction; alors le cas **B** est vérifié.

Si c'est le cas **A** qui est vérifié, alors M a une réduction quasi-faible infinie par la proposition 2.2.12. Nous examinons ensuite les cas restants. On définit les contextes D_{xL} et E_L par:

$$D_{xL} = (\lambda x.(\mathbf{c}x)L)[] \qquad E_L = ([]L)$$

et on suppose que F, G, H sont de la forme D_{xL} ou E_L. On réunit les cas **B** et **C** comme suit:

$$M \overset{\star}{\to}_o F[R_0] \to_o F[R_1] \to_o \ldots \to_o F[R_m] \to_o \ldots$$

Toute déduction $F[R_i] \to_o F[R_{i+1}]$ consiste en un nombre fini d'applications des règles (3) et (4) et en une application de (1) ou (2). C'est-à-dire, $\exists n_i \geq 0\ \exists G_1^i, \ldots, G_{n_i}^i\ \exists S_i, S_i'$ t.q. $S_i \to_o S_i'$ par la règle (1) ou par la règle (2) et

$$R_i = G_1^i[G_2^i[..[G_{n_i}^i[S_i]]..]]$$

$$R_{i+1} = G_1^i[G_2^i[..[G_{n_i}^i[S_1']]..]]$$

On écrit $G_1^i[..[G_{n_i}^i[..]]..]$ comme $G_1^i..G_{n_i}^i[..]$. Puisque pour tout i on a $R_i \uparrow_o$, il existe un plus grand k_i, appelé le *niveau d'imbrication* de R_i, avec $1 \leq k_i \leq n_i$, t.q.

$$R_i = G_1^i..G_{k_i}^i[X_i] \ \text{ où } X_i \uparrow_o$$

L'étape suivante consiste à prouver

$(*)$ Si $X_i \uparrow_o$ par **A** alors $R_i \uparrow_o$ a une réduction quasi-faible infinie.

On procède par récurrence sur k_i. Si $k_i = 0$, alors $R_i \uparrow_o$ par **A**; donc R_i a une réduction quasi-faible infinie. Dans le cas où $k_i \geq 1$, par récurrence, $G_2..G_{k_i}[X_i]$ a une réduction quasi-faible infinie $G_2..G_{k_i}[X_i] \to_{\beta c} Y_1 .. \to_{\beta c} Y_j \to_{\beta c} \ldots$ Si $G_1 = (\lambda x.(\mathbf{c}x)L)[]$, la réduction suivante est infinie et quasi-faible:

$$R_i = (\lambda x.(\mathbf{c}x)L)(G_2..G_{k_i}[X_i]) \to_c (\mathbf{c}G_2..G_{k_i}[X_i])(L[G_2..G_{k_i}[X_i]/x]) \to_{\beta c}$$

$$(\mathbf{c}Y_1)(L[G_2..G_{k_i}[X_i]/x])...\rightarrow_{\beta c}(\mathbf{c}Y_j)(L[G_2..G_{k_i}[X_i]/x])\rightarrow_{\beta c}...$$

Sinon, $G_1 = []L$; donc

$$R_i = (G_2..G_{k_i}[X_i])L\rightarrow_{\beta c}(Y_1 L)...\rightarrow_{\beta c}(Y_j L)\rightarrow_{\beta c}...$$

est une réduction quasi-faible infinie.

La réduction de M peut avoir deux formes:

1. Si $\forall i\ k_i = 0$, alors on montre $\forall i\ R_i{\uparrow}_o$ par **A**. Soit par exemple $i = 0$ et supposons que $R_0{\uparrow}_o$ par **B** ou **C**. Alors, il existe un N fini qui vérifie soit $R_N = D_{xL}[L']$ soit $R_N = E_L[L']$, où $L'{\uparrow}_o$. Par conséquent, $k_N \geq 1$, ce qui contredit l'hypothèse. Par $(*)$, $F[R_0]$ a une réduction quasi-faible infinie. Alors M en a une aussi.

2. Supposons $\exists i\ k_i > 0$. Considérons la séquence d'indices $m_0, \ldots m_i, ..$ t.q. :

   - m_0 est le plus petit j t.q. $k_j > 0$
   - $\forall i \forall z \in [m_i, m_{i+1} - 1]\ k_{m_i} < k_{m_{i+1}}\ \&\ k_z = k_{m_i}$

   Par la suite, p_i dénote k_{m_i} et nous omettons les indices des contextes G. On distingue deux cas:

   (a) La séquence d'indices associée à M est finie. C'est-à-dire, il existe un plus grand N fini t.q. $\forall j > p_N\ k_j = p_N$. Autrement dit, $\forall j > p_N\ R_j = G_1..G_{p_N}[X_j]$. Comme dans (1), il est clair que $X_{m_N}{\uparrow}_o$ par **A**; sinon le niveau d'imbrication de R_{m_N} dépasserait p_N. En conclusion, $FG_1..G_{p_N}[X_{m_N}]$ a une réduction quasi-faible infinie par la propriété $(*)$. Donc M en a une.

   (b) La séquence d'indices associée à M est infinie. C'est-à-dire,

   $$M\overset{\star}{\rightarrow}_o F[G_1..G_{p_0}[X_{m_0}]]\overset{\star}{\rightarrow}_o F[G_1..G_{p_0}G_{p_0+1}..G_{p_1}[X_{m_1}]]...\overset{\star}{\rightarrow}_o$$

   $$F[G_1..G_{p_0}..G_{p_i}G_{p_i+1}..G_{p_{i+1}}[X_{m_{i+1}}]]\overset{\star}{\rightarrow}_o...$$

   où $X_{m_i}{\uparrow}_o$ et $X_{m_i} \overset{+}{\rightarrow}_o G_{p_i+1}..G_{p_{i+1}}[X_{m_{i+1}}]$ pour tout i.

   Il faut remarquer que les réductions des X_{m_i} sont de longueur > 0. Supposons qu'il existe j qui vérifie $X_{m_j} = G_{p_j+1}..G_{p_{j+1}}[X_{m_{j+1}}]$. Alors

   $$R_{m_j} = G_1..G_{p_j}G_{p_j+1}..G_{p_{j+1}}[X_{m_{j+1}}]$$

   et le niveau d'imbrication de R_{m_j} dépasse $p_j = k_{m_j}$ car $p_{j+1} > p_j$, ce qui est absurde par hypothèse.

   Maintenant nous allons construire une réduction de M quasi-faible et infinie de la forme

   $$\text{(IQL)} \qquad M\overset{\star}{\rightarrow}_o Q_0[X_0] \ldots \overset{\star}{\rightarrow}_{\beta c} Q_0..Q_i[X_i]\overset{\star}{\rightarrow}_{\beta c}...$$

avec $Q_0, \dots Q_i$... des contextes faibles. On définit $l = k_{i+1} - k_i$ et $H_1 = G_{k_i+1}$ $\dots$ $H_l = G_{k_{i+1}}$.

Par récurrence sur l on a

$$(**) \quad H_1 \dots H_l[X] \overset{\star}{\rightarrow}_c Q[X]$$

Nous indiquons les étapes de la preuve du cas de récurrence. Soit $l > 1$ et Q' le contexte faible associé à $H_2 \dots H_l[X]$

- ou bien
  $$H_1 \dots H_l[X] = (\lambda y.(\mathbf{c}y)N)[])H_2 \dots H_l[X] \rightarrow_c$$
  $$(\mathbf{c}(H_2 \dots H_l[X]))(N[(H_2 \dots H_l[X])/y]),$$
  auquel cas $Q = (\mathbf{c}Q')(N[(H_2 \dots H_l[X])/y])$ est un contexte faibles
- ou bien $H_1 \dots H_l[X] = (H_2 \dots H_l[X])N$. Alors $Q = (Q'N)$ est un contexte faible.

On procède par cas sur X_{m_i} pour montrer qu'il existe un contexte faible Q_{i+1} t.q. $X_{m_i} \overset{\star}{\rightarrow}_{\beta c} Q_{i+1}$ avec au moins une étape de réduction faible. Donc (IQL) est une réduction quasi-faible infinie par la remarque 2.2.11.

(a) Si $X_{m_i} = (\lambda x.(\mathbf{c}x)L)V$ avec $V \in \mathbb{V}_c$, le caractère déterministe de $\rightarrow_o$ implique

$$X_{m_i} \overset{(2)}{\rightarrow}_o L[V/x] \overset{\star}{\rightarrow}_o G_{p_i+1}..G_{p_{i+1}}[X_{m_{i+1}}]$$

Par la proposition 2.2.12 et la propriété $(**)$, la réduction suivante fait au moins trois étapes de réduction faible:

$$X_{m_i} \overset{(\beta)}{\rightarrow}_c (\mathbf{c}V)(L[V/x]) \overset{(C2)}{\rightarrow}_c \mathbf{I}(L[V/x]) \overset{(\beta)}{\rightarrow}_c L[V/x] \overset{\star}{\rightarrow}_{\beta c}$$

$$G_{p_i+1}..G_{p_{i+1}}[X_{m_{i+1}}] \overset{\star}{\rightarrow}_c Q_{i+1}$$

(b) Si $X_{m_i} = (\lambda x.L)N$ with $L \neq (\mathbf{c}x)L'$, par un argument similaire à celui donné pour (a) on obtient une réduction de X_{m_i} avec, dans ce cas, au moins une étape faible.

(c) Si $X_{m_i} = (\lambda x.(\mathbf{c}x)L)N$ où $N \notin \mathbb{V}_c$, alors $N \Downarrow_c V$ implique:

$$X_{m_i} \overset{\star}{\rightarrow}_o (\lambda x.(\mathbf{c}x)L)V \overset{+}{\rightarrow}_o G_{p_i+1}..G_{p_{i+1}}[X_{m_{i+1}}]$$

C'est le cas (a) déjà montré. Remarquons que N ne peut pas diverger car on aurait $X_i = N$.

(d) Si $X_{m_i} = (N_1 N_2)$ avec $N_1 \notin \mathbb{V}_c$, alors $N_1 \downarrow_o$ (sinon $X_{m_i} = N_1$.) Soit V t.q. $N_1 \Downarrow_c V$; on a

$$X_{m_i} \xrightarrow{\star}_o (VN_2) \xrightarrow{+}_o G_{p_i+1}..G_{p_{i+1}}[X_{m_{i+1}}]$$

La forme de (VN) correspond au cas (a) ou au cas (b).

□

Preuve du théorème 2.2.6, partie $\Leftarrow$: supposons $M \Uparrow_v$. Donc $M \uparrow_o$. Par le lemme 2.2.14, M a une réduction quasi-faible infinie. On conclut $M \Uparrow_c$ par la proposition 2.2.12. □

2.2.4 Le calcul λ_j

Nous présentons ici une variante du lambda calcul faible λ_j pour les fonctions parallèles défini par Boudol [14], obtenu par combinaison de $\lambda_\oplus$ et λ_c. Le test de convergence parallèle y est définissable: il suffit de poser $\mathbf{p} = \lambda xy.(\mathrm{c}x \oplus \mathrm{c}y)$.

Comme nous l'avons signalé, la combinaison de λ_s et $\lambda_\oplus$ serait aussi convenable. Par contre, et malgré la simulation mutuelle des calculs λ_c, λ_s et λ_v, l'appel par valeur n'est pas compatible avec le choix non-déterministe. En effet, le calcul qui résulte de l'incorporation de l'opérateur $\oplus$ à λ_v, noté $\lambda_v^\oplus$, ne vérifie pas le lemme des contextes, nécessaire pour donner une interprétation fonctionnelle au calcul. Les tests applicatifs associés à $\lambda_v^\oplus$ sont de la forme

$$A ::= [] \mid \lambda x.A \mid \lambda^v x.A \mid (AM)$$

Lemme 2.2.15 *Il existe $M, N \in \Lambda_v^\oplus$ t.q. $M \sqsubseteq_{\mathcal{A}} N$ et $\neg(M \sqsubseteq_v^\oplus N)$*

Preuve.[16] Soit M et N les termes suivants

$$M = \lambda^v z.(K_v \oplus F_v) \qquad N = (\lambda^v z.K_v) \oplus (\lambda^v z.F_v)$$

où K_v et F_v sont les versions strictes de K et F; i.e.

$$K_v = \lambda^v xy.x \qquad F_v = \lambda^v xy.y$$

On remarquera que M est une valeur mais pas N. Il est facile de voir que $M \simeq_{\mathcal{A}} N$ puisque $\forall k \forall N_1 \ldots N_k \,.\, MN_1 \ldots N_k \Downarrow_v^\oplus \Leftrightarrow NN_1 \ldots N_k \Downarrow_v^\oplus$. Pour $k = 0$ la preuve est immédiate. Pour $k > 0$ on montre par récurrence sur k la propriété suivante:

$$M\tilde{L} \Downarrow_v^\oplus V \Leftrightarrow N\tilde{L} \Downarrow_v^\oplus V$$

---

[16] L'exemple est de G. Boudol.

Or on peut construire un contexte non-applicatif tel que $C[M] \Downarrow_v^\oplus$ mais $C[N] \Uparrow_v^\oplus$. On définit:

$$\begin{cases} \Omega_0 = \Delta\Delta \qquad \text{où } \Delta = \lambda x.xx \\ \Omega_{n+1} = \lambda^v x.\Omega_n \end{cases}$$

$$C = (\lambda^v x.((x\Omega_1\Omega_2\Omega_1)(x\Omega_1\Omega_1\Omega_2\Omega_1)))\,[]$$

On a $(\Omega_2\Omega_1) \Downarrow_v^\oplus \lambda^v x.\Omega_0$ et aussi $\Omega_2(\Omega_2\Omega_1) \Downarrow_v^\oplus \Omega_1$ tandis que $(\Omega_1\Omega_1) \Uparrow_v^\oplus$ et $\Omega_1(\Omega_2\Omega_1) \Uparrow_v^\oplus$. Par conséquent :

$$\begin{aligned} C[M] \Downarrow_v^\oplus \quad & \Leftrightarrow \quad (M\Omega_1\Omega_2\Omega_1)(M\Omega_1\Omega_1\Omega_2\Omega_1) \Downarrow_v^\oplus \\ & \Leftrightarrow \quad ((K_v \oplus F_v)\Omega_1\Omega_2\Omega_1)((K_v \oplus F_v)\Omega_1\Omega_1\Omega_2\Omega_1) \Downarrow_v^\oplus \end{aligned}$$

Le dernier terme converge en prenant K_v pour la première occurrence de $(K_v \oplus F_v)$ et F_v pour la deuxième: on obtient $\Omega_2(\Omega_2\Omega_1) \Downarrow_v^\oplus$ après quelques étapes d'évaluation. En ce qui concerne $C[N]$, le premier pas consiste à calculer une valeur pour N, c'est-à-dire, à choisir entre sa composante gauche et sa composante droite. Mais ni $C[\lambda^v z.K_v]$ ni $C[\lambda^v z.F_v]$ ne convergent : du premier terme on obtient $\Omega_2(\Omega_1\Omega_1) \Uparrow_v^\oplus$ et $\Omega_1(\Omega_2\Omega_1) \Uparrow_v^\oplus$ du second. Alors $C[N] \Uparrow_v^\oplus$. $\square$

Le contre-exemple construit dans la preuve précédente n'en est pas un si l'appel par valeur est simulé par le test de convergence. Dans ce cas,

$$(C)_v^c = (\lambda x.(\mathbf{c}x)((x\Omega_1\Omega_2\Omega_1)(x\Omega_1\Omega_1\Omega_2\Omega_1)))[]$$

et le problème ne se présente pas car x est substitué par le terme N entier.

Finalement, observons que si $\|$ remplace $\oplus$ il n'y a pas non plus de difficulté à montrer que M et N ne sont pas séparables; la raison est ici que $(\lambda^v z.K_v) \parallel (\lambda^v z.F_v)$ est une valeur.

En conclusion, à exception de λ_v et $\lambda_\oplus$, toute combinaison de λ_c, λ_v et λ_s avec $\lambda_\oplus$ et $\lambda_\|$ est susceptible de vérifier un lemme de contextes. Plus loin dans ce chapitre nous discuterons sur le choix de $\oplus$ au lieu de $\|$ dans la définition du calcul λ_j, motivé par la structure du codage dans π.

Syntaxe

La version de λ_j étudiée ici fait intervenir des substitutions semi-explicites; les abstractions et les variables sont définies comme des clôtures - i.e. paires constituées d'une valeur fonctionnelle ou d'une variable et d'une substitution. La présentation des variables comme clôtures entraîne une modification de la notion d'évaluation par rapport à la définition originale, dont l'utilité se manifeste dans la preuve d'adéquation du codage de λ_j dans π. Il s'agit de rendre explicite le processus de substitution d'une variable par un terme afin de le rapprocher du mécanisme employé par le codage.

Les termes et les valeurs de λ_j sont définis par les grammaires :

$$\begin{array}{ll} (\Lambda_j) & M ::= < U, \sigma > \,|\, (MM) \,|\, (M \oplus M) \,|\, \mathbf{c} \\ \\ & U ::= x \,|\, \lambda x.M \\ \\ (\Sigma_j) & \sigma ::= \varepsilon \,|\, [M/x]\sigma \\ \\ (\mathbb{V}_j) & K ::= < \lambda x.M, \sigma > \,|\, \mathbf{c} \end{array}$$

où σ dénote une substitution, possiblement vide (ε). Remarquons que $M\sigma$ n'est pas un terme de λ_j. Les composantes $[M/x]$ d'une substitution σ sont des *entrées de substitution*. On définit

$$Var(\sigma) = \{x \,/\, [M/x] \text{ est une entrée de substitution de } \sigma\}$$

L'image de x dans σ, notée $\sigma(x)$, est donnée par :

$$\varepsilon(x) = x \qquad ([M/y]\sigma)(x) = \begin{cases} M & \text{si } y = x \\ \sigma(x) & \text{sinon} \end{cases}$$

Désormais, le *corps* d'une clôture $< U, \sigma >$ est U et on appelle simplement *abstractions* les clôtures de la forme $< \lambda x.M, \sigma >$. On utilise toujours $\mathbf{I}$ pour dénoter l'identité $< \lambda x.x, \epsilon >$.

L'ensemble $fv(M)$ des variables libres de M est défini comme suit:

$$\begin{array}{rcl} fv(< x, \varepsilon >) & = & \{x\} \\ fv(< x, [M/y]\sigma >) & = & fv(< x, \sigma >) \\ fv(< x, [M/x]\sigma >) & = & fv(M) \\ fv(< \lambda x.M, \sigma >) & = & \{y \,/\, \exists z.\ z \in fv(M) - \{x\} \ \&\ y \in fv(\sigma(z))\} \\ fv(\mathbf{c}) & = & \emptyset \\ fv(MN) & = & fv(M \oplus N) = fv(M) \cup fv(N) \end{array}$$

L'ensemble $bv(M)$ est défini par:

$$\begin{array}{rcl} bv(\varepsilon) & = & \emptyset \\ bv([M/x]\sigma) & = & \{x\} \cup bv(M) \cup bv(\sigma) \\ bv(< x, \sigma >) & = & bv(\sigma) \\ bv(< \lambda x.M, \sigma >) & = & \{x\} \cup bv(M) \cup bv(\sigma) \\ bv(\mathbf{c}) & = & \emptyset \\ bv(MN) & = & bv(M \oplus N) = bv(M) \cup bv(N) \end{array}$$

L'application d'une substitution à un terme, opération sur laquelle se base la β-conversion, est définie par l'égalité $=_s$:

$$\begin{array}{rcl} < x, \sigma > \rho & =_s & < x, (\sigma \circ \rho) > \\ \mathbf{c}\rho & =_s & \mathbf{c} \\ < \lambda x.M, \sigma > \rho & =_s & < \lambda x.M, (\sigma \circ \rho) > \\ (MN)\rho & =_s & (M\rho\ N\rho) \\ (M \oplus N)\rho & =_s & (M\rho \oplus N\rho) \end{array}$$

Par abus de notation, on note $\sigma =_s \rho$ l'égalité de substitutions, définie par :

$$\begin{array}{rcl} (\varepsilon \circ \rho) & =_s & \rho \\ ([N/x]\sigma) \circ \rho & =_s & [N\rho/x](\sigma \circ \rho) \end{array}$$

L'extension de Λ_j qui incorpore les termes $M\sigma$, notée Λ_{ej}, est de'finie par :

$$(\Lambda_{ej}) \qquad M ::=< U, \sigma > \ | \, M\sigma \,|\, (MM) \,|\, (M \oplus M) \,|\, \mathbf{c}$$

et l'ensemble des substitutions où la composition $\circ$ fait partie de la syntaxe est :

$$(\Sigma_{ej}) \qquad \sigma ::= \varepsilon \,|\, [M/x]\sigma \,|\, (\sigma \circ \sigma)$$

Les termes de Λ_{ej} et les substitutions de Σ_{ej} ont des représentants uniques dans Λ_j et Σj. C'est-à-dire,

- Pour tout $\sigma \in \Sigma_{ej}$, il existe un unique $\tau \in \Sigma_j$ t.q. $\sigma =_s \tau$.
- Pour tout $M \in \Lambda_{ej}$, il existe un unique terme $N \in \Lambda_j$ t.q. $M =_s N$.

Évaluation

La relation d'évaluation faible $\rightarrow_j$ du calcul est définie par le système de règles de la figure 2.10. On remarquera que $\rightarrow_j$ est elle-même une relation transitive. La règle (trans) est utile dans l'évaluation des termes de la forme $\mathbf{c}N$: elle permet de ne pas observer les étapes intermédiaires qui correspondent à l'évaluation de N dans le système usuel, défini dans la section 2.2.3. C'est la règle $N \rightarrow_j N' \Rightarrow$

$$\begin{array}{ll} (\beta) \quad < \lambda x.M, \sigma > N \rightarrow_j M[N/x]\sigma & (\text{saisie}) \quad < x, [M/x]\sigma > \rightarrow_j M \\ (\text{chl}) \quad (M \oplus N) \rightarrow_j M & (\text{chr}) \quad (M \oplus N) \rightarrow_j N \\ \hline (\text{pop}) \ \dfrac{< x, \sigma > \rightarrow_j N}{< x, [M/y]\sigma > \rightarrow_j N} & (\text{app}) \ \dfrac{M \rightarrow_j M'}{(MN) \rightarrow_j (M'N)} \\ (\text{obs}) \ \dfrac{N \rightarrow_j V \in \mathbb{V}_j}{\mathbf{c}N \rightarrow_j I} & (\text{trans}) \ \dfrac{M \rightarrow_j L \quad L \rightarrow_j N}{M \rightarrow_j N} \end{array}$$

FIG. 2.10 - *Évaluation dans λ_j*

$(\mathbf{c}N)\rightarrow_j(\mathbf{c}N')$ qui poserait des problèmes dans la preuve d'adéquation du codage de λ_j dans π. Nous reviendrons sur ce point plus loin. Par ailleurs, les règles (saisie) et (pop) ainsi que l'égalité $< x, \sigma > \rho =_s < x, (\sigma \circ \rho) >$ ne font pas partie de la définition du calcul donnée dans [15]. Dans notre cadre, comme dans [15], les substitutions se comportent comme des environnements : la valeur de x dans $\sigma = [M_1/y_1] \ldots [M_n/y_n]\varepsilon$ est le premier terme M_i, de gauche à droite, qui vérifie $x = y_i$. Mais cette définition est donnée dans [15] en posant $x\sigma =_s \sigma(x)$. Ce sont les règles (pop) et (saisie) qui permettent une simulation assez directe de la substitution par le codage. Le prédicat de convergence $\Downarrow_j$, est l'union des prédicats $\Downarrow_c$ et $\Downarrow_\oplus$ (cf. figures 2.7 et 2.6) plus les règles de la figure 2.11. On vérifie $M\Downarrow_j V \Leftrightarrow M\rightarrow_j V$.

$$\frac{M\Downarrow_j V}{< x, [M/x]\sigma > \Downarrow_j V} \qquad \frac{< x, \sigma > \Downarrow_j V}{< x, [M/y]\sigma > \Downarrow_j V}$$

FIG. 2.11 - *Prédicat de convergence* $\Downarrow_j$

Nous considérons le préordre extensionnel $\sqsubseteq_j$ défini sur des λ_j-contextes où la constante [] n'apparaît pas dans la partie substitution des clôtures. C'est-à-dire, les contextes ont la forme suivante :

$$C ::= [] \mid < x, \sigma > \mid < \lambda x.C, \sigma > \mid (CC) \mid (C \oplus C) \mid \mathbf{c}$$

Définition 2.2.16 *Une substitution* $\sigma = [H_1/y_1] \ldots [H_n/y_n]\varepsilon$ *est* normalisée *si* $y_j \notin fn(H_i)$ *quelque soit* i, j.

Par la suite, nous travaillons sous l'hypothèse que les substitutions qui composent les clôtures sont normalisées. Il faut noter que cette restriction ne modifie pas l'expressivité du calcul; il s'agit d'une forme d'α-conversion. Étant donné $< V, \sigma >$, on peut toujours définir une nouvelle clôture $< V', \sigma' >$ où σ' est normalisée. On obtient σ' en renommant les entrées de substitution de σ avec des variables neuves; V' est le résultat de l'application de ce même renommage à V. Le sens de la clôture de départ ne se voit pas changé puisque chaque variable libre du corps d'une clôture utilise au plus une entrée de substitution. Cette restriction est essentielle pour garantir que le codage de λ_j dans π effectue correctement les substitutions[17].

2.2.5 Autres formes étendues du λ-calcul

L'introduction du non-déterminisme dans le lambda calcul est un vieux thème qui répond à deux motivations principales : l'étude des fonctions multivaluées

---

[17] Il existe d'autres codages qui ne nécessitent pas de substitutions normalisées.

d'une part, qui correspondent aux domaines des parties, et la solution du problème d'adéquation complète des modèles à la Scott. Le but de cette section est de situer les calculs étudiés auparavent dans le cadre des λ-calculs étendus. Pour cela, nous présentons une classification de ces calculs - qui ne prétend pas être exhaustive - selon les fonctionnalités ajoutées : choix non-déterministe, composition parallèle, test de convergence, etc. Nous avons dessiné trois tableaux, qui concernent le lambda calcul faible (tableau 2.1), le lambda calcul classique où la réduction ne se limite pas à une stratégie particulière (tableau 2.2), et le lambda calcul typé (tableau 2.3). La plupart des opérateurs nouveaux ont déjà été introduits. Nous définirons brievèment ceux qui ne l'ont pas été.

calcul	cf.	appel par nom	appel par valeur	$\oplus$	$\parallel$	**c**	**p**	$\uplus$	type sémantique opérationnelle
λ_p	[2]	●					●		may testing
λ_j	[14]	●		●		●			may testing
$\lambda_{\uplus}$	[72]	●						●	bisimulation
λ_1	[15]	●	●		asynch				may testing
λ_2	[4]	●	●	●	synch				must testing
λ_3	[61, 62]	●	●	●	synch				must testing
λ_4	[29, 30]	●	●	●	synch				must testing

TAB. 2.1 - λ-*calculs faibles non-déterministes*

calcul	cf.	$\oplus$	$\parallel$	type sémantique opérationnelle
λ_5	[49]	●		must testing
λ_6	[28, 31]	●	synch	must testing

TAB. 2.2 - λ-*calculs classiques non-déterministes*

Pour ce qui est du calcul $\lambda_{\uplus}$ de Sangiorgi, l'opérateur unaire $\uplus$ remplace le choix non-déterministe habituel. Il est défini par

$$\uplus M \rightarrow_{\uplus} M \quad \text{et} \quad \uplus M \rightarrow_{\uplus} \Omega$$

Nous en reparlerons dans la section 2.4. On remarquera aussi qu'il y a deux approches pour l'introduction de la composition parallèle $\parallel$: asynchrone, utilisée par Boudol et étudiée dans la section 2.2.1, et synchrone. La version asynchrone permet le calcul dans une seule branche de la composition, c'est-à-dire, $M \parallel N$ peut se réduire sur $M' \parallel N$ si M se réduit sur M'. Dans le cadre d'un calcul avec une sémantique de type must testing, comme λ_2 à λ_6, où on veut

$$(M \oplus N) \Downarrow \; \Leftrightarrow M \Downarrow \text{ et } N \Downarrow$$

le choix non-déterministe $\oplus$ et le $\|$ asynchrone constituent le même opérateur. La version synchrone de $\|$ est utilisée pour distinguer la composition parallèle du choix non-déterministe : si M et N se réduisent sur M' et N', respectivement, alors le terme $(M \| N)$ se réduit sur $(M' \| N')$; l'évaluation dans une seule branche n'est autorisée que si l'autre branche correspond à un terme irréductible.

Le tableau suivant concerne les langages typés, tous variantes du PCF de Plotkin, sauf le dernier dû à Hennessy.

langage	cf.	appel par nom	appel par valeur	"if" parallèle	"or"	up_τ
PCF	[66]	●		●		
NDL	[6]	●			●	
PCF_l	[13]	●		●		●
PCF_v	[76]		●	●		
PCF_{nv}	[77]		●	●	●	
définitions par récurrence non-déterministes	[39]	"run-time choice" et "call-time choice"	●		●	

TAB. 2.3 - *Langages typés non-déterministes*

L'opérateur up_τ est l'équivalent du test de convergence, pour chaque type τ. Le "if" parallèle permet l'évaluation de deux expressions en même temps. Le "or" remplace le choix non-déterministe des calculs non-typés.

Dans le dernier langage mentionné, le mécanisme d'appel par nom est divisé en deux stratégies, appelées "call-time choice" et "run-time choice" par l'auteur. Il s'agit de deux formes différentes de passage de paramètres, qui concernent les arguments de la forme M or N. En effet, la question se pose de comment évaluer $F(0 \text{ or } 1)$ étant donné

$$F(x) = \text{ if } x = 0 \text{ then } x \text{ else } 2$$

La stratégie "call-time-choice" d'appel par nom correspond à l'évaluation de $(0 \text{ or } 1)$ au moment de l'appel, i.e. avant de procéder à la substitution; les résultats possibles sont 0 et 2 :

$$F(0 \text{ or } 1) \rightarrow F(0) \rightarrow \ldots \rightarrow 0$$

et

$$F(0 \text{ or } 1) \rightarrow F(1) \rightarrow \ldots \rightarrow 2$$

Quant à la stratégie “run-time choice” d'appel par nom, elle correspond à l'évaluation de l'argument (0 or 1) après substitution, ce qui donne trois résultats possibles :

$$\begin{array}{lcl} F(0 \text{ or } 1) & \rightarrow & \text{if } (0 \text{ or } 1) = 0 \text{ then } (0 \text{ or } 1) \text{ else } 2 \rightarrow \\ & & \text{if } 0 = 0 \text{ then } (0 \text{ or } 1) \text{ else } 2 \rightarrow (0 \text{ or } 1) \rightarrow \left\{ \begin{array}{l} 0 \\ \text{et} \\ 1 \end{array} \right. \end{array}$$

et

$$\begin{array}{lcl} F(0 \text{ or } 1) & \rightarrow & \text{if } (0 \text{ or } 1) = 0 \text{ then } (0 \text{ or } 1) \text{ else } 2 \rightarrow \\ & & \text{if } 1 = 0 \text{ then } (0 \text{ or } 1) \text{ else } 2 \ldots \rightarrow 2 \end{array}$$

2.3 Codage de λ_j dans π

2.3.1 Définition

Le codage de λ_j dans π est une fonction $[\![-]\!] : (\Lambda_j \times \mathcal{N}) \cup \Sigma \rightarrow \Pi$, définie selon l'approche de Milner [54]. La construction de la fonction de traduction est guidée par la simulation de l'évaluation des termes de λ_j dans le π-calcul. Le rôle du canal qui accompagne le terme est crucial: dans les lambda calculs, la règle β spécifie la communication entre un argument et une abstraction par juxtaposition; dans le π-calcul cette communication est réalisée à travers un canal qui lie la fonction à son prochain argument.

Du fait que les substitutions de Σ se comportent comme des environnements, notre fonction $[\![-]\!]$ code les substitutions sous la forme de contextes dans lesquels les termes seront plongés. En particulier, le contexte [] représente la substitution vide ε. La notation adoptée est la suivante: $[\![\sigma]\!]$ dénote un π-contexte avec un seul trou, et dans $[\![\sigma]\!](P)$, la constante [] est remplacée par le processus P.

La traduction d'une clôture revient à plonger le corps (abstraction ou variable) dans le contexte qui traduit la substitution:

$$[\![< V, \sigma >]\!]u = [\![\sigma]\!]([\![V]\!]u)$$

Le codage des entrées de substitution est en rapport avec le codage des variables. L'entrée $[H/x]$ (notée aussi $x := H$) est représentée dans π par la ressource $x(w)[\![H]\!]w$, tandis que $[\![x]\!]u = \overline{x}u$ émet la continuation u. La traduction de $< x, [H/x]\varepsilon >$ est donc la composition parallèle de $x(w)[\![H]\!]w$ et $\overline{x}u$, ce qui permet de produire $[\![H]\!]u$ après une étape de réduction dans π.

Mais il est clair que le codage de $< x, [H/x][J/x]\varepsilon >$ ne peut pas être $(\overline{x}u \mid x(w)[\![H]\!]w \mid x(w)[\![J]\!]w)$, car ce processus donne $[\![H]\!]u$ et aussi $[\![J]\!]u$ par réduction. La solution de ce problème consiste en la création des noms privés pour chaque entrée de substitution, suivant l'imbrication de la substitution (i.e. l'ordre des entrées de gauche à droite). Ainsi, $\sigma = [H_1/y_1]\dots[H_n/y_n]\varepsilon$ est transformé en

$$(\nu y_n)(\,(\nu y_{n-1})(\,\dots(\nu y_1)(\,[]\mid [\![y_1 := H_1]\!]\,)\dots\mid [\![y_{n-1} := H_{n-1}]\!]\,)\mid [\![y_n := H_n]\!]\,)$$

Avec cette représentation, l'accès à $[\![J]\!]u$ dans

$$(\nu x)(\,(\nu x)(\overline{x}u \mid [\![x := H]\!]) \mid [\![x := J]\!]\,)$$

est bloqué parce que l'occurrence de x en $\overline{x}u$ est liée par la restriction (νx) la plus proche (qui correspond à $x := H$ dans ce cas.) C'est-à-dire, le processus est α-convertible en $(\nu x)(\,(\nu y)(\overline{y}u \mid [\![y := H]\!]) \mid [\![x := J]\!]\,)$ avec y nouveau.

Il y a encore un point a considérer: la non-linéarité des lambda termes. Remarquons que, pour chaque occurrence libre de la variable x dans M, la traduction de $M[x/H]\varepsilon$ doit mettre à disposition une ressource $[\![x := H]\!]$. En utilisant la réplication, le codage des entrées de substitution devient

$$[\![x := H]\!] = !x(w)[\![H]\!]w$$

Nous examinons maintenant le codage des abstractions et des applications. Les abstractions correspondent à des processus gardés par une entrée qui figent les corps, préservant ainsi la notion de valeur puisque les gardes d'entrée bloquent le calcul. La traduction de $\lambda x.M$ est celle de Milner [54]: le processus $[\![\lambda x.M]\!]u$ reçoit par u le nom du canal qui garde le processus correspondant au premier argument de $\lambda x.M$. Ensuite, le processus consomme le nom de canal à travers lequel il peut communiquer avec un deuxième argument potentiel.

$$[\![\lambda x.M]\!]u = u(x).u(v)[\![M]\!]v$$

De façon complémentaire, pour simuler la β-conversion $(\lambda x.M)N \rightarrow_j M[N/x]$, l'argument envoie à la fonction, d'abord, un nom privé z qui garde N - et qui substituera x, puis un nom w pour accéder a l'argument suivant de $(\lambda x.M)N$. La substitution $[N/z]$ n'est pas réalisée; le corps de l'abstraction est plongé dans l'environnement $(\nu z)([]|[\![z := N]\!])$. Ainsi, chaque instance de z peut consommer une copie de l'argument (rappelons que $[\![z := N]\!]$ est la réplication $!z(w)[\![N]\!]w$).

$$[\![MN]\!]w = (\nu u)([\![M]\!]u \mid push(N)uw) \quad \text{où}$$

$$push(N)uw = (\nu z)(\overline{u}z.\overline{u}w \mid [\![z := N]\!])$$

Pour le test de convergence on prend la définition donnée dans [54]:

$$[\![\mathrm{c}]\!]u = u(x).u(v).(\nu w)\overline{x}w.(\nu y)\overline{w}y.[\![I]\!]v$$

Le terme $[\![\mathrm{c}N]\!]u$ se comporte d'abord comme une application ordinaire (remarquons que $[\![\mathrm{c}]\!]u$ commence comme une abstraction) puis réalise les trois pas suivants:

- accéder à une copie de N à travers le nom privé qui remplace x (la garde $\overline{x}w$ est consommée); $[\![N]\!]w$ est alors en exécution (avec w un nom privé);

- attendre la convergence de N; c'est-à-dire, attendre un processus de la forme $w(y)\ldots$;
- si N est convergent, le processus obtenu consomme la garde de sortie $\overline{w}y$, ce qui donne comme résultat $\llbracket I \rrbracket v$.

La traduction du choix non-déterministe est très simple; on oblige à effectuer le choix entre M et N avant toute autre action:

$$\llbracket M \oplus N \rrbracket u = (\nu v)(\overline{v}u \mid v(w)\llbracket M \rrbracket w \mid v(w)\llbracket N \rrbracket w)$$

La figure 2.12 contient la définition complète de la fonction de codage.

$$\llbracket < V, \sigma > \rrbracket u = \llbracket \sigma \rrbracket(\llbracket V \rrbracket u)$$

$$\llbracket \varepsilon \rrbracket = []$$

$$\llbracket [M/x]\sigma \rrbracket = \llbracket \sigma \rrbracket(\,(\nu x)(\,[] \mid \llbracket x := M \rrbracket\,)\,)$$

$$\llbracket x \rrbracket u = \overline{x}u$$

$$\llbracket \lambda x.M \rrbracket u = u(x)u(v)\llbracket M \rrbracket v$$

$$\llbracket MN \rrbracket w = (\nu u)(\llbracket M \rrbracket u \mid push(N)uw)$$

$$push(N)uw = (\nu z)(\overline{u}z.\overline{u}w \mid \llbracket z := N \rrbracket)$$

$$\llbracket x := M \rrbracket = !x(w)\llbracket M \rrbracket w$$

$$\llbracket M \oplus N \rrbracket u = (\nu v)(\overline{v}u \mid v(w)\llbracket M \rrbracket w \mid v(w)\llbracket N \rrbracket w)$$

$$\llbracket \mathrm{c} \rrbracket u = u(x).u(v).(\nu w)\overline{x}w.(\nu y)\overline{w}y.\llbracket I \rrbracket v$$

FIG. 2.12 - *Codage de λ_j dans π*

Proposition 2.3.1 *Pour tout M terme clos de Λ_j,*

1. $M \in \mathbb{V}_j \Leftrightarrow \llbracket M \rrbracket u \downarrow u$
2. $\llbracket M \rrbracket u \overset{\star}{\triangleright} P \xrightarrow{\alpha} \quad \& \quad \alpha \neq \tau \;\Rightarrow\; \alpha = u(v)$

Preuve. La partie 1 est immédiate, par inspection de la définition de $[\![\,]\!]$. En ce qui concerne la partie 2, par récurrence sur M on montre que $[\![M]\!]u \overset{\star}{\triangleright} P$ implique $fn(P) = \{u\} \cup fn(M)$ et $\mathrm{Com}(P) \subseteq \{u\} \cup \overline{fn(M)}$. D'où l'on tire

$$\forall M, P \text{ clos } \; [\![M]\!]u \overset{\star}{\triangleright} P \Rightarrow fn(P) = \{u\} \;\&\; \mathrm{Com}(P) \subseteq \{u\} \tag{2.1}$$

Nous examinons le cas des clôtures à titre d'exemple. Soit $M =< \lambda x.M', \sigma >$. Remarquons que si $\sigma = [H_1/y_1] \ldots [H_n/y_n]$ est normalisée (i.e. $\forall i, j \;.\; y_j \notin fn(H_i)$), alors $fn([\![\sigma]\!](Q)) = \bigcup_{i=1}^{n} fn(H_i) \cup (fn(Q)/\{y_1, \ldots, y_n\})$. Donc

$$fn([\![\sigma]\!]([\![\lambda x.M']\!]u)) = \bigcup_{i=1}^{n} fn(H_i) \cup (fn([\![\lambda x.M']\!]u)/\{y_1, \ldots, y_n\})$$

Par la définition de $[\![\lambda x.M']\!]u$ et l'h.r. sur M', on a $fn([\![\lambda x.M']\!]u) = \{u\} \cup fn(\lambda x.M')$. Alors,

$$fn([\![\sigma]\!]([\![\lambda x.M']\!]u)) = \bigcup_{i=1}^{n} fn(H_i) \cup \{u\} \cup (fn(\lambda x.M')/\{y_1, \ldots, y_n\})$$

$$\begin{array}{rcl} \mathrm{Com}([\![M]\!]u) & = & \mathrm{Com}([\![\lambda x.M']\!]u)/\{y_1, \ldots, y_n, \overline{y_1}, \ldots, \overline{y_n}\} \\ & = & \{u\} \subseteq \{u\} \cup \overline{fn(M)} \end{array}$$

Supposons $[\![M]\!]u \overset{\star}{\triangleright} P \xrightarrow{\alpha}$. Par 1, il n'existe pas de nom r t.q. $\overline{r} \in \mathrm{Com}(P)$. Alors α est une transition d'entrée sur un canal w, où $w \in \mathrm{Com}(P)$. Comme M est clos, $w = u$. □

Il faut noter que la condition de normalisation sur les substitutions est essentielle. En effet, le codage des substitutions utilise la réplication, une construction pour représenter des éléments persistants, quand en réalité la substitution de λ_j s'efface une fois qu'elle a été utilisée (autant de fois que nécessaire). Le codage des substitutions normalisées reproduit ce comportement de la façon suivante:

Proposition 2.3.2 *Soit $[M/x]\sigma$ une substitution normalisée.*

$$[\![\sigma]\!]((\nu x)([\![M]\!]u \mid [\![x := M]\!])) \equiv [\![M]\!]u$$

Preuve. Puisque σ est normalisée, $x \notin fv(M)$. Donc la restriction (νx) concerne seulement $[\![x := M]\!]$ et on vérifie:

$$[\![\sigma]\!]((\nu x)([\![M]\!]u \mid [\![x := M]\!])) \equiv [\![\sigma]\!]([\![M]\!]u \mid (\nu x)[\![x := M]\!])$$

Une application réitérée de la loi pour réduire la portée d'un nom privé permet de conclure

$$[\![\sigma]\!]((\nu x)([\![M]\!]u \mid [\![x := M]\!])) \equiv ([\![M]\!]u \mid [\![\sigma]\!]((\nu x)[\![x := M]\!])) \equiv [\![M]\!]u$$

□

Dans la section 2.2.4, nous avons souligné que les règles usuelles d'évaluation concernant **c** posaient des problèmes. En effet, notre définition de $[\![\,]\!]$ n'admet pas la règle

$$\text{Si } N \to N' \text{ alors } (\mathbf{c}N) \to (\mathbf{c}N')$$

car celle-ci ne correspond pas à une séquence de réductions $\rhd$. Le résultat de correction 2.3.4 ne serait donc pas vérifié. Pour permettre une évaluation de $[\![N]\!]w$, le processus $[\![\mathbf{c}N]\!]u$ doit consommer quelques gardes de **c**. Par conséquent, $[\![\mathbf{c}N]\!]u$ ne peut pas devenir $[\![\mathbf{c}N']\!]u$.

Les calculs avec composition parallèle $\|$ ont été mis de côté en raison des propriétés des codages qu'ils auraient engendré. A priori, le codage de $(M \parallel N)$ pourrait être $([\![M]\!]u \mid [\![N]\!]u)$. Or, une telle définition pose deux problèmes. Observons tout d'abord que la règle de distribution de l'argument $(M \parallel N)L {\to_{\|}} (ML \parallel NL)$ n'aurait pas une séquence de réductions associée dans le π-calcul, mais une équivalence par bisimulation forte $(M \parallel N)L \sim_{\pi} (ML \parallel NL)$. Certainement, le système de règles de réduction de $\lambda_{\|}$ peut être modifié pour accommoder le codage, par l'ajout de règles du style $(< \lambda x.M, \sigma >\parallel N)L \to M[L/x]\sigma \parallel (NL)$. Une propriété plus importante de cet hypothétique codage, d'où l'on tire des nouveaux contre-exemples à l'adéquation complète, est la suivante : $[\![\lambda x.(M \parallel N)]\!]u$ peut être distingué de $[\![\lambda x.M \parallel \lambda x.N]\!]u$; en effet, le calcul π permet de compter le nombre de compositions parallèles d'un terme tandis que $\lambda x.(M \parallel N) {\simeq_{\|}} (\lambda x.M \parallel \lambda x.N)$.

Finalement, on remarquera que la notion de réduction $\rhd$ du π-calcul semble plus généreuse que nécessaire. Nous entendons par là que la restriction de la communication sur des noms privés n'affecterait pas nos preuves. La nouvelle formulation de $\mapsto$ à considérer serait

$$(\nu\vec{v})\{|\,...,x(y).R', \overline{x}z.T, ...\,|\} \mapsto (\nu\vec{v})\{|\,..., R'[z/y], T, ...\,|\} \qquad \text{avec } x \in \vec{v}$$

2.3.2 Substitutions

Nous examinons maintenant le processus de substitution défini par les équations $=_s$. Le première étape de la preuve de préservation des réductions par $[\![\,]\!]$, consiste à montrer que les substitutions de λ_j sont correctement modélisées par le codage. On étend le domaine de $[\![\,]\!]$ aux termes de Λ_{ej} et aux substitutions de Σ_{ej}. Les clauses ajoutées sont :

$$[\![M\sigma]\!]u = [\![\sigma]\!]([\![M]\!]u)$$

$$[\![\sigma \circ \rho]\!] = [\![\rho]\!]([\![\sigma]\!])$$

Nous rappelons que $M\sigma \in \Lambda_{ej}$ dénote le λ_j-terme, soit A, où σ a été distribué aux composants de M, selon les lois $=_s$. De plus, $\sigma \in \Sigma_{ej}$ dénote une substitution $\zeta \in \Sigma_j$. Dans le π calcul, $([\![M\sigma]\!]u, [\![M\zeta]\!]u)$ et $([\![M\sigma]\!]u, [\![A]\!]u)$ sont des équivalences par bisimulation quelque soit $M \in \Lambda_j$. Les propriétés $(*)$ et $(**)$ suivantes seront utiles dans la preuve de cette propriété, ainsi que les équivalences par bisimulation B1-B5 (prouvées à l'aide des équivalences A1 et A2.)

(∗) Si $[H/x]\zeta$ est normalisée alors $(N[H/x]\varepsilon)\zeta = N([H/x]\zeta)$

(∗∗) $[\![(M\zeta)\rho]\!]u = [\![M(\zeta \circ \rho)]\!]u$

A1 $!P \sim_\pi !P|!P$

B1 $(\nu x)(P|Q|!x(w).F) \sim_\pi (\nu x)(P|!x(w).F)|(\nu x)(Q|!x(w)F)$ si x apparaît dans P, Q, et F seulement comme nom de sortie.

Prouvée dans [55], cet équivalence permet la distribution des substitutions.

A2 $(\nu x)(P|!z(w)(Q|[\![x := H]\!])|[\![x := H]\!]) \sim_\pi$ $(\nu x)(P|[x := H)|(\nu x)!z(w)(Q|[\![x := H]\!])$ si x apparaît dans P et Q seulement comme nom de sortie.

La relation $\mathcal{R}$ ci-dessous est une bisimulation à $\sim_\pi$ près (en utilisant **A1**):

- ($(\nu x)(P|!z(w)(Q|[\![x := H]\!])|[\![x := H]\!])$, $(\nu x)(P|[\![x := H]\!]|(\nu x)!z(w)(Q|[\![x := H]\!])$) $\in \mathcal{R}$, si P, Q vérifient les conditions mentionnées.
- $(A, B) \in \mathcal{R}_1 \Rightarrow (A, B) \in \mathcal{R}$, où $\mathcal{R}_1$ est la bisimulation utilisée dans la preuve de B1.
- $(A, B) \in \mathcal{R} \Rightarrow ((\nu y)A\, , (\nu y)B) \in \mathcal{R}$.
- $(A, B) \in \mathcal{R} \Rightarrow (U|A\, , U|B) \in \mathcal{R}$ pour tout U.
- $(A, B) \in \mathcal{R}$ et $(B, C) \in \mathcal{R} \Rightarrow (A, C) \in \mathcal{R}$.

B2 $(\nu x)(v(z).P|!x(w).F) \sim_\pi (\nu x)v(z').(P[z'/z]|!x(w).F)$ si $v \neq x$ et z' est un nom nouveau.

Preuve immédiate.

B3 $(\nu x)(!v(z).P|!x(w).F) \sim_\pi (\nu x)!v(z').(P[z'/z]|!x(w).F)$ lorsque $v \neq x$ et z' sont des noms nouveaux.

Preuve immédiate .

B4 $(\nu x)r(z).P \sim_\pi r(z').(\nu x)P[z'/z]$ si $r \neq x$ et z' est un nom nouveau.

La relation $\mathcal{R}$ composée des paires

- $(\,(\nu x)(!v(z)P\,|\,!x(w)F)\,,\,(\nu x)!v(z')(P[z'/z]\,|\,!x(w)F)\,)$
- $(\,(\nu x)(S\,|\,!x(w)F\,|\,!v(z)P\,|\,!x(w)F)\,,$ $(\nu x)(S\,|\,!x(w)F\,|\,!v(z')(P[z'/z]\,|\,!x(w)F))\,)$

où $v \neq x$ et z' est nom nouveau, est une bisimulation à $\sim_\pi$ près (en utilisant **A1**).

B5 $(\nu x)!z(w).([\![N]\!]w|[\![x := H]\!]) \sim_\pi !(\nu x)z(w).([\![N]\!]w|[\![x := H]\!]$ lorsque $z \neq x$.

La relation $\mathcal{R}$ composée des paires de la forme

- $(\,(\nu x)!z(w)([\![N]\!]w|[\![x := H]\!])\,,\,!(\nu x)z(w)([\![N]\!]w|[\![x := H]\!]\,)$
- $(\,(Q|(\nu x)!z(w)([\![N]\!]w|[\![x := H]\!]))\,,\,(Q|!(\nu x)z(w)([\![N]\!]w|[\![x := H]\!])\,)$

est une bisimulation à $\sim_\pi$ près (en utilisant **A2**).

Théorème 2.3.3 *Soit* $M \in \Lambda_{ej}$ *et* $\sigma \in \Sigma_{ej}$.

1. *Si* $M\sigma =_s A$ *où* $A \in \Lambda_j$, *alors* $[\![M\sigma]\!]u \sim_\pi [\![A]\!]u$.

2. *Si* $\sigma =_s \zeta$ *où* $\zeta \in \Sigma_j$, *alors* $\forall M \in \Lambda_{ej}.\ [\![M\sigma]\!]u \sim_\pi [\![M\zeta]\!]u$.

Preuve.

1. Nous montrons que la relation $\sim_s$, constituée des paires suivantes, est une bisimulation forte à $\sim_\pi$ près:

$$\begin{array}{ll}
(a) & (\,[\![(MN)\sigma]\!]u\,,\,[\![(M\sigma\, N\sigma)]\!]u\,) \\
(b) & (\,[\![(M \oplus N)\sigma]\!]u\,,\,[\![(M\sigma \oplus N\sigma)]\!]u\,) \\
(c) & (\,[\![< \lambda x.M, \eta > \sigma]\!]u\,,\,[\![< \lambda x.M, \eta \circ \sigma >]\!]u\,) \\
(d) & (\,[\![\mathbf{c}\sigma]\!]u\,,\,[\![\mathbf{c}]\!]u\,) \\
(e) & (\,[\![< x, \eta > \sigma]\!]u\,,\,[\![< x, \eta \circ \sigma >]\!]u\,)
\end{array}$$

(a) Deux cas se présentent selon la forme σ:

- Soit σ une substitution simple, sans composition $\circ$. On procède par récurrence sur σ. Si $\sigma = \varepsilon$, l'énoncé est immédiat : $[\![\varepsilon]\!](push(N)wu) \sim_\pi push(N\varepsilon)wu$. Dans le cas où $\sigma = [H/x]\zeta$, l'hypothèse de récurrence donne, pour tout $M, N \in \Lambda_{ej}$,

  $$(\,[\![(MN)\zeta]\!]u\,,\,[\![(M\zeta\, N\zeta)]\!]u\,) \text{ et } [\![\zeta]\!](push(N)wu) \sim_\pi push(N\zeta)wu$$

  Nous montrons d'abord $[\![\sigma]\!](push(N)wu) \sim_\pi push(N\sigma)wu$. Soit r et v

des noms nouveaux:

$$\begin{array}{rcl}
 & & (\nu x)(\,push(N)wu \mid \llbracket x := H\rrbracket\,) \\
 & = & (\nu x)(\,(\nu z)(\overline{w}z.\overline{w}u|\llbracket z := N\rrbracket) \mid \llbracket x := H\rrbracket\,) \\
 & \equiv & (\nu z)(\nu x)(\,\overline{w}z.\overline{w}u|\llbracket z := N\rrbracket|\llbracket x := H\rrbracket\,) \\
\text{par (B1)} & \sim_\pi & (\nu z)(\,(\nu x)(\overline{w}z.\overline{w}u|\llbracket x := H\rrbracket) \mid \\
 & & (\nu x)(\llbracket z := N\rrbracket|\llbracket x := H\rrbracket)\,) \\
 & \equiv & (\nu z)(\,\overline{w}z.\overline{w}u \mid (\nu x)(\llbracket z := N\rrbracket|\llbracket x := H\rrbracket)\,) \\
\text{par (B3)} & \sim_\pi & (\nu z)(\,\overline{w}z.\overline{w}u \mid (\nu x)!z(r).(\llbracket N\rrbracket r|\llbracket x := H\rrbracket)\,) \\
\text{par (B5)} & \sim_\pi & (\nu z)(\,\overline{w}z.\overline{w}u \mid !(\nu x)z(r).(\llbracket N\rrbracket r|\llbracket x := H\rrbracket)\,) \\
\text{par (B4)} & \sim_\pi & (\nu z)(\,\overline{w}z.\overline{w}u \mid !z(v).(\nu x)(\llbracket N\rrbracket v|\llbracket x := H\rrbracket)\,) \\
 & = & (\nu z)(\,\overline{w}z.\overline{w}u \mid !z(v).\llbracket N[H/x]\varepsilon\rrbracket v\,) \\
 & = & push(N[H/x]\varepsilon)wu
\end{array}$$

Par conséquent,

$$\begin{array}{rcl}
\llbracket\sigma\rrbracket(push(N)wu) & = & \llbracket\zeta\rrbracket((\nu x)(\,push(N)wu \mid \llbracket x := H\rrbracket\,)) \\
 & \sim_\pi & \llbracket\zeta\rrbracket(push(N[H/x]\varepsilon)wu) \\
\text{par h.r.} & \sim_\pi & push((N[H/x]\varepsilon)\zeta)wu) \\
 & = & (\nu z)(\overline{w}z.\overline{w}u|!z(r).\llbracket(N[H/x]\varepsilon)\zeta\rrbracket r) \\
\text{par } (*) & = & (\nu z)(\overline{w}z.\overline{w}u|!z(r).\llbracket\sigma\rrbracket(\llbracket N\rrbracket r)) \\
 & = & push(N\sigma)wu
\end{array}$$

La preuve de (a) est alors:

$$\begin{array}{rcl}
\llbracket(MN)\sigma\rrbracket u & = & \llbracket\zeta\rrbracket((\nu x)(\,(\nu w)(\llbracket M\rrbracket w|push(N)wu) \mid \llbracket x := H\rrbracket)) \\
 & \equiv & \llbracket\zeta\rrbracket((\nu w)(\nu x)(\llbracket M\rrbracket w|push(N)wu|\llbracket x := H\rrbracket)) \\
\text{par (B1)} & \sim_\pi & \llbracket\zeta\rrbracket((\nu w)(\,(\nu x)(\llbracket M\rrbracket w|\llbracket x := H\rrbracket) \mid \\
 & & (\nu x)(push(N)wu|\llbracket x := H\rrbracket)\,)) \\
 & = & \llbracket\zeta\rrbracket((\nu w)(\llbracket M[H/x]\varepsilon\rrbracket w \mid \\
 & & (\nu x)(push(N)wu|\llbracket x := H\rrbracket)\,)) \\
 & \sim_\pi & \llbracket\zeta\rrbracket((\nu w)(\llbracket M[H/x]\varepsilon\rrbracket w \mid push(N[H/x]\varepsilon)wu)) \\
 & = & \llbracket\zeta\rrbracket(\llbracket M[H/x]\varepsilon\, N[H/x]\varepsilon\rrbracket u) \\
\text{par h.r.} & \sim_\pi & \llbracket(M[H/x]\varepsilon)\zeta\,(N[H/x]\varepsilon)\zeta\rrbracket u \\
\text{par } (*) & = & \llbracket M\sigma\, N\sigma\rrbracket u
\end{array}$$

- Si $\sigma = \zeta \circ \rho$, la preuve est par récurrence sur le nombre n d'occurrences de $\circ$ dans σ. Le cas de base, $n = 0$, correspond à la partie (a) de l'énoncé. Si $n > 0$, on a

  $$\llbracket(MN)(\zeta \circ \rho)\rrbracket u = \llbracket\zeta \circ \rho\rrbracket(\llbracket MN\rrbracket u) =$$

  $$(\llbracket\rho\rrbracket(\llbracket\zeta\rrbracket)(\llbracket MN\rrbracket u) = \llbracket\rho\rrbracket(\llbracket\zeta\rrbracket(\llbracket MN\rrbracket u))$$

  En appliquant deux fois l'hypothèse de récurrence, puis $(**)$, on obtient:

  $$\llbracket\rho\rrbracket(\llbracket\zeta\rrbracket(\llbracket MN\rrbracket u)) \sim_\pi \llbracket(M\zeta)\rho\,(N\zeta)\rho\rrbracket u \sim_\pi \llbracket M\sigma\, N\sigma\rrbracket u$$

En ce qui concerne les équivalences restantes (b)-(e), nous nous limitons à étudier le cas des substitutions $[H/x]\varepsilon$. L'extension à $[H/x]\zeta$ et le traitement des substitutions composées sont comme dans le cas (a).

(b) Ici,

$$\begin{array}{rcl}
\llbracket(M \oplus N)\sigma\rrbracket u & = & (\nu x)((\nu v)(\overline{v}u|v(w).\llbracket M\rrbracket w|v(w).\llbracket N\rrbracket w)|\llbracket x := H\rrbracket) \\
 & \equiv & (\nu v)(\nu x)(\overline{v}u|v(w).\llbracket M\rrbracket w|v(w).\llbracket N\rrbracket w|\llbracket x := H\rrbracket) \\
\text{par (B1)} & \sim_{\pi} & (\nu v)(\,(\nu x)(\overline{v}u|\llbracket x := H\rrbracket)\,|\,(\nu x)(v(w).\llbracket M\rrbracket w\,| \\
 & & \llbracket x := H\rrbracket)\,|\,(\nu x)(v(w)\llbracket N\rrbracket w|\llbracket x := H\rrbracket)\,) \\
 & \equiv & (\nu v)(\,\overline{v}u\,|\,(\nu x)(v(w).\llbracket M\rrbracket w|\llbracket x := H\rrbracket)\,| \\
 & & (\nu x)(v(w).\llbracket N\rrbracket w|\llbracket x := H\rrbracket)\,) \\
\text{par (B2)} & \sim_{\pi} & (\nu v)(\,\overline{v}u\,|\,(\nu x)v(w).(\llbracket M\rrbracket w|\llbracket x := H\rrbracket)\,| \\
 & & (\nu x)v(w).(\llbracket N\rrbracket w|\llbracket x := H\rrbracket)\,) \\
\text{par (B4)} & \sim_{\pi} & (\nu v)(\,\overline{v}u\,|\,v(w).(\nu x)(\llbracket M\rrbracket w|\llbracket x := H\rrbracket)\,| \\
 & & v(w).(\nu x)(\llbracket N\rrbracket w|\llbracket x := H\rrbracket)\,) \\
 & = & (\nu v)(\,\overline{v}u\,|\,v(w).\llbracket M\sigma\rrbracket w\,|\,v(w).\llbracket N\sigma\rrbracket\,) \\
 & = & \llbracket(M\sigma \oplus N\sigma)\rrbracket u
\end{array}$$

(c) Il découle de la définition de $\llbracket\eta \circ \sigma\rrbracket$ que :

$$\llbracket< \lambda x.M, \eta > \sigma\rrbracket u = \llbracket\sigma\rrbracket(\llbracket\eta\rrbracket(\llbracket\lambda x.M\rrbracket u)) =$$

$$(\llbracket\sigma\rrbracket(\llbracket\eta\rrbracket))(\llbracket\lambda x.M\rrbracket u) = \llbracket< \lambda x.M, \eta \circ \sigma >\rrbracket u$$

(d) Puisque $fv(\mathbf{c}) = \emptyset$, on a :

$$\llbracket\mathbf{c}\sigma\rrbracket u = (\nu x)(\,\llbracket\mathbf{c}\rrbracket u|\llbracket x := H\rrbracket\,) \equiv \llbracket\mathbf{c}\rrbracket u$$

(e) Similaire au cas (c).

2. L'énoncé est conséquence de

$$\llbracket M(\varepsilon \circ \rho)\rrbracket u \equiv \llbracket M\rho\rrbracket u \quad \text{et} \quad \llbracket M([H/y]\eta \circ \rho)\rrbracket u \sim_{\pi} \llbracket M[H\rho/x](\eta \circ \rho)\rrbracket u$$

pourvu que $[H/y]\eta$ et ρ soient des substitutions normalisées. La première de ces équivalences est évidente; nous montrons la seconde par récurrence sur M. Soit $\sigma = \varepsilon \circ \rho$ et $\zeta = [H\rho/x](\eta \circ \rho)$.

- Si $M =< x, \varepsilon >$, alors dans le cas où $x \neq y$ il est facile de voir :

  $$\llbracket< x, [H/y]\eta \circ \rho >\rrbracket u \sim_{\pi} \llbracket< x, \eta \circ \rho >\rrbracket u \sim_{\pi} \llbracket< x, [L/y](\eta \circ \rho) >\rrbracket u$$

  quelque soit le terme L t.q. $fv(L) \cap Var(\eta, \rho) = \emptyset$; ceci est vrai en particulier pour $L = H\rho$, par l'hypothèse de normalisation sur les substitutions. Si $x = y$, alors

  $$\llbracket< x, [H/x]\eta \circ \rho >\rrbracket u \sim_{\pi} \llbracket< x, [H\rho/x]\varepsilon >\rrbracket u \sim_{\pi} \llbracket< x, [H\rho/x]\theta >\rrbracket u$$

pour toute substitution θ normalisée t.q. $fv(H\rho) \cap Var(\theta) = \emptyset$, en particulier pour $\theta = \eta \circ \rho$, car $[H/y]\eta$ et ρ sont normalisées.

- Nous donnons un aperçu de la preuve pour $M =< \lambda x.N, \psi >$. Dans les cas restants, la preuve utilise l'hypothèse de récurrence et la partie (1) de l'énoncé. On a,

  $$[\![M\sigma]\!]u \sim_\pi [\![< \lambda x.N, \psi \circ \sigma >]\!]u = [\![\psi \circ \sigma]\!](u(x).u(v).[\![N]\!]v)$$

  Par B2, B4, $(**)$ et h.r., on vérifie:

  $$\begin{gathered} [\![M\sigma]\!]u \sim_\pi u(x').u(v').[\![(N[x'/x])(\psi \circ \sigma)]\!]v' \sim_\pi \\ u(x').u(v').[\![((N[x'/x])\psi)[H\rho/x](\eta \circ \rho)]\!]v' \sim_\pi \\ u(x').u(v').[\![(N[x'/x])(\psi \circ \zeta)]\!]v' \sim_\pi \\ [\![\psi \circ \zeta]\!](u(x').u(v').[\![N[x'/x]]\!]v') \equiv [\![\zeta]\!]([\![< \lambda x.N, \psi >]\!]u) = [\![M\zeta]\!]u \end{gathered}$$

□

2.3.3 Adéquation du codage

Le codage de λ_j dans π est adéquat, c'est-à-dire,

$$[\![M]\!]p \sqsubseteq_\pi [\![N]\!]p \Rightarrow M \sqsubseteq_\lambda N$$

Ce résultat découle de l'adéquation de la convergence, i.e. $M{\Downarrow_j} \Leftrightarrow [\![M]\!]u{\Downarrow_\pi}$. Nous montrons d'abord comment $[\![\,]\!]$ reproduit les réductions dans λ_j.

Il faut remarquer à ce point que certaines réductions de λ_j ne correspondent pas à des réductions dans π, puisque la substitution indiquée par la règle (β) n'est pas vraiment réalisée par le codage. Cependant, $[\![M\sigma]\!]$ et le terme où la substitution est réalisée sont fortement bisimilaires (i.e. sont capables exactement des mêmes actions) par le théorème 2.3.3. On prouve donc un résultat de correction qui implique la partie $\Rightarrow$ de l'adéquation de la convergence.

Lemme 2.3.4 *Si $L_1 \rightarrow_j L_2$, alors $[\![L_1]\!]u \overset{\star}{\rhd}\sim_\pi [\![L_2]\!]u$ pour tout canal u.*

Preuve. Par cas sur la dernière règle appliquée dans la dérivation de $L_1 \rightarrow_j L_2$. Toutes les substitutions sont normalisées.

(β) Ici, $L_1 =< \lambda x.M, \sigma > N \rightarrow_j M[N/x]\sigma =_s L_2$.

Soit $(\{x\} \cup Var(\sigma)) \cap fv(N) = \emptyset$.

$$\begin{array}{rcl}
\llbracket(<\lambda x.M,\sigma>)N\rrbracket w & = & (\nu u)(\llbracket\sigma\rrbracket(\llbracket\lambda x.M\rrbracket u)|push(N)uw) \\
Var(\sigma)\cap fv(N)=\emptyset & \equiv & (\nu u)(\llbracket\sigma\rrbracket(\llbracket\lambda x.M\rrbracket u|push(N)uw)) \\
z\notin fv(MN) & = & (\nu u)(\llbracket\sigma\rrbracket(u(x).u(v).\llbracket M\rrbracket v|(\nu z)(\overline{u}z.\overline{u}w|\llbracket z:=N\rrbracket)) \\
 & \triangleright & (\nu u)(\llbracket\sigma\rrbracket((\nu z)(u(v).\llbracket M\rrbracket v[z/x]|\overline{u}w|\llbracket z:=N\rrbracket))) \\
x\notin fv(N) & \equiv & (\nu u)(\llbracket\sigma\rrbracket((\nu x)(u(v).\llbracket M\rrbracket v\mid\overline{u}w|\llbracket x:=N\rrbracket))) \\
 & \triangleright & \llbracket\sigma\rrbracket((\nu x)(\llbracket M\rrbracket w|\llbracket x:=N\rrbracket) \\
 & = & \llbracket M\,[N/x]\sigma\rrbracket w \\
\text{par le théorème 2.3.3} & \sim_\pi & \llbracket L_2\rrbracket w
\end{array}$$

$(saisie)$ Ici, $L_1 =< x, [L_2/x]\sigma > \rightarrow_j L_2$.

$$\begin{array}{rcl}
\llbracket< x,[L_2/x]\sigma >\rrbracket u & = & \llbracket\sigma\rrbracket((\nu x)(\overline{x}u|\llbracket x:=L_2\rrbracket)) \\
 & \triangleright & \llbracket\sigma\rrbracket((\nu x)(\llbracket L_2\rrbracket u|\llbracket x:=L_2\rrbracket)) \\
\text{par 2.3.2, puisque } [L_2/x]\sigma \text{ est normalisée} & \equiv & \llbracket L_2\rrbracket u
\end{array}$$

(chl) Ici, $L_1 = (L_2 \oplus N)\rightarrow_j L_2$.

$$\begin{array}{rcl}
\llbracket(L_2\oplus N)\rrbracket u & = & (\nu v)(\overline{v}u\mid v(w).\llbracket L_2\rrbracket w\mid v(w).\llbracket N\rrbracket w) \\
 & \triangleright & (\nu v)(\llbracket L_2\rrbracket u\mid v(w).\llbracket N\rrbracket w) \\
 & \equiv & \llbracket L_2\rrbracket u\mid(\nu v)v(w).\llbracket N\rrbracket w \\
\text{par la proposition 2.1.12} & \sim_\pi & \llbracket L_2\rrbracket u
\end{array}$$

(chr) Comme dans le cas précédent.

(pop) Ici, $L_1 =< x, [M/y]\sigma > \rightarrow_j L_2$ avec $< x, \sigma > \rightarrow_j L_2$.

$$\begin{array}{rcl}
\llbracket< x,[N/y]\sigma >\rrbracket u & = & \llbracket\sigma\rrbracket((\nu y)(\llbracket x\rrbracket u|\llbracket y:=N\rrbracket)) \\
x\neq y \text{ et proposition } 2.1.12 & \sim_\pi & \llbracket\sigma\rrbracket(\llbracket x\rrbracket u) \\
\text{par h.r.} & \triangleright & \llbracket L_2\rrbracket u
\end{array}$$

(app) Ici, $L_1 = (MN)\rightarrow_j(M'N) = L_2$ avec $M\rightarrow_j M'$.

$$\begin{array}{rcl}
\llbracket MN\rrbracket u & = & (\nu w)(\llbracket M\rrbracket w\mid(\nu z)(\overline{w}z.\overline{w}u\mid\llbracket z:=N\rrbracket) \\
\text{par h.r.} & \overset{\star}{\triangleright}\sim_\pi & (\nu w)(\llbracket M'\rrbracket w\mid(\nu z)(\overline{w}z.\overline{w}u\mid\llbracket z:=N\rrbracket) = \llbracket M'N\rrbracket u
\end{array}$$

(obs) Ici, $L_1 = \mathbf{c}N\rightarrow_j I = L_2$ avec $N\rightarrow_j V$.

$$\begin{array}{rcl}
\llbracket\mathbf{c}N\rrbracket u & = & (\nu w)(w(x).w(v).(\nu r)\overline{x}r.(\nu y)\overline{r}y.\llbracket I\rrbracket v\mid \\
 & & (\nu z)(\overline{w}z.\overline{w}u\mid\llbracket z:=N\rrbracket)) \\
 & \overset{\star}{\triangleright} & (\nu z)((\nu r)(\overline{z}r.(\nu y)\overline{r}y.\llbracket I\rrbracket u)\mid\llbracket z:=N\rrbracket) \\
 & \triangleright & (\nu z)(\nu r)(\llbracket N\rrbracket r\mid(\nu y)\overline{r}y.\llbracket I\rrbracket u\mid\llbracket z:=N\rrbracket) \\
\text{par h.r.} & \overset{\star}{\triangleright}\sim_\pi & (\nu z)(\nu r)(\llbracket V\rrbracket r\mid(\nu y)\overline{r}y.\llbracket I\rrbracket u\mid\llbracket z:=N\rrbracket) \\
 & \triangleright & \llbracket I\rrbracket u\mid(\nu z)\llbracket z:=N\rrbracket\mid(\nu r)(\nu y)... \\
\text{par la proposition } 2.1.12 & \sim_\pi & \llbracket I\rrbracket u
\end{array}$$

$(trans)$ Ici, $L_1\rightarrow_j L_2$ avec $L_1\rightarrow_j L$ et $L\rightarrow_j L_2$.

Par la transitivité de $\triangleright$ et $\sim_\pi$.

□

Théorème 2.3.5 *(Adéquation de la convergence)*
Pour tout terme $M \in \Lambda_j^o$, $M\Downarrow_j \Leftrightarrow [\![M]\!]u\Downarrow_\pi$.

Preuve.
Cas $\Rightarrow$**:** Soit $M\Downarrow_j V$; i.e. $M\rightarrow_j V$. On a $[\![M]\!]u \overset{\star}{\rhd}\sim_\pi [\![V]\!]u$ par application du lemme 2.3.4. Comme nous l'avons souligné dans 2.3.1, $[\![V]\!]u \downarrow u$. Donc $[\![M]\!]u\Downarrow_\pi$.
Cas $\Leftarrow$**:** On prouve le raffinement suivant de l'énoncé:

$$\forall M \in \Lambda_j^o \ \forall P \in \Pi \ \ [\![M]\!]u \ \overset{\star}{\rhd} P \overset{u(v)}{\rightarrow} \quad \text{implique un des point suivants}$$

1. $P \sim_\pi [\![< \lambda x.A, \rho >]\!]u$ et $M\rightarrow_j < \lambda x.A, \rho >$
2. $P \sim_\pi [\![\mathbf{c}]\!]u$ et $M\rightarrow_j \mathbf{c}$.

Par la suite, $\overset{n}{\rhd}$ dénote une réduction de longueur n. Supposons $[\![M]\!]u \ \overset{n}{\rhd} P \overset{u(v)}{\rightarrow}$. la preuve est par récurrence sur n.

Le cas $n = 0$ est immédiat, comme nous l'avons remarqué dans 2.3.1(1). Dans le cas où $n = k + 1$, on analyse la structure de M :

- si $M =< x, \sigma >$, alors $[\![M]\!]u \rhd [\![M']\!]u \overset{k}{\rhd} P$, car la réduction de $[\![M]\!]u$ n'est pas vide. Le terme M' est t.q. $[M'/x]$ est la première entrée de substitution pour x dans σ, de gauche à droite. Par l'hypothèse de récurrence, on a :

  1. soit $P \sim_\pi [\![< \lambda x.A, \rho >]\!]u$ et $M'\rightarrow_j < \lambda x.A, \rho >$
  2. soit $P \sim_\pi [\![\mathbf{c}]\!]u$ et $M'\rightarrow_j \mathbf{c}$.

  D'où l'énoncé, puisque $M =< x, \sigma > \rightarrow_j M'$ par application de (pop) et de (saisie).

- si $M = M_1M_2$, examinons la définition de $[\![M]\!]u$, $(\nu r)([\![M_1]\!]r \mid push(M_2)ru)$: Pour ramener u à la surface, $push(M_2)ru$ doit être consommé; la réduction de $[\![M]\!]u$ est divisée en :

  $$\begin{array}{lcl} (\nu r)([\![M_1]\!]r \mid push(M_2)ru) & \overset{n_1}{\rhd} & (\nu r)(R \mid push(M_2)ru) \ = \ S \\ & \overset{n_2}{\rhd} & P \end{array}$$

  où $n = n_1 + n_2$, $n_1 \geq 0$ et $[\![M_1]\!]r \ \overset{n_1}{\rhd} \ R \overset{r(v)}{\rightarrow}$. Par h.r. on a

  - soit $S \sim_\pi [\![< \lambda x.B, \delta > M_2]\!]u$ et $M_1\rightarrow_j < \lambda x.B, \rho >$
  - soit $S \sim_\pi [\![\mathbf{c}M_2]\!]u$ et $M_1\rightarrow_j \mathbf{c}$.

  Dans le premier cas, il existe $P'\overset{u(v)}{\rightarrow}$ t.q.

  $$S \sim_\pi [\![< \lambda x.B, \delta > M_2]\!]u \overset{n_2}{\rhd} P' \sim_\pi P$$

De plus, la réduction de $\llbracket < \lambda xB, \delta > M_2 \rrbracket u$ ne peut être qu'une simulation de la règle (β). Soit $M' \in \Lambda_j$ t.q. $M' =_s B[M_2/x]\delta$; le lemme 2.3.4 nous permet d'affirmer :

$$\llbracket < \lambda x.B, \delta > M_2 \rrbracket u \overset{+}{\triangleright} \sim_\pi \llbracket M' \rrbracket u \overset{n_3}{\triangleright} P'$$

Remarquons que $n_3 < n$. Alors, l'hypothèse de récurrence sur $\llbracket M' \rrbracket u$ donne :

1. soit $P \sim_\pi P' \sim_\pi \llbracket < \lambda x.A, \rho > \rrbracket u$ et $M \rightarrow_j (< \lambda x.B, \delta >) M_2 \rightarrow_j M' \rightarrow_j < \lambda x.A, \rho >$,
2. soit $P \sim_\pi P' \sim_\pi \llbracket \mathbf{c} \rrbracket u$ et $M \rightarrow_j (< \lambda x.B, \delta >) M_2 \rightarrow_j M' \rightarrow_j \mathbf{c}$.

Dans le cas 2, il existe $P' \overset{u(v)}{\rightarrow}$ t.q.

$$S \sim_\pi \llbracket \mathbf{c} M_2 \rrbracket u \overset{n_2}{\triangleright} P' \sim_\pi P$$

De plus, la réduction de $\llbracket \mathbf{c} M_2 \rrbracket u$ doit être:

$$\llbracket \mathbf{c} M_2 \rrbracket u \overset{n_4}{\triangleright} (\nu z)(\nu r)(T \,|\, (\nu y)\overline{r}y.\llbracket I \rrbracket u \,|\, \llbracket z := M_2 \rrbracket) \overset{+}{\triangleright} P' \overset{u(v)}{\rightarrow}$$

Pour permettre une transition d'entrée sur u, $\llbracket M_2 \rrbracket r \overset{+}{\triangleright} T \overset{r(v)}{\rightarrow}$. Alors $P \sim_\pi P' \sim_\pi \llbracket I \rrbracket u$ par h.r. Par conséquent, $M \rightarrow_j (\mathbf{c} M_2) \rightarrow_j I$ puisque par h.r. on a

- soit $T \sim_\pi \llbracket < \lambda x.L, \psi > \rrbracket r$ et $M_2 \rightarrow_j < \lambda x.L, \psi >$,
- soit $T \sim_\pi \llbracket \mathbf{c} \rrbracket r$ et $M_2 \rightarrow_j \mathbf{c}$.

- si $M = (M_1 \oplus M_2)$, alors le codage de M oblige à choisir une des branches, i.e.

$$\llbracket M \rrbracket u \triangleright \llbracket M_i \rrbracket u \overset{+}{\triangleright} P$$

avec $i = 1$ ou $i = 2$. L'énoncé découle de l'hypothèse de récurrence sur M_i et du fait que $M \rightarrow_j M_i$ par (chl) ou (chr).

□

Proposition 2.3.6 *Pour tout $M, N \in \Lambda_j$ et λ_j-contexte C, si $\llbracket M \rrbracket p \sqsubseteq_\pi \llbracket N \rrbracket p$ alors $\llbracket C[M] \rrbracket p \sqsubseteq_\pi \llbracket C[N] \rrbracket p$.*

Preuve. La définition compositionnelle du codage garantit que pour tout λ_j-contexte C, il existe un π-contexte D et un canal u t.q. $\llbracket C[M] \rrbracket p = D[\llbracket M \rrbracket u]$. Soit $E[\llbracket C[M] \rrbracket p] \Downarrow_\pi$, où E est un π-contexte. Alors $E[D[\llbracket M \rrbracket u]] \Downarrow_\pi$. Par hypothèse $E[D[\llbracket N \rrbracket u]] \Downarrow_\pi$, donc $E[\llbracket C[N] \rrbracket p] \Downarrow_\pi$. □

Théorème 2.3.7 ***(Adéquation du codage)***
Pour tout $M, N \in \Lambda_j$, si $\llbracket M \rrbracket p \sqsubseteq_\pi \llbracket N \rrbracket p$ alors $M \sqsubseteq_j N$.

Preuve. Supposons $C[M] \Downarrow_j$. Par le théorème 2.3.5 on a $\llbracket C[M] \rrbracket p \Downarrow_\pi$, d'où

$$\llbracket C[M] \rrbracket p \sqsubseteq_\pi \llbracket C[N] \rrbracket p$$

par la proposition 2.3.6. Alors $\llbracket C[N] \rrbracket p \Downarrow_\pi$, et une nouvelle application du théorème 2.3.5 nous permet de conclure $C[N] \Downarrow_j$. □

2.4 Le rôle des ressources dans le λ-calcul

En ce qui concerne la question de l'adéquation complète du codage, nous avons mentionné dans l'introduction de ce chapitre l'impossibilité de briser la η-conversion avec les contextes du (des) lambda calcul(s). Par contre, le π-calcul permet la construction de contextes qui réalisent des substitutions partielles dans le terme codé. Prenons un exemple : soit $M = x(\lambda y.xy)$ et $N = xx$; il est clair que $M \sqsubseteq_j N$. Cependant, en plongeant les traductions de ces termes dans le π-contexte $C = (\nu x)([]|x(w).[\![\mathbf{I}]\!]w)$, on vérifie

$$C[[\![M]\!]u] \triangleright (\nu x)([\![\mathbf{I}]\!]w|push(x)wu) \overset{\star}{\triangleright} (\nu x)[\![x]\!]u \Uparrow_{\pi}$$

$$C[[\![N]\!]u] \triangleright (\nu x)([\![\mathbf{I}]\!]w|push(\lambda y.xy)wu) \overset{\star}{\triangleright} (\nu x)[\![\lambda y.xy]\!]u \downarrow u$$

C'est exactement ce type de contexte que le calcul avec ressources permet de définir. Si l'on se souvient de la syntaxe de λ_r présentée dans l'introduction, le λ_r-contexte qui joue le rôle de C est tout simplement $[]\langle \mathbf{I}/x \rangle$, où $\mathbf{I}$ désigne *une* ressource, qui ne peut être utilisée qu'une seule fois pendant l'évaluation.

Il est peut-être utile d'indiquer pourquoi l'approche de Sangiorgi [72] évite ce genre de contre-exemple. Il considère le lambda calcul faible $\lambda_{\uplus}$, augmenté avec l'opérateur non-déterministe $\uplus$ qui vérifie les réductions

$$\uplus M \rightarrow_{\uplus} M \quad \text{et} \quad \uplus\; M \rightarrow_{\uplus} \Omega$$

L'extension de la bisimulation applicative définie par Sangiorgi, $\simeq_{\uplus}$, est la suivante : $M \simeq_{\uplus} N$ ssi il existe une relation binaire symétrique $\mathcal{R}$, sur les $\lambda_{\uplus}$-termes, t.q. les deux clauses ci-dessous sont vérifiées :

- si $M \overset{\star}{\rightarrow}_{\uplus} \lambda x.M'$ alors $N \overset{\star}{\rightarrow}_{\uplus} \lambda x.N'$ et $\forall L \in \Lambda^{o}_{\uplus}.\ M'[L/x] \,\mathcal{R}\, N'[L/x]$
- si $M \overset{\star}{\rightarrow}_{\uplus} M'$ alors $N \overset{\star}{\rightarrow}_{\uplus} N'$ et $M' \,\mathcal{R}\, N'$

Il faut souligner que la bisimulation applicative étendue ne coïncide pas avec l'équivalence observationnelle à la Morris. Le résultat est que, sur les λ-termes clos, $\lambda_{\uplus}$ a le même pouvoir de discrimination que π, considéré avec une sémantique par bisimulation[18] :

$$\forall M, N \in \Lambda^{o} \;\; [\![M]\!]u \sim^{w} [\![N]\!]u \Leftrightarrow M \simeq_{\uplus} N$$

Revenons au contre-exemple; pourquoi $\lambda x.x(\lambda y.xy) \not\simeq_{\uplus} \lambda x.xx$? En posant $L = \uplus I$ dans la première clause de la définition de $\simeq_{\uplus}$ il faudrait prouver $\lambda y.(\uplus I)y \simeq_{\uplus} \uplus I$; mais ceci n'est pas vérifié puisque $\uplus I$ se réduit sur Ω.

---

[18] Sur les lambda termes la sémantique par bisimulation et la sémantique contextuelle $\sqsubseteq_{\pi}$ coïncident [73].

Chapitre 3

Ressources dans le lambda calcul

Nous étudions dans ce chapitre divers lambda calculs avec ressources, du point de vue de la sémantique opérationnelle (ou observationnelle).

Les deux premières sections présentent le calcul avec ressources λ_r, non-déterministe, et le calcul avec multiplicités λ_m, sous-calcul déterministe de λ_r, définis par Boudol [17]. Ces calculs sont des extensions propres du lambda calcul faible; ils introduisent la notion de blocage par le moyen d'arguments "épuisables", constitués d'un nombre possiblement fini de ressources. L'évaluation est présentée sous forme d'une machine abstraite qui retarde les substitutions; i.e. la substitution d'une variable par un terme se fait seulement en position de tête. Les substitutions explicites de λ_r et λ_m jouent un rôle essentiel dans la définition de ce mécanisme.

Dans la section 3.3 nous définissons le calcul avec paquets λ_d, dont le langage de termes est comme dans λ_r mais sans substitutions explicites. L'évaluation repose sur une substitution non-déterministe. Nous montrons que l'évaluation de λ_r est une stratégie normalisante de l'évaluation dans λ_d : le pouvoir de discrimination ne diffère pas d'un calcul à l'autre. La définition des équations sémantiques que les modèles du calcul avec ressources doivent vérifier est motivée par la notion de substitution dans λ_d. De plus, les approximants des termes de λ_r, étudiés dans la section 3.8, sont définis sur λ_d.

La section 3.4 contient la définition du calcul λ_r^c avec ressources et test de convergence.

La section 3.5 contient la définition de la sémantique observationnelle des calculs avec ressources. Nous montrons des lemmes des contextes pour λ_r^c et λ_d, et des propriétés de la sémantique observationnelle qui seront utilisées par la suite.

La théorie faible $\simeq_r$ associée à λ_r est caractérisée dans la section 3.6. Nous adaptons les définitions de degré de fonctionnalité et de non-résolution pour λ_r et montrons que $\simeq_r$ est une théorie "fully-lazy" maximale. La section se termine par une comparaison entre le pouvoir de discrimination de λ_r et celui de λ_r^c. On prouve que λ_r^c, sur des termes de λ_r, est strictement plus discriminant que λ_r.

Cette propriété est à la source du résultat de non complète adéquation du modèle de cônes pour λ_r (nous traiterons ce point dans la section 5.6.3.)

Dans la section 3.7 on construit les approximants des termes avec ressources et on montre que la sémantique algébrique engendrée, intensionnelle, est adéquate (mais pas complètement) par rapport à la sémantique observationnelle. Nous montrons aussi qu'une notion d'approximant pour le calcul de multiplicités n'est pas facilement concevable.

Dans la section 3.8 on énonce d'abord les résultats de [18, 19, 20] à propos du pouvoir de discrimination des multiplicités sur les λ-termes. Nous nous intéressons ensuite à deux problèmes de définissabilité : du test de convergence dans λ_r, et du choix non-déterministe dans λ_m. Felleisen [33] développe une théorie de la définissabilité qui a comme (méta)théorème principal l'énoncé suivant : si un constructeur altère l'équivalence observationnelle du langage étendu, il est impossible d'exprimer ce constructeur dans le sous-langage restreint. Cependant, l'énoncé inverse n'est pas vérifié. La non-définissabilité du test de convergence dans λ_r est un corollaire immédiat de ce théorème et du contre-exemple mentionné auparavant. On donne aussi une preuve syntaxique simple de ce résultat. En ce qui est du choix non-déterministe, on prouve que la syntaxe du calcul des multiplicités ne permet pas de définir cet opérateur, même en considérant des stratégies d'évaluation non-déterministes. La question de savoir si le choix non-déterministe permet de séparer davantage de termes de Λ_m que λ_m même reste ouverte.

3.1 Lambda calcul avec ressources λ_r

La syntaxe des termes de λ_r est définie par la grammaire suivante :

$$(\Lambda_r) \quad M ::= x \mid \lambda x.M \mid (MP) \mid M\langle P/x\rangle$$

$$P ::= \mathbf{1} \mid M \mid (P \mid P) \mid M^{\infty}$$

On adopte la convention de noter $L, M, N, \ldots$ les termes du langage et $P, Q, \ldots$ les arguments ou *paquets*. On désignera par T termes ou arguments sans distinction. De plus, $R, S, \ldots$ représentent indistinctement des arguments ou des entrées de substitution $\langle P/x\rangle$; on utilise $\tilde{R}$ comme abréviation de la séquence $R_1 \ldots R_n$ lorsqu'il n'y a pas d'ambiguïté sur la valeur de n, et on définit $|\tilde{R}|$ comme le nombre d'arguments de la séquence $\tilde{R}$, sans compter ses entrées de substitution. Pour des séquences composées exclusivement d'entrées de substitution on écrit souvent $\langle\tilde{P}/\tilde{x}\rangle$, ou encore $\rho, \rho' \ldots$ au lieu de $\langle P_1/x_1\rangle \ldots \langle P_n/x_n\rangle$.

À la caractérisation habituelle des variables libres et des variables liées d'un terme dans le lambda calcul, on ajoute que les variables libres (liées) des arguments P et des entrées de substitution $\langle P/x\rangle$ sont les variables libres (liées) de

P, et que les occurrences libres de x dans M sont liées dans $M\langle P/x\rangle$. Évidemment, $fv(\mathbf{1}) = bv(\mathbf{1}) = var(\mathbf{1}) = \emptyset$ et aussi $fv(P \mid Q) = fv(P) \cup fv(Q)$ et $bv(P \mid Q) = bv(P) \cup bv(Q)$.

On considère les termes de λ_r à α-conversion près. L'α-conversion $M{=_\alpha}N$ est la congruence engendrée par les clauses

$$\begin{array}{rcl} \lambda x.M & = & \lambda z.M[z/x] \quad \text{où } z \notin var(M) \\ M\langle P/x\rangle & = & (M[z/x])\langle P/z\rangle \quad \text{où } z \notin var(M) \end{array}$$

où le renommage de x par z dans M, noté $M[z/x]$, avec $z \notin var(M)$ est défini comme suit :

$$\begin{array}{rcl} y[z/x] & = & \begin{cases} z & \text{si } y = x \\ y & \text{sinon} \end{cases} \\[2ex] (\lambda y.M)[z/x] & = & \begin{cases} \lambda y.M & \text{si } y = x \\ \lambda y.M[z/x] & \text{sinon} \end{cases} \\[2ex] (MP)[z/x] & = & (M[z/x])(P[z/x]) \\[2ex] (M\langle P/y\rangle)[z/x] & = & \begin{cases} M\langle P[z/x]/y\rangle & \text{si } y = x \\ (M[z/x])\langle P[z/x]/y\rangle & \text{sinon} \end{cases} \\[2ex] \mathbf{1}[z/x] & = & \mathbf{1} \\[2ex] (P \mid Q)[z/x] & = & (P[z/x] \mid Q[z/x]) \\[2ex] (M^\infty)[z/x] & = & ((M[z/x])^\infty) \end{array}$$

Le caractère de multi-ensemble des paquets et la signification des ressources infinies sont exprimés dans la congruence $\equiv \subseteq \Lambda_r \times \Lambda_r$, qui permet de permuter les composantes des paquets et de développer les paquets infinis d'un terme.

$$\boxed{\begin{array}{cc} (\mathbf{1} \mid P) \equiv P & M^\infty \equiv (M \mid M^\infty) \\[2ex] (P \mid Q) \equiv (Q \mid P) & (P \mid (Q \mid T)) \equiv ((P \mid Q) \mid T) \\[2ex] \multicolumn{2}{c}{P \equiv Q \ \Rightarrow \begin{cases} MP \equiv MQ \\ M\langle P/x\rangle \equiv M\langle Q/x\rangle \end{cases}} \end{array}}$$

FIG. 3.1 - *Équivalence structurelle* $\equiv$

La relation d'évaluation $\rightarrow_r$ du calcul λ_r répond à la stratégie faible adoptée par Abramsky et Ong pour le lambda calcul : ni les corps des abstractions, ni les arguments dans les applications, ni les entrées de substitution ne sont évalués. Les règles de évaluation se divisent en deux parties : la première, définie dans la figure 3.2, formalise la β-réduction faible dans un calcul avec substitutions explicites; la seconde, définie dans la figure 3.3, établit un mécanisme de *saisie* de ressources, noté $\succ$, qui sert à réaliser la substitution de façon différée.

$$(\beta)\ (\lambda x.M)P \rightarrow_r M\langle P/x\rangle \qquad (v)\ (\lambda x.M)\langle P/z\rangle \rightarrow_r \lambda x.(M\langle P/z\rangle)\ \ x \notin fv(z,P)$$

$$\frac{N \rightarrow_r N' \quad (M =_\alpha N \ \text{ ou } \ M \equiv N)}{M \rightarrow_r N'} \qquad \frac{M \rightarrow_r M'}{MP \rightarrow_r M'P} \qquad \frac{M \rightarrow_r M'}{M\langle P/x\rangle \rightarrow_r M'\langle P/x\rangle}$$

FIG. 3.2 - *Évaluation dans* λ_r

$$\text{(saisie)}\ \frac{M\langle N/x\rangle \succ M' \quad x \notin fv(N)}{M\langle (N \mid Q)/x\rangle \rightarrow_r M'\langle Q/x\rangle}$$

$$x\langle N/x\rangle \succ N \qquad \frac{M\langle N/x\rangle \succ M'}{(MP)\langle N/x\rangle \succ M'P} \qquad \frac{M\langle N/x\rangle \succ M' \quad (x \neq z \ \&\ z \notin fv(N))}{M\langle P/z\rangle\langle N/x\rangle \succ M'\langle P/z\rangle}$$

FIG. 3.3 - *Mécanisme de saisie*

Les dérivations $M \succ M'$, où $M = xR_1 \dots R_n\langle N/x\rangle$ et l'occurrence de tête de la variable x dans M n'est pas liée par $\tilde{R}$, sont de la forme

$$\begin{array}{c} x\langle N/x\rangle \succ N \\ \hline xR_1\langle N/x\rangle \succ NR_1 \\ \vdots \\ \hline xR_1 \dots R_{n-1}\langle N/x\rangle \succ NR_1 \dots R_{n-1} \\ \hline M = xR_1 \dots R_n\langle N/x\rangle \succ M' = NR_1 \dots R_n \end{array}$$

En raison des conditions sur les variables qui accompagnent certaines règles, une telle dérivation n'est possible que si aucune variable libre de N n'est capturée.

Remarquons que les arguments de λ_r se comportent effectivement comme des multi-ensembles de termes : l'opération de saisie ne suit aucun critère dans la consommation de ressources; le choix de la copie à utiliser est non-déterministe grâce à la règle $N \rightarrow_r N'$ & $M \equiv N \Rightarrow M \rightarrow_r N'$ de la figure 3.2. Ce non-déterminisme du processus d'évaluation est mis en évidence par le “codage” du choix interne [17] :

$$(M \oplus N) \overset{def}{=} x\langle (M \mid N)/x \rangle$$

Par la commutativité de la composition parallèle et la définition de saisie dans le calcul, les réductions $(M \oplus N) \rightarrow_r M\langle N/x\rangle$ et $(M \oplus N) \rightarrow_r N\langle M/x\rangle$ sont toutes deux démontrables, pourvu que $x \notin fv(M, N)$. Les résultats $M\langle N/x\rangle$ et $N\langle M/x\rangle$ sont essentiellement M et N, respectivement.

Nous avons souligné dans l'introduction que les termes irréductibles sont de deux types, à savoir : termes bloqués - où il n'y a pas de ressource disponible pour la variable de tête, ou abstractions. Les valeurs du calcul constituent en général un sous-ensemble des termes irréductibles. L'approche standard de la sémantique observationnelle est basée sur une notion d'observation qui assimile les termes bloqués aux termes divergents; les termes observables sont les abstractions :

$$\mathbb{V}_r = \{\lambda x.M \,/\, M \in \Lambda_r\}$$

Le prédicat de convergence $\Downarrow_r$ sur les termes clos est défini par :

$$M \Downarrow_r V \overset{def}{\Leftrightarrow} (V \in \mathbb{V}_r \ \& \ M \overset{\star}{\rightarrow}_r V)$$

Avec an abus de notation, $M \Downarrow_r$ ssi $\exists V \,, M \Downarrow_r V$, et on écrit $M \Downarrow_r^n V$ ou simplement $M \Downarrow_r^n$ lorsque l'évaluation convergente de M (vers V) est de longueur n à utilisation de la règle (v) près.

Définition 3.1.1 ***(Compatibilité entre termes)***

Étant donné $M \in \Lambda_r$, on définit les fonctions abs, tt et arg par :

$$M = \lambda x_1 \ldots x_m . x_i \tilde{R} \quad \Rightarrow \quad \begin{cases} abs(M) = m \\ tt(M) = i \quad si\ i \leq m \\ tt(M) = 0 \quad si\ x_i \in fv(M) \\ arg(M) = |\,\tilde{R}| \end{cases}$$

$$M = \lambda x_1 \ldots x_m . \Omega \quad \Rightarrow \quad \begin{cases} abs(M) = m \\ tt(M) = 0 \\ arg(M) = 0 \end{cases}$$

Deux termes $N, N' \in \Lambda_r$ sont compatibles*, noté $N \smile N'$, ssi $tt(N) = tt(N') \neq 0$, $abs(N) = abs(N')$ et $arg(N) = arg(N')$.*

3.2 Lambda calcul avec multiplicités λ_m

Le calcul avec multiplicités λ_m est défini sur l'ensemble des termes suivants :

$$(\Lambda_m) \quad M ::= x \mid \lambda x.M \mid (MN^k) \mid M\langle N^k/x\rangle \qquad \text{où } k \in \mathbb{N} \cup \{\infty\}$$

L'évaluation faible $\rightarrow_m$ est définie par les règles des figures 3.2 et 3.3 à l'exception de la règle de saisie, qui doit être remplacée par

$$\text{(saisie)}\ \frac{M\langle N/x\rangle \succ M' \ \ (x \notin fv(N))}{M\langle N^{k+1}/x\rangle \rightarrow_m M'\langle N^k/x\rangle}$$

Par abus de notation, lorsque $k = \infty$, on suppose $k = k + 1$.

Le calcul avec multiplicités est un sous-calcul de λ_r si l'on établit la correspondance suivante : l'argument M^0 de λ_m correspond à **1**, et pour une multiplicité k supérieure à zéro, M^k s'écrit $\underbrace{(M \mid \ldots \mid M)}_{k \text{ copies}}$. Le paquet M^∞ désigne le même objet infini dans λ_m que dans λ_r, inépuisable par évaluation.

Comme dans λ_r, une évaluation dans λ_m termine sur un terme V irréductible ssi V est soit une abstraction soit un terme bloqué. La forme de ces derniers est par exemple $x\tilde{P}\langle M^0/x\rangle$. Par contre, l'évaluation $\rightarrow_m$ est déterministe à α-conversion près. Il suffit d'observer que l'opération de saisie est déterministe : tant qu'il y a des copies de l'argument à consommer, la substitution est réalisée; et c'est seulement lorsque l'argument a été épuisé qu'un blocage se produit.

Proposition 3.2.1 *(Proposition 2.3 [18])*
Si $M \rightarrow_m N$ et $M \rightarrow_m N'$ alors $N =_\alpha N'$.

3.3 Lambda calcul avec paquets λ_d

La présence des substitutions explicites dans le calcul de ressources permet la définition d'un mécanisme de substitution retardée qui engendre la stratégie d'évaluation normalisante $\rightarrow_r$. Nous entendons par cela que, du point de vue de la convergence, toute autre façon de réaliser la substitution est moins performante. Pour formaliser cette affirmation, nous définissons un lambda calcul avec paquets λ_d, sans substitutions explicites, où le processus de substitution est une opération qui ne privilégie aucune stratégie de distribution des ressources (c'est la β-conversion qui est non-déterministe). Une des motivations de la définition du calcul avec paquets est que le traitement des environnements dans les équations sémantiques vérifiées par les modèles de λ_r, définies dans le chapitre 4, correspond exactement à la notion de substitution dans λ_d.

Nous montrons comment simuler la convergence de λ_r dans λ_d et vice-versa. Un résultat de ce type a été prouvé par Abadi et al. [1, 26] pour le calcul faible[1] $\lambda\sigma$ et le lambda calcul faible pur, qui jouent les rôles de λ_r et de λ_d respectivement. Ils montrent que l'évaluation d'un terme a dans $\lambda\sigma$ termine sur une fntf b ssi l'évaluation de $\sigma(a)$ dans le calcul pur termine sur une fntf a' vérifiant $a' = \sigma(b)$, où la fonction $\sigma(c)$ désigne le terme qui résulte du calcul des substitutions explicites dans c.

Malgré la similitude des énoncés, on ne peut s'attendre à pouvoir transposer les preuves car le résultat de [1, 26] repose sur la normalisation forte du calcul des substitutions explicites. Or, notre substitution doit rendre compte de *toutes* les distributions possibles des ressources. Autrement dit, le calcul λ_d est non-déterministe tandis que le calcul faible pur est confluent.

3.3.1 Syntaxe et évaluation

L'ensemble des termes du calcul λ_d est le sous-ensemble de Λ_r sans substitutions explicites, et les valeurs considérées par la sémantique standard sont toujours les abstractions :

$$\begin{array}{ll}
(\Lambda_d) & M ::= x \mid \lambda x.M \mid (MP) \\
 & P ::= \mathbf{1} \mid M \mid (P \mid P) \mid M^{\infty} \\
 & \\
(\mathbb{V}_d) & V ::= \lambda x.M
\end{array}$$

Pour ce qui est de la syntaxe, la seule modification apportée au lambda calcul pur est l'introduction d'arguments composés de termes hétérogènes. Les fonctions fv, bv, var ainsi que le renommage $M[z/x]$ et l'α-conversion sont définis comme d'habitude.

Définition 3.3.1 ***(Substitution dans λ_d)***

Soit $M, P \in \Lambda_d$. La substitution de x par P dans M, notée $M\lceil P/x\rceil$ est la

---

[1] Ici faible désigne l'évaluation qui s'arrête face à une forme normale de tête faible (i.e. une abstraction), mais qui autorise le calcul dans les arguments. Voir théorème 3.6 dans [26].

partie de Λ_d définie par :

$$\begin{array}{rcl}
x\lceil P/x\rceil & = & \{M \,/\, P \equiv (M \mid Q)\} \cup \{\Omega\} \\
y\lceil P/x\rceil & = & \{y\} \\
(\lambda y.M)\lceil P/x\rceil & = & \left\{\begin{array}{ll} \{\lambda z.M' \,/\, \forall z \;\; z \notin fv(\lambda y.M, P, x)\ \& & \\ \qquad\qquad M' \in M[z/y]\lceil P/x\rceil\} & \textit{si } y \neq x \\ \{\lambda y.M\} & \textit{sinon} \end{array}\right. \\
(MQ)\lceil P/x\rceil & = & \{(M'Q') \,/\, P \equiv (P_1 \mid P_2)\ \& \\
 & & \qquad M' \in M\lceil P_1/x\rceil\ \&\ Q' \in Q\lceil P_2/x\rceil\} \\
 & & \\
\mathit{1}\lceil P/x\rceil & = & \{\mathit{1}\} \\
(M \mid Q)\lceil P/x\rceil & = & \{(M' \mid Q') \,/\, P \equiv (P_1 \mid P_2)\ \& \\
 & & \qquad M' \in M\lceil P_1/x\rceil\ \&\ Q' \in Q\lceil P_2/x\rceil\} \\
M^\infty\lceil P/x\rceil & = & \{(M_1 \mid \ldots \mid M_k \mid N^\infty) \,/\, P \equiv (P_1 \mid \ldots \mid P_k)\ \& \\
 & & \qquad M_i \in M\lceil P_i/x\rceil\ \&\ N \in M\lceil \mathit{1}/x\rceil\}
\end{array}$$

La substitution $M\lceil P/x\rceil$ distribue les ressources de P aux sous-termes de M. Une formulation équivalente de la première clause divise la définition en deux cas, selon que le paquet à substituer soit vide ou pas :

$$x\lceil P/x\rceil = \left\{\begin{array}{ll} \{\Omega\} & \text{si } P \equiv \mathbf{1} \\ \{M \,/\, P \equiv (M \mid Q)\} & \text{sinon} \end{array}\right.$$

Par abus de notation on pose $(\lambda y.M)\lceil P/x\rceil = \lambda z.(M[z/y]\lceil P/x\rceil)$ si $y \neq x$.

Lemme 3.3.2 *Soit $z \notin var(M\lceil P/y\rceil)$; alors pour tout N*

$$N \in M\lceil P/y\rceil \;\;\Rightarrow\;\; N[z/x] \in \left\{\begin{array}{ll} (M[z/x])\lceil P[z/x]/y\rceil & \textit{si } x \neq y, \\ M\lceil P[z/x]/y\rceil & \textit{sinon} \end{array}\right.$$

Preuve. La preuve est immédiate, par inspection de la définition de $M\lceil P/y\rceil$ et du renommage. □

Les règles de réduction de λ_d, dans la figure 3.4, sont sont celles de λ, avec une β-conversion non-déterministe.

$$\boxed{(\beta)\ \frac{N \in M\lceil P/x\rceil}{(\lambda x.M)P \rightarrow_d N} \qquad\qquad \frac{M \rightarrow_d M'}{MP \rightarrow_d M'P}}$$

FIG. 3.4 - *Évaluation dans λ_d*

Le prédicat de convergence $\Downarrow_d$ sur Λ_d° est défini dans la figure 3.5.

Lemme 3.3.3 *Pour tout $M \in \Lambda_d$, on a $M \Downarrow_d V$ ssi $M \overset{\star}{\rightarrow}_d V$*

Preuve. Directe, par récurrence sur les tailles des dérivations $M \Downarrow_d V$ et $M \overset{\star}{\rightarrow}_d V$. □

$$\boxed{\lambda x.M \Downarrow_d \lambda x.M \qquad \frac{M\Downarrow_d \lambda x.M' \quad N \in M'\lceil P/x\rceil \quad N\Downarrow_d V}{MP\Downarrow_d V}}$$

FIG. 3.5 - *Prédicat de convergence* $\Downarrow_d$

3.3.2 Transformation de termes de λ_r en termes de λ_d

Étant donné un terme $M \in \Lambda_r$, on peut calculer un sous-ensemble de Λ_d par distribution des substitutions explicites présentes dans M, selon l'opération de substitution $\lceil P/x\rceil$:

Définition 3.3.4 *La transformation* $\{\![-]\!\} : \Lambda_r \to \mathcal{P}(\Lambda_d)$ *est définie par:*

$$\begin{array}{rcl}
\{\![x]\!\} & = & \{x\} \\
\{\![\lambda x.M]\!\} & = & \{\lambda x.N \,/\, N \in \{\![M]\!\}\} \quad \textit{noté } \lambda x.\{\![N]\!\} \\
\{\![MP]\!\} & = & \{NQ \,/\, N \in \{\![M]\!\} \;\&\; Q \in \{\![P]\!\}\} \quad \textit{noté } \{\![M]\!\}\{\![P]\!\} \\
\{\![M\langle P/x\rangle]\!\} & = & \bigcup N\lceil Q/x\rceil \quad \textit{où } N \in \{\![M]\!\} \;\&\; Q \in \{\![P]\!\} \\
 & & \\
\{\![\boldsymbol{1}]\!\} & = & \{\boldsymbol{1}\} \\
\{\![(P \mid Q)]\!\} & = & \{(P' \mid Q') \,/\, P' \in \{\![P]\!\} \;\&\; Q' \in \{\![Q]\!\}\} \\
\{\![M^\infty]\!\} & = & \{(N_1^\infty \mid \ldots \mid N_k^\infty) \,/\, N_i \in \{\![M]\!\} \;\&\; k \in \mathbb{N}\}
\end{array}$$

Par abus de notation, lorsque $P = (Q_1 \mid \ldots \mid Q_n)$ et Q_i est soit un terme M soit un terme avec multiplicité infinie M^∞, on note $P\langle \mathbf{1}/x\rangle$ le paquet de termes $(Q'_1 \mid \ldots \mid Q'_n)$ où, pour tout i on a

$$Q'_i = \begin{cases} M\langle \mathbf{1}/x\rangle & \text{si } Q_i = M \\ (M\langle \mathbf{1}/x\rangle)^\infty & \text{si } Q_i = M^\infty \end{cases}$$

On étend l'α-conversion de λ_d aux ensembles $\{\![M]\!\}$ comme suit :

$$\forall M, M' \in \Lambda_r \;\; \{\![M]\!\} =_\alpha \{\![M']\!\} \overset{def}{\Leftrightarrow}$$

$$(\forall N \in \{\![M]\!\}\; \exists N' \in \{\![M']\!\} \;\; N =_\alpha N') \;\&\; (\forall N' \in \{\![M']\!\}\; \exists N \in \{\![M]\!\} \;\; N' =_\alpha N)$$

Il est clair que la transformation n'introduit pas de variables libres, i.e. $N \in \{\![M]\!\} \;\Rightarrow\; fv(N) \subseteq fv(M)$. Parmi les propriétés de la transformation $\{\![\,]\!\}$ on a

Lemme 3.3.5 *1.* $\{\![(MP)\langle \boldsymbol{1}/x\rangle]\!\} = \{\![(M\langle \boldsymbol{1}/x\rangle)(P\langle \boldsymbol{1}/x\rangle)]\!\}$

2. $x \neq y \;\Rightarrow\; \{\![(M\langle P/y\rangle)\langle \boldsymbol{1}/x\rangle]\!\} = \{\![(M\langle \boldsymbol{1}/x\rangle)(\langle P\langle \boldsymbol{1}/x\rangle)/y\rangle]\!\}$.

3. $\{\![(MP)\langle N/x\rangle]\!\} = \{\![(M\langle N/x\rangle)(P\langle \boldsymbol{1}/x\rangle)]\!\} \cup \{\![(M\langle \boldsymbol{1}/x\rangle)(P\langle N/x\rangle)]\!\}$

4. $x \neq y \ \& \ y \notin fv(N) \ \Rightarrow$

$$\{\![(M\langle P/y\rangle)\langle N/x\rangle]\!\} = \{\![(M\langle N/x\rangle)(\langle P/y\rangle\langle \boldsymbol{1}/x\rangle)]\!\} \cup \{\![(M\langle \boldsymbol{1}/x\rangle)(\langle P/y\rangle\langle N/x\rangle)]\!\}$$

5. $\{\![M]\!\} \subseteq \{\![N]\!\} \ \& \ \{\![P]\!\} \subseteq \{\![Q]\!\} \ \Rightarrow \ \{\![MP]\!\} \subseteq \{\![NQ]\!\} \ \& \ \{\![M\langle P/x\rangle]\!\} \subseteq \{\![N\langle Q/x\rangle]\!\}$

6. $N\lceil P/x\rceil = \{\![N\langle P/x\rangle]\!\}$

7. $x \notin fv(M) \ \Rightarrow \ \forall P \ \{\![M\langle P/x\rangle]\!\} = \{\![M]\!\}$

8. $P \equiv (M \mid P') \ \& \ P \in \{\![Q]\!\} \ \Rightarrow \ \exists N \ \exists Q' \ \ (Q \equiv (N \mid Q') \ \& \ M \in \{\![N]\!\})$

Preuve. Les preuves sont immédiates par inspection de la définition de la transformation. □

Lemme 3.3.6 *1.* $N \in \{\![M]\!\} \ \Rightarrow \ N[z/x] \in \{\![M[z/x]]\!\}$

2. $M {=_\alpha} M' \ \Rightarrow \ \{\![M]\!\} {=_\alpha} \{\![M']\!\}$.

Preuve.

1. Par récurrence sur M. Si M est une variable, alors $N = M$. Si $M = \lambda y.L$ et $x \neq y$, alors $M[z/x] = \lambda y.L[z/x]$. D'autre part, $N = \lambda y.N'$ où $N' \in \{\![L]\!\}$. Donc $N[z/x] = \lambda y.N'[z/x]$ et l'énoncé est vérifié par l'hypothèse de récurrence. Si $x = y$, alors $M[z/x] = M$. Puisque $N = \lambda y.N'$, avec $N' \in \{\![L]\!\}$, on a $N[z/x] = N$. Pour $M = M'P$, on utilise l'hypothèse de récurrence directement. Le dernier cas est $M = M'\langle P/y\rangle$. Alors $N \in N'\lceil Q/y\rceil$ où $N' \in \{\![M']\!\}$ et $Q \in \{\![P]\!\}$. Par le lemme 3.3.2 et l'hypothèse de récurrence,

$$N[z/x] \in (N'\lceil Q/y\rceil)[z/x] \subseteq \{\![M'\langle P/y\rangle[z/x]]\!\}$$

2. On prouve l'énoncé pour les lois de l'alpha-conversion dans λ_r. Soit $z \notin fv(M)$:

$$M = \lambda x.M_0 {=_\alpha} \lambda z.M_0[z/x] = M'$$

Alors, $\{\![M]\!\} = \{\lambda x.N \,/\, N \in \{\![M_0]\!\}\}$ et $\{\![M]\!\} = \{\lambda z.N' \,/\, N' \in \{\![M_0[z/x]]\!\}\}$. Par la partie (1) de ce lemme, $N \in \{\![M_0]\!\}$ implique $N[z/x] \in \{\![M_0[z/x]]\!\}$.

Dans le cas où $M = M_0\langle P/x\rangle {=_\alpha} M_0[z/x]\langle P/z\rangle = M'$, on a $N \in N'\lceil Q/x\rceil$ avec $N' \in \{\![M_0]\!\}$ et $Q \in \{\![P]\!\}$. Par la partie (2) du lemme, $N[z/x] \in N'[z/x]\lceil Q/z\rceil$, donc $N[z/x] \in \{\![M']\!\}$.

□

3.3.3 De l'évaluation $\to_r$ à l'évaluation $\to_d$

Dans cette section on montre comment la convergence dans λ_r peut être simulée dans λ_d.

Lemme 3.3.7

1. *Si* $M\langle N/x\rangle \succ\!\!- L$ *où* $x \notin fv(N)$*, alors* $\{[L\langle \mathbf{1}/x\rangle]\} \subseteq \{[M\langle N/x\rangle]\}$.
2. *Si* $M \overset{saisie}{\to}_r M'$*, alors* $\{[M']\} \subseteq \{[M]\}$.
3. *Si* $M \overset{\beta}{\to}_r M'$*, alors* $\forall N' \in \{[M']\}\ \exists N \in \{[M]\}\ \ N \overset{*}{\to}_d N'$.
4. *Si* $M \overset{v}{\to}_r M'$*, alors* $\{[M]\} = \{[M']\}$.

Preuve.

1. Par récurrence sur la taille de la dérivation de la saisie. Si $M = x$, alors $L = N$. Par le lemme 3.3.5(7) et la définition de la transformation, on a

   $$\{[L\langle \mathbf{1}/x\rangle]\} = \{[L]\} \subseteq \{[L]\} \cup \{\Omega\} = \{[x\langle L/x\rangle]\}$$

   Dans le cas où $M = M'P$, on a $M'\langle N/x\rangle \succ\!\!- L'$ et $L = L'P$: par l'hypothèse de récurrence, $\{[L'\langle \mathbf{1}/x\rangle]\} \subseteq \{[M'\langle N/x\rangle]\}$. Donc, par le lemme 3.3.5 (1),(5),(3)

   $$\{[(L'P)\langle \mathbf{1}/x\rangle]\} = \{[L'\langle \mathbf{1}/x\rangle P\langle \mathbf{1}/x\rangle]\} \subseteq \{[M'\langle N/x\rangle P\langle \mathbf{1}/x\rangle]\}$$

   Le cas où $M = M'\langle P/y\rangle$ et $x \neq y$ et $y \notin fv(N)$, on a $M'\langle N/x\rangle \succ\!\!- L'$ et $L = L'\langle P/y\rangle$. L'énoncé découle de l'hypothèse de récurrence plus des propriétés 3.3.5(2),(5),(4).

2. Par récurrence sur $M \overset{saisie}{\to}_r M'$. Si la règle (saisie) est directement appliquée, on a $M = L\langle P/x\rangle$, $P \equiv (N \mid Q)$, $L\langle N/x\rangle \succ\!\!- L'$ et $M' = L'\langle Q/x\rangle$. Par le point (1) et la propriété 3.3.5(5),

   $$\{[L'\langle Q/x\rangle]\} = \{[L'\langle \mathbf{1} \mid Q/x\rangle]\} \subseteq \{[L\langle N \mid Q/x\rangle]\} = \{[M]\}$$

   Dans le cas où M serait une application ou un terme avec substitution explicite, l'énoncé découle de l'hypothèse de récurrence plus la propriété 3.3.5(5).

3. Par récurrence sur $M \overset{\beta}{\to}_r M'$. Le cas de base correspond à $M = (\lambda x.M_0)P$ et $M' = M_0\langle P/x\rangle$. Alors, quelque soit $N' \in \{[M']\}$, il existe $L \in \{[M_0]\}$ et $Q \in \{[P]\}$ t.q. $N' = L\lceil Q/x\rceil$. Le terme $N \in \{[M]\}$ associé à N' est $(\lambda x.L)Q$: il suffit d'observer que $(\lambda x.L)Q \to_d H$ pour tout $H \in L\lceil Q/x\rceil$, en particulier pour $H = N'$.

On examine maintenant le cas de $M = LP$, $M' = L'P$ et $L \overset{\beta}{\rightarrow}_r L'$. Soit $N' \in \{[L'P]\}$; par définition, $N' = H'Q$ où $H' \in \{[L']\}$ et $Q \in \{[P]\}$. Par l'hypothèse de récurrence, il existe $H \in \{[L]\}$ t.q. $H \overset{\star}{\rightarrow}_d H'$. Le terme $N \in \{[M]\}$ recherché est donc HQ. Le même raisonnement s'applique dans le cas où $M = L\langle P/x\rangle$.

4. Si $M \overset{v}{\rightarrow}_r M'$, alors $M = (\lambda x.M_0)\langle P/y\rangle\tilde{R}$ et $M' = (\lambda x.M_0\langle P/y\rangle)\tilde{R}$ avec $x \notin fv(P, y)$. D'autre part, pour tout $L \in \{[M_0]\}$ et tout $Q \in \{[P]\}$ on a $(\lambda x.L)\lceil Q/y\rceil = \lambda x.L\lceil Q/y\rceil$. On conclut $\{[M]\} = \{[M']\}$.

□

Théorème 3.3.8 *Soit $M, M' \in \Lambda_r$. Alors,*

$$M \overset{\star}{\rightarrow}_r M' \;\Rightarrow\; \forall N' \in \{[M']\}\ \exists N \in \{[M]\}\ \ N \overset{\star}{\rightarrow}_d N'$$

Preuve. Par récurrence sur la longueur de la dérivation $M \overset{\star}{\rightarrow}_r M'$. Dans le cas où c'est zéro, l'énoncé est immédiat car $M = M'$. Supposons que $M \rightarrow_r M_0 \overset{\star}{\rightarrow}_r M'$. L'énoncé est conséquence directe des résultats suivants, obtenus par application du lemme 3.3.7 et de l'hypothèse de récurrence, respectivement :

$$\forall N'' \in \{[M_0]\}\ \exists N \in \{[M]\}\ \ N \overset{\star}{\rightarrow}_d N''$$

$$\forall N' \in \{[M']\}\ \exists N'' \in \{[M_0]\}\ \ N'' \overset{\star}{\rightarrow}_d N'$$

□

Corollaire 3.3.9 *1. Si $M \in \Lambda_r$, alors $(M\Downarrow_r \;\Rightarrow\; \exists N \in \{[M]\}\ \ N\Downarrow_d)$*

2. Si $M \in \Lambda_d$, alors $(M\Downarrow_r \;\Rightarrow\; M\Downarrow_d)$

Preuve.

1. Soit $M \in \Lambda_r$ et $M \overset{\star}{\rightarrow}_r V$ où $V \in \mathbb{V}_r$. Par le théorème 3.3.8, pour tout $W \in \{[V]\}$ il existe $N \in \{[M]\}$ t.q. $N \overset{\star}{\rightarrow}_d W$. Puisque V est une abstraction, W aussi. Donc $N\Downarrow_d$.

2. Ce point est un cas particulier de (1). Il suffit de noter que $M \in \Lambda_d$ implique $\{[M]\} = \{M\}$.

□

3.3.4 De l'évaluation $\rightarrow_d$ à l'évaluation $\rightarrow_r$

Pour montrer comment passer d'une séquence convergente dans λ_d à une séquence convergente dans λ_r il nous faut quelques résultats concernant la forme des éléments des ensembles $\{\![M]\!\}$.

Lemme 3.3.10 *Soit $H \in \Lambda_d$, et $\tilde{P}, R$ des paquets de λ_d (donc sans entrées de substitution.)*

1. *Si $(\lambda x.M)\tilde{Q} \in (z\tilde{P})\lceil R/y\rceil$, alors $z = y$ & $R \equiv ((\lambda x.M)Q_0 \ldots Q_i \mid R')$ & $Q_{i+1} \ldots Q_n \in \tilde{P}\lceil R'/y\rceil$.*

2. *Si $(\lambda x.M)\tilde{Q} \in ((\lambda v.H)\tilde{P})\lceil R/y\rceil$, alors $R \equiv (R' \mid R'')$ & $x \notin fv(\lambda v.H, R, y)$ & $M \in H[x/v]\lceil R'/y\rceil$ & $\tilde{Q} \in \tilde{P}\lceil R''/y\rceil$.*

Preuve.

1. Par définition, $R \equiv (R' \mid R'')$ et $(\lambda x.M)\tilde{Q} \in \{\![(y\lceil R'/y\rceil)(\tilde{P}\lceil R''/y\rceil)]\!\}$. Donc $R' \equiv (N \mid R''')$ et $(\lambda x.M)\tilde{Q} \in N(\tilde{P}\lceil R''/y\rceil)$; c'est-à-dire, $(\lambda x.M)\tilde{Q} = N\tilde{P}'$ où $\tilde{P}' \in \tilde{P}\lceil R''/y\rceil$. D'où $N = (\lambda x.M)\tilde{Q}'$ et $\tilde{P} = \tilde{Q}'\tilde{P}'$.

2. Par récurrence sur la longueur n de $\tilde{Q}$. Si $n = 0$, alors par définition $\tilde{P}$ est la séquence vide. On a soit $v = y$, et donc $\lambda x.M = \lambda v.H$ (i.e. $x = v$ et $M = H$); soit $v \neq y$, et alors :

   $$\lambda x.M \in \{\lambda z.M' \,/\, z \notin fv(\lambda v.H, R, y) \;\&\; M' \in H[z/v]\lceil R/y\rceil\}$$

   C'est-à-dire, $x \notin fv(\lambda v.H, R, y)$ et $M \in H[x/v]\lceil R/y\rceil$.

   Dans le cas où $n = k + 1$, les séquences $\tilde{Q}$ et $\tilde{P}$ ont la même longueur. Par définition, $R \equiv (R_0 \mid R_1)$ et

   $$(\lambda x.M)Q_0 \ldots Q_k \in ((\lambda v.H)P_0 \ldots P_k)\lceil R_0/y\rceil \quad \text{et} \quad Q_{k+1} \in P_{k+1}\lceil R_1/y\rceil$$

   Par l'hypothèse de récurrence, $R_0 \equiv (R' \mid R_2)$ et

   $$M \in H[x/v]\lceil R'/y\rceil \quad \text{et} \quad Q_0 \ldots Q_k \in (P_0 \ldots P_k)\lceil R_2/y\rceil$$

   Donc $\tilde{Q} \in (P_0 \ldots P_k)\lceil R_2/y\rceil P_{k+1}\lceil R_1/y\rceil \subseteq \tilde{P}\lceil (R_2 \mid R_1)/y\rceil$ par définition de la substitution.

□

Lemme 3.3.11 *Soit $\tilde{P}, R$ des paquets du calcul λ_d (donc sans entrées de substitution.)*

1. *Si $x\tilde{Q} \in (y\tilde{P})\lceil R/y\rceil$, alors il existe i t.q. $R \equiv ((xQ_0 \ldots Q_i) \mid R')$ & $Q_{i+1} \ldots Q_n \in \tilde{P}\lceil R'/y\rceil$.*

2. *Si* $x\tilde{Q} \in (x\tilde{P})\lceil R/y \rceil$, *où* $x \neq y$, *alors* $\tilde{Q} \in \tilde{P}\lceil R/y \rceil$.

Preuve. Soit n, m les longueurs des séquences $\tilde{P}$ et $\tilde{Q}$, respectivement. On procède par récurrence sur n. Si $n = 0$, dans le cas (1), par définition, $R \equiv (x\tilde{Q} \mid R')$. Dans le cas (2), $x\tilde{Q} \in \{x\}$ implique $m = 0$.

Si $n = k + 1$, par définition on a $m = k' + 1$.

1. Ici, $R \equiv (R_0 \mid R_1)$ et

$$xQ_0 \dots Q_{k'} \in (yP_0 \dots P_k)\lceil R_0/y \rceil \quad \text{et} \quad Q_m \in P_n\lceil R_1/y \rceil$$

De l'hypothèse de récurrence on déduit

$$\exists i \;\; R_0 \equiv (xQ_0 \dots Q_i \mid R_2) \;\&\; Q_{i+1} \dots Q_{k'} \in (P_0 \dots P_k)\lceil R_2/y \rceil$$

Par conséquent, $R \equiv (xQ_0 \dots Q_i \mid R_2 \mid R_1)$ et

$$Q_{i+1} \dots Q_{k'} Q_m \in (P_0 \dots P_k)\lceil R_2/y \rceil (P_n\lceil R_1/y \rceil \subseteq (P_0 \dots P_n)\lceil (R_2 \mid R_1)/y \rceil$$

2. Par hypothèse de récurrence, comme pour la partie (1).

□

Les deux lemmes qui suivent sont des généralisations des lemmes 3.3.11 et 3.3.10. D'abord une définition :

Définition 3.3.12 ***(Partage des substitutions)***

Étant donnée une séquence d'entrées de substitution ρ, *un* n-partage *de* ρ, *noté* $p_n(\rho)$, *est défini par :*

$$\begin{array}{rcl} \rho = \emptyset & \Rightarrow & p_n(\rho) = (\underbrace{\emptyset, \dots, \emptyset}_{n\ fois}) \\ \rho \equiv \langle P_1 \mid \dots \mid P_n/x \rangle \rho' & \Rightarrow & p_n(\rho) = (\langle P_1/x \rangle, \dots, \langle P_n/x \rangle) \star p_n(\rho') \end{array}$$

où l'opération $\star$ *est la concatenation de listes composante à composante, définie comme suit :*

$$\begin{array}{rcl} (\rho_0, \dots, \rho_n) \star (\emptyset, \dots, \emptyset) & = & (\rho_0, \dots, \rho_n) \\ (\rho_0, \dots, \rho_n) \star ((\xi_0, \dots, \xi_n) \star p_n(\rho)) & = & (\rho_0\xi_0, \dots, \rho_n\xi_n) \star p_n(\rho) \end{array}$$

Lemme 3.3.13 ***(Généralisation du lemme 3.3.11)***

Si $x\tilde{P} \in \{[T]\}$, *alors*

$$T \overset{\star}{\rightarrow}_r x\rho_0 Q_1 \rho_1 \dots Q_n \rho_n$$

où l'occurrence de tête de la variable x *est libre et*

$$\forall i \, \exists P'_i \;\; P_i {=_\alpha} P'_i \;\;\&\;\; \tilde{P}' \in \{[Q_1\rho_1 \dots Q_n\rho_n]\}$$

Preuve. Soit n la longueur de la séquence $\tilde{P}$. On procède par récurrence sur T. Si T est une variable, alors $n = 0$. Par définition de la transformation, T ne peut être une abstraction. Dans le cas où T est une application, l'énoncé découle directement de l'hypothèse de récurrence. Il reste à examiner le cas où $T = T'\langle Q/y\rangle$. Par la définition de la transformation, $x\tilde{P} \in J\lceil R/y\rceil$ avec $J \in \{[T']\}$ et $R \in \{[Q]\}$. Selon le lemme 3.3.11, les possibilités sont :

- $J = (y\tilde{P}'')$, $R \equiv ((xP_1 \ldots P_i) \mid R')$ et $P_{i+1} \ldots P_n \in \tilde{P}''\lceil R'/y\rceil$.

  Par hypothèse de récurrence, $T' \overset{\star}{\rightarrow}_r y S'_i Q'_{i+1}\rho'_{i+1} \ldots Q'_n\rho'_n$, et

  $$\forall j \in [i+1, n]\ \exists P^*_j\ \ \tilde{P}'' =_\alpha \tilde{P}^* \in \{[Q'_{i+1}\rho'_{i+1} \ldots Q'_n\rho'_n]\}$$

  De plus, $R \equiv (N \mid R'')$ où $(xP_1 \ldots P_i) \in \{[N]\}$ et $R' \in \{[R'']\}$. Alors, en faisant les α-conversions nécessaires, et par l'hypothèse de récurrence (N est un sous-terme de T), on a

  $$T \overset{\star}{\rightarrow}_r N S''_i Q_{i+1}\rho_{i+1} \ldots Q_n\rho_n\langle R''/v\rangle \overset{\star}{\rightarrow}_r$$

  $$x\tilde{S}_0 Q_1 \tilde{S}_1 \ldots Q_i \rho'''_i S''_i Q_{i+1}\rho_{i+1} \ldots Q_n\rho_n\langle R''/v\rangle$$

  et $\forall j \in [1, i] \exists P'_j\ \ P_1 \ldots P_i =_\alpha P'_1 \ldots P'_i \in .\{[Q_1\tilde{S}_1 \ldots Q_i\rho'''_i]\}$. Par ailleurs,

  $$\{[Q'_{i+1}\rho'_{i+1} \ldots Q'_n\rho'_n\langle R''/y\rangle]\} =_\alpha \{[Q_{i+1}\rho_{i+1} \ldots Q_n\rho_n\langle R''/v\rangle]\} = \{[T'']\}$$

  implique $\exists P'_1, \ldots, P'_i\ \ P_1 \ldots P_i =_\alpha P'_1 \ldots P'_i \in \{[T'']\}$.

- $J = (x\tilde{P}'')$ et $\tilde{P} \in \tilde{P}''\lceil R/y\rceil$. L'énoncé découle immédiatement de l'hypothèse de récurrence sur $J \in \{[T']\}$.

□

Lemme 3.3.14 ***(Généralisation du lemme 3.3.10)***
Si $(\lambda x.M)Q_1 \ldots Q_n \in \{[T]\}$, *alors*

$$T \overset{\star}{\rightarrow}_r (\lambda z.M')\rho_0 Q'_1\rho_1 \ldots Q'_n\rho_{n+1}$$

et pour tout $j \in [1, n+1]$ *il existe un* 2*-partage* $p_2(\rho_j) = (\rho'_j, \rho''_j)$ *t.q.*

$$\lambda x.M \in \{[(\lambda z.M')\rho_0\rho'_1 \ldots \rho'_{n+1}]\}\ \ \&$$

$$\forall i\, \exists P_i\ Q_i =_\alpha P_i\ \ \&\ \ \tilde{P} \in \{[Q'_1\rho''_1 \ldots Q'_n\rho''_{n+1}]\}$$

Preuve. Par récurrence sur T, qui n'est pas une variable par définition de la transformation. Si $T = \lambda z.L$, alors $z = x$, $n = 0$ et $M \in \{[L]\}$. Si $T = T'R$, alors $n \geq k+1$. Par définition, $(\lambda x.M)P_1 \ldots P_k \in \{[T']\}$ et $P_n \in \{[R]\}$. L'énoncé découle de l'hypothèse de récurrence.

Finalement, dans le cas où $T = T'\langle R/y\rangle$, par la définition de la transformation,

$$(\lambda x.M)\tilde{Q} \in J\lceil R'/y\rceil \quad \text{où } J \in \{[T']\} \ \& \ R' \in \{[R]\}$$

Le terme J est une application dont le terme de tête est une variable ou une abstraction. D'après le lemme 3.3.10, les possibilités sont :

- $J = (y\tilde{P'})\lceil R'/y\rceil$, où $R' \equiv ((\lambda x.M)Q_1 \dots Q_i \mid R'')$ et

  $$(1) \qquad Q_{i+1} \dots Q_n \in \tilde{P'}\lceil R''/y\rceil$$

  Par le lemme 3.3.13 ($J \in \{[T']\}$), $T' \xrightarrow{\star}_r y\rho_0 W_{i+1}\rho_{i+1} \dots W_n\rho_n$ où l'occurrence de tête de la variable x est libre et

  $$(2) \qquad \forall j \in [i+1, n]\ \exists P''_j\ \ P'_j {=_\alpha} P''_j \ \ \& \ \ \tilde{P''} \in \{[W_{i+1}\rho_{i+1} \dots W_n\rho_n]\}$$

  D'autre part, il existe N t.q. $R \equiv (N \mid R^*)$ et $(\lambda x.M)Q_1 \dots Q_i \in \{[N]\}$. Par conséquent,

  $$T'\langle R/y\rangle \xrightarrow{\star}_r y\rho_0 W_{i+1}\rho_{i+1} \dots W_n\rho_n\langle R/y\rangle \rightarrow_r N\rho'_0 W'_{i+1}\rho'_{i+1} \dots W'_n\rho'_n\langle R^*/v\rangle$$

  après quelques α-conversions. Par hypothèse de récurrence (N est un sous-terme de T), on a $N \xrightarrow{\star}_r (\lambda z.M')\rho_0 Q'_1\rho_1 \dots Q'_i\rho_{i+1}$, et, pour tout $j \in [1, i+1]$, il existe un 2-partage $p_2(\rho_j) = (\rho'_j, \rho''_j)$ t.q.

  $$\lambda x.M \in \{[(\lambda z.M')\tilde{S}_0\rho'_1 \dots \rho'_{i+1}]\} \ \ \&$$

  $$\forall i\, \exists P_i\ Q_i {=_\alpha} P_i \ \ \& \ \ \tilde{P} \in \{[Q'_1\rho''_1 \dots Q'_n\rho''_{i+1}]\}$$

  Donc, $T \xrightarrow{\star}_\lambda (\lambda z.M')\rho_0 Q'_1\rho_1 \dots Q'_i\rho_{i+1}\rho'_0 W'_{i+1}\rho'_{i+1} \dots W'_n\rho'_n\langle R^*/v\rangle$. Il reste à montrer

  $$\exists P_{i+1} \dots P_n \ \ Q_{i+1} \dots Q_n {=_\alpha} P_{i+1} \dots P_n \in \{[W'_{i+1}\rho'_{i+1} \dots W'_n\rho'_n\langle R^*/v\rangle]\}$$

  De (1) et (2) on déduit $\tilde{P'}\lceil R''/y\rceil {=_\alpha} \tilde{P''}\lceil R''/y\rceil$. Donc, $\forall j \in [i+1, n]\ \exists U_j$

  $$Q_{i+1} \dots Q_n {=_\alpha} U_{i+1} \dots U_n \in \tilde{P''}\lceil R''/y\rceil \subseteq \{[(W_{i+1}\rho_{i+1} \dots W_n\rho_n)\langle R''/y\rangle]\}$$

  De plus,

  $$y\rho_0 W_{i+1}\rho_{i+1} \dots W_n\rho_n\langle R^*/y\rangle {=_\alpha} v\rho'_0 W'_{i+1}\rho'_{i+1} \dots W'_n\rho'_n\langle R^*/v\rangle \quad \text{implique}$$

  $$W_{i+1}\rho_{i+1} \dots W_n\rho_n {=_\alpha} W''_{i+1}\rho''_{i+1} \dots W''_n\rho''_n \quad \text{et } \tilde{W'}\rho' = \tilde{W''}\rho''[v/y]$$

  C'est-à-dire,

  $$W_{i+1}\rho_{i+1} \dots W_n\rho_n\langle R^*/y\rangle {=_\alpha} W'_{i+1}\rho'_{i+1} \dots W'_n\rho'_n\langle R^*/v\rangle$$

  Par conséquent, il existe P_j t.q. $U_j {=_\alpha} P_j$ et

  $$P_{i+1} \dots P_n \in \{[(W'_{i+1}\rho'_{i+1} \dots W'_n\rho'_n)\langle R^*/v\rangle]\}$$

  Par la transitivité de $=_\alpha$, on a $Q_{i+1} \dots Q_n {=_\alpha} P_{i+1} \dots P_n$.

- $J = ((\lambda v.H)\tilde{P}')\lceil R'/y\rceil$ où $R' \equiv (R_0 \mid R_1)$, $x \notin fv(\lambda v.H, R', y)$ et $\lambda x.M \in (\lambda v.H)\lceil R_0/y\rceil$ & $\tilde{Q} \in \tilde{P}'\lceil R_1/y\rceil$. Donc la taille de $\tilde{P}'$ est n.

  Par hypothèse de récurrence ($J \in \{[T']\}$ et T' sous-terme de T),

  $$T' \stackrel{\star}{\rightarrow}_r (\lambda z.M')\rho_0 Q'_1\rho_1 \ldots Q'_n\rho_n$$

  où $(\lambda v.H) \in \{[(\lambda z.M')\rho_0\rho']\}$ et $\tilde{P}' {=_\alpha} \tilde{P}'' \in \{[Q_1\rho''_1 \ldots Q'_n\rho''_n]\}$. Les séquences ρ' et $\rho''_1 \ldots \rho''_n$ définissent un partage de $\rho_1 \ldots \rho_n$. On en déduit :

  $$(\lambda x.M) \in \{[(\lambda z.M')\rho_0\rho'\langle R_0/y\rangle]\}$$

  $$T = T'\langle R/y\rangle \stackrel{\star}{\rightarrow}_r (\lambda z.M')\rho_0 Q'_1\rho_1 \ldots Q'_n\rho_n\langle R/y\rangle \ = \ T''$$

  Par ailleurs, $R \equiv (R'_0 \mid R'_1)$ où $R_0 \in \{[R'_0]\}$ et $R_1 \in \{[R'_1]\}$. Donc, $(\lambda x.M) \in \{[(\lambda z.M')\rho_0\rho'\langle R'_0/y\rangle]\}$. Puisque $\tilde{Q} \in \tilde{P}'\lceil R_1/y\rceil {=_\alpha} \tilde{P}''\lceil R_1/y\rceil$, en utilisant le lemme 3.3.5, on a

  $$\exists \tilde{P}\ \tilde{Q} {=_\alpha} \tilde{P} \in \{[(Q'_1\rho''_1 \ldots Q'_n\rho''_n)\langle R'_1/y\rangle]\}$$

  Il est clair que $\rho'\langle R'_0/y\rangle$ et $\rho''_1 \ldots \rho''_n\langle R'_1/y\rangle$ constituent un 2-partage de la séquence $\rho_1 \ldots \rho_n\langle R/y\rangle$.

□

Corollaire 3.3.15 *Soit* $T \in \Lambda_r$. *Si* $(\lambda x.M)P_1 \ldots P_n \in \{[T]\}$ *avec* $n \geq 1$, *alors*

$$\exists T' \in \Lambda_r\ \exists N \in \Lambda_d\ \ T \stackrel{\star}{\rightarrow}_r T'\ \ \&\ \ (M\lceil P_1/x\rceil P_2 \ldots P_n) {=_\alpha} N \in \{[T']\}$$

Preuve. On suppose $(\lambda x.M)P_1 \ldots P_n \in \{[T]\}$. Donc, par le lemme 3.3.14

$$T \stackrel{\star}{\rightarrow}_r (\lambda z.L)\rho_0 Q_1\rho_1 \ldots Q_n\rho_n$$

où $(\lambda x.M) \in \{[(\lambda z.L)\rho_0\tilde{S}']\}$ et $\tilde{P} {=_\alpha} \tilde{P}' \in Q_1\rho''_1 \ldots Q_n\rho''_n$ (ici, $\tilde{S}'$ et $\rho''_1 \ldots \rho''_n$ définissent un 2-partage de $\rho_1 \ldots \rho_n$.) Par la définition de la traduction, on déduit $x \notin fv(\lambda z.L, \tilde{S}_0, \rho')$. Donc,

$$T \stackrel{\star}{\rightarrow}_r (\lambda x.L[x/z])\rho_0 Q_1\rho_1 \ldots Q_n\rho_n \stackrel{\star}{\rightarrow}_r$$

$$(\lambda x.L[x/z]\rho_0)Q_1\rho_1 \ldots Q_n\rho_n \rightarrow_r$$

$$L[x/z]\rho_0\langle Q_1/x\rangle\rho_1 \ldots Q_n\rho_n = T'$$

D'autre part, il existe un 2-partage $\rho''_1 \ldots \rho''_n$, désigné par ρ''' et $\rho'''_1 \ldots \rho'''_n$, t.q.

$$\begin{array}{rcl} P_1 & =_\alpha & P'_1 \in \{[Q_1\rho''']\} \\ P_2 \ldots P_n & =_\alpha & P'_2 \ldots P'_n \in \{[Q_2\rho'''_2 \ldots Q_n\rho'''_n]\} \end{array}$$

Il est clair que $M\lceil P_1/x\rceil P_2 \ldots P_n {=_\alpha} M\lceil P'_1/x\rceil P'_2 \ldots P'_n$ et que ce dernier terme est contenu dans $\{[L[x/z]\rho_0\rho'\langle Q_1\rho'''/x\rangle\rho'''_1 Q_2 \ldots Q_n\rho'''_n]\} \subseteq \{[T']\}$. □

Théorème 3.3.16 *Soit* $M, M', N \in \Lambda_d$ *et* $T \in \Lambda_r$. *Alors,*

$$M =_\alpha M' \in \{[T]\} \ \&\ M \overset{\star}{\to}_d N \ \Rightarrow\ \exists N', N''\ T \overset{\star}{\to}_r N' \ \&\ N =_\alpha N'' \in \{[N']\}$$

Preuve. Par récurrence sur $M \overset{\star}{\to}_d N$. Si la longueur de la dérivation est zéro, i.e. $M = N$, on pose $N' = T$ et $N'' = M'$. Dans le cas de récurrence, $M \to_d M_0 \overset{\star}{\to}_d N$ avec $M = (\lambda x.L)PQ_1 \ldots Q_n$ et $M_0 \in L\lceil P/x \rceil \tilde{Q}$.

Par le corollaire 3.3.15, $\exists T' \in \Lambda_r\ \exists M_0' \in \Lambda_d\ \ T \overset{\star}{\to}_r T'\ \&\ M_0 =_\alpha M_0' \in \{[T']\}$. Par hypothèse de récurrence ($M_0 \overset{\star}{\to}_r N$), on a $\exists N', N''\ T' \overset{\star}{\to}_r N'\ \&\ N =_\alpha N'' \in \{[N']\}$. Donc $T \overset{\star}{\to}_r N'$. $\square$

Corollaire 3.3.17 *Si* $M \in \Lambda_d$, *alors* $M\Downarrow_d \ \Rightarrow\ M\Downarrow_r$.

Preuve. Soit $M, V \in \Lambda_d$ t.q. $M \overset{\star}{\to}_d V$. On montre qu'il existe $W \in \lambda_r$ t.q. $V \in \{[W]\}$ et $M\Downarrow_r W$. Par le théorème 3.3.16, puisque $M \in \{[M]\}$, on a

$$\exists N', N''\ \ M \overset{\star}{\to}_r N'\ \&\ V =_\alpha N'' \in \{[N']\}$$

Comme $V = \lambda x.N$, le terme N'' est aussi une abstraction, soit $\lambda z.L$. Une application du lemme 3.3.14 permet de conclure $N' \overset{\star}{\to}_r (\lambda y.L')\rho_0 \Downarrow_r$. $\square$

En résumant, pour M sans substitutions explicites, la convergence dans λ_r équivaut à la convergence dans λ_d, et pour M dans λ_r, M converge ssi il y a un représentant de M dans λ_d qui converge. Ces résultats seront utilisés dans la preuve du lemme d'approximation pour λ_r, donnée dans la section 3.7.

3.4 Lambda calcul avec ressources et test de convergence λ_r^c

Le langage λ_r^c est une extension du lambda calcul avec ressources qui incorpore le test de convergence sous la forme d'un opérateur unaire c sur les paquets. Le comportement de c est étroitement lié à la notion de *valeur* dans le calcul, i.e. forme normale significative obtenue par évaluation. Le fait que c soit un opérateur unaire, et non pas 0-aire comme dans λ_c, permet de conserver la notion de valeur standard. Le combinateur **c** de [3] est $\lambda x.\mathrm{c}x$. Dans le cadre de la sémantique standard, celle du lambda calcul faible [2], les seules valeurs sont les abstractions. Le terme $\mathrm{c}M$ rend la fonction identité lorsque le processus d'évaluation transforme M en un terme observable. Nous introduisons cette construction en admettant des paquets comme arguments de c. Les termes et les valeurs de λ_r^c sont construits selon la syntaxe suivante :

$$\begin{array}{ll} (\Lambda_{rc}) & M ::= x \mid \lambda x.M \mid (MP) \mid M\langle P/x \rangle \mid \mathrm{c}P \\ & P ::= \mathbf{1} \mid M \mid (P \mid P) \mid M^\infty \end{array}$$

$$\begin{array}{ll} (\mathbb{V}_{rc}) & V ::= \lambda x.M \end{array}$$

Les variables libres (liées) d'un terme $\mathrm{c}P$ sont les variables libres (liées) de P. On ajoute $(\mathrm{c}P)[z/x] = \mathrm{c}(P[z/x])$ au renommage de variables, où $z \notin var(P)$. L'α-conversion reste inchangée et on incorpore la clause suivante à la définition de $\equiv$:

$$P \equiv Q \;\Rightarrow\; \mathrm{c}P \equiv \mathrm{c}Q$$

L'évaluation $\rightarrow_{rc}$ est un raffinement de $\rightarrow_r$ (cf. figure 3.2), capable de gérer le nouvel opérateur c; les nouvelles règles sont écrites dans la figure 3.6.

$$(\mathrm{c}1)\ \mathrm{c}(\lambda x.M) \rightarrow_{rc} \mathbf{I} \qquad (\mathrm{c}2)\ \frac{P \equiv (M \mid Q)}{\mathrm{c}P \rightarrow_{rc} \mathrm{c}M} \qquad \frac{M \rightarrow_{rc} M'}{\mathrm{c}M \rightarrow_{rc} \mathrm{c}M'}$$

$$\frac{M\langle N/x\rangle \succ M'}{(\mathrm{c}M)\langle N/x\rangle \succ \mathrm{c}M'}$$

FIG. 3.6 - *Évaluation dans* λ_r^c

La première règle indique que le test de convergence réussit face à une abstraction; le terme entier devient l'identité, ce qui permet de continuer l'évaluation, par exemple dans des termes de la forme $(\mathrm{c}(\lambda x.M))R_1 \ldots R_n$. La seconde règle sélectionne de façon non-déterministe une composante M d'un paquet P, éliminant en même temps le reste du paquet. Le terme M devient l'argument de c. On observera qu'une présentation alternative de cette règle est

$$\mathrm{c}(M \mid Q) \rightarrow_{rc} \mathrm{c}M$$

puisque $\mathrm{c}P \equiv \mathrm{c}(M \mid Q)$ et que $\rightarrow_{rc}$ a été défini à équivalence structurelle près. La troisième règle est une règle structurelle qui permet d'évaluer l'argument de c. La quatrième concerne l'opération de saisie.

Remarquons que, même si c est un opérateur sur les paquets, le test de convergence est effectué seulement lorsque l'argument de c est devenu un terme simple. Ce choix n'est pas essentiel; nous pourrions aussi bien travailler avec l'ensemble de règles suivant :

$$\mathrm{c}(\lambda x.M \mid P) \rightarrow_{rc} \mathbf{I} \qquad \frac{M \rightarrow_{rc} M'}{\mathrm{c}(M \mid P) \rightarrow_{rc} \mathrm{c}(M' \mid P)} \qquad \frac{M\langle N/x\rangle \succ M'}{(\mathrm{c}(M \mid P))\langle N/x\rangle \succ \mathrm{c}(M' \mid P)}$$

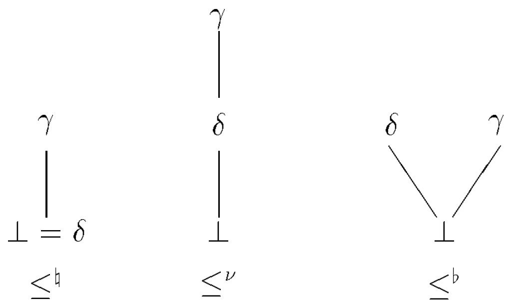
3.5 Sémantique observationnelle des calculs avec ressources

3.5.1 Définition

La sémantique standard des calculs avec ressources, basée sur la notion de convergence, ne fournit pas de moyen de détecter le blocage d'un terme, seulement les abstractions sont observables. Dans le but d'étudier la sémantique du λ-calcul engendré par le codage dans le calcul π, Boudol et Laneve [19, 20] ont élargi la notion de préordre observationnel en admettant deux sortes de valeurs, pour représenter les abstractions et les termes bloqués. Ils définissent :

Définition 3.5.1 *(Domaine des observations)*

Soit $\mathrm{O} = \{\gamma, \delta, \perp\}$. *Les objets de ce domaine représentent les évaluations qui terminent en une abstraction* (γ)*, les évaluations qui terminent en un terme bloqué* (δ) *et les évaluations infinies ou divergentes* ($\perp$)*. L'ordre standard* $\leq^{\natural}$*, l'ordre vertical* $\leq^{\nu}$ *et l'ordre plat* $\leq^{\flat}$ *sur* O *partent tous de la base que la divergence ne fournit aucune information; ce sont :*



L'extension des préordres $\leq^{\natural}$, $\leq^{\nu}$ *et* $\leq^{\flat}$ *- regroupés dans la notation* $\leq$ *- aux parties de* O *est :*

$$o \leq o' \overset{def}{\Leftrightarrow} \forall a \in o \ \exists a' \in o' \ a \leq a'$$

Définition 3.5.2 ***(Fonction d'observation)***

La fonction d'observation $obs(-) : \Lambda_r^o \to \mathcal{P}(\mathrm{O})$ *est définie par :*

- $\gamma \in obs(M)$ *ssi* $M \overset{\star}{\to}_r \lambda x.M'$
- $\delta \in obs(M)$ *ssi* $M \overset{\star}{\to}_r N \not\to_r$ *et* $N \neq \lambda x.N'$
- $\perp \in obs(M)$ *ssi pour tout* $n \in \mathbb{N}$ *il existe* N *tel que* $M \overset{\star}{\to}_r N$ *est une évaluation en* n *étapes.*

Définition 3.5.3 ***(Préordre observationnel généralisé)***
Les préordres observationnels sur λ_r, désignés par $\sqsubseteq^\natural$, $\sqsubseteq^\nu$ et $\sqsubseteq^\flat$, suivent le schéma de définition

$$M \sqsubseteq N \overset{def}{\Leftrightarrow} \forall C \;\; obs(C[M]) \leq obs(C[N])$$

où $\leq$ est respectivement $\leq^\natural$, $\leq^\nu$ ou $\leq^\flat$. Il est sous-entendu ici que $C[M]$ et $C[N]$ sont des termes clos.

Remarquons que $\mathrm{obs}(M)$ est un ensemble non-vide quelque soit $M \in \Lambda_r^o$. Donc $\bot \leq \bot, \delta, \gamma$ implique $\{\bot\} \leq \mathrm{obs}(M)$; comme on le verra, $\Omega \sqsubseteq M$ est vérifié dans tous les cas de $\sqsubseteq$. Il est clair que $M\Downarrow_r$ ssi $\gamma \in \mathrm{obs}(M)$; par conséquent, $\sqsubseteq^\natural$ est équivalent à $\sqsubseteq_r$. Dans le cadre de λ_m, le déterminisme de $\rightarrow_m$ fait de $\mathrm{obs}(M)$ un ensemble réduit à un singleton. Les résultats de [19, 20] sur les préordres $\sqsubseteq^\nu$, $\sqsubseteq^\flat$, et $\sqsubseteq^\natural$ du calcul λ_m seront énoncés dans la section 3.8.1. Sur λ_r, nous nous limitons à l'étude du préordre standard.

3.5.2 Lemmes des contextes

Dans cette section on prouve un lemme des contextes pour la sémantique standard du langage λ_r^c. Comme conséquence des résultats de la section 3.3, on déduit un lemme des contextes pour le calcul λ_d.

Une présentation alternative de $\sqsubseteq_{rc}$ en fonction de contextes (pseudo)applicatifs est possible. La structure de ces contextes est bien moins généreuse que celle des contextes définis sur la syntaxe des termes; en particulier, la constante [] n'y apparaît qu'une fois au plus, et seulement en position de tête. La grammaire des contextes applicatifs est :

$$A ::= [] \mid AP \mid A\langle P/x\rangle \mid \mathrm{c}A$$

Dans les preuves, nous considérons la description des contextes applicatifs basée sur les ensembles $\mathbb{A}_i$, définis par récurrence comme suit :

$$\begin{aligned} \mathbb{A}_0 &= \{[]\tilde{R} \;\; \text{pour tout } \tilde{R}\ \} \\ \mathbb{A}_{n+1} &= \{(\mathrm{c}A)\tilde{R} \;\; \text{pour tout } A \in \mathbb{A}_n \text{ et } \tilde{R}\ \} \end{aligned}$$

Il est évident que A est un contexte applicatif ssi $\exists n\ A \in \mathbb{A}_n$. La forme générale des contextes applicatifs est $\underbrace{\mathrm{c}\ldots\mathrm{c}}_{m}([]\tilde{R}_0)\tilde{R}_1\ldots\tilde{R}_m$, avec $m \geq 0$. Afin de simplifier la notation, nous écrivons souvent c^m et $\tilde{R}^m$ au lieu de $\underbrace{\mathrm{c}\ldots\mathrm{c}}_{m}$ et $\tilde{R}_1\ldots\tilde{R}_m$ respectivement, ce qui donne $\mathrm{c}^m([]\tilde{R}_0)\tilde{R}^m$.

Nous utilisons le même genre d'écriture raccourcie pour des termes et des contextes arbitraires. Il suffit d'observer que pour tout terme M il existe $m > 0$ maximal tel que

$$M = \underbrace{\mathrm{c}(\ldots\mathrm{c}((\mathrm{c}}_{m\geq 0}T)\tilde{R}_1)\tilde{R}_2\ldots)\tilde{R}_m$$

où T est un terme simple ou un paquet de termes. C'est-à-dire, T ne commence pas par c; plus précisément,

- soit $T = N\tilde{R}_0$ avec N une variable x ou une abstraction $\lambda x.N'$
- soit $T = (P \mid Q)$ ou $T = N^\infty$.

En suivant la convention adoptée pour les contextes applicatifs, $M = \mathrm{c}^m T\tilde{R}^m$. Selon le même principe, tout contexte est de la forme $\mathrm{c}^m W\tilde{B}^m$ où W est un contexte de termes $x\tilde{B}_0$, $[]_i\tilde{B}_0$ ou $(\lambda x.D)\tilde{B}_0$, ou un contexte d'argument $(B_0 \mid B_1)$ ou D^∞.

Lemme 3.5.4 *Si $M \sqsubseteq_{\mathcal{A}} N$ et $z \notin var(M, N)$, alors $M[z/x] \sqsubseteq_{\mathcal{A}} N[z/x]$.*

Preuve. Soit $M \sqsubseteq_{\mathcal{A}} N$ et A un contexte applicatif clos qui ferme $M[z/x]$, $N[z/x]$ et qui vérifie $A[M[z/x]]\Downarrow_{rc}$. Si $[] \notin A$, l'énoncé est trivial. Sinon, nous examinons deux cas:

- le premier a lieu quand $z \notin fv(M[z/x], N[z/x])$ (i.e. $x \notin fv(M, N)$). Clairement, $M[z/x] = M$ et $N[z/x] = N$, alors $A[N]\Downarrow_{rc}$ par hypothèse.
- le second suppose que z est libre dans au moins un des termes $M[z/x]$ et $N[z/x]$. Si l'on écrit $A = \mathrm{c}^m([]\tilde{R}_0)\tilde{R}^m$, il existe un plus petit i tel que $\tilde{R}_i = \tilde{S}_1\langle Q/z\rangle\tilde{S}_2$ et $\tilde{S}_1$ n'a pas d'entrée de substitution pour z :

  $(*)$ $\tilde{R}_0, \ldots, \tilde{R}_{i-1}\tilde{S}_1$ ne contient pas d'entrée de substitution pour z.

  Par récurrence et en utilisant $(*)$, on montre que

  $$(\mathrm{c}^i([]\tilde{R}_0)[x/z]\tilde{R}_{i-1}\tilde{S}_1) = \mathrm{c}^i([]\tilde{R}'_0)\tilde{R}'_{i-1}\tilde{S}'_1$$

  où $\tilde{R}'_j = (Q_1[x/z])\ldots(Q_r[x/z])$ si $\tilde{R}_j = Q_1\ldots Q_r$. Appelons A' le contexte

  $$\mathrm{c}^m(M\tilde{R}'_0)\tilde{R}'_{i-1}\tilde{S}'_1\langle Q/x\rangle\tilde{S}_2\tilde{R}_{i+1}\ldots\tilde{R}_m$$

  Comme $A[M[z/x]]=_\alpha A'[M]$, on a $A'[N]\Downarrow_{rc}$ par hypothèse. D'autre part, puisque $A'[N]=_\alpha A[N[z/x]]$, le dernier terme converge par définition d'évaluation.

□

Lemme 3.5.5 ***(Lemme des contextes pour λ_r^{c})***

$$\forall M, N \in \Lambda_{rc} \;\; M \sqsubseteq_{rc} N \Leftrightarrow M \sqsubseteq_{\mathcal{A}} N$$

Preuve. La partie $\Leftarrow$ est triviale. La partie $\Rightarrow$ est un cas particulier de la proposition suivante:

$$\forall M_1, \ldots, M_p \ \forall N_1, \ldots, N_p \ (\forall i \ M_i \sqsubseteq_{\mathcal{A}} N_i) \ \Rightarrow \ (\forall C \ C[\tilde{M}]\Downarrow_{rc} \ \Rightarrow \ C[\tilde{N}]\Downarrow_{rc})$$

où C est un λ_r^c-contexte à plusieurs trous.

Soit C un contexte qui ferme $\tilde{M}, \tilde{N}$ et tel qu'il existe une valeur V vérifiant $C[\tilde{M}] \overset{\star}{\rightarrow}_{rc} V$ en un nombre fini l d'étapes d'évaluation. Nous montrons $C[\tilde{N}]\Downarrow_{rc}$ par récurrence sur (l, h) (selon l'ordre lexicographique), où h est le nombre de trous de C, appelé $t(C)$ dorénavant.

$l = 0$: le terme $C[\tilde{M}]$ est lui-même une valeur. On a soit $C = \lambda z.L$, auquel cas $C[N]$ est aussi une valeur, soit $C = [\,]_i$ et M_i est une abstraction. Alors M_i et N_i sont des termes clos et $N_i = C[\tilde{N}]\Downarrow_{rc}$ est vérifié, car $M_i \sqsubseteq_{\mathcal{A}} N_i$.

$l > 0$: si $t(C) = 0$ alors C converge et $C = C[\tilde{N}]$. Supposons $t(C) \geq 1$ et $C = \mathrm{c}^m W \tilde{B}^1 \ldots \tilde{B}^m$; nous établissons la proposition par cas sur W :

1. $W = C_0 \tilde{B}^0$ où C_0 est une variable x, ou $[\,]_i$, ou $\lambda x.D$. On décompose l'analyse selon C_0 :

(a) $C_0 = x$: afin de simplifier la preuve, on écrit C comme $\mathrm{c}^m C_0 B_1 \ldots B_n$. Le premier pas d'évaluation à partir de $C[\tilde{M}]$ ne peut être qu'une saisie sur x: il existe un plus petit i tel que $B_i = \langle (D \mid B')/x \rangle$ où D est un contexte de termes et $C[\tilde{M}]$ consomme $D[\tilde{M}]$. En tenant compte des α-conversions éventuelles, la structure du terme obtenu par saisie est:

$$C[\tilde{M}] \rightarrow_{rc} \mathrm{c}^m D[\tilde{M}] B'_1[\tilde{M}^1] \ldots B'_{i-1}[\tilde{M}^{i-1}] \langle B'[\tilde{M}]/z \rangle B_{i+1}[\tilde{M}] \ldots B_n[\tilde{M}]\Downarrow_{rc}$$

où $z \notin var(C[\tilde{M}])$ et $B'_j, \tilde{M}^j$, pour $j = 1, \ldots, i-1$, résultent de $B_j, \tilde{M}$ par renommage de variables (x en particulier) et par α-conversions. Sans perte de généralité, on peut supposer que les variables utilisées dans ces α-conversions ne figurent pas parmi celles de $C[\tilde{N}]$, en particulier $z \notin var(C[N])$. La longueur de $\tilde{M}^j$, soit p_j, peut différer de p, celle de $\tilde{M}$.

Pour $j = 1, \ldots, i-1$, nous appelons $B^\star_j$ le contexte à trous multiples construit comme B'_j, où $[\,]_k$ est remplacé par $[\,]_{p+p_1+\ldots+p_{j-1}+k}$. Soit $C^\star$ le contexte

$$\mathrm{c}^m D B^\star_1 \ldots B^\star_{i-1} \langle B'/z \rangle B_{i+1} \ldots B_n$$

Il est évident que $C^\star[\tilde{M}, \tilde{M}^1, \ldots, \tilde{M}^{i-1}]\Downarrow_{rc}$ en $l-1$ étapes d'évaluation.

Effectués sur $\tilde{N}$, les renommages et α-conversions qui ont donné lieu aux séquences $\tilde{M}^j$ produisent $\tilde{N}^1, \ldots, \tilde{N}^i$. Avec un abus de notation, le lemme 3.5.4 implique $\tilde{M}^j \sqsubseteq_{\mathcal{A}} \tilde{N}^j$, pour tout $j = 1, \ldots, i-1$. Sur $C[\tilde{N}]$, la saisie donne

$$C[\tilde{N}] \rightarrow_{rc} C^\star[\tilde{N}, \tilde{N}^1, \ldots, \tilde{N}^i]$$

et ce dernier terme converge par hypothèse de récurrence. Donc, $C[\tilde{N}]\Downarrow_{rc}$.

(b) $C_0 = \lambda x.D$: on examine deux situations possibles. La première correspond à $\tilde{B}^0$ vide. Le fait d'avoir $l > 0$ implique $m \geq 1$. L'évaluation de $C[\tilde{M}]$ est

$$C[\tilde{M}] \rightarrow_{rc} \mathrm{c}^{m-1}(\mathbf{I})\tilde{B}^1 \ldots \tilde{B^m}[\tilde{M}]\Downarrow_{rc} \text{ en } l-1 \text{ étapes}$$

Clairement, $C_0[\tilde{N}]$ est aussi une valeur, alors $C[\tilde{N}] \rightarrow_{rc} \mathrm{c}^{m-1}(\mathbf{I})\tilde{B}^1 \ldots \tilde{B^m}[\tilde{N}]\Downarrow_{rc}$ par hypothèse de récurrence.

Par contre, si $\tilde{B}^0 = B_1 \ldots B_q$ n'est pas vide, le premier pas d'évaluation dépend de B_1, un contexte de substitution ou un contexte d'argument. Pour $B_1 = \langle B_0/y \rangle$ et $\tilde{B}' = B_2 \ldots B_q$, on a

$$C[\tilde{M}] \rightarrow_{rc} \mathrm{c}^m(\lambda z.(D'[\tilde{M}']\langle B_0/y \rangle)\tilde{B}')\tilde{B}^1 \ldots \tilde{B^m}[\tilde{M}]\Downarrow_{rc} \text{ en } l-1 \text{ étapes}$$

où, sans perte de généralité, $z \notin var((C_0B_1)[\tilde{M}])$ et $z \notin var((C_0B_1)[\tilde{N}])$. Le contexte D' et les termes $\tilde{M}'$ s'obtiennent par renommage de x par z dans D et $\tilde{M}$ respectivement.

On note $D^\star$ le contexte construit comme D' mais où $[]_j$ est remplacé par $[]_{p+j}$ pour $j = 1, \ldots, p$. Soit

$$C^\star = \mathrm{c}^m(\lambda z.(D^\star\langle B_0/y \rangle)\tilde{B}')\tilde{B}^1 \ldots \tilde{B^m}$$

Le renommage de x par z effectué sur $\tilde{N}$ donne $\tilde{N}'$; alors $\tilde{M}' \sqsubseteq_{\mathcal{A}} \tilde{N}'$ par application du lemme 3.5.4. Puisque $C^\star[\tilde{M}, \tilde{M}']\Downarrow_{rc}$ en $l-1$ pas d'évaluation, $C^\star[\tilde{N}, \tilde{N}']\Downarrow_{rc}$ est conséquence de l'hypothèse de récurrence. Donc $C[\tilde{N}] \rightarrow_{rc} C^\star[\tilde{N}, \tilde{N}']$ implique $C[\tilde{N}]\Downarrow_{rc}$.

Supposons maintenant que B_1 est un contexte d'argument. Soit

$$C' = \mathrm{c}^m(D\langle B_1/x \rangle\tilde{B}')\tilde{B}^1 \ldots \tilde{B^m}$$

alors $C[\tilde{M}] \rightarrow_{rc} C'[\tilde{M}]$ et ce terme converge en $l-1$ pas. Par récurrence, on a $C'[\tilde{N}]\Downarrow_{rc}$. Ce qui implique $C[\tilde{N}]\Downarrow_{rc}$, car $C[\tilde{N}] \rightarrow_{rc} C'[\tilde{N}]$.

(c) $C_0 = []_i$: nous abrégeons la preuve en posant $C = \mathrm{c}^m C_0 B_1 \ldots B_n$. Soit $C' = \mathrm{c}^m M_i B_1 \cdots B_n$ et $A = \mathrm{c}^m []B_1[\tilde{N}] \cdots B_n[\tilde{N}]$. D'une part, $C[\tilde{M}] = C'[\tilde{M}]\Downarrow_{rc} V$ en l étapes. Mais, $t(C') = t(C) - 1$, alors $A[M_i] = C'[\tilde{N}]\Downarrow_{rc}$ par hypothèse de récurrence. D'autre part, A ferme N_i, car $A[N_i] = C[\tilde{N}]$, et aussi M_i: si l'on suppose $\exists x \in fv(M_i)$ tel que $x \in fv(A[M_i])$, alors il n'y a pas de contexte de substitution pour x parmi les $B_1, \ldots, B_n$, donc x est libre dans $C[\tilde{M}]$, ce qui contredit l'hypothèse. Finalement, $M_i \sqsubseteq_{rc} N_i$ et $A[N_i] = C[\tilde{N}]$ impliquent $A[N_i]\Downarrow_{rc}$. Ce qui revient à dire $C[\tilde{N}]\Downarrow_{rc}$.

2. $W = (W_0 \mid W_1)$: Dans ce cas, $m \geq 1$ et il existe D, un contexte de termes, et W', un contexte d'argument, tel que $W \equiv (D \mid W')$ et

$$C[\tilde{M}]\rightarrow_{rc}(\mathrm{c}^m D B_1 \dots B_n)[\tilde{M}]\Downarrow_{rc} \text{ en } l-1 \text{ pas d'évaluation}$$

alors, par récurrence, $C[\tilde{N}]\rightarrow_{rc}(\mathrm{c}^m D B_1 \dots B_n)[\tilde{N}]\Downarrow_{rc}$.

3. $W = D^{\infty}$: dans ce cas, la preuve de l'énoncé est complètement similaire à celle du point (2). Le terme choisi est $D[\tilde{M}]$. □

Les lemmes des contextes pour les sémantiques standards de λ_m et λ_r sont corollaires du lemme 3.5.5. Des preuves directes ont été données dans [17, 18]. Par ailleurs, la sémantique $\sqsubseteq^{\nu}$ sur λ_r (donc sur λ_m) est aussi caractérisée par des contextes applicatifs (cf. [19]). Il n'est pas difficile d'étendre le lemme des contextes pour la sémantique standard de λ_r^c aux sémantiques verticale et plate.

Comme conséquence du lemme des contextes, l'évaluation dans λ_r^c est décroissante par rapport au préordre applicatif :

Lemme 3.5.6

$$\forall M, N \in \Lambda_{rc} \;\; M\rightarrow_{rc}N \;\Rightarrow\; N\sqsubseteq_{\mathcal{A}}M$$

Preuve. Soit $M, N \in \Lambda_{rc}$ et A un λ_r^c-contexte applicatif t.q. $A[N]\Downarrow_{rc}$. La forme de A est $\mathrm{c}^m[]\tilde{R}_0\tilde{R}_m$, donc $M\rightarrow_{rc}N$ implique $A[M]\rightarrow_{rc}A[N]\Downarrow_{rc}$. □

Quant au calcul de paquets, le lemme des contextes est corollaire du lemme des contextes pour λ_r et des résultats de la section 3.3. Les contextes applicatifs associés à λ_d sont :

$$A ::= x \mid \lambda x.A \mid AP$$

Lemme 3.5.7 ***(Lemme des contextes pour λ_d)***

$$\forall M, N \in \Lambda_d \;\; M\sqsubseteq_d N \Leftrightarrow M\sqsubseteq_{\mathcal{A}}N$$

Preuve. On note ici $\sqsubseteq_{\mathcal{A}}^r$ le préordre applicatif associé à λ_r. À chaque Λ_r-contexte C on associe Λ_d-contexte $\lfloor M \rfloor$ défini par :

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\lfloor [] \rfloor & = & [] & \lfloor x \rfloor & = & x \\
\lfloor \lambda x.C \rfloor & = & \lambda x.\lfloor C \rfloor & \lfloor CQ \rfloor & = & \lfloor C \rfloor \lfloor Q \rfloor \\
\lfloor C\langle Q/x \rangle \rfloor & = & (\lambda x.\lfloor C \rfloor)\lfloor Q \rfloor & \lfloor \mathbf{1} \rfloor & = & \mathbf{1} \\
\lfloor (Q_0 \mid Q_1) \rfloor & = & (\lfloor Q_0 \rfloor \mid \lfloor Q_1 \rfloor) & \lfloor C^{\infty} \rfloor & = & \lfloor C \rfloor^{\infty}
\end{array}$$

On montre d'abord $\forall M, N \in \Lambda_d \;\; M\sqsubseteq_{\mathcal{A}}N \;\Rightarrow\; M \sqsubseteq_{\mathcal{A}}^r N$.

Soit A un contexte dans λ_r t.q. $A[M]\Downarrow_r$. Il est clair que pour tout $M \in \Lambda_d$, $A[M]\Downarrow_r \Leftrightarrow \lfloor A \rfloor [M]\Downarrow_r$. Puisque $\lfloor A \rfloor [M] \in \Lambda_d$, par le corollaire 3.3.9(2), on a

$\lfloor A\rfloor[M]\Downarrow_d$. Par hypothèse, $\lfloor A\rfloor[N]\Downarrow_d$. Alors $\lfloor A\rfloor[N]\Downarrow_r$ par le corollaire 3.3.17. D'où $A[N]\Downarrow_r$.

Ces corollaires permettent aussi de montrer $M \sqsubseteq_r N \ \Rightarrow\ M \sqsubseteq_d N$ pour tout $M, N \in \Lambda_d$. Le lemme des contextes pour λ_r permet de conclure

$$\forall M,N \in \Lambda_d \;\; M \sqsubseteq_{\mathcal{A}} N \;\Rightarrow\; M \sqsubseteq^r_{\mathcal{A}} N \;\Rightarrow\; M \sqsubseteq_r N \;\Rightarrow\; M \sqsubseteq_d N$$

Remarquons que pour des termes clos de λ_d, le préordre applicatif coïncide avec la bisimulation applicative d'Abramsky [2]. □

3.5.3 Quelques propriétés de la sémantique standard

Dans cette section nous regroupons plusieurs résultats concernant la sémantique standard du calcul de ressources λ_r^c que nous utiliserons par la suite. Les preuves sont en général des longues récurrences, que nous ne détaillerons pas toujours.

Déplacements des entrées de substitution

Il est clair qu'un terme supporte certaines permutations d'entrées de substitution ainsi que l'effacement des entrées qui ne lient aucune variable sans que soit affectée pour autant sa sémantique observationnelle. Ce fait est formalisé par la relation $\asymp$, qui contient $\equiv \cup =_\alpha$ et satisfait :

$$\begin{array}{rcll}
M\langle R/x\rangle & \asymp & M & x \notin fv(M) \\
(MP)\langle R/x\rangle & \asymp & (M\langle R/x\rangle)P & x \notin fv(P) \\
(M\langle P/z\rangle)\langle R/x\rangle & \asymp & (M\langle R/x\rangle)\langle P/z\rangle & z \neq x, x \notin fv(P), z \notin fv(R) \\
(\mathrm{c}M)\langle R/x\rangle & \asymp & \mathrm{c}(M\langle R/x\rangle) & \\
M \asymp M' & \Rightarrow & (MP) \asymp (M'P) & \\
M \asymp M' & \Rightarrow & (M\langle P/x\rangle) \asymp (M'\langle P/x\rangle) & \\
M \asymp M' & \Rightarrow & (\mathrm{c}M) \asymp (\mathrm{c}M') &
\end{array}$$

La preuve de $M \asymp N \ \Rightarrow\ M \simeq_{\mathcal{A}} N$ nécessite la définition et les lemmes suivants.

Définition 3.5.8 *Soit $\tilde{T}$ une séquence dans* Π *(composée de paquets et de substitutions). Soit $z, v_1, \ldots, v_n$ des variables t.q. $v_i \notin var(\tilde{T}, z)$ pour tout i. On dit que $\tilde{T}$ est $z\tilde{v}$-*indépendant *ssi l'un des cas suivants est vérifié:*

- *$\tilde{T}$ est vide,*

- *$\tilde{T} = P\tilde{T}'$ et $\tilde{T}'$ est $z\tilde{v}$-indépendant,*

- *$\tilde{T} = \langle P/x\rangle\tilde{T}'$, $x \neq z, \tilde{v}$ et $\tilde{T}'$ est $z, \tilde{v}$-indépendant.*

De plus, si $\tilde{T}$ est $z\tilde{v}$-indépendant et $w \in \tilde{v}$, alors $\tilde{T}[w/z]$ désigne la séquence définie par :

$$\tilde{T}[w/z] = \begin{cases} \textit{vide} & \textit{si } \tilde{T} \textit{ est vide} \\ (P[w/z])(\tilde{T}'[w/z]) & \textit{si } \tilde{T} = P\tilde{T}' \\ \langle (P[w/z])/x\rangle(\tilde{T}'[w/z]) & \textit{si } \tilde{T} = \langle P/x\rangle\tilde{T}' \textit{ et } x \neq z, w \end{cases}$$

Lemme 3.5.9 *Pour tout $L, H, M_0, L_v, L_z \in \Lambda_{rc}$, on a*

1. $(L[v/z])\langle M_0/v\rangle \succ\!\!- L_v \ \ \& \ \ v \notin var(L,z) \cup fv(M_0) \ \Rightarrow$

   $$(\forall w \notin var(L,z) \cup fv(M_0) \forall \tilde{T} \textit{ séquence } zvw\textit{-indépendante}$$

   $$\exists L_w \ (L[w/z])\langle M_0/w\rangle \succ\!\!- L_w \ \ \& \ \ L_w(\tilde{T}[w/z])\langle Q/w\rangle =_\alpha L_v(\tilde{T}[v/z])\langle Q/v\rangle)$$

2. $L\langle M_0/z\rangle \succ\!\!- L_z \ \ \& \ \ L \asymp H \ \ \& \ \ \neg(L =_\alpha H) \ \Rightarrow \ \exists H_z \ H\langle M_0/z\rangle \succ\!\!- H_z \ \ \& \ \ L_z \asymp H_z$

Preuve. On prouve (1) et (2) en même temps, par récurrence sur les profondeurs de dérivation de $(L[v/z])\langle M_0/v\rangle \succ\!\!- L_v$ et de $L\langle M_0/z\rangle \succ\!\!- L_z$ respectivement. Considérons $w \notin var(L,z) \cup fv(M_0)$ et $\tilde{T}$ une séquence zvw-indépendante. On examine la dernière règle utilisée :

(a) si $L = z$ alors $L_v = L_z = M_0$. D'un coté on a $L_w = M_0$, de l'autre H est z. D'où l'on conclut (1) et (2) facilement.

(b) si $L = L_0P_0$ alors $L_v = L'_v(P_0[v/z])$ et $L_z = L'_zP_0$, où $(L_0[v/z])\langle M_0/v\rangle \succ\!\!- L'_v$ et $L_0\langle M_0/z\rangle \succ\!\!- L'_z$.

1. Par h.r., il existe L'_w vérifiant $(L_0[w/z])\langle M_0/w\rangle \succ\!\!- L'_w$, donc on a $L_w = L'_w(P_0[w/z])$. De plus, l'équation

   $$L'_v(P_0[v/z])(\tilde{T})[v/z])\langle Q/v\rangle =_\alpha L'_w(P_0[w/z])(\tilde{T}[w/z])\langle Q/w\rangle$$

   implique $L_v\tilde{T}\langle Q/v\rangle =_\alpha L_w\tilde{T}\langle Q/w\rangle$.

2. Par la définition de $\asymp$, le terme H ne peut être qu'une application (H_0P_0) où $L_0 \asymp H_0$. Par h.r., il y a donc un terme $H'_z \asymp L'_z$ qui vérifie $H_0\langle M_0/z\rangle \succ\!\!- H'_z$. En utilisant la même règle on prouve $H\langle M_0/z\rangle \succ\!\!- H'_zP_0 = H_z$. Par conséquent, $H_z \asymp L_v$.

(c) si $L = L_0\langle P_0/x\rangle$, alors $L_v = L'_v\langle P_0[v/z]/x\rangle$ et $L_z = L'_z\langle P_0/x\rangle$, avec $z \neq x$, $L_0\langle M_0/z\rangle \succ\!\!- H'_z$ et $(L_0[v/z])\langle M_0/v\rangle \succ\!\!- L'_v$.

1. La preuve est identique à celle du point (b.1) précédent.

2. Trois cas se présentent, selon la forme de H. Pour $H = H_0\langle P_0/x\rangle$ avec $L_0 \asymp H_0$ on procède comme pour le cas (b.2) précédent.

   La deuxième possibilité consiste à avoir $H = H_0\langle P_0/x\rangle\langle Q_0/x'\rangle$ et $L_0 = H_0\langle Q_0/x'\rangle$, avec $x \neq x'$, $x' \notin fv(P_0)$, $x \notin fv(Q_0)$. L'inférence de L'_z est donc telle que $L'_z = L''_z\langle Q_0/x'\rangle$ et $H_0\langle M_0/z\rangle \succ\!\!- L''_z$.

   On dérive facilement $H_z = L''_z\langle P_0/x\rangle\langle Q_0/x'\rangle$. A noter que les conditions sur les variables sont satisfaites parce que $\neg(L{=_\alpha}H)$ implique $fv(H) = fv(L)$ et $bv(H) = bv(L)$. On en déduit $H_z \asymp L_z$, car $x \neq x'$.

   Le dernier cas, celui de $H = \mathrm{c}(H_0\langle P_0/x\rangle)$ et $L_0 = (\mathrm{c}H_0)$, est complètement similaire au précédent.

(d) si $L = \mathrm{c}L_0$, alors $L_z = \mathrm{c}L'_z$ et $H = \mathrm{c}H_0$, avec $H_0 \asymp L_0$. On procède comme auparavant, en utilisant l'hypothèse de récurrence et ensuite les règles utilisées dans les dérivations des saisies et de $H \asymp L$. $\square$

Lemme 3.5.10 *Pour tout $M, N, M' \in \Lambda_{rc}$, on a*

$$M \asymp N \ \&\ M {\to_{rc}} M' \ \Rightarrow\ \exists N'\ N {\to_{rc}} N' \ \&\ M' \asymp N'$$

Preuve. Dans la démonstration, on peut se restreindre à des termes M, N tels que l'inférence de $M \asymp N$ n'utilise qu'un axiome. Il suffit d'observer que si la transitivité de la relation $\asymp$ intervient, le lemme résulte d'un nombre fini d'applications du lemme restreint.

La preuve est par récurrence sur la taille de la démonstration de $M \asymp N$. Si $M \equiv N$ ou $M{=_\alpha}N$, la proposition est vraie par définition de $\to_{rc}$. C'est aussi vrai pour les axiomes tant que l'évaluation ne concerne que la partie “inamovible” du terme M; i.e. L dans le cas où M et N sont respectivement : $L\langle R/x\rangle$ et L ou $(LP)\langle R/x\rangle$ et $(L\langle R/x\rangle)P$ ou $(L\langle P/z\rangle)\langle R/x\rangle$ et $(L\langle R/x\rangle)\langle P/z\rangle$, ou encore $(\mathrm{c}L)\langle R/x\rangle$ et $\mathrm{c}(L\langle R/x\rangle)$.

Dans tous les autres cas, on procède par récurrence sur la taille de la dérivation de $M{\to_{rc}}M'$. La base (taille 0) suppose $M' = M$; alors on a $N' = N$. Pour le cas de récurrence, nous examinons la dernière règle utilisée. Indépendamment de M et N, tant que l'évaluation est de la forme

$$\frac{\begin{array}{c}\vdots\\ M^*{\to_{rc}}M'\end{array}}{M{\to_{rc}}M'}\ M{=_\alpha}M^* \text{ ou } M \equiv M^*$$

on a $N \asymp M^*$ et, par h.r., $\exists N'\ N{\to_{rc}}N' \asymp M'$. Nous nous limitons donc à étudier l'utilisation des règles restantes.

- Si $M = (LP)\langle R/x\rangle \asymp (L\langle R/x\rangle)P = N$ et $x \notin fv(P)$, l'évaluation est due soit à une saisie soit à une réduction β.

  - Saisie: Supposons que L consomme une ressource M_0 dans $R \equiv (M_0 \mid R_0)$; la dérivation, guidée par la syntaxe, donne :

    $$M' = (M''P)\langle R_0/x\rangle \quad \text{où} \quad L\langle M_0/x\rangle \succ M''$$

    car $x \notin fv(M_0)$ est vérifié par hypothèse. La même règle, appliquée à N, donne $N' = (M''\langle R_0/x\rangle)P$. Or, $M' \asymp N'$ est une instance d'axiome.

  - β: Alors $L = \lambda y.L'$ et $M' = L'\langle P/y\rangle\langle R/x\rangle$. Par ailleurs, si $z \notin fv(L', R, x)$, on déduit :

    $$N =_\alpha (\lambda z.(L'[z/y])\langle R/x\rangle)P \rightarrow_{rc} (L'[z/y])\langle R/x\rangle\langle P/z\rangle \;=\; N'$$

    $$N' \asymp (\alpha_y^z(L')\langle P/z\rangle)\langle R/x\rangle \quad \text{car } x \notin fv(P)$$

    Donc $(L'[z/y])\langle P/z\rangle =_\alpha L'\langle P/y\rangle$ implique $N' \asymp M'$ par transitivité de la relation.

- Si $M = (\mathrm{c}L)\langle R/x\rangle \asymp \mathrm{c}(L\langle R/x\rangle) = N$, l'évaluation saisit une ressource M_0 dans R. Le raisonnement est semblable à celui du point précédent.

- Si $M = (L\langle P/z\rangle)\langle R/x\rangle \asymp (L\langle R/x\rangle)\langle P/z\rangle = N$ et $z \notin fv(R, x), x \notin fv(P)$, l'évaluation $M \rightarrow_{rc} M'$ correspond à une saisie, sur la variable z ou sur la variable x.

  - Sur la variable x : soit $R \equiv (M_0 \mid R_0)$. Si $L\langle M_0/x\rangle \succ L'$, on obtient $M' = (L'\langle P/z\rangle)\langle R_0/x\rangle$ car $x \neq z$ et $z \notin fv(M_0)$ par hypothèse. Donc,

    $$L\langle R/x\rangle \rightarrow_{rc} L'\langle R_0/x\rangle \;\Rightarrow\; N \rightarrow_{rc} N' = (L'\langle R_0/x\rangle)\langle P/z\rangle \asymp M'$$

  - Sur la variable z : l'inférence de $M \rightarrow_{rc} M'$ repose sur l'évaluation d'un certain terme $K =_\alpha L\langle P/z\rangle$. C'est-à-dire, $K \rightarrow_{rc} M''$ et $M' = M''\langle R/x\rangle$. Or, $K = (L'[v/z])\langle P'/v\rangle$ où $L =_\alpha L'$ et $P =_\alpha P'$. Le terme M'' est ainsi:

    $$\frac{\begin{array}{c}\vdots\\ (L'[v/z])\langle M_0'/v\rangle \succ L_v\end{array}}{K \rightarrow_{rc} L_v\langle Q'/v\rangle \;=\; M''}\; P' \equiv (M_0' \mid Q')$$

    Soit $w \notin var(L'\langle R/x\rangle, z) \cup fv(M_0')$. D'après le lemme 3.5.9(1),

    $$\exists L_w \; (L'[w/z])\langle M_0'/w\rangle \succ L_w$$

Par ailleurs, $(L'\langle R/x\rangle)[w/z] = (L'[w/z])\langle R/x\rangle$ car $w \neq x, z \notin fv(R)$. Notre terme N' est dérivé comme suit :

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\\(L'[w/z])\langle M'_0/w\rangle\succ L_w\end{matrix}}{(L'[w/z])\langle R/x\rangle\langle M'_0/w\rangle\succ L_w\langle R/x\rangle}\; w\neq x, x\notin fv(M'_0)}{(L'[w/z])\langle R/x\rangle\langle P'/w\rangle\rightarrow_{rc}L_w\langle R/x\rangle\langle Q'/w\rangle}}{N=_\alpha L'\langle R/x\rangle\langle P'/z\rangle\rightarrow_{rc}L_w\langle R/x\rangle\langle Q'/w\rangle \;=\; N'}$$

D'autre part, on sait aussi que $L_v\langle Q'/v\rangle=_\alpha L_w\langle Q'/w\rangle$. Par conséquent, $M' = L_v\langle Q'/v\rangle\langle R/x\rangle=_\alpha L_w\langle Q'/w\rangle\langle R/x\rangle \asymp N'$.

Supposons maintenant que la longueur de dérivation de $M \asymp N$ est supérieure à un. Soit $L \asymp L'$. Si l'évaluation de M repose sur une évaluation de L, on applique l'hypothèse de récurrence. Sinon,

- ou bien $M = (LP) \asymp (L'P) = N$, $L = \lambda x.L_0$ et $M' = L_0\langle P/x\rangle$: par définition de $\asymp$, on sait que $P=_\alpha P'$ et que $L'=_\alpha\lambda x.L_0$; i.e. $L' = \lambda z.(L'_0[z/x])$ et $L'_0=_\alpha L_0$. La β-réduction de N donne alors $L'P\rightarrow_{rc}(L'_0[z/x])\langle P'/z\rangle =\ N' \asymp M'$.

- ou bien $M = (L\langle P/x\rangle) \asymp (L'\langle P/x\rangle) = N$ et M est réduit par saisie d'une ressource dans P. C'est-à-dire, $P \equiv (M_0 \mid Q)$ et $L\langle M_0/x\rangle\succ L_x \ \ \& \ \ M' = L_x\langle Q/x\rangle$.

Dans le cas où $L=_\alpha L'$, on procède comme dans le point précédent. Sinon, par le lemme 3.5.9 : $\exists L'_x\ \ L'\langle M_0/x\rangle\succ L'_x \ \ \& \ \ L_x \asymp L'_x$, et on déduit $N\rightarrow_{rc}L'_x\langle Q/x\rangle = N'$. Donc $M' \asymp N'$.

□

Lemme 3.5.11 *Pour tout* $M, N \in \Lambda_{rc}$, $M \asymp N \ \Rightarrow\ M\simeq_{\mathcal{A}}N$.

Preuve. Soit $M \asymp N$ et A un contexte applicatif. Étant donné la forme générale de ces contextes, à savoir $c^k([]\tilde{S}_0)\tilde{S}^k$, on a $A[M] \asymp A[N]$. Supposons que $A[M]\xrightarrow{\star}_{rc}V$, où V est une valeur; par le lemme 3.5.10, il existe N' t.q. $A[N]\xrightarrow{\star}_{rc}N'$ et $V \asymp N'$. De la définition de $\asymp$ on déduit $N'=_\alpha V$, donc $A[N]\rightarrow_{rc}V$. C'est-à-dire, $A[N]\Downarrow_{rc}$. □

Élargissement des paquets

On démontre que la capacité de convergence d'un terme n'est pas modifiée par l'élargissement des paquets qui constituent ses sous-termes.

Définition 3.5.12 *L'ordre* $T \Subset T'$ *- lisez* T est contenu dans T', *est défini par :*

$$P \quad \Subset \quad (P \mid Q)$$

$$P \Subset P' \quad \Rightarrow \quad \begin{cases} (MP) \Subset (MP') \\ M\langle P/x\rangle \Subset M\langle P'/x\rangle \\ cP \Subset cP' \end{cases}$$

$$M \Subset N \quad \Rightarrow \quad \begin{cases} (MR) \Subset (NR) \\ M\langle R/x\rangle \Subset N\langle R/x\rangle \\ \lambda x.M \Subset \lambda x.N \end{cases}$$

Lemme 3.5.13 *Pour tout* $M, N \in \Lambda_{rc}$, *si* $M \Subset N$ *alors* $M \sqsubseteq_{rc} N$.

Preuve. La preuve de l'énoncé est suffisamment similaire à celle du lemme 3.5.11 pour que nous puissions l'omettre. En effet, la démonstration s'articulerait comme suit : on devrait montrer, d'abord, l'analogue du lemme 3.5.10 pour notre ordre $\Subset$, i.e.

$$\forall M, N, M' \in \Lambda_{rc} \;\; M \Subset N \;\&\; M \overset{\star}{\rightarrow}_{rc} M' \;\Rightarrow\; \exists N' \in \Lambda_{rc} \;\; N \overset{\star}{\rightarrow}_{rc} N' \;\&\; M' \Subset N'$$

ensuite, la propriété

$$\forall V \in \mathbb{V}_{rc} \; \forall L \in \Lambda_{rc} \;\; V \Subset L \;\Rightarrow\; L \in \mathbb{V}_{rc}$$

qui est évidente car les valeurs sont des abstractions et $\Subset$ préserve la structure globale du terme (i.e. le nombre d'abstractions, d'applications, d'entrées de substitution et de tests de convergence). Par conséquent, si $A[M] \overset{\star}{\rightarrow}_{rc} V$, on déduit $A[N]{\Downarrow}_{rc}$. □

Partage des entrées de substitution

Le préordre observationnel est décroissant par rapport à certains partages de substitutions.

Lemme 3.5.14 *Soit* $\rho, \rho', \rho_1, \ldots, \rho_n$ *des entrées de substitution* $\langle P/x\rangle$ *avec* P *clos, portant sur la même variable* x. *Si* $P \equiv (N_1 \mid \ldots \mid N_n \mid Q)$ *alors*

1. $M\rho_0\tilde{S}_0\rho_1\tilde{S}_1 \sqsubseteq_{\mathcal{A}} M\tilde{S}_0(\rho_0 \mid \rho_1)\tilde{S}_1$
2. $M\langle (N_1\rho_1 \mid \ldots \mid N_n\rho_n \mid Q)/z\rangle\rho \sqsubseteq_{\mathcal{A}} (M\langle P/z\rangle)(\rho \mid \rho_1 \mid \ldots \mid \rho_n)$

3. $(M\rho)(N_1\rho_1 \mid \ldots \mid N_n\rho_n \mid Q) \sqsubseteq_{\mathcal{A}} (MP)(\rho \mid \rho_1 \mid \ldots \mid \rho_n)$

Preuve. Dans le cadre de λ_r, on pourrait montrer ces énoncés en utilisant la relation qui existe entre λ_d et λ_r. Puisqu'il s'agit ici du préordre applicatif associé à $\sqsubseteq_{rc}$, on donne des preuves directes. Remarquons que les trois énoncés à démontrer ont la même structure. Nous appelons C_i, C'_i les contextes applicatifs pour lesquels nous montrerons $C_i[M] \sqsubseteq_{\mathcal{A}} C'_i[M]$, avec $i = 1, 2, 3$ correspondant à chacun des points ci-dessus. Dans des observations d'ordre général, l'indice sera supprimé. Tout au long de la preuve, P' désigne le paquet $(N_1\rho_1 \mid \ldots \mid N_n\rho_n \mid Q)$,

$$\rho = \langle R/x \rangle \quad \text{et} \quad \rho_i = \langle R_i/x \rangle \;\; i = 1, \ldots, n$$

On sous-entend que ρ' et ρ'_i sont les entrées de substitution $\langle R/x' \rangle$ et $\langle R_i/x' \rangle$, pour tout i.

Soit un contexte applicatif $A = \mathrm{c}^k([]\tilde{W}_0)\tilde{W}^k$ tel que $A[C[M]]\Downarrow_{rc}$ en l étapes d'évaluation à (v)-réductions près. La preuve de $A[C'[M]]\Downarrow_{rc}$ est par récurrence sur l.

Si $l = 0$, alors $k = 0$; i.e. $A = []\tilde{W}_0$, où $\tilde{W}_0$ est vide ou composé de substitutions exclusivement. C'est aussi le cas de $\tilde{S}_0$ et $\tilde{S}_1$ dans la partie (1) de l'énoncé. Pour (1) et (2), $M \overset{\star}{\rightarrow}_{rc} \lambda y.M'$ en utilisant l'axiome (v). Les dérivations suivantes, où w est une variable neuve, montrent $C'_i[M]\tilde{W}_0\Downarrow_{rc}$, $i = 1, 2$:

$$C'_1[M] \overset{\star}{\rightarrow}_{rc} \lambda w.((M'[w/y])\tilde{S}_0(\rho_0 \mid \rho_1)\tilde{S}_1\tilde{W}_0)$$

$$C'_2[M] \overset{\star}{\rightarrow}_{rc} \lambda w.((M'[w/y])\langle P/z \rangle(\rho \mid \rho_1 \mid \ldots \mid \rho_n)\tilde{W}_0)$$

La partie (3) ne vérifie pas la prémisse.

Si $l > 0$, supposons que $M \overset{\star}{\rightarrow}_{rc} M'$ par application de (v) et $\neg(M' \overset{(v)}{\rightarrow}_{rc})$. Il existe des termes T_i, $i = 1, 2, 3$, correspondant aux parties (1), (2) et (3), tel que

$$A[C[M]] \overset{\star}{\rightarrow}_{rc} A[C[M']] \overset{\mu}{\rightarrow}_{rc} T_i \overset{\star}{\rightarrow}_{rc} V$$

où μ désigne l'axiome utilisé pour arriver à T_i (différent de (v) par hypothèse).

Dans le cas où μ s'applique à M', i.e. $M' \overset{\mu}{\rightarrow}_{rc} M''$, on déduit $T_i = A[T(M'')]$ et $A[C'[M]] \overset{+}{\rightarrow}_{rc} A[C'[M'']]$. L'hypothèse de récurrence donne $A[C'[M'']]\Downarrow_{rc}$. Par conséquent, $A[C'[M]]\Downarrow_{rc}$.

Quand l'utilisation de μ est partagée entre M' et C (ou A), on procède par cas sur l'axiome donnant lieu à μ. Celui-ci doit être autre que (c2), car la définition de contexte applicatif (donc de A) n'autorise pas de termes de tête de la forme $\mathrm{c}(Q \mid Q')$. Remarquons que si μ s'obtient par saisie d'une ressource alors $M' = \mathrm{c}^l(y\tilde{Z}_0)\tilde{Z}^l$ tandis que si μ s'obtient par (β) ou par (c1) alors $M' = \lambda y.N$. On prouve (1)-(3) séparément :

(1) Ici, $C = []\rho_0\tilde{S}_0\rho_1\tilde{S}_1$ et $C' = []\tilde{S}_0(\rho_0 \mid \rho_1)\tilde{S}_1$.

(*saisie*) Deux cas sont à considérer:

- si $y \in fv(M)$ et $y = x$, il existe L clos tel que $R_0 \equiv (L \mid Q)$, $M'' = \mathrm{c}^l(L\tilde{Z}_0)\tilde{Z}^l$ et $A[C[M]] \xrightarrow{\mu}_{rc} A[M''\langle Q/x\rangle\tilde{S}_0\rho_1\tilde{S}_1]\Downarrow_{rc}$ en $l-1$ étapes. La convergence de $A[C'[M]]$ découle du fait que l'évaluation ne modifie ni $\tilde{Z}_0$ ni $\tilde{Z}^l$, car L est clos, et
$$A[T'(M)] \xrightarrow{saisie}_{rc} A[M''\tilde{S}_0(\langle Q/x\rangle \mid \rho_1)\tilde{S}_1]\Downarrow_{rc} \text{ par h.r.}$$
- si $y \in fv(M)$ et $y \neq x$, l'évaluation saisie une ressource de $\tilde{S}_0\tilde{S}_1\tilde{W}_0\ldots\tilde{W}_k$. Supposons que $\tilde{S}_1 \equiv \tilde{S}_2\langle(L \mid Q)/y\rangle\tilde{S}_3$ et que y est libre dans $\tilde{S}_0\tilde{S}_2$. On a,
$$A[C[M]] \xrightarrow{\mu}_{rc} A[\mathrm{c}^l(L\tilde{Z}')\tilde{Z}'^l\rho_0'\tilde{S}_0'\rho_1'(\tilde{S}_2'\langle Q/z\rangle\tilde{S}_3)]\Downarrow_{rc} \quad \text{en } l-1 \text{ étapes}$$
  Par hypothèse de récurrence, et parce que
$$A[C'[M]] \xrightarrow{saisie}_{rc} A[\mathrm{c}^l(L\tilde{Z}')\tilde{Z}'^l\tilde{S}_0'(\rho_0' \mid \rho_1')(\tilde{S}_2'\langle Q/z\rangle\tilde{S}_3)]$$
  on conclut $A[T'(M)]\Downarrow_{rc}$.

  Si y est lié par $\tilde{S}_0$ ou par un des $\tilde{W}_i$, le raisonnement est similaire.

(β) L'argument à consommer est pris dans $\tilde{S}_0\tilde{S}_1\tilde{W}_0$. Nous examinons le cas où $\tilde{S}_0$ et $\tilde{S}_1$ sont constitués d'entrées de substitution et $W_0 = \langle\tilde{W}'/\tilde{v}\rangle P\tilde{W}''$. Alors,
$$A[C[M]]\xrightarrow{\star}_{rc}\xrightarrow{\mu}_{rc} A'[\alpha_y^w(N)\rho_0\tilde{S}_0\rho_1\tilde{S}_1\langle\tilde{W}'/\tilde{v}\rangle]\Downarrow_{rc}$$
avec $A' = \mathrm{c}^k([]\langle P/w\rangle\tilde{W}'')\tilde{W}^k$. Puisque
$$A[C'[M]]\xrightarrow{\star}_{rc}\xrightarrow{\beta}_{rc} A'[\alpha_y^w(N)\tilde{S}_0(\rho_0 \mid \rho_1)\tilde{S}_1\langle\tilde{W}'/\tilde{v}\rangle]$$
l'hypothèse de récurrence donne $A[C'[M]]\Downarrow_{rc}$.

(c1) Dans ce cas, $\tilde{S}_0$ et $\tilde{S}_1$ ne contiennent que des substitutions. De plus, $k > 0$; ce qui implique
$$A[C'[M]]\xrightarrow{\star}_{rc}A[\lambda x.(\alpha_y^x(N)\rho_0\tilde{S}_0\rho_1\tilde{S}_1)] \xrightarrow{\text{(C1)}}_{rc} \mathrm{c}^{k-1}(\mathbf{I}\tilde{W}_0)\tilde{W}_1\ldots\tilde{W}_k\Downarrow_{rc}$$

(2) Ici, $C = []\langle(N_1\rho_1 \mid \ldots \mid N_n\rho_n \mid Q)/z\rangle\rho$ et $C' = ([]\langle P/z\rangle)(\rho \mid \rho_1 \mid \ldots \mid \rho_n)$

(saisie) il faut considérer deux cas:

- si $y \in fv(M)$ et $y = z$, supposons que la ressource consommée est $N_1\rho_1$:
$$A[C(M)] \xrightarrow{\mu}_{rc} A[(\mathrm{c}^l(N_1\rho_1\tilde{Z}_0')\tilde{Z}'^l)\langle P_0/z'\rangle\rho]\xrightarrow{\star}_{rc}V$$

où $P_0 = (N_2\rho_2 \mid \ldots N_n\rho_n \mid Q)$ et z' est une variable neuve. Si P_1 désigne $(N_2 \mid \ldots \mid N_n \mid Q)$, par hypothèse de récurrence on a

$$A[(\mathrm{c}^l(N_1\rho_1\tilde{Z}'_0)\tilde{Z'}^l)\langle P_1/z'\rangle(\rho \mid \rho_2 \mid \ldots \mid \rho_n)]{\Downarrow_{rc}}$$

La partie (1) implique $\mathrm{c}((N_1\rho_1)\tilde{Z}'_0) \sqsubseteq_{\mathcal{A}} \mathrm{c}((N_1\tilde{Z}'_0)\rho_1)$. Par le lemme 3.5.11 on a $\mathrm{c}((N_1\tilde{Z}'_0)\rho_1) \simeq_{rc} (\mathrm{c}(N_1\tilde{Z}'_0))\rho_1$. Après itération,

$$\begin{array}{rl}
\mathrm{c}^l((N_1\rho_1)\tilde{Z}'_0)\tilde{Z'}^l\langle P_1/z'\rangle(\rho \mid \rho_2 \mid \ldots \mid \rho_n) & \sqsubseteq_{\mathcal{A}} \\
(\mathrm{c}(\mathrm{c}^{l-1}(N_1\tilde{Z}'_0)\tilde{Z'}^{l-1}))\rho_1\tilde{Z}'_l\langle P_1/z'\rangle(\rho \mid \rho_2 \mid \ldots \mid \rho_n) & \sqsubseteq_{\mathcal{A}} \text{ par (1)} \\
\mathrm{c}^l(N_1\tilde{Z}'_0)\tilde{Z'}^l\langle P_1/z'\rangle(\rho_1 \mid \rho \mid \rho_2 \mid \ldots \mid \rho_n) & \sqsubseteq_{\mathcal{A}} \\
\mathrm{c}^l(N_1\tilde{Z}'_0)\tilde{Z'}^l\langle P_1/z'\rangle(\rho \mid \rho_1 \mid \rho_2 \mid \ldots \mid \rho_n) &
\end{array}$$

D'autre part, $A[C'[M]] \xrightarrow{\mu}_{rc} A[\mathrm{c}^l(N_1\tilde{Z}'_0)\tilde{Z'}^l\langle P_1/z'\rangle(\rho \mid \rho_1 \mid \rho_2 \mid \ldots \mid \rho_n)]$ et ce dernier terme converge par h.r., ce qui implique $A[C'[M]]{\Downarrow_{rc}}$.

- si $y \in fv(M)$ et $y = x$, l'évaluation saisit une ressource dans ρ. C'est-à-dire, il existe L clos t.q. $R \equiv (L \mid Q')$ et $M'' = \mathrm{c}^l(L\tilde{Z}_0)\tilde{Z}^l$ où

  $$A[C(M)] \xrightarrow{\mu}_{rc} A[M''\langle P'/z\rangle\langle Q'/x\rangle] \xrightarrow{\star}_{rc} V$$

  On remarquera que cette évaluation ne génère pas d'α-conversions, car L est un terme clos. L'hypothèse de récurrence donne $A[M''\langle P/z\rangle(\langle Q'/x\rangle \mid \rho_1 \mid \ldots \mid \rho_n)]{\Downarrow_{rc}}$. Donc, $A[C'[M]]{\Downarrow_{rc}}$.

- si $y \in fv(M)$, $y \neq z$ et $y \neq x$, l'évaluation consomme une ressource dans $\tilde{Z}_0\tilde{Z}^l\tilde{W}_0\tilde{W}^k$. Dans ce cas, l'argumentation est similaire à celle de la partie (1).

(β) et (c1) Essentiellement par hypothèse de récurrence, comme dans la partie (1). Dans le cas de (β), la ressource doit être prise dans $\tilde{W}_0$.

(3) Ici, $C = ([]\rho)(N_1\rho_1 \mid \ldots \mid N_n\rho_n \mid Q)$ et $C' = ([]P)(\rho \mid \rho_1 \mid \ldots \mid \rho_n)$.

(saisie) Par hypothèse de récurrence.

(β) Dans ce cas,

$$A[C[M]] \xrightarrow{\star}_{rc} A[(\lambda y.N)\rho P'] \to_{rc} A[(\lambda z.(\alpha^z_x(N)\rho))P'] \xrightarrow{\mu}_{rc}$$

$$A[[z/x](N)\rho\langle P'/z\rangle] \xrightarrow{\star}_{rc} V$$

Par le lemme 3.5.11, $(N[z/x])\rho\langle P'/z\rangle \asymp (N[z/x])\langle P'/z\rangle\rho$ implique

$$A[(N[z/x])\langle P'/z\rangle\rho]\Downarrow_{rc}$$

avec le même nombre de pas que $A[(N[z/x])\rho\langle P'/z\rangle]$.

Puisque $A[(N[z/x])\langle P/z\rangle(\rho \mid \rho_1 \mid \ldots \mid \rho_n)]\Downarrow_{rc}$, on conclut $A[C'[M]]\Downarrow_{rc}$ par h.r.

(c1) Par vacuité : l'évaluation ne peut pas utiliser cet axiome en premier parce que l'argument P' doit être consommé. $\square$

η-expansion

Le calcul avec ressources (dans toutes ses variantes), comme tout calcul faible, vérifie une forme particulière de l'égalité η, à savoir :

$$M\Downarrow_{rc} \;\Rightarrow\; M \simeq_{rc} \lambda y.My^{\infty} \quad \text{pour tout } y \notin fv(M)$$

Dans la proposition suivante on montre $MP{\simeq_{rc}}My^{\infty}\langle P/y\rangle$, où $y \notin fv(M) \cup fv(P)$. La partie $\sqsubseteq_{rc}$ est immédiate. Pour montrer la partie ${}_{rc}\sqsupseteq$ on utilise le fait que $My^{\infty}\langle P/y\rangle\Downarrow_{rc}$ suppose $M\Downarrow_{rc}$, car l'application My^{∞} doit disparaître. Alors, on utilise l'égalité η mentionnée auparavant.

Proposition 3.5.15 *1.* $M\langle\tilde{P}/\tilde{x}\rangle Q{\simeq_{rc}}MQ\langle\tilde{P}/\tilde{x}\rangle$, *si* $\tilde{x} \notin fv(Q)$.

2. $M\langle P/x\rangle{\simeq_{rc}}M\langle y^{\infty}/x\rangle\langle P/y\rangle$, *où* $y \notin fv(M, P, x)$.

3. $MP{\simeq_{rc}}My^{\infty}\langle P/y\rangle$, *où* $y \notin fv(M) \cup fv(P)$.

Preuve. Par le lemme des contextes 3.5.5, il suffit de considérer des contextes applicatifs. Soit $A = \mathrm{c}^m([]\tilde{R}_0)\tilde{R}^m$ et $M = \mathrm{c}^k W\tilde{S}^k$ avec k maximal.

1. Cette partie de la proposition est conséquence du lemme 3.5.11, car

   $$M\langle\tilde{P}/\tilde{x}\rangle Q \asymp MQ\langle\tilde{P}/\tilde{x}\rangle$$

   est vérifié dès que $\tilde{x} \notin fv(Q)$ l'est.

2. Nous nous limitons à prouver la partie $\Rightarrow$ de l'équivalence, la partie $\Leftarrow$ lui étant similaire. Soit $A[M\langle P/x\rangle]\Downarrow_{rc}$ et $y \notin fv(M,P)$; nous montrons $A[M\langle y^{\infty}/x\rangle\langle P/y\rangle]\Downarrow_{rc}$ par récurrence sur le nombre l de réductions (β), $(fetch)$, (c1) et (c2) de l'évaluation normalisante de $A[M\langle P/x\rangle]$.

   $l = 0$: ici, $m = 0$, $\tilde{R}_0$ est vide ou constitué d'entrées de substitution et $M = (\lambda z.M')\langle\tilde{Q}/\tilde{w}\rangle$ (i.e. $k = 0$). Ceci implique, pour toute variable z neuve,

   $$(\lambda z.M')\langle\tilde{Q}/\tilde{w}\rangle\langle y^{\infty}/x\rangle\langle P/y\rangle\tilde{R}_0 \xrightarrow{*}_{rc} \lambda z'.(M'[z'/z]\langle\tilde{Q}/\tilde{w}\rangle\langle y^{\infty}/x\rangle\langle P/y\rangle\tilde{R}_0)\Downarrow_{rc}$$

$l > 0$: on montre l'énoncé par cas sur W (qui, par définition, ne commence pas par c):

- si $W = x\tilde{S}_0$ et M ne lie pas cette occurrence de x, alors le premier pas correspond à la saisie d'une ressource dans P; c'est-à-dire, $P \equiv (N \mid Q)$ et
  $$A[M\langle P/x\rangle] \rightarrow_{rc} A[(\mathrm{c}^k(N\tilde{S}'_0)\tilde{S'}^{k})\langle Q/z\rangle]\Downarrow_{rc} \text{ en } l-1 \text{ étapes}$$
  où z est une variable neuve et les $\tilde{S}'_i$, $i = 0, \ldots, k$, vérifient
  $$(\mathrm{c}^k(x\tilde{S}_0)\tilde{S}^k)\langle P/x\rangle =_\alpha ((\mathrm{c}^k(x\tilde{S}_0)[z/x]\tilde{S}^k))\langle P/z\rangle =_\alpha (\mathrm{c}^k(z\tilde{S}'_0)\tilde{S'}^{k})\langle P/z\rangle$$
  De l'autre côté,
  $$\begin{aligned} A[M\langle y^\infty/x\rangle\langle P/y\rangle] \;\rightarrow_{rc}\; & A[(\mathrm{c}(y\tilde{S}''_0)\tilde{S''}^{k})\langle y^\infty/x\rangle\langle P/y\rangle] \rightarrow_{rc} \\ & A[(\mathrm{c}^k(N\tilde{S}^*_0)\tilde{S^*}^{k})\langle y^\infty/z\rangle\langle Q/y\rangle] \end{aligned}$$
  où $y \notin bv(\tilde{S''}_0, \ldots, \tilde{S''}_k)$, $\tilde{S^*}_0 \ldots \tilde{S^*}_k$ ne capture pas des variables libres de N et
  $$\begin{aligned} (\mathrm{c}^k(x\tilde{S}_0)\tilde{S}^k)\langle P/x\rangle \;& =_\alpha\; \mathrm{c}^k(x\tilde{S}''_0)\tilde{S''}^{k} \\ (\mathrm{c}^k(y\tilde{S}''_0)\tilde{S''}^{k})\langle y^\infty/x\rangle\langle P/y\rangle \;& =_\alpha\; (\mathrm{c}^k(y\tilde{S}''_0)\tilde{S''}^{k})[z/x]\langle y^\infty/z\rangle\langle P/y\rangle =_\alpha \\ & \; (\mathrm{c}^k(y\tilde{S}^*_0)\tilde{S^*}^{k})\langle y^\infty/z\rangle\langle P/y\rangle \end{aligned}$$
  La différence essentielle entre $\tilde{S}'_0\tilde{S'}^{k}$ et $\tilde{S}^*_0\tilde{S^*}^{k}$ est que la première séquence lie possiblement y tandis que la seconde n'y est pas autorisée; donc
  $$(\mathrm{c}^k(N\tilde{S}'_0)\tilde{S'}^{k})\langle y^\infty/z\rangle\langle P/y\rangle =_\alpha (\mathrm{c}^k(N\tilde{S}^*_0)\tilde{S^*}^{k})\langle y^\infty/z\rangle\langle P/y\rangle$$
  Par hypothèse de récurrence, $A[(\mathrm{c}^k(N\tilde{S}'_0)\tilde{S'}^{k})\langle y^\infty/z\rangle\langle P/y\rangle]\Downarrow_{rc}$. Ce qui implique $A[(\mathrm{c}^k(N\tilde{S}^*_0)\tilde{S^*}^{k})\langle y^\infty/z\rangle\langle P/y\rangle]\Downarrow_{rc}$.
- si $W = z\tilde{S}_0$ avec $z \neq x$ (ou $z = x$, mais cette occurrence est liée par M), la suite $\tilde{S}_0 \ldots \tilde{S}_k\tilde{R}_0 \ldots \tilde{R}_m$ contient une entrée de substitution $\langle Q/z\rangle$ pour z. En supposant $\tilde{S}_i = \tilde{T}_0\langle Q/z\rangle\tilde{T}_1$, la séquence $\tilde{S}_0 \ldots \widetilde{S_{i-1}}\tilde{T}_0$ ne fournit pas de substitution pour z.

  Dans le cas où $Q \equiv (N \mid Q')$, on a $A[M\langle P/x\rangle] \rightarrow_{rc} A[M'\langle P/x\rangle]\Downarrow_{rc}$ en $l-1$ étapes d'évaluation, où
  $$M' = \mathrm{c}^k(N\tilde{S}'_0)\tilde{S}'_1 \ldots \widetilde{S'_{i-1}}(\tilde{T}'_0\langle Q'/v\rangle\tilde{T}_1)\widetilde{S_{i+1}} \ldots \tilde{S}_k$$
  Par hypothèse de récurrence,
  $$A[M\langle y^\infty/x\rangle\langle P/y\rangle] \rightarrow_{rc} A[M'\langle y^\infty/x\rangle\langle P/y\rangle] \Downarrow_{rc}$$

Considérons maintenant le cas de $\tilde{R}_i = \tilde{T}_0\langle Q/z\rangle\tilde{T}_1$; c'est-à-dire, le cas où $\tilde{S}_0 \ldots \tilde{S}_k\tilde{R}_0 \ldots \tilde{R}_{i-1}\tilde{T}_0$ n'offre pas de substitution pour z. Si $Q \equiv (N \mid Q')$ et v, w sont des variables neuves, le terme $A[M\langle P/x\rangle]$ converge en $l-1$ étapes de la façon suivante:

$$A[M\langle P/x\rangle]\rightarrow_{rc}\mathrm{c}^m(M'\langle P'/w\rangle)\tilde{R}'_0 \ldots \tilde{R}'{}_{i-1}\tilde{T}'_0\langle Q'/v\rangle\tilde{T}_1\tilde{R}_{i+1} \ldots \tilde{R}_m \Downarrow_{rc}$$

avec $M' = \mathrm{c}^k(N\tilde{S}'_0)\tilde{S}'^k$. En conclusion, $A[M\langle y^\infty/x\rangle\langle P/y\rangle] \Downarrow_{rc}$ par hypothèse de récurrence.

- si $W = (\lambda z.N)\tilde{S}_0$, on analyse deux cas. Le premier correspond à $\tilde{S}_0$ sans composante d'argument; alors $k > 0$ et on utilise la règle (c1) comme suit:

  $$A[M\langle P/y\rangle]\xrightarrow{\star}_{rc}A[(\mathrm{c}^k(\lambda z'.(N[z'/z])\tilde{S}_0)\tilde{S}^k)\langle P/x\rangle]\rightarrow_{rc}A[M'\langle P/x\rangle] \Downarrow_{rc}$$

  en $l-1$ étapes d'évaluation au plus, avec $M' = \mathrm{c}^{k-1}\mathbf{I}\tilde{S}^k$. Par h.r., on a

  $$A[M'\langle y^\infty/x\rangle\langle P/y\rangle] \Downarrow_{rc}$$

  Donc $A[M\langle y^\infty/x\rangle\langle P/y\rangle] \Downarrow_{rc}$, car

  $$A[M\langle y^\infty/x\rangle\langle P/y\rangle] \xrightarrow{+}_{rc} A[M'\langle y^\infty/x\rangle\langle P/y\rangle]$$

  Le second cas se présente quand $\tilde{S}_0$ a des composantes d'argument. L'évaluation de $A[M\langle P/x\rangle]$ commence par une β réduction, possiblement après quelques applications de (v). Comme dans le cas précédent, à partir de $A[M\langle y^\infty/x\rangle\langle P/y\rangle]$ on peut faire ces mêmes pas de réduction; l'énoncé est vérifié par hypothèse de récurrence.

- si $W = (Q_1 \mid Q_2)$ ou $W = Q^\infty$, le premier pas de l'évaluation normalisante de $A[M\langle P/x\rangle]$ consiste à sélectionner un terme N dans le paquet W. On peut toujours faire cette sélection dans $A[M\langle y^\infty/x\rangle\langle P/y\rangle]$, ce qui implique, par h.r., que ce terme converge.

3. Soit A un contexte applicatif; nous montrons la partie $\Rightarrow$ par récurrence sur la longueur d'inférence de $A[MP]\Downarrow_{rc}$. La preuve de la direction $\Leftarrow$ est complètement similaire.

   $l = 0$: Par vacuité.

   $l > 0$: Comme pour la partie (2) de l'énoncé. Si W est $z\tilde{S}_0$ ou $(Q_0 \mid Q_1)$ ou bien Q^∞, l'énoncé est vérifié essentiellement par h.r. On examine le cas $W = (\lambda z.N)\tilde{S}_0$:

   - si $k = 0$ et $\tilde{S}_0 = \langle\tilde{Q}/\tilde{v}\rangle$ alors, pour x une variable neuve,

     $$A[MP]\xrightarrow{\star}_{rc}A[(\lambda x.(N[x/z])\tilde{S}_0)P]\rightarrow_{rc}A[(N[x/z])\tilde{S}_0\langle P/x\rangle] \Downarrow_{rc}$$

en moins de l étapes. De plus,

$$A[My^\infty\langle P/y\rangle] \overset{\star}{\to}_{rc} A[(\lambda x.(N[x/z])\tilde{S}_0)y^\infty\langle P/y\rangle] \to_{rc}$$

$$A[(N[x/z])\tilde{S}_0\langle y^\infty/x\rangle\langle P/y\rangle]$$

Ce dernier terme converge par la partie (2) de l'énoncé. Donc

$$A[My^\infty\langle P/y\rangle] \Downarrow_{rc}$$

- si $k > 0$ et $\tilde{S}_0 = \langle \tilde{Q}/\tilde{v}\rangle$ alors, pour x une variable neuve,

  $$A[MP] \overset{\star}{\to}_{rc} A[(\mathrm{c}^k(\lambda x.(N[x/z])\tilde{S}_0)\tilde{S}^k)P] \to_{rc} A[(\mathrm{c}^{k-1}(\mathbf{I}\tilde{S}_1)\tilde{S}_2\ldots\tilde{S}_k)P] \Downarrow_{rc}$$

  en moins de l étapes. Par h.r.,

  $$A[My^\infty\langle P/y\rangle] \overset{+}{\to}_{rc} A[(\mathrm{c}^{k-1}(\mathbf{I}\tilde{S}_1)\tilde{S}_2\ldots\tilde{S}_k)y^\infty\langle P/y\rangle] \Downarrow_{rc}$$

- si $\tilde{S}_0 = \langle \tilde{Q}/\tilde{v}\rangle P'\tilde{S}'_0$, une argumentation similaire à celle des cas précédents permet de prouver l'énoncé.

$\square$

3.6 Théories $\simeq_r$ et $\simeq_{rc}$

Le non-déterminisme de l'évaluation $\to_r$ empêche une classification des termes comme dans le lambda calcul faible pur, où les termes sont soit résolubles, soit non-résolubles. Néanmoins, on peut définir les degrés de fonctionnalité et de non-résolution d'un terme et en tirer quelques conclusions, comme la "fully-laziness" de la théorie engendrée par $\simeq_r$, appelée $\mathcal{T}_r$. Nous utiliserons la notation Th pour l'opération de clôture par rapport à la égalité dans λ_r (cf. Barendregt [8]).

Définition 3.6.1 ***(Degrés de fonctionnalité dans λ_r)***

1. *Le degré de fonctionnalité de M est 0, noté $M \in \mathcal{O}^r_0$, ssi*

   $$\neg(\exists N \,.\, M \overset{\star}{\to}_r \lambda x.N)$$

2. *Le degré de fonctionnalité de M est $n+1$, noté $M \in \mathcal{O}^r_{n+1}$, ssi*

   $$\exists F \in \mathcal{O}^r_n \,.\, M \overset{\star}{\to}_r \lambda x.F$$

On écrit aussi, $M\,\mathcal{O}^r_{n+1}\lambda x.F$.

3. Le degré de fonctionnalité de M est ∞, noté $M \in O^r_\infty$, ssi

$$\forall n \in \mathbb{N}\,.\, M \notin O^r_n$$

Définition 3.6.2 ***(Degrés de non-résolution dans λ_r)***

1. *M est non-résoluble de degré 0, noté $M \in PO^r_0$, ssi*

$$M \in O^r_0 \ \& \ \neg(\exists \vec{R}\,.\, M \overset{\star}{\rightarrow}_r x\vec{R})$$

   On écrit $M PO^r_0 M'$ pour tout M' t.q. $M \overset{\star}{\rightarrow}_r M'$.

2. *M est non-résoluble de degré n, noté $M \in PO^r_{n+1}$, ssi*

$$\exists F \in PO^r_n\,.\, M \overset{\star}{\rightarrow}_r \lambda x.F$$

   On écrit $M PO^r_{n+1} \lambda x.F$.

3. *M est non-résoluble de degré ∞, noté $M \in PO^r_\infty$, ssi $M \in O^r_\infty$.*

Il faut remarquer qu'un terme de Λ_r peut avoir plusieurs degrés de fonctionnalité et de non-résolution.

Définition 3.6.3 *Un terme $M \in \Lambda_r$ a un* degré maximal de fonctionnalité *n, notation $M \in mO^r_n$, ssi pour tout k tel que $M \in O^r_k$, on a $n \geq k$.*

Un terme $M \in \Lambda_r$ a un degré maximal de non-résolution *n, notation $M \in mPO^r_n$, ssi pour tout k tel que $M \in PO^r_k$, on a $n \geq k$.*

Un terme $M \in \Lambda_r$ est strictement non-résoluble de degré *n, notation $M \in sPO^r_n$ ssi $M \in mPO^r_n$ et $\forall k \;\; M \in O^r_k / PO^r_k \;\Rightarrow\; k = \infty$.*

Remarque 3.6.4 *Pour tout $M \in \Lambda_r$, on a $M \in PO^r_0 \Leftrightarrow M \in sPO^r_0$.*

Toute évaluation des termes dans la classe mPO^r_n termine sur un terme non-résoluble (i.e. dans PO^r_k pour $k \geq 0$.) Suivant la terminologie usuelle, on dit que les termes de λ_r dans la classe sPO^r_0 sont fortement non-résolubles.

Lemme 3.6.5 *Soit $M, N, P \in \Lambda_r$.*

1. *$M \in PO^r_0 \Rightarrow (M \overset{\star}{\rightarrow}_r N \;\Rightarrow\; N \in PO^r_0)$*
2. *$M \in PO^r_0 \Rightarrow (M =_\alpha N \;\Rightarrow\; N \in PO^r_0)$*
3. *$M \in PO^r_0 \Rightarrow (M\langle P/x\rangle \in PO^r_0 \ \&\ MP \in PO^r_0)$*

Lemme 3.6.6 *Pour tout $M \in \Lambda^o_r$, on a $M \in sPO^r_0$ ssi $M \Uparrow_r$.*

Preuve. Soit M un terme clos. Partie $\Rightarrow$: $M \in \mathrm{PO}_0^r$ et $M{\Downarrow_r}$ est une contradiction parce que $M \in \mathrm{O}_0^r$ ssi il n'y a pas d'évaluation de M qui termine sur une abstraction. Partie $\Leftarrow$: soit $M{\Uparrow_r}$; i.e. $\neg(\exists x, M'\ M \overset{\star}{\rightarrow}_r \lambda x.M')$. Donc $M \in \mathrm{mO}_0^r$. Comme M est clos, $M \in \mathrm{PO}_0^r$. C'est-à-dire, $M \in \mathrm{sPO}_0^r$. □

Théorème 3.6.7 ***($\mathcal{T}_r$ est "fully lazy")***

La théorie $\mathcal{T}_r$ identifie uniquement les termes strictement non-résolubles de même degré. C'est-à-dire,

$$\forall M \in sPO_m^r\ \forall N \in sPO_n^r\ \ M \sqsubseteq_r N \Leftrightarrow m \leq n$$

Preuve. Nous montrons d'abord la partie $\Rightarrow$. Soit

$$\mathbb{A}_q(M,N) = \{[\,]\langle\tilde{\Omega}/\tilde{z}\rangle \underbrace{\mathbf{I}\ldots\mathbf{I}}_{q \text{ fois}} \;/\; \{z_1,\ldots,z_p\} \supseteq fv(M,N)\}$$

Supposons $M \in \mathrm{sPO}_m^r$, $N \in \mathrm{sPO}_n^r$, et $m > n \geq 0$. Nous montrons, par récurrence sur n,

$$\forall q\ \forall A \in \mathbb{A}_q\ \ (n \leq q \leq m-1\ \Rightarrow\ A[M]{\Downarrow_r}\ \&\ A[N]{\Uparrow_r})$$

Si $n = 0$ alors $m = k+1$, $0 \leq q \leq k$ et $M \overset{\star}{\rightarrow}_r \lambda x_1.M'$ avec $M \in \mathrm{PO}_k^r$. Par le lemme 3.6.5, $A[N] = N\langle\tilde{\Omega}/\tilde{z}\rangle\,\mathbf{I}\ldots\mathbf{I}{\Uparrow_r}$. On prouve $A[M]{\Downarrow_r}$ par récurrence sur q. Si $q = 0$, alors $A[M] \overset{\star}{\rightarrow}_r (\lambda x.M')\langle\tilde{\mathbf{I}}/\tilde{z}\rangle \overset{\star}{\rightarrow}_r (\lambda x.M'\langle\tilde{\Omega}/\tilde{z}\rangle){\Downarrow_r}$. Si $q \geq 1$ alors $m \geq 2$; c'est-à-dire,

$$A[M] \overset{\star}{\rightarrow}_r M'\langle\tilde{\Omega}/\tilde{z}\rangle\langle\mathbf{I}/x\rangle\ \ \underbrace{\mathbf{I}\ldots\mathbf{I}}_{q-1 \text{ fois}}\ = A'[M']$$

De $M' \in \mathrm{PO}_k^r$ et $q \leq k+1$, par récurrence on déduit $A'[M']{\Downarrow_r}$. Donc $A[M]{\Downarrow_r}$.

Dans le cas où $n = k+1$, on a $m \geq k+2$. Par définition, $M \overset{\star}{\rightarrow}_r \lambda x.M'$ et $M' \in \mathrm{sPO}_{m-1}^r$. On définit $A' = [\,]\langle\tilde{\mathbf{I}}/\tilde{z}\rangle\langle\mathbf{I}/x\rangle\,\underbrace{\mathbf{I}\ldots\mathbf{I}}_{q \text{ fois}}$, qui vérifie $A[M] \overset{+}{\rightarrow}_r A'[M']$. D'autre part, sans perte de généralité (voir lemme 3.6.5), $N \overset{\star}{\rightarrow}_r \lambda x.N'$ où $N' \in \mathrm{sPO}_h^r$, $h \leq k$. Donc, $A_n[N] \overset{+}{\rightarrow}_r A'[N']$. Il est clair que $A' \in \mathbb{A}_{q-1}(M',N')$. On remarque que $h \leq k < k+1 \leq q \leq m-1$ implique $h \leq q-1 \leq m-2$. Par hypothèse de récurrence, $A'[M']{\Downarrow_r}$ et $A'[N']{\Uparrow_r}$ pour tout N'. Donc $A[M]{\Downarrow_r}$ et $A[N]{\Uparrow_r}$.

La preuve de la partie $\Leftarrow$ est basée sur la proposition suivante :

$$M \in \mathrm{sPO}_m^r\ \Rightarrow\ \forall A\ (A = [\,]\tilde{R}\ \&\ |\tilde{R}| = q\ \Rightarrow\ A[M] \in \mathrm{sPO}_{m-q}^r)$$

(ici, $0 - q$ désigne 0.) Par récurrence sur m :

- si $m = 0$, par le lemme 3.6.5, $A[M] \in \mathrm{sPO}_0^r$.

- si $m = k + 1$, en supposant $\tilde{R} = \rho P \tilde{R}'$ et en négligeant les α-conversions, on a

  $$A[M] \overset{\star}{\rightarrow}_r (\lambda x.M')\tilde{R} \overset{\star}{\rightarrow}_r M'\rho\langle P/x\rangle \tilde{R}' = A'[M']$$

  où $M \overset{\star}{\rightarrow}_r \lambda x.M'$ et $M' \in \mathrm{sPO}^r_h$ avec $h \leq k$. (Remarquons que c'est le seul type de dérivation issue de $A[M]$.) Par hypothèse de récurrence, $A'[M'] \in \mathrm{sPO}^r_{h-(q-1)}$. C'est-à-dire, $A'[M'] \in \mathrm{sPO}^r_{(h+1)-q}$. Donc $A[M] \in \mathrm{PO}^r_{(h+1)-q}$ pour tout h t.q. $M \in \mathrm{PO}^r_{h+1}$; en particulier pour $h = k$. D'où l'on conclut $A[M] \in \mathrm{sPO}^r_{(k+1)-q}$.

Soit $A[M]\Downarrow_r$; alors $\exists t \ A[M] \in \mathrm{sPO}^r_t \ \& \ t > 0$. Par l'énoncé précédent, $t = m - q$ où q est le nombre d'arguments du contexte A. Donc $q < m$. D'autre part, toujours par cet énoncé, $A[N] \in \mathrm{sPO}^r_{n-q}$. On a $n - q > 0$ puisque $q < m \leq n$. I.e. $A[N]\Downarrow_r$.
□

La "maximalité" de la théorie $\mathcal{T}_r$, énoncé dans le théorème 3.6.9 ci-dessous, repose sur le lemme suivant :

Lemme 3.6.8 *Soit $M \in \Lambda^o_r$. Pour tout $k \in \mathbb{N}$, si $k > 0$ et $M \in O^r_k/PO^r_k$, alors*

$$\exists \textit{ contexte applicatif } B \ \ \boldsymbol{I} \sqsubseteq_r B[M]$$

Théorème 3.6.9 ***(Théorie maximale)***
La théorie $\mathcal{T}_r$ est maximale. C'est-à-dire,

$$\forall M, N \in \Lambda_r \ \neg(M \sqsubseteq_r N) \ \Rightarrow$$

Th($\mathcal{T}_r \cup M \sqsubseteq_r N$) n'est pas "fully lazy" ou est incohérente

Preuve. Soit $\neg(M \sqsubseteq_r N)$; i.e. il existe un contexte applicatif A tel que $A[M]\Downarrow_r$ mais $A[N]\Uparrow_r$. Par le lemme 3.6.6, $A[N] \in \mathrm{sPO}^r_0$. Nous examinons deux cas :

- si $A[M] \in \mathrm{sPO}^r_m$ avec $m > 0$, on conclut $\mathrm{Th}(\mathcal{T}_r \cup M \sqsubseteq_r N)$ n'est pas "fully lazy" par le théorème 3.6.7;

- si $A[M] \in \mathrm{O}^r_m/\mathrm{PO}^r_m$ avec $m \in \mathbb{N}$ et $m > 0$, par le lemme 3.6.8 on a $\exists B \ \mathbf{I} \sqsubseteq_r B[A[M]]$. En ce qui concerne $A[N]$, par le lemme 3.6.5, $C[A[N]] \in \mathrm{sPO}^r_0$ quelque soit C. Donc $B[A[N]] \sqsubseteq_r \Omega$.

  En conclusion, la théorie $\mathrm{Th}(\mathcal{T}_r \cup M \sqsubseteq_r N)$ nous permet de déduire $\mathbf{I} \sqsubseteq_r \Omega$, d'où $\mathbf{I}X \sqsubseteq_r \Omega X$ pour tout terme X, c'est-à-dire, $X \sqsubseteq_r \Omega$ est prouvable pour tout X. Cette théorie est donc incohérente, car on a déjà $\Omega \sqsubseteq_r Y$ pour tout terme Y.

□

Nous entamons la preuve du lemme 3.6.8. Dans un calcul déterministe, si M est un terme résoluble clos, alors il existe une seule suite de dérivations (à α-conversion près) de la forme

$$\begin{gathered} M \overset{\star}{\rightarrow}_r \lambda x_1.M_1 \\ M_1 \overset{\star}{\rightarrow}_r \lambda x_2.M_2 \\ \vdots \\ M_{m-1} \overset{\star}{\rightarrow}_r \lambda x_m.z\tilde{R} \end{gathered}$$

t.q. il existe $j \in [1, m]$ qui vérifie $z = x_j$. C'est le cas par exemple du calcul avec multiplicités λ_m. Il est assez simple de constater que le contexte B suivant [2] transforme M en l'identité : soit $p = |\tilde{R}|$; on pose

$$B = []\,\underbrace{\mathbf{I}\ldots\mathbf{I}}_{j-1}(\lambda z_1 \ldots z_p.\mathbf{I})\,\underbrace{\mathbf{I}\ldots\mathbf{I}}_{m-j-1}$$

En effet, $B[M] \simeq_m \mathbf{I}$ grâce à la confluence du calcul. Ce n'est évidement pas le cas dans λ_r, où un terme résoluble clos appartient, en général, à plusieurs classes O^r_k et PO^r_g. Par exemple $M = (\lambda x.x)(\lambda x.xP \mid \lambda xy.y \mid \lambda x.\Omega)$ est à la fois dans O^r_1, O^r_2 et PO^r_1. Dans le cadre de λ_r, le contexte B pour le calcul λ_r dépend du degré de fonctionnalité (différent de 0); on ne peut donc montrer que $B[M] \Downarrow_r \mathbf{I}$.

Preuve du lemme 3.6.8. Soit $M \in \Lambda^o_r$ et $k \geq 0$ t.q. $M \in \mathrm{O}^r_k/\mathrm{PO}^r_k$, et supposons que $M \in \mathrm{O}^r_k E$ avec $E = \lambda x_1 \ldots x_k.x_j\tilde{R}$ et $p = \mid \tilde{R} \mid$.

On définit $B_k = []\,\underbrace{\mathbf{I}\ldots\mathbf{I}}_{j-1}(\lambda z_1 \ldots z_p.\mathbf{I})\underbrace{\mathbf{I}\ldots\mathbf{I}}_{k-j-1}$; ce terme a une seule évaluation convergente. En effet, l'évaluation du terme $B_k[E]$ est déterministe :

$$B_k[E] \overset{\star}{\rightarrow}_r (\lambda z_1 \ldots z_p.\mathbf{I})\tilde{R}'\langle \tilde{P}/\tilde{x} \rangle \overset{\star}{\rightarrow}_r \mathbf{I}$$

où $P_i = \mathbf{I}$ pour tout $i \neq j$ et $P_j = \mathbf{1}$. Par conséquent, $B[M] \Downarrow_r \mathbf{I}$ est vérifié. Donc $\mathbf{I} \sqsubseteq_r B[M]$. □

Des degrés de fonctionnalité et de non-résolution d'un terme dans Λ_d peuvent être définis suivant les schémas des définitions 3.6.1 et 3.6.2; il suffit de remplacer l'évaluation $\rightarrow_r$ par $\rightarrow_d$. Nous utiliserons O^d_k et PO^d_k comme notation pour les ensembles de termes résolubles et non-résolubles de degré k, respectivement.

Simplification d'arguments

Nous présentons ici une nouvelle classe de termes de λ_r qui sont dans la relation $\sqsubseteq_r$, motivés par l'observation que les sous-termes non-résolubles d'un

---

[2] C'est le contexte utilisé par Boudol et Laneve [18] dans la preuve de maximalité de la théorie $\simeq_m$.

terme ne sont "utiles" qu'une fois dans les dérivations convergentes du terme. Plus précisément, quelque soit l'évaluation convergente d'un terme celle-ci n'évalue pas plus d'un terme non-résoluble (possiblement aucun). La conséquence est que des sous-termes non-résolubles d'un terme donné peuvent être effacés sans affecter la convergence potentielle du terme.

Définition 3.6.10 ***(Ordre de simplification)***
L'ordre strict $M \lesssim N$ - lisez M est une simplification de *N, est :*

$$(\lambda x_1 \ldots x_n.\Omega \mid Q) \quad \lesssim \quad (\lambda x_1 \ldots x_n.\Omega \mid \lambda x_1 \ldots x_m.\Omega \mid Q) \text{ si } n \geq m$$

$$P \lesssim P' \quad \Rightarrow \quad \begin{cases} (MP) \lesssim (MP') \\ M\langle P/x\rangle \lesssim M\langle P'/x\rangle \end{cases}$$

$$M \lesssim N \quad \Rightarrow \quad \begin{cases} (MP) \lesssim (NP) \\ M\langle P/x\rangle \lesssim N\langle P/x\rangle \\ \lambda x.M \lesssim \lambda x.N \end{cases}$$

Il est évident que $M \lesssim N \Rightarrow M \sqsubseteq_r N$, car $M \lesssim N$ implique $M \Subset N$, donc, par le lemme 3.5.13, $M \sqsubseteq_r N$. Nous montrons ensuite que $M \lesssim N \Rightarrow N \sqsubseteq_r M$ est aussi vérifiée. La validité de cette implication s'explique par le lemme suivant

Lemme 3.6.11 *Soit $(\lambda x_1 \ldots x_m.\Omega)\tilde{R}\Downarrow_r$ et $|\tilde{S}| = |\tilde{R}|$. Alors*

$$n \geq m \Rightarrow (\lambda x_1 \ldots x_n.\Omega)\tilde{S}\Downarrow_r$$

Preuve. Soit $\tilde{x} = x_1 \ldots x_n$. On procède par récurrence sur le nombre k d'arguments dans $\tilde{R}$. Si $k = 0$, pour toute séquence $\tilde{S} = \langle\tilde{Q}/\tilde{z}\rangle$ on a $(\lambda\tilde{x}.\Omega)\tilde{S} \asymp (\lambda\tilde{x}.\Omega)\Downarrow_r$, car $(\lambda\tilde{x}.\Omega)$ est un terme clos. Alors $(\lambda\tilde{x}.\Omega)\tilde{S}\Downarrow_r$ par le lemme 3.5.11.

Dans le cas où $k \geq 1$, i.e. $\tilde{R} = \langle\tilde{Q}/\tilde{z}\rangle P\tilde{R}'$, soit $\tilde{S} = \langle\tilde{Q}'/\tilde{z}\rangle P'\tilde{S}'$ compatible avec $\tilde{R}$. Il est clair que $(\lambda x_1 \ldots x_m.\Omega)\tilde{R} \asymp (\lambda x_1 \ldots x_m.\Omega)P\tilde{R}'$. Donc le lemme 3.5.11 implique $(\lambda x_1 \ldots x_m.\Omega)P\tilde{R}'\Downarrow_r$. Toute évaluation convergente de ce terme commence par une application de (β) :

$$(\lambda x_1 \ldots x_m.\Omega)P\tilde{R}' \rightarrow_r (\lambda x_2 \ldots x_m.\Omega)\langle P/x_1\rangle\tilde{R}'\Downarrow_r$$

Puisque $\tilde{R}'$ a exactement $k-1$ arguments, par hypothèse de récurrence on a $(\lambda x_2 \ldots x_n.\Omega)\langle P'/x_1\rangle\tilde{S}'\Downarrow_r$. Or,

$$(\lambda\tilde{x}.\Omega)\langle\tilde{Q}'/\tilde{z}\rangle P'\tilde{S}' \asymp (\lambda\tilde{x}.\Omega)P'\tilde{S}' \rightarrow_r (\lambda x_2 \ldots x_n.\Omega)\langle P'/x_1\rangle\tilde{S}'$$

Ainsi, $(\lambda\tilde{x}.\Omega)\tilde{S}\Downarrow_r$ est vérifié. □

Le résultat qui nous intéresse est :

Lemme 3.6.12 ***(Lemme de simplification)***
Si $M, N \in \Lambda_r$, alors $M \lesssim N \Rightarrow N \sqsubseteq_r M$.

Preuve. Premièrement, nous allons montrer les points (i) et (ii) suivants, sous l'hypothèse $M \lesssim N$:

$$(i) \quad N \in \mathrm{PO}^r_k \;\Rightarrow\; \exists h \; M \in \mathrm{PO}^r_h \;\&\; h \geq k$$

$$(ii) \quad N\mathrm{O}^r_k/\mathrm{PO}^r_k N' \;\Rightarrow\; \exists M'' \; M\mathrm{O}^r_k/\mathrm{PO}^r_k M''$$

Soit $P = (\lambda x_1 \ldots x_n.\Omega \mid Q)$ et $P' = (\lambda x_1 \ldots x_n.\Omega \mid \lambda y_1 \ldots y_m.\Omega \mid Q)$ où $n \geq m$. On procède par récurrence sur la preuve de $M \lesssim N$, puis par récurrence sur k :

1. $M = L\langle P/x\rangle$ et $N = L\langle P'/x\rangle$. On montre (i) par récurrence sur Q, puis par récurrence sur k. Supposons $Q = \mathbf{1}$. Si $N \in \mathrm{PO}^r_0$, on a $L \in \mathrm{PO}^r_0$. Donc $M \in \mathrm{PO}^r_0$. Si $N \in \mathrm{PO}^r_k$ avec $k > 0$, alors $L \in \mathrm{O}^r_h$. On doit considérer trois cas :

   - si $N \overset{\star}{\to}_r (\lambda z.L')\langle P'/x\rangle \to_r \lambda v.(L'[z/v]\langle P'/x\rangle) \in \mathrm{PO}^r_k$, avec $L \overset{\star}{\to}_r (\lambda z.L')$, alors $(L'[z/v]\langle P'/x\rangle) \in \mathrm{PO}^r_{k-1}$. Par hypothèse de récurrence, on a $(L'[z/v]\langle P/x\rangle) \in \mathrm{PO}^r_{k-1}$. Donc $M \in \mathrm{PO}^r_k$.
   - si $N \overset{\star}{\to}_r x\tilde{R}\langle P'/x\rangle \to_r (\lambda\tilde{x}.\Omega)\tilde{R}\langle \lambda\tilde{y}.\Omega \mid Q/x\rangle \in \mathrm{PO}^r_k$, où $L \overset{\star}{\to}_r x\tilde{R}$, alors $k = n - |\tilde{R}|$. Puisque $M \overset{\star}{\to}_r (\lambda\tilde{x}.\Omega)\tilde{R}\langle Q/x\rangle$, on a $M \in \mathrm{PO}^r_k$.
   - si $N \overset{\star}{\to}_r x\tilde{R}\langle P'/x\rangle \to_r (\lambda\tilde{y}.\Omega)\tilde{R}\langle \lambda\tilde{x}.\Omega \mid Q/x\rangle \in \mathrm{PO}^r_k$, alors $k = m - |\tilde{R}|$. On déduit $M \in \mathrm{PO}^r_h$, avec $h = n - |\tilde{R}| \geq k$.

   Dans le cas où Q est différent de $\mathbf{1}$, deux possibilités sont encore à examiner (en plus de ce qui a été fait) :

   - $N \in \mathrm{PO}^r_0$ et $L \overset{\star}{\to}_r x\tilde{R}$: il existe H t.q. $Q \equiv (H \mid Q')$ et

     $$N \overset{\star}{\to}_r H\tilde{R}'\langle \lambda\tilde{x}.\Omega \mid \lambda\tilde{y}.\Omega \mid Q'/x\rangle \in \mathrm{PO}^r_0$$

     Par hypothèse de récurrence, $M \overset{\star}{\to}_r H\tilde{R}'\langle \lambda\tilde{x}.\Omega \mid\mid Q'/x\rangle \in \mathrm{PO}^r_0$.
   - $N \overset{\star}{\to}_r x\tilde{R}\langle P'/x\rangle$ et il existe H t.q. $Q \equiv (H \mid Q')$ et

     $$N \overset{\star}{\to}_r H\tilde{R}'\langle(\lambda\tilde{y}.\Omega \mid \lambda\tilde{x}.\Omega \mid Q'/x\rangle = N' \in \mathrm{PO}^r_k$$

     Par hypothèse de récurrence $M \overset{\star}{\to}_r H\tilde{R}'\langle \lambda\tilde{x}.\Omega \mid Q'/x\rangle \in \mathrm{PO}^r_h$ avec $h \geq k$.

   On montre (ii), par récurrence sur Q et sur k. Supposons $Q = \mathbf{1}$. Quand $N\mathrm{O}^r_0/\mathrm{PO}^r_0 N'$, on vérifie $L \overset{\star}{\to}_r z\tilde{R}$, où l'occurrence de tête de z est libre, $z \neq x$ et $N' = z\tilde{R}\langle P'/x\rangle$ il est clair que $M\mathrm{O}^r_0 z\tilde{R}\langle P/x\rangle$. Si $k > 0$, un des cas suivants est vérifié :

   - si $L \overset{\star}{\to}_r x\tilde{R}$, et si l'occurrence de tête de x est libre, on procède comme dans le cas $k = 0$.

- si $L\xrightarrow{\star}_r \lambda z.L'$, alors $N\xrightarrow{\star}_r \lambda v.L'[v/z]\langle P'/x\rangle \mathrm{O}^r_k/\mathrm{PO}^r_k \lambda v.N''$ implique

$$L'[v/z]\langle P'/x\rangle \mathrm{O}^r_{k-1} N''$$

  Par récurrence, $L'[v/z]\langle P/x\rangle \mathrm{O}^r_{k-1}/\mathrm{PO}^r_{k-1} M''$ avec $M'' \lesssim N''$. Donc $M \mathrm{O}^r_k/\mathrm{PO}^r_k \lambda v.M''$.

Dans le cas où Q est différent de $\mathbf{1}$, il faut encore considérer une possibilité : $L\xrightarrow{\star}_r x\tilde{R}$, l'occurrence de tête de x est libre, et $\exists H\ Q \equiv (H \mid Q')$ t.q.

$$N\xrightarrow{\star}_r H\tilde{R}'\langle \lambda\tilde{x}.\Omega \mid \lambda\tilde{y}.\Omega \mid Q'/v\rangle \mathrm{O}^r_0/\mathrm{PO}^r_0 N'$$

Par hypothèse de récurrence (Q' est plus petit que Q),

$$M\xrightarrow{\star}_r H\tilde{R}'\langle \lambda\tilde{x}.\Omega \mid\mid Q'/v\rangle \mathrm{O}^r_0/\mathrm{PO}^r_0 M'$$

où $M' \lesssim N'$.

2. $M = LP$ et $N = LP'$. Si $N \in \mathrm{PO}^r_0$, une des possibilités est $L \in \mathrm{PO}^r_0$. Donc $M \in \mathrm{PO}^r_0$. Sinon, $L \in \mathrm{O}^r_{g+1}$, i.e. $N\xrightarrow{\star}_r (\lambda z.L')P' \rightarrow_r L'\langle P'/z\rangle \in \mathrm{PO}^r_0$, où $L\xrightarrow{\star}_r \lambda z.L'$. Par le point précédent, $L'\langle P/z\rangle \in \mathrm{PO}^r_h$ avec $h \geq 0$. Donc $M \in \mathrm{PO}^r_h$.

   Dans tous les autres cas, $N \in \mathrm{O}^r_k$ avec $k > 0$, le raisonnement est similaire (on utilise directement le résultat précédent.)

3. $M = L\langle T/x\rangle$ et $N = L'\langle T/x\rangle$ où $L \lesssim L'$. Comme précédemment, les preuves sont par récurrence sur T puis sur k. À titre d'exemple, pour T différent de $\mathbf{1}$, on a : si $N \in \mathrm{PO}^r_0$ et $L \in \mathrm{O}^r_0/\mathrm{PO}^r_0$, alors

$$N\xrightarrow{\star}_r x\tilde{R}\langle T/x\rangle \rightarrow_r H\tilde{R}'\langle T'/v\rangle \in \mathrm{PO}^r_0$$

   où $L'\xrightarrow{\star}_r x\tilde{R}$, $T \equiv (H \mid T')$. Par hypothèse de récurrence (sur $L \lesssim L'$), $L\mathrm{O}^r_0/\mathrm{PO}^r_0 x\tilde{S} \lesssim x\tilde{R}$. De la définition de l'ordre $\lesssim$, on déduit $H\tilde{S}'\langle T'/v\rangle \lesssim H\tilde{R}'\langle T'/v\rangle$. Par conséquent, $M\xrightarrow{\star}_r H\tilde{S}'\langle T'/v\rangle \in \mathrm{PO}^r_0$.

4. $M = LT$ et $N = L'T$ où $L \lesssim L'$. On procède comme pour le cas (2), en utilisant (3).

5. $M = \lambda x.L$ et $N = \lambda x.L'$ où $L \lesssim L'$. Directement par hypothèse de récurrence (dans le cas (i), on a $k \geq 0$).

Soit A un contexte applicatif. Par définition de l'ordre $\lesssim$, on vérifie $M \lesssim N \Rightarrow A[M] \lesssim A[N]$. Si $A[N]\Downarrow_r$, alors $A[N] \in \mathrm{O}^r_k$, $k > 0$. Par (i) et (ii), on déduit $A[M] \in \mathrm{O}^r_h$, avec $h \geq k$ i.e. $A[M]\Downarrow_r$. $\Box$

La propriété de simplification que nous avons énoncée pour λ_r, n'est pas vérifiée dans λ_r^c, au moins dans toute sa généralité. La taille des paquets, plus précisément le nombre d'éléments non-résolubles qui les composent n'est pas toujours négligeable dans λ_r^c.

Contre-exemple 3.6.13 *Soit* $P = (\lambda y.\Omega)^m$ *et* $P' = (\lambda y.\Omega)^n$, *où* $m < n \geq 1$. *Donc,* $P \lesssim P'$ *implique* $(\lambda zx.xz^\infty)P \lesssim (\lambda zx.xz^\infty)P'$. *Par le lemme 3.6.12, on a*

$$(\lambda zx.xz^\infty)P' \sqsubseteq_r (\lambda zx.xz^\infty)P$$

Cependant, $\neg((\lambda zx.xz^\infty)P' \sqsubseteq_{rc} (\lambda zx.xz^\infty)P)$. *On définit* $A = [\,](\lambda w.c^n(w))$, *où,* $c^i(w) = \underbrace{(cw)\ldots(cw)}_{i \textit{ fois}}$ *est une sorte de test de convergence d'arité i. Le contexte A sépare les deux termes mentionnés auparavant; i.e.*

$$\begin{array}{lll}
A[(\lambda zx.xz^\infty)P'] & \overset{\star}{\rightarrow}_{rc} & (xz^\infty)\langle P'/z\rangle\langle \lambda w.c^n(v))/x\rangle \\
 & \rightarrow_{rc} & (\lambda v.c^n(w))z^\infty\langle P'/z\rangle\langle \boldsymbol{1}/x\rangle \\
 & \rightarrow_{rc} & c^n(w)\langle z^\infty/w\rangle\langle P'/z\rangle\langle \boldsymbol{1}/x\rangle \\
 & \overset{\star}{\rightarrow}_{rc} & (c(\lambda y.\Omega))c^n(w)\langle z^\infty/w\rangle\langle(\lambda y.\Omega)^{n-1}/z\rangle\langle \boldsymbol{1}/x\rangle \\
 & \rightarrow_{rc} & \boldsymbol{I}c^n(w)\langle z^\infty/w\rangle\langle(\lambda y.\Omega)^{n-1}/z\rangle\langle \boldsymbol{1}/x\rangle \\
 & \vdots & \\
 & \rightarrow_{rc} & \boldsymbol{I}\langle z^\infty/w\rangle\langle \boldsymbol{1}/z\rangle\langle \boldsymbol{1}/x\rangle \\
 & \asymp & \boldsymbol{I}\Downarrow_{rc}
\end{array}$$

mais, puisque $n-m \geq 1$, *l'évaluation de* $A[(\lambda zx.xz^\infty)P]$ *est bloquée par le manque de termes dans P; i.e.*

$$\begin{array}{lll}
A[(\lambda zx.xz^\infty)P] & \overset{\star}{\rightarrow}_{rc} & c^n(w)\langle z^\infty/w\rangle\langle(\lambda y.\Omega)^m/z\rangle\langle \boldsymbol{1}/x\rangle \\
 & \overset{\star}{\rightarrow}_{rc} & c^{n-m}(w)\langle z^\infty/w\rangle\langle \boldsymbol{1}/z\rangle\langle \boldsymbol{1}/x\rangle \Uparrow_{rc}
\end{array}$$

Nous avons souligné dans l'introduction que les modèles que nous savons construire pour λ_r ne sont pas complètement adéquats. Ce résultat repose essentiellement sur le contre-exemple 3.6.13, qui implique la non-définissabilité de c dans λ_r (cf. [33].) Nous rediscuterons ce point dans le chapitre 5.

3.7 Approximants dans le calcul avec ressources

Le but de cette section est de définir une interprétation algébrique du calcul avec ressources, basée sur la notion d'approximant d'un terme, tel que le préordre engendré constitue une sémantique adéquate de λ_r. Nous montrons comment le programme de Lévy [47] pour le lambda calcul faible peut être adapté à λ_r. Au départ, l'idée était d'appliquer ce programme directement sur λ_m, dont l'atout principal serait d'avoir une évaluation déterministe. Nous verrons que la syntaxe des termes dans Λ_m est trop contraignante.

En gros, un approximant de $M \in \Lambda_r$ est un terme en forme normale par rapport à une notion de réduction forte, obtenu en éliminant les radicaux de chaque sous-terme de M, soit par évaluation - en utilisant les règles (β) et de saisie, soit par remplacement des radicaux par $\perp$ (appelée souvent $\perp$-réduction.) La première étape consiste donc en la définition de la réduction forte, dont l'évaluation $\rightarrow_r$ doit être une sorte de stratégie faible. La différence avec le lambda calcul pur, où l'on peut prendre l'égalité $=_\beta$, est que l'élimination sûre des radicaux de la forme $(MP)\langle Q/x\rangle$ nécessite la distribution de la substitution $\langle Q/x\rangle$ aux sous-termes M et P, une capacité que $\rightarrow_r$ ne possède pas. La réduction forte, $\rhd_d$, sera donc définie sur le calcul des paquets λ_d.

Il semble que la relation $\rhd_d$ - quelque soit sa définition, échappe au cadre des multiplicités. La raison est la suivante: supposons $M = (x(\lambda z.yz^1)^2)\langle \mathbf{I}^1/y\rangle$ et essayons de déterminer, rapidement, ses approximants. Il est clair que $\perp$, $x\perp^1$ et aussi $x(\lambda z.\perp)^2$, $x(\lambda z.\perp)^1$ et $x(\lambda z.\mathbf{I}z^1)^1$ sont des approximants de M. Pour expliquer le dernier terme, on peut dire que M est meilleur que $(x(\lambda z.yz^1)^1)\langle \mathbf{I}^1/y\rangle$, et que, intuitivement

$$(x(\lambda z.yz^1)^1)\langle \mathbf{I}^1/y\rangle \;\rhd_d\; x((\lambda z.yz^1)^1\langle \mathbf{I}^1/y\rangle) \;\rhd_d\; x(\lambda z.y\langle \mathbf{I}^1/y\rangle z^1)^1 \;\rhd_d\; x(\lambda z.\mathbf{I}z^1)^1$$

Cependant, si $C = [\,]\langle \lambda v.vv^1/x\rangle$, aucun de ces approximants A ne vérifie $C[A]\Downarrow_m$ tandis que $C[M]\Downarrow_m$. Le problème est que la distribution de l'entrée de substitution $\langle \mathbf{I}/y\rangle$ à l'argument $(\lambda z.yz^1)^2$ ne peut pas se faire correctement sans considérer l'argument comme le paquet de ressources $(\lambda z.yz^1 \mid \lambda z.yz^1)$, de façon à pouvoir obtenir $(\lambda z.\mathbf{I}z^1 \mid \lambda z.\perp)$ par réduction. Mais l'approximant $B = x(\lambda z.\mathbf{I}z^1 \mid \lambda z.\perp)$, pour lequel on a $C[B]\Downarrow_m$, n'est pas un terme du calcul avec multiplicités.

Définition 3.7.1 ***(Règles de réduction $\rhd_d$)*** *La relation $\rhd_d \subset \Lambda_d \times \Lambda_d$ est définie par:*

$$\frac{N \in M\lceil P/x\rceil}{(\lambda x.M)P \rhd_d N} \qquad \frac{M \rhd_d M'}{MP \rhd_d M'P} \qquad \frac{M \rhd_d M'}{\lambda x.M \rhd_d \lambda x.M'}$$

$$\frac{P \rhd_d P'}{MP \rhd_d MP'} \qquad \frac{M \rhd_d M' \quad P \equiv (M \mid P')}{P \rhd_d (M' \mid P')}$$

La relation de réduction forte $\rhd_d$ sur le calcul avec paquets étend l'évaluation $\rightarrow_d$. Il s'agit donc d'une relation non-déterministe. Le blocage est assimilé à la divergence. La définition d'approximant pour λ_r s'appuie sur cette relation. En fait, les réductions fortes de $M \in \Lambda_r$, seront les réductions fortes des éléments de $\{[M]\}$.

Définition 3.7.2 ***(Approximants)***

L'ensemble $\mathcal{L}_r$ *des approximants des termes du calcul avec ressources* λ_r *est le plus petit sous-ensemble de* $\Lambda_d \cup \{\bot\}$ *qui contient* $\lambda x_1 \ldots x_n.\bot$ *et* $\lambda\vec{x}.x\vec{A}_1 \ldots \vec{A}_s$ *lorsque* $\vec{A}_i = (A_1 \mid \ldots \mid A_n)$ *et* $A_j \in \mathcal{L}_r$*. Il faut noter que les paquets des approximants n'ont que des multiplicités finies.*

Les approximants directs de $T \in \Lambda_d$ *sont les termes* $\overline{\omega}(T) \subseteq \mathcal{L}_r$ *définis par récurrence comme suit :*

$$\overline{\omega}(\lambda\tilde{x}.(\lambda y M)P\tilde{R}) = \{\lambda\tilde{x}.\bot\}$$

$$\overline{\omega}(\lambda\tilde{x}.yP_1 \ldots P_s) = \{\lambda\tilde{x}.y\overline{\omega}(P_1) \ldots \overline{\omega}(P_s)\}$$

$$\overline{\omega}(P) = \begin{cases} \{\bot\} & si\ P = \boldsymbol{1} \\ \bigcup(M' \mid Q') & si\ P = (M \mid Q)\ et\ M' \in \overline{\omega}(M)\ et\ Q' \in \overline{\omega}(Q) \\ \bigcup \overline{\omega}(M)^m & si\ P = M^\infty\ et\ m \in \mathbb{N} \end{cases}$$

L'ensemble des approximants d'un terme $M \in \Lambda_r$ *est :*

$$\mathcal{A}(M) = \bigcup\{\overline{\omega}(N) \,/\, \exists M' \in \{[M]\}\ \ M' \overset{\star}{\rhd}_d N\}$$

Quel est le préordre $\leq$ sur $\mathcal{L}_r$, engendré par $\Omega \leq A$? Nous devons préciser ce qu'est le préordre sur les paquets d'approximants. La définition la plus restrictive à laquelle on peut penser est : $\vec{A} \leq \vec{B}$ si $\vec{A} \equiv (A_1 \mid \ldots \mid A_m)$, $\vec{B} \equiv (B_1 \mid \ldots \mid B_m)$, et $A_i \leq B_i$ quelque soit $i \in [1, m]$. On observera que avec cette définition, on vérifie

$$x(A \mid \bot) \not\leq x(A \mid B)$$

car on doit tenir compte des ressources Ω au même titre que le reste. Pourtant, $x(A \mid \Omega) \sqsubseteq_r x(A \mid B)$. Dans la définition que nous donnons ci-dessous, $x(A \mid \bot) \leq x(A \mid B)$.

Définition 3.7.3 ***(Ordre*** $\leq$ ***sur*** $\mathcal{L}_r$***)***

L'ordre $\leq$ *sur les (paquets d') approximants est la plus petite relation transitive qui vérifie :*

$$\begin{array}{rcll} \bot & \leq & A & pour\ tout\ A \in \mathcal{L}_r \\ \lambda x.A & \leq & \lambda x.B & si\ A \leq B \\ x\vec{A}_1 \ldots \vec{A}_n & \leq & x\vec{B}_1 \ldots \vec{B}_n & si\ \vec{A}_i \leq \vec{B}_i\ pour\ tout\ i \end{array}$$

où $\vec{A} \leq \vec{B}$ *ssi* $\exists A_1, \ldots, A_m \neq \bot\ \exists B_1, \ldots B_m, \vec{B}'$ *t.q.*

$$\vec{A} \equiv (A_1 \mid \ldots \mid A_m \mid \bot \mid \ldots \mid \bot)\ \ \&\ \ \vec{B} \equiv (B_1 \mid \ldots \mid B_m \mid \vec{B}')$$

et $\forall i \in [1, m]\ \ A_i \leq B_i$

Nous verrons plus bas que $\leq$ engendre un préordre sur λ_r qui est adéquat par rapport à $\sqsubseteq_r$. Cependant, la définition de $\leq$ est trop restrictive pour engendrer un préordre complètement adéquat. Par exemple, l'ordre $\leq$ permet de compter les composantes non-résolubles des paquets, ce qui est impossible dans le cadre de la sémantique standard de λ_r (cf. lemme de simplification 3.6.12 et contre-exemple 3.6.13.)

L'ensemble des approximants d'un terme de λ_r n'est pas dirigé par rapport au préordre $\leq$, comme l'est $\mathcal{A}(M)$ dans le cadre du lambda calcul pur [47]. Par exemple, si $M = K \oplus F$, alors $\{K, F\} \subseteq \mathcal{A}(M)$ mais il n'existe pas $A \in \mathcal{A}(M)$ t.q. $K \leq A$ et $F \leq A$. La cause est ici le non-déterminisme de la relation de réduction $\rhd_d$. Un autre type d'exemple provient de la disponibilité limitée des arguments. Par exemple, si $M = (xyy)\langle K/y \rangle$, alors $A_1 = xK\bot$ et $A_2 = x\bot K$ sont deux approximants de M qui n'ont pas de borne supérieure dans $\mathcal{A}(M)$.

Il est évident que le théorème de continuité standard, $C[M]\Downarrow_r$ ssi $\exists A \in \mathcal{A}(M)\ C[A]\Downarrow_r$, n'est pas vérifié (th. 5.7 [47]) : comme le calcul avec ressources n'est pas confluent, il n'y a pas nécessairement un approximant qui peut être pris comme représentant d'un terme, dans un contexte C donné. Néanmoins, pour reproduire la convergence de $C[M]$, on va utiliser un approximant de $\mathcal{A}(M)$ chaque fois que le terme M (ou un terme obtenu à partir de M par renommages) apparaît en position de tête dans l'évaluation convergente. Puisque l'évaluation est finie, le nombre d'approximants nécessaires est fini.

Pourtant, le préordre intensionnel $\leq_{\mathcal{L}_r}$ défini par $M \leq_{\mathcal{L}_r} N \overset{def}{\Leftrightarrow} \mathcal{A}(M) \subseteq \mathcal{A}(N)$ n'admet pas une caractérisation en fonction du préordre $\leq$. En effet, $\mathcal{A}(M) \subseteq \mathcal{A}(N) \Rightarrow \forall A \in \mathcal{A}(M)\ \exists B \in \mathcal{A}(N)\ \ A \leq B$ (on prend $B = A$), mais l'implication inverse n'est pas vérifiée. Ceci est dû au fait que $\mathcal{A}(M)$ n'est pas clos par le bas par rapport au préordre $\leq$; en particulier, il existe $M \in \Lambda_r$ t.q. $\bot \notin \mathcal{A}(M)$. Par exemple, si $M = \mathbf{I}$. La différence avec la définition de Lévy est que, dans le lambda calcul pur, la propriété de Church-Rosser permet de conclure $\mathcal{A}(M) = \mathcal{A}(N)$ si $M =_\beta N$. En particulier, $\mathbf{I}M =_\beta M$ et $\bot \in \mathcal{A}(\mathbf{I}M)$ impliquent $\bot \in \mathcal{A}(M)$ pour tout M. Toutefois, il est possible construire un domaine algébrique pour l'interprétation de λ_r sur la base du treillis des approximants $\mathcal{L}_r$ et le préordre $\leq$. Pour cela, il suffit de faire la complétion par cônes non-vides[3] de $\mathcal{L}_r$. On note $\downarrow\!\mathcal{A}(M)$ le plus petit cône non-vide qui contient $\mathcal{A}(M)$. On définit

$$M \leq_{\mathcal{L}_r} N \overset{def}{\Leftrightarrow} \downarrow\!\mathcal{A}(M) \subseteq \downarrow\!\mathcal{A}(N)$$

et on démontre

$$M \leq_{\mathcal{L}_r} N \Leftrightarrow \forall A \in \mathcal{A}(M)\ \exists B \in \mathcal{A}(N)\ \ A \leq B$$

Le lemme d'approximation que nous allons prouver dit

$$\forall M \in \Lambda_r\ \ C[M]\Downarrow_r \Leftrightarrow \exists A_1, \ldots, A_n \in \downarrow\!\mathcal{A}(M)\ C[A_1 \oplus \ldots \oplus A_n]\Downarrow_r$$

---

[3] On entend par cône non-vide sur $(\mathcal{L}_r, \leq)$, tout sous-ensemble X de $\mathcal{L}_r$ t.q. $\bot \in X$, et si $A \in X$ et $B \leq A$, alors $B \in X$.

où $\oplus$ désigne le choix interne dans le calcul de ressources; i.e. $A_1 \oplus \ldots \oplus A_n = x\langle A_1 \mid \ldots \mid A_n/x\rangle$. Un corollaire de ce résultat est $M \leq_{\mathcal{L}_r} N \Rightarrow M \sqsubseteq_r N$. La preuve du lemme d'approximation nécessite quelques résultats que nous montrons maintenant.

Lemme 3.7.4 *1.* $A \in \overline{\varpi}(N) \Rightarrow A \sqsubseteq_r N$

2. $A \leq B \Rightarrow A \sqsubseteq_r B$

Preuve.

1. Il n'est pas difficile de voir que $\Omega \sqsubseteq_r N$ est vérifié pour tout N: c'est un corollaire du théorème 3.6.7 de caractérisation de la théorie $\mathcal{T}_r$. L'énoncé découle donc du fait que $\sqsubseteq_r$ est une precongruence et du lemme 3.5.13.

2. La preuve de cette partie de l'énoncé est par récurrence sur la dérivation de $A \leq B$. Si $A = \Omega$, alors $A \sqsubseteq_r B$ pour tout B comme nous l'avons souligné dans la partie (1). Si $A = \lambda x.A' \leq \lambda x.B'$ et $A' \leq B'$, par h.r. $C[A'] \sqsubseteq_r C[B']$ pour tout contexte C. En particulier pour $C = \lambda x.[]$.

   Supposons maintenant $A = y\vec{A}_1 \ldots \vec{A}_n \leq y\vec{B}_1 \ldots \vec{B}_n$, avec $\vec{A}_i \leq \vec{B}_i$ pour tout i. On se restreint au cas $n = 1$, suffisant pour illustrer le raisonnement. On note $\vec{A}$ et $\vec{B}$ les arguments de y et on suppose, sans perte de généralité, que

   $$\vec{A} = (A_1 \mid \ldots \mid A_k \mid \Omega \mid \ldots \mid \Omega) \quad \text{et} \quad \vec{B} = (B_1 \mid \ldots \mid B_k \mid \vec{B}')$$

   où $A_i \leq B_i$ pour tout $i = 1, \ldots, k$. Par h.r., $C[A_i] \sqsubseteq_r C[B_i]$ quelque soit le contexte C. Il existe des contextes $C_1, \ldots, C_{k+1}$ t.q.

   $$y\vec{A} = C_1[A_1] \sqsubseteq_r C_1[B_1] = C_2[A_2] \sqsubseteq_r \ldots \sqsubseteq_r C_{k-1}[B_{k-1}] =$$

   $$C_k[A_k] \sqsubseteq_r C_{k+1}[B_k] = y(B_1 \mid \ldots \mid B_k \mid \Omega \ldots \mid \Omega) \sim_r y(B_1 \mid \ldots \mid B_k) \sqsubseteq_r y\vec{B}$$

   où la dernière inégalité est conséquence du lemme 3.5.13 d'élargissement des paquets.

□

Lemme 3.7.5 *Soit $L, M, M', N, N' \in \Lambda_d$ et P, Q des paquets de λ_d. Alors,*

1. $M\lceil P/x\rceil\lceil Q/y\rceil = \bigcup_{Q \equiv (Q_0 | Q_1)} M[z/x]\lceil Q_0/y\rceil\lceil P\lceil Q_1/y\rceil/z\rceil,$

où $z \notin fv(M, Q_0, x, y)$.

2. $M \triangleright_d N \ \& \ N' \in N\lceil Q/y\rceil \Rightarrow \exists M' \ M' \triangleright_d N' \ \& \ M' \in M\lceil Q/y\rceil$

3. $M \triangleright_d N \ \& \ N' \in L\lceil (N \mid Q)/y\rceil \Rightarrow \exists M' \ M' \triangleright_d N' \ \& \ \in L\lceil (M \mid Q)/y\rceil$

Preuve.

1. Par inspection de la définition de la substitution dans λ_d. À titre d'exemple, nous traitons le cas de M égal à une variable et celui de M égal à une application. Soit $L \in M\lceil P/x\rceil\lceil Q/y\rceil$ et $H \in M\lceil P\lceil Q/y\rceil/x\rceil$. On montre $L \in M\lceil P\lceil Q/y\rceil/x\rceil$ et $H \in M\lceil P/x\rceil\lceil Q/y\rceil$. Si L ou H sont Ω, l'énoncé est immédiat.

   • Supposons $M = x$. Dans les deux cas, $x = y$ et $x \neq y$, on a $L \in N\lceil Q/y\rceil$ & $P \equiv (N \mid P')$. Pour tout $R \in P'\lceil \mathbf{1}/y\rceil$, on vérifie $(L \mid R) \in P\lceil Q/y\rceil$. Donc, $L \in M\lceil P\lceil Q/y\rceil/x\rceil$. Quant à l'autre inclusion, observons qu'il existe $R \equiv (H \mid R')$ et $R \in P\lceil Q/y\rceil$. C'est-à-dire, il existe une division de Q, soit $Q \equiv (Q_0 \mid Q_1)$, et une division de P, soit $P \equiv (N \mid P')$, t.q. $H \in N\lceil Q_0/y\rceil$. Par la définition de la substitution, $H \in N\lceil Q/y\rceil$ aussi. Donc $H \in M\lceil P/x\rceil\lceil Q/y\rceil$.

   • Si $M = M'R$, il existe L' et R' t.q. $L = L'R'$, $L' \in M'\lceil P_0/x\rceil\lceil Q_0/y\rceil$ et $R' \in R\lceil P_1/x\rceil\lceil Q_1/y\rceil$, avec $P \equiv (P_0 \mid P_1)$ et $Q \equiv (Q_0 \mid Q_1)$. Par récurrence, $L' \in M'\lceil P_0\lceil Q_0/y\rceil/x\rceil$ et $R' \in R\lceil P_1\lceil Q_1/y\rceil/x\rceil$, d'où

   $$L'R' \in (M'R)\lceil P_0\lceil Q_0/y\rceil \mid P_1\lceil Q_1/y\rceil/x\rceil \subseteq (M'R)\lceil P\lceil Q/y\rceil/x\rceil$$

   Quant à H, il existe H' et R' t.q. $H = H'R'$, $H' \in M'\lceil S_0/x\rceil$ et $R' \in R\lceil S_1/x\rceil$, où $(S_0 \mid S_1) \in P\lceil Q/y\rceil$. Par définition, $P \equiv (P_0 \mid P_1)$ et $Q \equiv (Q_0 \mid Q_1)$ t.q. $S_i \in P_i\lceil Q_i/y\rceil$, $i = 0, 1$. Par récurrence, $H' \in M'\lceil P_0/x\rceil\lceil Q_0/y\rceil$ et $R' \in R\lceil P_1/x\rceil\lceil Q_1/y\rceil$. Donc, $(H'R') \in (M'R)\lceil P/x\rceil\lceil Q/y\rceil$.

2. Par récurrence sur la taille de la dérivation de $M \triangleright_d M_1$. Seuls quelques cas seront analysés; le reste peut être traité de façon complètement similaire.

   • Si $M = (\lambda x.M_0)P$ et $N \in M_0\lceil P/x\rceil$, alors $N' \in M_0\lceil P/x\rceil Q/y$. Par la partie (1) de l'énoncé, $N' \in M_0[z/x]\lceil Q_0/y\rceil\lceil P\lceil Q_1/y\rceil/z\rceil$ où $Q \equiv (Q_0 \mid Q_1)$. Par la définition de la substitution,

   $$\exists H, P' \;\; N' \in H\lceil P'/z\rceil \;\; \& \;\; H \in M_0[z/x]\lceil Q_0/y\rceil \;\; \& \;\; P' \in P\lceil Q_1/y\rceil$$

   Alors, $M' = (\lambda z.H)P' \triangleright_d N'$ et $M' \in M\lceil Q/y\rceil$.

   • Si $M = M_0P$, $M_0 \triangleright_d M'_0$ et $N = M'_0P$. Par hypothèse, $N' = LR$ où $L \in M'_0\lceil Q_0/y\rceil$ et $R \in P\lceil Q_1/y\rceil$. Par l'hypothèse de récurrence, il existe $L' \in M_0\lceil Q_0/y\rceil$ t.q. $L' \triangleright_d L$. D'autre part, $L'R \in M\lceil Q/y\rceil$ par la définition de la substitution. On conclut $L'R \triangleright_d LR = M'_1$.

   • Si $M = M_0(N_0 \mid P)$, $N_0 \triangleright_d N_1$ et $M_1 = M_0(N_1 \mid P)$, alors $N' = L(H \mid P')$ où $L \in M_0\lceil Q_0/y\rceil$, $H \in N_1\lceil Q_1/y\rceil$ et $P' \in P\lceil Q_2/y\rceil$. Par récurrence, il existe $H' \in N_0\lceil Q_1/y\rceil$ t.q. $H' \triangleright_d H$. Alors $L(H' \mid P') \in M\lceil Q/y\rceil$ et $L(H' \mid P') \triangleright_d L(H \mid P') = M'_1$.

3. On procède par récurrence sur L :

   • si $L = y$, alors $N' = N$, ou $N' = \Omega$, ou encore $N' = H$ avec $Q \equiv (H \mid Q')$. Le terme M' recherché est soit M, soit Ω, soit H, respectivement. Si $L = x$, alors $N' = x$ et $M' = x$.

   • si $L = (\lambda x.L')$, alors $N' = \lambda z.N''$ où $N'' \in L'[z/x]\lceil (N \mid Q)/y \rceil$. Par récurrence, il existe $H \in L'[z/x]\lceil (M \mid Q)/y \rceil$ t.q. $H \rhd_d N''$. Donc $\lambda z.H \rhd_d \lambda z.N''$. Par la définition de la substitution, il est clair que $\lambda z.H \in L\lceil (M \mid Q)/y \rceil$.

   • si $L = (L'P)$, alors $N' = N''P'$ où $(N \mid Q) \equiv (R_0 \mid R_1)$, $N'' \in L'\lceil R_0/y \rceil$ et $P' \in P\lceil R_1/y \rceil$. Si N est une composante du paquet R_0, on conclut par l'hypothèse de récurrence comme dans le cas précédent. Si N appartient à R_1, par la définition de substitution sur les paquets, il existe H tel que $P \equiv (H \mid P_0)$, $R_1 \equiv (N \mid R' \mid R'')$ et $P' \equiv (H' \mid P'')$ où $H' \in H\lceil (N \mid R')/y \rceil$. Par récurrence, il existe $H'' \in H\lceil (M \mid R')/y \rceil$ t.q. $H'' \rhd_d H'$. Le terme cherché est donc $M' = N''(H'' \mid P'')$.

□

Lemme 3.7.6 *Pour tout* $M, N, V \in \Lambda_d$ *t.q.* $M \rhd_d N \rightarrow_d V$*, il existe* $W \in \Lambda_d$ *t.q. le diagramme suivant commute :*

$$\begin{tikzcd}
M \arrow[r, phantom, "\rhd_d"] \arrow[d, "d"'] & N \arrow[d, "d"'] \\
W \arrow[r, phantom, "\rhd_d"] & V
\end{tikzcd}$$

où $\rightarrow_d$ *est l'évaluation faible du calcul des paquets définie dans la figure 3.4.*

Preuve. Par récurrence sur la taille de l'arbre de dérivation de $M \rhd_d N$.

- Si $M = (\lambda x.L)P$ et $N \in L\lceil P/x \rceil$, alors $M \rightarrow_d N$. De plus, $N \rightarrow_d V$ implique $N \rhd_d V$.

- Si $M = LP$, $L \rhd_d L'$ et $N = L'P$, alors $L' = (\lambda x.H)P_1 \dots P_n$ où $n \geq 0$. On montre l'énoncé par récurrence sur n. Si $n = 0$, alors $V \in H\lceil P/x \rceil$. D'autre part, la réduction de L vers L' peut être de deux types : supposons $L = \lambda x.H'$ et $H' \rhd_d H$; par le lemme 3.7.5(2), il existe $W \in H'\lceil P/x \rceil$ t.q. $W \rhd_d V$. Donc, $M = LP \rightarrow_d W \rhd_d V$. Supposons maintenant $L = (\lambda y.L'')Q$ et $L' \in L''\lceil Q/y \rceil$. Alors, $M = LP \rightarrow_d L'P = (\lambda x.H)P \rhd_d V$.

  Dans le cas où $n > 0$, $V \in H\lceil P_1/x \rceil P_2 \dots P_n P$. On a aussi

  $$L \rhd_d L' \rightarrow_d H\lceil P_1/x \rceil P_2 \dots P_n = V'$$

  donc, par hypothèse de récurrence, il existe W' t.q. $L \rightarrow_d W' \rhd_d V'$. Le terme W cherché est $W'P$, car $M = LP \rightarrow_d W'P \rhd_d V'P = V$.

- Si $M = \lambda x.L$, $L \rhd_d L'$ et $N = \lambda x.L'$, alors $V = N$. Par conséquent, $W = M$.

- Si $M = M_0(L \mid P)$, $L \rhd_d L'$ et $N = M_0(L' \mid P)$, alors $N \overset{\star}{\to}_d V$ implique $M_0 = (\lambda x.M_1)P_1 \dots P_n$, avec $n \geq 0$. On procède par récurrence sur n. Si $n = 0$, alors $V \in M_1 \lceil (L' \mid P)/x \rceil$. Par le lemme 3.7.5(3), il existe $W \in M_1 \lceil (L \mid P)/x \rceil$ t.q. $W \rhd_d V$. Donc $M = (\lambda x.M_1)(L \mid P) \to_d W \rhd_d V$. Dans le cas où $n \geq 0$, on a, pour toute variable z neuve,

  $$V \in M_1 \lceil P_1/x \rceil P_2 \dots P_n(L' \mid P) = M_1[z/x]P_2 \dots P_n(L' \mid P)\lceil P_1/z \rceil$$

  De plus, puisque $M_1[z/x]P_2 \dots P_n(L \mid P) \rhd_d M_1[z/x]P_2 \dots P_n(L' \mid P)$, par le lemme 3.7.5(2) on a :

  $$\exists W \in M_1[z/x]P_2 \dots P_n(L \mid P)\lceil P_1/z \rceil \;\; W \rhd_d V$$

  On conclut $W \in M_1 \lceil P_1/x \rceil P_2 \dots P_n(L \mid P)$ et $M \to_d W \rhd_d V$.

□

Lemme 3.7.7 ***(Stratégie normalisante de $\rhd_d$)***
L'évaluation $\to_d$ est une stratégie normalisante pour $\rhd_d$. Plus précisément :

$$\forall M \in \Lambda_d \;\; M \overset{\star}{\rhd}_d N \overset{\star}{\to}_d V \;\&\; V \in \mathbb{V}_d \;\Rightarrow\; \exists W \in \mathbb{V}_d \;\; M \overset{\star}{\to}_d W \overset{\star}{\rhd}_d V$$

Preuve. Par récurrence sur (l, l'), selon l'ordre lexicographique, où l est la longueur de la réduction $M \overset{\star}{\rhd}_d N$ et l' la longueur de $N \overset{\star}{\to}_d V$.

- Si $l = 0$, alors $W = M$.

- Si $l > 0$ et $l' = 0$, supposons $M \overset{\star}{\rhd}_d M' \rhd_d V = \lambda y.N'$. Par hypothèse de récurrence, il existe $W \in \mathbb{V}_d$ qui vérifie $M \overset{\star}{\to}_r W \overset{\star}{\rhd}_d V$.

- Si $l > 0$ et $l' > 0$, alors

  $$M \overset{\star}{\rhd}_d M' \rhd_d N \to_r N' \overset{\star}{\to}_r V$$

  Par le lemme 3.7.6, il existe H tel que $M' \to_r H \rhd_d N'$, ce qui implique $H \rhd_d N' \overset{\star}{\to}_r V$. Par hypothèse de récurrence (sur la longueur de l'évaluation),

  $$\exists W' \in \mathbb{V}_d \;\; H \overset{\star}{\to}_r W' \overset{\star}{\rhd}_d V$$

  Alors $M \overset{\star}{\rhd}_d H \overset{\star}{\to}_r W'$, d'où, par hypothèse de récurrence (sur la longueur de la réduction), $\exists W \in \mathbb{V}_d \;\; M \overset{\star}{\to}_r W \overset{\star}{\rhd}_d W' \overset{\star}{\rhd}_d V$.

□

Proposition 3.7.8 *1.* $M \overset{\star}{\rhd}_d N \;\Rightarrow\; N \sqsubseteq_r M$

2. $\forall M\ \forall$ *contexte applicatif* $A\ \ (A[M] \Downarrow_r^m\ \Rightarrow\ \exists L \in \{[M]\}\ \exists l \leq m\ A[L] \Downarrow_r^l)$

3. $\forall M \in \Lambda_r\ \forall L\ \ (L \in \{[M]\}\ \Rightarrow\ L \sqsubseteq_r M)$

Preuve.

1. Observons d'abord que la réduction est décroissante par rapport au préordre $\sqsubseteq_d$, i.e.
$$M \overset{\star}{\rhd}_d N\ \Rightarrow\ N \sqsubseteq_d M$$
Soit A un λ_d-contexte applicatif tel que $A[N] \Downarrow_d V$. On a $A[M] \rhd_d A[N] \overset{\star}{\to}_d V$, car $M \rhd_d N$. Par le lemme 3.7.7, il existe $W \in \mathbb{V}_d$ t.q. $A[M] \Downarrow_d W$.

   Rappelons enfin que nous avons montré $N \sqsubseteq_d M\ \Rightarrow\ N \sqsubseteq_r M$ dans la preuve du lemme des contextes pour λ_d 3.5.7.

2. Soit A t.q. $A[M] \Downarrow_r^m V$. Par le théorème 3.3.8,
$$\forall W \in \{[V]\}\ \exists N \in \{[A[M]]\}\ \ N \overset{\star}{\to}_d W$$
Par construction de $\{[A[M]]\}$, il existe $L \in \{[M]\}$ qui vérifie $N \in \{[A[L]]\}$. Alors, en utilisant le théorème 3.3.16,
$$\exists W' \in \Lambda_r\ \ A[L] \overset{\star}{\to}_r W'\ \ \&\ \ W \in \{[W']\}$$
Comme W est une abstraction, W' est aussi une abstraction, éventuellement suivie des quelques entrées de substitution, ce qui implique $A[L] \Downarrow_r$.

3. La preuve de cette partie de l'énoncé suit le schéma de celle de la partie 2; il suffit d'échanger M et L. Remarquons que le raisonnement est correct parce que, quelque soit $L \in \{[M]\}$, $N \in \{[A[L]]\}$ implique $N \in \{[A[M]]\}$.

□

Définition 3.7.9 *Un contexte* C *est* linéaire *en* $[]_i$ *si cette constante n'apparaît qu'une fois dans* C.

Lemme 3.7.10 ***(Lemme d'approximation)***
Pour tout λ_r*-contexte* C *et tout* $M \in \Lambda_r$ *t.q.* $C[M] \in \Lambda_r^o$*, on vérifie*

$$C[M] \Downarrow_r\ \Leftrightarrow\ \exists A_1, \ldots, A_n \in \downarrow\!\mathcal{A}(M)\ C[A_1 \oplus \ldots \oplus A_n] \Downarrow_r$$

De plus, si C *est linéaire en* $[]$*, alors on peut prendre* $n = 1$*.*

Preuve. Tout au long de la preuve, on notera $\mathbb{A}$ les termes de la forme $A_1 \oplus \ldots \oplus A_n$; la notation $\mathbb{A} \in \downarrow\mathcal{A}(M)$ indique $A_i \in \downarrow\mathcal{A}(M)$ pour tout i.

On montre la partie $\Leftarrow$: soit $C[A_1 \oplus \ldots \oplus A_n]\Downarrow_r$, avec $A_i \in \downarrow\mathcal{A}(M)$. Rappelons que pour chaque A_i il existe $B_i \in \mathcal{A}(M)$ tel que $A_i \leq B_i$. Par ailleurs, pour tout i, il existe M_i, N_i t.q. $B_i \in \overline{\varpi}(N_i)$, $M_i \in \{[M]\}$ et $M_i \overset{\star}{\rhd}_d N_i$. Par le lemme 3.7.4(1) et (2), on a $A_i \sqsubseteq_r B_i \sqsubseteq_r N_i$ quelque soit $i = 1, \ldots, n$. Par la proposition 3.7.8(1), $N_i \sqsubseteq_r M_i$ est vérifié; par la proposition 3.7.8(3) on a $M_i \sqsubseteq_r M$. Par conséquent, $A_i \sqsubseteq_r M$ pour tout $i = 1, \ldots, n$. Puisque $\sqsubseteq_r$ est une precongruence, $C[M \oplus \ldots \oplus M]\Downarrow_r$. Donc $C[M]\Downarrow_r$, car $M \simeq_r M \oplus \ldots \oplus M$.

La partie $\Rightarrow$ est corollaire de l'énoncé plus général suivant : soit C un λ_r-contexte à plusieurs trous et $M_1, \ldots, M_p \in \Lambda_r$ tels que $C[M_1, \ldots, M_p] \in \Lambda_r^o$. Alors,

$$C[M_1, \ldots, M_p]\Downarrow_r \ \Rightarrow \ \forall i \in [1,p] \ \exists \mathbb{A}_i \in \downarrow\mathcal{A}(M) \ \ C[\mathbb{A}_1, \ldots, \mathbb{A}_p]\Downarrow_r$$

De plus, pour tout i tel que C est linéaire en $[]_i$, on peut prendre $\mathbb{A}_i = A_i \in \downarrow\mathcal{A}(M_i)$. La notation $C[\tilde{M}]$ est une abréviation de $C[M_1, \ldots, M_p]$. Avec an abus de terminologie, on note $\tilde{\mathbb{A}} \in \downarrow\mathcal{A}(\tilde{M})$ lorsque $\mathbb{A}_i \in \downarrow\mathcal{A}(M_i)$ pour chaque i. La forme générale d'un λ_r-contexte C est $C_0 C_1 \ldots C_n$ où C_0 est soit une constante $[]_i$, soit une variable x, soit un contexte $\lambda x.D$, et C_j, $j > 0$, est ou bien un paquet de contextes $(D_1^{m_1} \mid \ldots \mid D_q^{m_q})$ ou bien une substitution explicite faite de contextes $\langle D_1^{m_1} \mid \ldots \mid D_q^{m_q}/x\rangle$.

Soit $C[M_1, \ldots, M_p]\Downarrow_r^l$. Supposons, sans perte de généralité, que C est linéaire dans $[]_i$ pour $i = k, \ldots, p$ avec $0 \leq k \leq p+1$ (le cas $k = p+1$ correspond à C non-linéaire en aucune constante $[]_i$.) On procède par récurrence sur l.

$l = 0$. Ici, $C[\tilde{M}]$ se réduit vers une abstraction par application de la règle (v); c'est-à-dire, $C_1, \ldots, C_n$ sont des contextes de substitution. On a, soit $C_0 = \lambda x.D$, soit $C_0 = []_i$. Dans le premier cas, $C[\tilde{\bot}]\Downarrow_r$ (i.e. on peut prendre l'élément minimal de $\downarrow\mathcal{A}(M_i)$ pour tout i). Dans le deuxième cas, M_1 doit être une abstraction $\lambda x.M_1'$; alors, on prend $A_1 = \lambda x.\bot \in \mathcal{A}(M_1)$ et $A_i = \bot$ pour tout $i \neq 1$. On conclut $A_1 C_1 \ldots C_n[\tilde{\bot}] \overset{\star}{\rightarrow}_r \lambda x.(\bot C_1 \ldots C_n[\tilde{\bot}])\Downarrow_r$.

$l > 0$. On montre l'énoncé par cas sur C_0.

1. $C_0 = \lambda x.D$. Soit C_i le premier contexte de paquets dans $C_1 \ldots C_n$. L'évaluation convergente de $C[\tilde{M}]$ commence par :

   $$C[\tilde{M}] \overset{\star}{\rightarrow}_r (D[\tilde{M}][z/x] C_1 \ldots C_{i-1} \langle C_i/z\rangle C_{i+1} \ldots C_n)[\tilde{M}]\Downarrow_r^{l'} \quad \text{avec } l' < l$$

   où z une variable neuve. On définit D' comme D avec les constantes $[]_i$, pour $i = 1, \ldots, k-1$, remplacées par $[]_{p+i}$, et

   $$C' = D'[z/x] C_1 \ldots C_{i-1} \langle C_i/z\rangle C_{i+1} \ldots C_n$$

Remarquons que le contexte C est non-linéaire en les constantes $[]_i$ remplacées. Quelque soit la suite $\tilde{N}$ de longueur p qui vérifie $fv(\tilde{N}) \subseteq fv(\tilde{M})$, on a

$$(*) \quad C[\tilde{N}] \overset{\star}{\to}_r C'[\tilde{N}, \tilde{N}'[z/x]] \Downarrow_r^{l-1}$$

où $\tilde{N}' = N_1, \ldots, N_{k-1}$. Par hypothèse de récurrence, $C'[\tilde{\mathbb{A}}, \tilde{\mathbb{B}}] \Downarrow_r$ où $\tilde{\mathbb{A}} \in \downarrow\mathcal{A}(\tilde{M})$ et $\tilde{\mathbb{B}} \in \downarrow\mathcal{A}(\tilde{M}'[z/x])$.

Il est facile de voir qu'il existe $\tilde{\mathbb{A}}' \in \downarrow\mathcal{A}(\tilde{M}')$ t.q. $\tilde{\mathbb{B}} = \tilde{\mathbb{A}}'[z/x]$. On définit $\tilde{\mathbb{A}}''$ comme suit :

$$\mathbb{A}_i'' = \begin{cases} \mathbb{A}_i \oplus \mathbb{A}_i' & \text{si } i = 1, \ldots k-1 \\ \mathbb{A}_i & \text{si } i = k, \ldots, p \end{cases}$$

Alors, par $(*)$, on a

$$C[\tilde{\mathbb{A}}''] \overset{\star}{\to}_r C'[\tilde{\mathbb{A}}'', \mathbb{A}_1''[z/x], \ldots, \mathbb{A}_{k-1}''[z/x]]$$

Par le lemme 3.5.13 d'élargissement des paquets,

$$C'[\tilde{\mathbb{A}}, \tilde{\mathbb{B}}] \sqsubseteq_r C'[\tilde{\mathbb{A}}'', \mathbb{A}_1''[z/x], \ldots, \mathbb{A}_{k-1}''[z/x]]$$

Donc $C'[\tilde{\mathbb{A}}'', \mathbb{A}_1''[z/x], \ldots, \mathbb{A}_{k-1}''[z/x]]$, ce qui implique $C[\tilde{\mathbb{A}}''] \Downarrow_r$.

2. $C_0 = []_i$. Il y a trois cas à analyser, qui dépendent de la forme de M_i.

   (a) $M_i = \lambda x.M'$. Soit C_j le premier paquet de la suite $C_1, \ldots, C_n$ et z une variable neuve. L'évaluation de $C[\tilde{M}]$ est alors

   $$C[\tilde{M}] \overset{\star}{\to}_r M'[z/x] C_1 \ldots C_{j-1} \langle C_j / z \rangle C_{j+1} \ldots C_n[\tilde{M}] \Downarrow_r^{l'} \quad \text{où } l' < l$$

   On examine deux cas : si C n'est pas linéaire en $[]_i$ (i.e. $i < k$), alors on définit

   $$C' = []_{p+1} C_1 \ldots C_{j-1} \langle C_j / z \rangle C_{j+1} \ldots C_n$$

   qui est donc linéaire en $[]_k, \ldots, []_p, []p+1$. Pour tout $\tilde{N}$, de longueur p, qui vérifie $fv(\tilde{N}) \subseteq fv(\tilde{M})$ et $N_i = \lambda x.N'$, on a

   $$(**) \quad C[\tilde{N}] \overset{\star}{\to}_r C'[\tilde{N}, N'[z/x]] \Downarrow_r^{l'}$$

   Par l'hypothèse de récurrence, $C'[\tilde{\mathbb{A}}, B] \Downarrow_r$ où $\tilde{\mathbb{A}} \in \downarrow\mathcal{A}(\tilde{M})$ et $B \in \downarrow\mathcal{A}(M'[z/x])$. Il existe $B' \in \downarrow\mathcal{A}(M')$ qui vérifie $B = B'[z/x]$. Donc, $\lambda x.B' \in \downarrow\mathcal{A}(M)$.
   On définit $\tilde{\mathbb{A}}'$ tel que

   $$\mathbb{A}_j' = \begin{cases} \mathbb{A}_j & \text{si } j \neq i \\ \mathbb{A}_i \oplus \lambda x.B' & \text{si } j = i \end{cases}$$

   Alors,

   $$C[\tilde{\mathbb{A}}'] \overset{\star}{\to}_r C'[\tilde{\mathbb{A}}', B'[z/x]]$$

   De plus, $C'[\tilde{\mathbb{A}}, B] \sqsubseteq_r C'[\tilde{\mathbb{A}}', B]$ car on peut toujours élargir les paquets et préserver la convergence[4]. On en déduit $C[\tilde{\mathbb{A}}'] \Downarrow_r$ car $C'[\tilde{\mathbb{A}}, B] \Downarrow_r$. On

---

[4] Le caractère “may testing” de la sémantique est ici essentiel.

observera que le fait que C ne soit pas linéaire en $[]_i$ oblige à construire un approximant $\mathbb{A}'_i$ pour M_i à l'aide du choix non-déterministe.

Dans le cas où C est linéaire en $[]_i$, on définit

$$C' = []_i C_1 \dots C_{j-1} \langle C_j / z \rangle C_{j+1} \dots C_n$$

On a alors $C[\tilde{N}] \overset{\star}{\rightarrow}_r C'[N_1, \dots, N'[z/x], \dots, N_p] \Downarrow_r^{l'}$. Par hypothèse de récurrence, $C'[\mathbb{A}_1, \dots, B[z/x], \dots, \mathbb{A}_p] \Downarrow_r$. On conclut

$$C[\mathbb{A}_1, \dots, \lambda x.B, \dots, \mathbb{A}_p] \Downarrow_r$$

(b) $M_i = x R_1 \dots R_q$ et $\tilde{R}$ ne lie pas l'occurrence de tête de x. Puisque certains des R_i sont possiblement des entrées de substitution, on ne peut pas, comme dans le cas du lambda calcul pur, construire les approximants M_i par un raisonnement sur chacun des R_i.
Soit $\dot{C} = []_{p+1} C_1 \dots C_n$. Il est clair que $\dot{C}[\tilde{M}, M_i] = C[\tilde{M}]$. Par la proposition 3.7.8,

$$\exists M'_i \in \{[M_i]\} \;\; \dot{C}[\tilde{M}, M'_i] \Downarrow_r^{l'} \quad \text{où } l' \leq l$$

La forme générale de M'_i est $x P_1 \dots P_t$ où $P_{j+1} = (N_{s_j+1} \mid \dots \mid N_{s_{j+1}})$ avec $1 = s_0 < s_1 \dots < s_t$. On note $\tilde{N}$ la séquence $N_{s_0}, \dots N_{s_1}, \dots N_{s_t}$. Notre point de départ est donc l'évaluation convergente de $\dot{C}[\tilde{M}, M'_i]$. Soit C_j le premier contexte de substitution de la forme $\langle (D \mid D')/x \rangle$ dans $C_1 \dots C_n$, et soit z une variable neuve. On se restreint au cas où il n'y a pas d'α-conversion à faire. L'évaluation convergente est donc

$$\dot{C}[\tilde{M}, M'_i] \overset{\star}{\rightarrow}_r D \tilde{P} C_1 \dots C_{j-1} \langle D'/z \rangle C_{j+1} \dots C_n[\tilde{M}] \Downarrow_r^{l''} \quad \text{où } l'' < l$$

On définit $C' = D D_1 \dots D_t C_1 \dots C_{j-1} \langle D'/z \rangle C_{j+1} \dots C_n$ où

$$D_{j+1} = ([]_{s_j+p+1} \mid \dots \mid []_{s_{j+1}+p}) \qquad j = 0, \dots, t-1$$

En résumant, $\dot{C}[\tilde{M}, M'_i] \overset{\star}{\rightarrow}_r C'[\tilde{M}, \tilde{N}] \Downarrow_r^{l''}$. Remarquons que le contexte C' est linéaire en $[]_j$, $j = k+1, \dots, s_t + p$. Par hypothèse de récurrence, $C'[\tilde{\mathbb{A}}, B_1, \dots, B_{s_t}] \Downarrow_r$, avec $\tilde{\mathbb{A}} \in \downarrow\mathcal{A}(\tilde{M}')$ et $B_j \in \downarrow\mathcal{A}(N_i)$. Soit

$$\vec{B}^{j+1} = (B_{s_j+1} \mid \dots \mid B_{s_{j+1}})$$

Le terme $x \vec{B}^1 \dots \vec{B}^t$ est un approximant de M'_i; par la définition d'approximant, $x \vec{B}^1 \dots \vec{B}^t \in \downarrow\mathcal{A}(M_i)$ puisque $M'_i \in \{[M_i]\}$. Donc

$$\dot{C}[\tilde{\mathbb{A}}, x \vec{B}^1 \dots \vec{B}^t] \overset{\star}{\rightarrow}_r C'[\tilde{\mathbb{A}}, B_1, \dots, B_{s_t}]$$

ce qui implique $\dot{C}[\tilde{\mathbb{A}}, x\vec{B}^1 \ldots \vec{B}^t]\Downarrow_r$. On définit $\tilde{\mathbb{A}}'$ par

$$\mathbb{A}'_j = \begin{cases} \mathbb{A}_j & \text{si } j \neq i \\ (\mathbb{A}_i \oplus x\vec{B}^1 \ldots \vec{B}^t) & \text{si } j = i \end{cases}$$

On a $\tilde{\mathbb{A}}' \in \downarrow\mathcal{A}(\tilde{M})$ et $C[\tilde{\mathbb{A}}']\Downarrow_r$ car

$$\dot{C}[\tilde{\mathbb{A}}, x\vec{B}^1 \ldots \vec{B}^t]\sqsubseteq_r C'[\mathbb{A}_1, \ldots, (\mathbb{A}_i \oplus x\vec{B}^1 \ldots \vec{B}^t), \ldots, \mathbb{A}_p, x\vec{B}^1 \ldots \vec{B}^t]\sqsubseteq_r C[\tilde{\mathbb{A}}']$$

(c) Si M_i n'est pas en forme normale, la première étape de l'évaluation convergente de $C[M]$ correspond à une évaluation de M_i, soit $M_i \overset{\star}{\rightarrow}_r N$, avec N en forme normale (comme dans les cas (a) et (b) précédents). On a

$$C[M] = M_i C_1 \ldots C_n \overset{\star}{\rightarrow}_r N C_1 \ldots C_n \Downarrow_r^{l'} \quad \text{où } l' < l$$

Par conséquent, si $C' = [\,]_{p+1} C_1 \ldots C_n$, par l'hypothèse de récurrence il existe $\tilde{\mathbb{A}}' \in \downarrow\mathcal{A}(\tilde{M})$ et $\mathbb{B} \in \downarrow\mathcal{A}(N)$ tel que $C'[\tilde{A}', B]\Downarrow_r$. Donc

$$C[\mathbb{A}'_1, \ldots, (\mathbb{A}'_i \oplus \mathbb{B}), \ldots, \mathbb{A}'_p]\Downarrow_r$$

3. $C_0 = x$. L'énoncé est vérifié comme cas particulier de (b).

□

Un corollaire du lemme d'approximation 3.7.10 est que l'interprétation de λ_r sur le domaine algébrique des cônes non-vides de $\mathcal{L}_r$ est adéquate par rapport à la sémantique observationnelle standard $\sqsubseteq_r$. C'est-à-dire, $\leq_{\mathcal{L}_r}$ est une precongruence (cf. th. 5.8 [47] dont la preuve repose sur la complétude des réductions "inside-out" de Welch, démontré à l'aide d'un calcul étiqueté.)

Théorème 3.7.11 ***Adéquation de la sémantique algébrique***
$\forall M, N \in \Lambda_r \;\; M \leq_{\mathcal{L}_r} N \;\Rightarrow\; M \sqsubseteq_r N$

Preuve. Soit $M, N \in \Lambda_r$ et C un λ_r-contexte qui ferme M et N. Supposons $M \leq_{\mathcal{L}_r} N$ et $C[M]\Downarrow_r$. Par le lemme d'approximation 3.7.10, il existe $A_1, \ldots, A_n$ dans $\downarrow\mathcal{A}(M)$ tel que $C[(A_1 \oplus \ldots \oplus A_n)]\Downarrow_r$. Par définition du préordre $\leq_{\mathcal{L}_r}$, $A_i \in \downarrow\mathcal{A}(N)$ pour tout $i = 1, \ldots, n$. On conclut $C[N]\Downarrow_r$ par le lemme d'approximation.
□

Le préordre $\leq_{\mathcal{L}_r}$ est trop restrictif pour être complètement adéquat par rapport à $\sqsubseteq_r$, c'est-à-dire

$$M \sqsubseteq_r N \;\not\Rightarrow\; M \leq_{\mathcal{L}_r} N$$

Trois types d'exemples montrent l'incomplétude :

- le premier fait intervenir la η-expasion (signalé par Lévy [47], voir aussi [18]) : on a

$$x \sqsubseteq_r \lambda y.xy^{\infty}$$

mais $x \in \mathcal{A}(x)$ tandis que $x \notin \mathcal{A}(\lambda y.xy^\infty)$, donc

$$x \not\leq_{\mathcal{L}_r} \lambda y.xy^\infty$$

- le deuxième problème posé par la définition de $\leq_{\mathcal{L}_r}$ est que le domaine n'a pas d'élément maximal : soit $\Xi = (\lambda fx.ff^\infty)(\lambda fx.ff^\infty)^\infty$; alors

  $$I \sqsubseteq_r \Xi$$

  mais $\lambda x.x \notin \mathcal{A}(\Xi)$, donc

  $$I \not\leq_{\mathcal{L}_r} \Xi$$

- un troisième type repose sur le fait que $\leq_{\mathcal{L}_r}$ est sensible au nombre de ressources d'un paquet, ce qui n'est pas toujours le cas pour $\sqsubseteq_r$, comme nous l'avons démontré dans la section 3.6 : on a

  $$x(\lambda y.\Omega \mid \lambda y.\Omega) \sqsubseteq_r x(\lambda y.\Omega)$$

  par la propriété de simplification, mais

  $$x(\lambda y.\Omega \mid \lambda y.\Omega) \not\leq_{\mathcal{L}_r} x(\lambda y.\Omega)$$

3.8 Comparaison des calculs

3.8.1 Pouvoir discriminant

Afin de donner un aperçu plus complet de la sémantique observationnelle du calcul λ_r, nous énonçons ici quelques résultats sur le pouvoir de discrimination des multiplicités et des ressources sur les λ termes et les connections que l'on peut établir avec les extensions standards du λ calcul faible, étudiées dans le chapitre 2, et avec le calcul π. Ces résultats, obtenus par Boudol et Laneve [18, 19, 20], reposent sur la caractérisation intensionnelle des préordres $\sqsubseteq_m$ et $\sqsubseteq^\flat{}_m$ sur λ et font appel à des résultats précédents de Ong [59] sur le λ calcul et de Sangiorgi [72, 73] sur le codage de λ dans π.

Remarquons d'abord que le λ-calcul est un sous-calcul de λ_m. Plus précisément, le sous-calcul de λ_m qui n'a que la multiplicité infinie ∞ dans ses arguments correspond exactement au λ-calcul. La plupart des énoncés concernent les λ-termes purs que nous noterons ici $E, F, \ldots$ sans faire référence explicite au codage. Nous gardons la notation M, N pour des termes dans Λ_m ou Λ_r. Le codage $[\![-]\!] : \Lambda \to \Lambda_m$, étudié dans [17], est le suivant :

$$[\![x]\!] = x \quad [\![\lambda x.E]\!] = \lambda x.[\![E]\!] \quad [\![(EF)]\!] = [\![E]\!][\![F]\!]^\infty$$

Les premiers résultats à signaler sont :

$$E \leq_{\mathcal{L}} F \;\Rightarrow\; E \sqsubseteq_m F \;\Rightarrow\; E \sqsubseteq_\lambda F$$

De plus, ces deux implications sont irréversibles. En ce qui concerne la seconde, le fait que les contextes avec multiplicités discriminent plus que les contextes usuels n'est pas surprenant. L'exemple typique montre que le contexte $C = [\,]\langle \mathbf{I}^1/x\rangle$ distingue les termes $x(\lambda y.xy)$ et xx, inséparables par des λ-contextes (cf. exemple 1.2.1 de l'introduction.) Pour ce qui est de la première implication, la raison de l'irréversibilité est double. D'une part les termes dans O_∞ ne sont pas des éléments maximaux du préordre $\leq_{\mathcal{L}}$. D'autre part, ce préordre ne valide pas la η-expansion; i.e. $\neg(x\leq_{\mathcal{L}}\lambda y.xy)$. Le préordre $\leq^\eta_{\mathcal{L}}$ de Ong [59][5] incorpore ces deux ingrédients :

Définition 3.8.1 *(Définition 5.1 [18])*
Le préordre $E\leq^\eta_{\mathcal{L}}F$ sur les λ-termes est :

$$E\leq^\eta_{\mathcal{L}}F \overset{def}{\Leftrightarrow} \forall k\in\mathbb{N}\ \ E\leq^\eta_k F$$

où $\leq^\eta_k$ est défini par :

1. *$E\leq^\eta_0 F$ pour tout E et F;*
2. *$E\leq^\eta_{k+1}F$ ssi*
   - *(a) $F\in PO_\infty$ ou*
   - *(b) $E\in PO_n$ et $F=_\beta\lambda x_1\ldots x_m.F'$ avec $m\geq n$, ou*
   - *(c) $E=_\beta\lambda x_1\ldots x_n.xE_1\ldots E_s$ et $F=_\beta\lambda x_1\ldots x_n y_1\ldots y_t.xF_1\ldots F_s Y_1\ldots Y_y$ avec $\tilde{E}$ et $\tilde{F}$ t.q. $E_i\leq^\eta_k F_i$ et $y_j\leq^\eta_k Y_j$ et $y_j\notin fv(x\tilde{E})$.*

Le sémantique observationnelle standard $\sqsubseteq_m$ sur les λ-termes est équivalente au préordre intentionnel $\leq^\eta_{\mathcal{L}}$ et strictement plus discriminante que $\sqsubseteq_{\mathrm{P}}$ et $\sqsubseteq_{\mathrm{C}}$:

$$E\sqsubseteq_m F \Leftrightarrow E\leq^\eta_{\mathcal{L}}F \;\Rightarrow\; E\sqsubseteq_{\mathrm{P}}F \;\Rightarrow\; E\sqsubseteq_{\mathrm{C}}F \;\Rightarrow\; E\sqsubseteq_\lambda F$$

L'équivalence entre $\sqsubseteq_m$ et $\leq^\eta_{\mathcal{L}}$ est prouvée dans [18]. La implication qui suit est montrée dans [59]. Une preuve indépendante de $E\sqsubseteq_m F \;\Rightarrow\; E\sqsubseteq_{\mathrm{P}}F$ a été proposée dans des versions préliminaires de [18]. Il est clair que l'implication $E\sqsubseteq_r F \;\Rightarrow\; E\sqsubseteq_m F$ permet de transposer ces résultats au calcul avec ressources. Nous montrons que les ressources ne discriminent pas plus que les multiplicités lorsqu'il s'agit de λ-termes, i.e.

$$E\sqsubseteq_m F \;\Rightarrow\; E\sqsubseteq_r F$$

Cette implication est corollaire de $E\leq^\eta_{\mathcal{L}}F \;\Rightarrow\; E\sqsubseteq_{rc}F$. Notre preuve, sémantique, est reportée au chapitre 5, où l'on introduit quelques outils techniques nécessaires.

---

[5] Ce préordre caractérise la structure locale du modèle de Plotkin-Scott-Engeler du λ-calcul.

Nous développons ensuite les deux exemples 1.2.2 et 1.2.3 de l'introduction, qui illustrent le pouvoir des multiplicités finies, par rapport à **c** et à **p**. Rappelons que les termes M et N des exemples ne peuvent pas être séparés par des λ-contextes. Dans un souci de clarté, on néglige les substitutions explicites sur des variables qui n'apparaissent pas libres (les substitutions $\langle P/x\rangle$ qui ne lient aucune occurrence de la variable x.)

Exemple 1.2.2. Supposons $B = x(\lambda y.\Omega)\Omega$ et

$$M = \lambda x.xB(\lambda y.\Omega) \;\text{ et }\; N = \lambda x.x(\lambda z.Bz)(\lambda y.\Omega)$$

Le contexte $A = []\mathbf{c}$ sépare M et N :

$$A[N]\to\mathbf{c}(\lambda z.\mathbf{c}(\lambda y.\Omega)\Omega z)(\lambda y.\Omega)\to\mathbf{I}(\lambda y.\Omega)\to\lambda y.\Omega \;\text{ converge}$$

$$A[M]\to\mathbf{c}(\mathbf{c}(\lambda y.\Omega)\Omega)(\lambda y.\Omega)\to\mathbf{c}(I\Omega)(\lambda y.\Omega)\to\mathbf{c}\Omega(\lambda y.\Omega) \;\text{ diverge}$$

Le λ_r-contexte $C = []U$, où $U = \lambda vw.v$, sépare aussi M et N :

$$\begin{array}{lll}
C[N] & \to_r & x(\lambda z.Bz)(\lambda y.\Omega)\langle U/x\rangle \to_r \\
 & & (\lambda vw.v)(\lambda z.Bz)(\lambda y.\Omega)\langle \mathbf{1}/x\rangle \overset{\star}{\to}_r \\
 & & v\langle \lambda z.Bz/v\rangle\langle \mathbf{1}/x\rangle \to_r \\
 & & (\lambda z.Bz)\langle \mathbf{1}/x\rangle \to_r \lambda z.(Bz)\langle \mathbf{1}/x\rangle{\Downarrow_r} \\
C[M] & \to_r & xB(\lambda y.\Omega)\langle U/x\rangle \to_r \\
 & & (\lambda vw.v)B(\lambda y.\Omega)\langle \mathbf{1}/x\rangle \overset{\star}{\to}_r \\
 & & v\langle B/v\rangle\langle \mathbf{1}/x\rangle \to_r \\
 & & x(\lambda y.\Omega)\Omega\langle \mathbf{1}/x\rangle \Uparrow_r
\end{array}$$

où le dernier terme diverge car il n'y a pas de ressource disponible pour x (l'argument U n'a que multiplicité 1). Comme nous avons souligné dans l'introduction, les termes M et N ne sont pas exactement des termes du lambda calcul faible; il faudrait ajouter la multiplicité infinie ∞ à chaque sous-terme en position d'argument, ce qui ne change en rien l'exemple.

Exemple 1.2.3. Supposons $B = x\Omega\Omega$ et

$$M = \lambda x.xB(xB\Omega) \;\text{ et }\; N = \lambda x.xB(x(\lambda z.Bz)\Omega)$$

Ces termes ne sont pas séparables dans λ_c [18].

Il est clair que le contexte $A = []\mathbf{p}$ sépare M et N :

$$A[N]\!\to \underbrace{\mathbf{p}(\underbrace{\mathbf{p}\Omega\Omega}_{\text{diverge}})\,\underbrace{\mathbf{p}(\underbrace{\lambda y.(\mathbf{p}\Omega\Omega)y}_{\text{converge}})\Omega}_{\text{converge}}}_{\text{converge}}$$

$$A[M]\!\to \underbrace{\mathbf{p}(\underbrace{\mathbf{p}\Omega\Omega}_{\text{diverge}})(\underbrace{\mathbf{p}(\underbrace{\mathbf{p}\Omega\Omega}_{\text{diverge}})\Omega}_{\text{diverge}})}_{\text{diverge}}$$

Le contexte $C = [](P \mid P)FK$, où $P = \lambda x_1x_2x_3.x_3x_1x_2$, $K = \lambda x_1x_2.x_1$ et $F = \lambda x_1x_2.x_2$, distingue aussi M et N.

$$\begin{array}{lll}
C[N] & \overset{\star}{\to}_r & x_3x_1x_2\langle B/x_1\rangle\langle x(\lambda y.By)\Omega/x_2\rangle\langle P/x\rangle\langle F/x_3\rangle K \to_r \\
 & & Fx_1x_2\langle B/x_1\rangle\langle x(\lambda y.By)\Omega/x_2\rangle\langle P/x\rangle K \overset{\star}{\to}_r \\
 & & x_2\langle x(\lambda y.By)\Omega/x_2\rangle\langle P/x\rangle K \to_r \\
 & & x(\lambda y.By)\Omega\langle P/x\rangle K \to_r P(\lambda y.By)\Omega\langle \mathbf{1}/x\rangle K \overset{\star}{\to}_r \\
 & & Kx_1x_2\langle \lambda y.By/x_1\rangle\langle\Omega/x_2\rangle\langle\mathbf{1}/x\rangle \overset{\star}{\to}_r \\
 & & \lambda y.(By\langle\mathbf{1}/x\rangle)\Downarrow_r \\
C[M] & \overset{\star}{\to}_r & x_3x_1x_2\langle B/x_1\rangle\langle xB\Omega/x_2\rangle\langle P/x\rangle\langle F/x_3\rangle K \to_r \\
 & & Fx_1x_2\langle B/x_1\rangle\langle xB\Omega/x_2\rangle\langle P/x\rangle K \overset{\star}{\to}_r \ldots \\
 & & PB\Omega\langle\mathbf{1}/x\rangle K \overset{\star}{\to}_r \\
 & & Kx_1x_2\langle B/x_1\rangle\langle\Omega/x_2\rangle\langle\mathbf{1}/x\rangle \overset{\star}{\to}_r B\langle\mathbf{1}/x\rangle = x(\lambda y.\Omega)\Omega\langle\mathbf{1}/x\rangle \Uparrow_r
\end{array}$$

Puisque $E{=}_{\mathcal{L}}F \Leftrightarrow E{\leq}^{\eta}_{\mathcal{L}}F$ & $F{\leq}^{\eta}_{\mathcal{L}}E$, le résultat de Sangiorgi [72] sur le codage du λ-calcul dans π permet de conclure

$$E \simeq_{\uplus} F \Leftrightarrow E{=}_{\mathcal{L}}F \Leftrightarrow E \simeq_m F \Leftrightarrow E \sim_\pi F$$

où $\sim_\pi$ est l'équivalence par bisimulation. Par ailleurs, le préordre $\leq_{\mathcal{L}}$ est prouvé [19] équivalent à la sémantique ${\sqsubseteq^\flat}_m$ non-standard où divergence et blocage sont distingués :

$$E{\leq_{\mathcal{L}}}F \Leftrightarrow E{\sqsubseteq^\flat}_mF \Leftrightarrow E{\sqsubseteq^\flat}_rF$$

Au contraire de $\sqsubseteq_m$, le préordre ${\sqsubseteq^\flat}_m$ ne valide pas la η-expansion et les termes de degré de fonctionnalité infini ne sont pas des éléments maximaux. D'une part le contexte $[]\langle I^0/x\rangle$ sépare les termes x et $\lambda y.xy$ (comme le calcul $\lambda_{\uplus}$, cf. section 2.4). En effet,

$$\delta \in \mathrm{obs}(x\langle I^0/x\rangle) \quad \text{mais} \quad \delta \notin \mathrm{obs}((\lambda y.xy)\langle I^0/x\rangle)$$

D'autre part, $\mathbf{I} \;{\not\sqsubseteq^\flat}_m \Xi = (\lambda fx.ff^\infty)(\lambda fx.ff^\infty)^\infty$ car

$$\mathbf{I}\delta \overset{\star}{\to}_r \delta \quad \text{mais} \quad \Xi\delta \overset{\star}{\to}_r (\lambda x.\Xi)\delta \overset{\star}{\to}_r \Xi \Uparrow_r$$

Finalement, en utilisant des résultats de [73] et un codage adéquat de λ_m dans π, Boudol et Laneve [20] montrent

$$E \sqsubseteq^{\flat}{}_m F \Leftrightarrow E \sqsubseteq_\pi F$$

où $\sqsubseteq_\pi$ est la sémantique contextuelle du π-calcul, en répondant ainsi à la question soulevée par Milner dans le papier fondateur [54]. Le codage de λ_m dans π utilisé a été conçu pour permettre au π-calcul de déterminer si un terme est convergent, ou s'il est bloqué.

3.8.2 Expressivité

Les énoncés de la section 3.8.1 illustrent le pouvoir de séparation des multiplicités (donc ressources) face aux λ-termes. Ils sont d'autant plus remarquables que λ_m ne contient ni le test de convergence ni les fonctions parallèles. Plus précisément, les combinateurs **c** et **p** ne sont définissables ni dans le calcul avec multiplicités ni dans le calcul avec ressources[6]. En ce qui concerne le test de convergence, ce résultat est conséquence immédiate du contre-exemple 3.6.13[7] et du résultat de Felleisen [33]. Une preuve syntaxique directe pour le calcul λ_m a été donnée dans [18]. Pour λ_r, voici une preuve syntaxique :

Lemme 3.8.2

$$\nexists T \in \Lambda_r^o \; \forall M \in \Lambda_r \; \begin{cases} TM{\Downarrow_r}\boldsymbol{I} & si\ M{\Downarrow_r} \\ TM{\Uparrow_r} & sinon \end{cases}$$

Preuve. Supposons qu'un tel T existe. Alors $T(\lambda x.\Omega){\Downarrow_r}\mathbf{I}$ et $T\Omega{\Uparrow_r}$. Or on prouve facilement

$$\forall T \in \Lambda_r^o \;\; T(\lambda x.\Omega){\Downarrow_r}\mathbf{I} \;\Rightarrow\; T\Omega{\Downarrow_r}$$

On montre d'abord $T\langle\lambda x.\Omega/z\rangle\xrightarrow{\star}_r\mathbf{I} \;\Rightarrow\; T\langle\Omega/z\rangle{\Downarrow_r}$. Par récurrence sur la longueur de l'évaluation de $T\langle\lambda x.\Omega/z\rangle$, à (v)-réductions près : le cas de base est trivial; lorsque $T\langle\lambda x.\Omega/z\rangle\rightarrow_r T'\langle\lambda x.\Omega/z\rangle\xrightarrow{\star}_r\mathbf{I}$, l'énoncé est vérifié par hypothèse de récurrence; si $T = z\tilde{R}$ alors

$$T\langle\lambda/x\rangle.\Omega z\rightarrow_r(\lambda x.\Omega)\tilde{R}\langle\mathbf{1}/z\rangle\xrightarrow{\star}_r\mathbf{I}$$

Il est clair que cette évaluation n'est pas réalisable même si $|\tilde{R}| = 0$. La conclusion est donc vérifiée.

Revenons à notre preuve : soit $T(\lambda x.\Omega){\Downarrow_r}\mathbf{I}$, alors il existe M t.q. $T\xrightarrow{\star}_r(\lambda z.M)$ et $M\langle\lambda x.\Omega/z\rangle{\Downarrow_r}\mathbf{I}$. On a $T\langle\Omega/z\rangle{\Downarrow_r}$ puisque $M\langle\Omega/z\rangle{\Downarrow_r}$. □

---

[6] De plus, l'exemple 1.2.2 semble indiquer que le test de convergence **c** n'est pas définissable dans λ_r, même si on ne cherche à simuler son effet que sur les λ-termes : la multiplicité de l'argument $U = \lambda vw.v$ dépend du nombre d'occurrences libres de la variables x.

[7] Il faut souligner que l'origine du contre-exemple 3.6.13 n'est pas la question de la définissabilité de **c** mais le problème de l'adéquation complète du modèle de cônes, traité dans les chapitres 4 et 5.

Comme corollaire, le combinateur de test de convergence parallèle **p** n'est pas définissable dans λ_r (donc dans λ_m). Cependant, le choix non-déterministe est définissable dans λ_r (cf. section 3.1) : $M \oplus N = x\langle M \mid N/x\rangle$. D'après le lemme 3.5.13, ce terme est le "join" de M et N. En effet, $M \oplus N \sqsubseteq_r L$ ssi $M \sqsubseteq_r L$ et $N \sqsubseteq_r L$.

Le calcul λ_m, déterministe, ne contient pas les fonctions parallèles. On montre ici que ce fait est indépendant du mécanisme d'évaluation utilisé. Même en considérant des stratégies d'évaluation plus libérales (non-déterministes), on ne peut simuler le parallélisme dans le langage des multiplicités.

On note λ_{md} le calcul par distribution où l'on restreint Λ_d à des termes avec multiplicités (Λ_{md}).

Définition 3.8.3 *Soit* $M, N \in \Lambda^0_{md}$*. Le terme* $M \oplus N$ *est définissable dans* λ_{md} *ssi* $\exists O \in \Lambda^0_{md}\ O \overset{\star}{\rightarrow}_d M$ & $O \overset{\star}{\rightarrow}_d N$*. Le choix non-déterministe* $\oplus$ *est définissable dans* λ_{md} *ssi*

$$\exists D \in \Lambda^0_{md}\ \forall M, N \in \Lambda_{md}\ \ DMN \overset{\star}{\rightarrow}_d M \ \&\ DMN \overset{\star}{\rightarrow}_d N$$

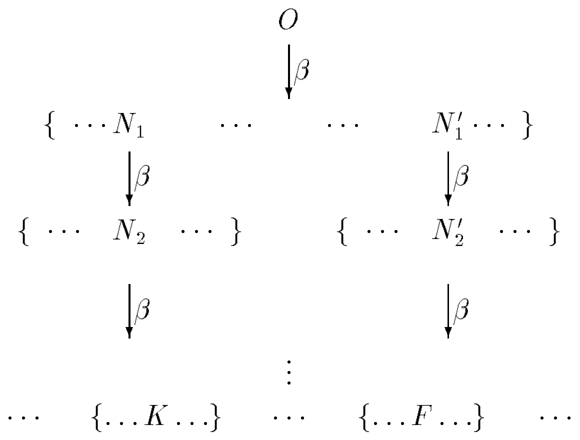
Lemme 3.8.4 *L'opérateur non-déterministe* $\oplus$ *est définissable dans* λ_{md} *si et seulement si* $K \oplus F$ *est définissable.*

Preuve. L'implication $\Rightarrow$ est évidente. Rappelons que $K = \lambda xy.x$ et $F = \lambda xy.y$. Supposons $K \oplus F$ définissable, i.e. $\exists O \in \Lambda^0_{md}\ O \overset{\star}{\rightarrow}_d K$ & $O \overset{\star}{\rightarrow}_d F$. Donc

$$\forall M, N \in \Lambda^0_{md}\ OMN \overset{\star}{\rightarrow}_d M \ \&\ OMN \overset{\star}{\rightarrow}_d N$$

C'est-à-dire, $\oplus$ est définissable dans λ_{md}. □

Prouver la non-définissabilité de l'opérateur $\oplus$ revient donc à prouver la non-définissabilité de $K \oplus F$. Autrement dit, qu'il n'existe pas de terme O dans Λ^0_{md} dont l'arbre de dérivation est de la forme suivante :



Lemme 3.8.5 *Soit* $L_0, L_1 \in \Lambda_m$, $M_0, M_1 \in \Lambda_d$. *Si* $L_0 {=_\alpha} L_1$ & $M_i \in \{[L_i]\}$, $i = 0, 1$, *alors*

$$M_0\, O^d_n/PO^d_n\, E \;\Rightarrow\; (M_1\, O^d_q/PO^d_q\, F \;\Rightarrow\; q = n \;\&\; E \smile F)$$

Preuve. Par récurrence sur $n \geq 0$.

- Si $n = 0$, par hypothèse $M_0 \overset{\star}{\rightarrow}_d E = y\tilde{N}$. Donc, $L_0 \overset{\star}{\rightarrow}_m y\tilde{R}$ où $|\,\tilde{N}\,| = |\,\tilde{R}\,|$ par application du lemme 3.3.13. Supposons $q > 0$. C'est-à-dire, $F = \lambda z.F'$ et $M_1 \overset{\star}{\rightarrow}_d \lambda z.M'$ où $M'\, \mathrm{O}^d_{q-1}/\mathrm{PO}^d_{q-1}\, F'$. Par le lemme 3.3.14, il y a une évaluation $\rightarrow_m$ du terme M_1 qui termine sur une abstraction, soit $\lambda v.H$. Comme M_1 et M_0 sont α-convertibles, $M_0 \overset{\star}{\rightarrow}_m \lambda v.H$. La contradiction provient du fait que $\rightarrow_m$ est déterministe, i.e. il faudrait vérifier $y\tilde{P} {=_\alpha} \lambda v.H$, ce qui est impossible. Donc $q = 0$ et $F = y'\tilde{N}'$. On arrive à $L_1 \overset{\star}{\rightarrow}_m y'\tilde{R}'$ où $|\,\tilde{N}'\,| = |\,\tilde{R}'\,|$ par le lemme 3.3.13. Donc, $L_0 {=_\alpha} L_1$ implique $y\tilde{R} {=_\alpha} y'\tilde{R}'$, d'où $y = y'$ et $|\,\tilde{R}\,| = |\,\tilde{R}'\,|$. C'est-à-dire, $E \smile F$.
- Dans le cas $n = k + 1$, le raisonnement est complètement similaire; on utilise le lemme 3.3.14, le déterminisme de $\rightarrow_m$ plus l'hypothèse de récurrence.

□

Corollaire 3.8.6 *$K \oplus F$ n'est pas définissable dans λ_{md}.*

Preuve. Soit $H \in {\Lambda_m}^o$ t.q. $H \overset{\star}{\rightarrow}_d K$ et $H \overset{\star}{\rightarrow}_d F$. Alors $H\mathrm{O}^r_2/\mathrm{PO}^r_2 K$ et $H\mathrm{O}^r_2/\mathrm{PO}^r_2 F$. Par le lemme 3.8.5, K et F ont la même variable de tête. Ce qui est contredit leur définition. □

Le problème de savoir si le choix non-déterministe sur Λ_m permet de séparer plus de termes que λ_m reste ouvert. Plus généralement, on voudrait déterminer la relation entre λ_m et λ_r (ou les versions respectives avec test de convergence), sur Λ_m.

Chapitre 4

Deux équations aux domaines

Nous nous proposons de déterminer le type d'équation aux domaines à la Scott [74, 75] permettant de définir des modèles à environnements du calcul de ressources, dans le cadre de la sémantique observationnelle standard. Pour commencer, rappelons que la théorie faible du λ-calcul est caractérisée par l'équation aux domaines $D = (D \to D)_\perp$, où l'élévation permet de distinguer les éléments à contenu fonctionnel significatif - abstractions dans le langage des termes, possiblement non-résolubles - des éléments divergents dont l'interprétation est l'élément $\perp$ ajouté. Cette distinction garde tout son sens dans le calcul de ressources; les termes bloqués sont assimilés aux termes divergents. Cependant, D ne remplit pas le rôle de domaine des arguments parce qu'il est essentiel de rendre compte de la multiplicité des composantes des paquets. En gros, il faut pouvoir distinguer la dénotation de M (comme argument) de celle de $(M \mid M)$. L'interprétation de la composition parallèle par le "join" dans le domaine, comme dans [17, 29] par exemple, détruirait cette différence, car le "join" est idempotent. La construction de l'équation aux domaines pour le calcul λ_r requiert donc

1. Un domaine D où interpréter les termes clos.
2. Un domaine $\mathcal{M}(D)$ où interpréter les arguments, des multi-ensembles de termes.

À condition que l'on sache construire $\mathcal{M}(D)$ en partant de D, le modèle canonique où interpréter les termes clos du calcul avec ressources sera la plus petite solution de l'équation

$$D = (\mathcal{M}(D) \to D)_\perp$$

Nous construisons dans ce chapitre deux équations de cette forme, qui se distinguent par le choix du domaine $\mathcal{M}(D)$:

$$(\mathcal{E}_i) \qquad D = (\mathcal{M}_i(D) \to D)_\perp \qquad i = 1, 2$$

Ces équations donnent lieu à deux fonctions d'interprétation des termes différentes. Remarquons que les environnements à considérer sont des fonctions à domaine fini dans Env $= [Var \to \mathcal{M}(D)]$. La notation utilisée pour l'interprétation ne change pas d'un domaine à l'autre, elle est

$$\mathcal{V}[\![-]\!]_{-} : \Lambda_r \times \mathrm{Env} \to D$$

Dans quelle catégorie faut-il chercher les solutions à l'équation aux domaines? Quelles propriétés doit vérifier $\mathcal{M}(D)$? Quelles équations sémantiques doivent satisfaire les modèles du lambda calcul avec ressources? Il faut préciser d'abord que nous entendons construire des structures applicatives, qui vérifient l'équation $\mathcal{V}[\![\lambda x.M]\!]_\rho \cdot d = \mathcal{V}[\![M]\!]_{\rho[x:=d]}$, et suffisamment riches pour constituer des modèles de λ_r (cf. [41, 8, 5] pour le cas du lambda calcul pur). Voici une esquisse de réponse aux questions soulevées. Partons de l'observation que notre définition d'environnement ne permet pas de valider l'équation usuelle en sémantique dénotationnelle, $\mathcal{V}[\![x]\!]_\rho = \rho(x)$, puisque les dénotations pour les variables sont prises dans $\mathcal{M}(D)$ et non pas dans D. Aussi la première condition pour $\mathcal{M}(D)$, indépendamment de l'équation, est l'existence d'une fonction de transformation j pour définir la dénotation des variables dans un environnement, i.e.

$$j : \mathcal{M}(D) \to D \qquad \text{et} \qquad \mathcal{V}[\![x]\!]_\rho = j(\rho(x))$$

Pour simplifier l'analyse de l'application, on suppose que l'on est en mesure de définir une fonction d'interprétation $\mathcal{W}[\![P]\!]_\rho$ des paquets P dans $\mathcal{M}(D)$. Cette hypothèse ne se rapporte en fait qu'à $\mathcal{E}_2$, mais les intuitions sont les mêmes pour $\mathcal{E}_1$. La loi usuelle concernant l'application, $\mathcal{V}[\![MP]\!]_\rho = \mathcal{V}[\![M]\!]_\rho \cdot \mathcal{W}[\![P]\!]_\rho$, entraîne une duplication indésirable des ressources associées aux variables libres du terme par l'environnement ρ. La manipulation de l'environnement doit en fait décrire le processus de substitution dans le calcul, qui correspond, dans notre cas, à la substitution non-déterministe du calcul λ_d. Par analogie avec ce calcul, $\mathcal{V}[\![MP]\!]_\rho$ doit être fonction de l'ensemble de dénotations $\mathcal{V}[\![M]\!]_{\rho_0} \cdot \mathcal{W}[\![P]\!]_{\rho_1}$ où ρ_0, ρ_1 constituent un partage de ρ. Malgré le non-déterminisme inhérent à λ_r, le caractère "may testing" de la sémantique observationnelle nous oblige à prendre la meilleure de ces dénotations approximées, c'est-à-dire, la borne supérieure de l'ensemble des dénotations approximées. Nous avons souligné dans le chapitre 3 que l'ensemble des approximants d'un terme de λ_r n'est pas dirigé donc la deuxième condition sur les équations aux domaines est que D soit un treillis complet; i.e. avec des bornes arbitraires. Ainsi, dans le cadre de l'équation $\mathcal{E}_2$ on aura :

$$\mathcal{V}[\![MP]\!]_\rho = \bigsqcup_{\rho_0,\rho_1} \mathcal{V}[\![M]\!]_{\rho_0} \cdot \mathcal{W}[\![P]\!]_{\rho_1}$$

L'interprétation des termes accompagnés d'une substitution explicite, toujours dans le cadre de $\mathcal{E}_2$, peut être déduite suivant un raisonnement similaire; on aura :

$$\mathcal{V}[\![M\langle P/x\rangle]\!]_\rho = \bigsqcup_{\rho_0,\rho_1} \mathcal{V}[\![M]\!]_{\rho_0[x:=\mathcal{W}[\![P]\!]_{\rho_1}]}$$

La façon de définir le partage des environnements dépend de la structure du domaine $\mathcal{M}(D)$.

Pour l'équation $\mathcal{E}_2$, on construit un domaine $\mathcal{M}_2(D)$ où interpréter les paquets, c'est-à-dire, la composition parallèle. Ce domaine est équipé d'un produit $\bullet$ de multi-ensembles, commutatif, associatif et avec élément neutre. L'équation sémantique pour la composition parallèle est :

$$\mathcal{W}[\![(P \mid Q)]\!]_\rho = \bigsqcup_{\rho_0,\rho_1} \mathcal{W}[\![P]\!]_{\rho_0} \bullet \mathcal{W}[\![Q]\!]_{\rho_1}$$

Donc, $\mathcal{M}_2(D)$ doit être un treillis complet. La dernière condition qu'il doit vérifier est l'existence d'une fonction i pour définir les paquets unitaires :

$$i : D \to \mathcal{M}_2(D) \qquad \text{et} \qquad \mathcal{W}[\![M]\!]_\rho = i(\mathcal{V}[\![M]\!]_\rho)$$

Les deux équations qui nous intéressent ont des solutions du type des modèles de filtres à la Coppo (cf. [22, 23, 24, 9] pour la théorie dans le cadre du lambda calcul standard et [3, 14, 15, 29] pour des calculs faibles).

Dans la section 4.1 on définit les notions de base de la théorie des domaines, utilisées dans la construction des équations.

La section 4.2 est consacrée à la définition de la première équation et à la présentation de sa solution canonique. Le domaine $\mathcal{M}_1(D)$ que l'on construit peut être vu comme la contrepartie du domaine des parties de Hoare pour les multi-ensembles. Cette construction ne permet pas d'interpréter les paquets, car elle n'a pas de bornes supérieures arbitraires. On interprète néanmoins les approximants finis des paquets (i.e. avec un nombre fini de composantes). On montre ensuite que la solution canonique de l'équation $\mathcal{E}_1$ a une présentation logique comme domaine de cônes sur une théorie de types avec produit (sans intersection).

Dans la section 4.3, on présente la construction $\mathcal{M}_2$ qui sert à définir la deuxième équation. Nous prouvons que $\mathcal{M}_2$, comme opération sur les treillis complets (p-algébriques) est universelle. On montre ensuite que la solution canonique de $\mathcal{E}_2$ a une présentation logique comme domaine de filtres sur une théorie de types avec intersection et produit.

4.1 Préliminaires

Nous rappelons quelques définitions de la théorie des domaines (cf. [67, 27] et aussi [58]).

Un *ensemble ordonné*[1] est une paire $(D, \sqsubseteq)$ où $\sqsubseteq$ est un ordre partiel sur l'ensemble D, i.e. une relation reflexive, transitive et anti-symétrique.

---

[1] poset

Un sous-ensemble non-vide X d'un ensemble ordonné $(D, \sqsubseteq)$ est *dirigé* si et seulement si pour tout $d, e \in X$ il existe $x \in X$ tel que $d \sqsubseteq x$ et $e \sqsubseteq x$. Un *ordre partiel complet (opc)* est un ensemble ordonné $(D, \sqsubseteq)$ avec *élément minimum* $\bot$ où tout sous-ensemble dirigé X de D a une *borne supérieure* $\bigsqcup X$. Un élément d d'un opc $(D, \sqsubseteq)$ est *compact* si et seulement si, pour tout X sous-ensemble dirigé de D, $d \sqsubseteq \bigsqcup X$ implique $d \sqsubseteq x$ pour $x \in X$. Par exemple, $\bot$ est compact. Nous désignons par $\mathcal{K}(D)$ l'ensemble des éléments compacts de D. Un opc D est *algébrique* si et seulement si, pour tout x, l'ensemble $\{d \in D \,/\, d \in \mathcal{K}(D) \;\&\; d \sqsubseteq x\}$ est dirigé et x est sa borne supérieure. Un opc D est *ω-algébrique* s'il est algébrique et $\mathcal{K}(D)$ est un ensemble dénombrable.

Un *cône inférieur* sur $(D, \sqsubseteq)$ est un sous-ensemble X de D fermé vers le bas, c'est-à-dire, vérifiant

$$x \in X \;\&\; y \sqsubseteq x \;\;\Rightarrow\;\; y \in X$$

La *complétion par cônes inférieurs* de $(D, \sqsubseteq)$ est l'ensemble des cônes de D ordonnés par inclusion. Un *idéal* est un cône inférieur dirigé. Étant donné $(D, \sqsubseteq)$, on note Idéal$(D, \sqsubseteq)$ sa *complétion par idéaux* (l'ensemble des idéaux de $(D, \sqsubseteq)$ ordonnés par inclusion), un opc ω-algébrique. De plus, tout opc algébrique $(D, \sqsubseteq)$ est isomorphe à Idéal$(\mathcal{K}(D), \sqsubseteq)$.

Un *treillis complet* est un ensemble ordonné $(D, \sqsubseteq)$ dans lequel tout sous-ensemble X a une borne supérieure $\bigsqcup X$. Un treillis complet a un élément minimum et un élément maximum, à savoir $\bot = \bigsqcup \emptyset$ et $\top = \bigsqcup D$. Dans ce cadre, un élément c est compact ssi pour tout $X \subseteq D$, $c \sqsubseteq \bigsqcup X \;\Rightarrow\; c \sqsubseteq \bigsqcup Y$ où Y est un sous-ensemble fini de X. Un élément p d'un treillis est *premier* si et seulement si pour tout sous-ensemble fini Y de D vérifiant $p \sqsubseteq \bigsqcup Y$, il existe $x \in Y$ tel que $p \sqsubseteq x$. On remarquera que $\bot$ n'est pas premier. Nous utilisons $\mathcal{KP}(D)$ comme notation pour l'ensemble des éléments à la fois compacts et premiers de D, et nous écrivons $\mathcal{KP}_\bot(D) = \mathcal{KP}(D) \cup \{\bot\}$. Un treillis complet $(D, \sqsubseteq)$ est *p-algébrique* si tout élément e de D est la borne supérieure des éléments à la fois compacts et premiers qu'il domine:

$$e = \bigsqcup \{c \,/\, c \in \mathcal{KP}(D) \;\&\; c \sqsubseteq e\}$$

Un élément compact d'un treillis est donc la borne supérieure d'un nombre fini d'éléments premiers, i.e. si c est compact alors on peut écrire $c = p_1 \sqcup \ldots \sqcup p_n$ avec $p_i \in \mathcal{KP}(D)$. Il faut noter que la définition de treillis p-algébrique aurait aussi bien pu être formulée en remplaçant $\mathcal{KP}(D)$ par $\mathcal{KP}_\bot(D)$.

Proposition 4.1.1 *([58]) Tout treillis p-algébrique $(D, \sqsubseteq)$ est isomorphe à la complétion par cônes inférieurs de $(\mathcal{KP}(D), \sqsubseteq)$, ou encore, à la complétion par cônes inférieurs non-vides de $(\mathcal{KP}_\bot(D), \sqsubseteq)$.*

Preuve. On définit les fonctions de transformation c et p, d'éléments du treillis en cônes inférieurs et vice versa par

$$x \in D \;\Rightarrow\; c(x) = \{y \,/\, y \in \mathcal{KP}(D) \;\&\; y \subseteq x\}$$

$$d \text{ un cône de } \mathcal{KP}(D) \ \Rightarrow\ p(d) = \sqcup x \in d$$

Il est clair que $c(x)$ est un cône inférieur et que $p(d)$ est un élément de D (tout ensemble a une borne supérieure). De plus, $x = \sqcup\{e \,/\, e \in \mathcal{KP}(D) \ \&\ e \sqsubseteq x\}$ implique $p(c(x)) = x$. Il ne reste à prouver que $c(p(d)) = d$: en effet,

$$c(p(d)) = \{y \,/\, y \in \mathcal{KP}(D) \ \&\ y \sqsubseteq \sqcup\{x \in d\}\,\}$$

En appliquant la définition d'élément compact et premier on obtient

$$c(p(d)) = \{y \,/\, y \in \mathcal{KP}(D) \ \&\ y \sqsubseteq x \ \&\ x \in d\}$$

Donc $c(p(d)) = \{y \,/\, y \sqsubseteq x \ \&\ x \in d\}$, car les éléments de d appartiennent à $\mathcal{KP}(D)$. C'est-à-dire, $c(p(d)) = d$. $\square$

Pour tout ensemble E, on note $\mathbf{Fin}(E)$ l'ensemble des parties finies de E. Soit $(D, \sqsubseteq)$ un ensemble ordonné; le préordre $\sqsubseteq^\flat$ (cf. [38]; ne pas confondre avec le préordre observationnel plat défini dans le chapitre 3) sur $\mathbf{Fin}(E)$ est défini par

$$u \sqsubseteq^\flat v \overset{def}{\Leftrightarrow} \forall x \in u \ \exists y \in v \ x \sqsubseteq y$$

L'ensemble vide est le minimum de ce préordre. Étant donné $(D, \sqsubseteq)$, l'*ensemble ordonné des parties*[2] $\mathcal{Q}(D)$ est Idéal$(\mathbf{Fin}(D), \sqsubseteq^\flat)$. Le *domaine des parties de Hoare*[3] sur un opc ω-algébrique $(D, \sqsubseteq)$ est $\mathcal{Q}(\mathcal{K}(D))$.

Proposition 4.1.2 *La complétion par cônes inférieurs de* $(D, \sqsubseteq)$ *est isomorphe à* $\mathcal{Q}(D)$*.*

Preuve. Pour un cône X de D, on définit $i(X) = \mathbf{Fin}(X)$. Il est facile de voir que cet ensemble est un idéal de Idéal$(\mathbf{Fin}(D), \sqsubseteq^\flat)$. Si $v \in \mathbf{Fin}(X)$ et $u \in \mathbf{Fin}(D)$ alors $u \sqsubseteq^\flat v \ \&\ x \in u \ \Rightarrow\ \exists y \in v \ x \sqsubseteq y$ d'où $x \in X$ par définition de cône. Donc $u \in \mathbf{Fin}(X)$, c'est-à-dire, $\mathbf{Fin}(X)$ est un cône inférieur. De plus, il est dirigé: si $u, v \in \mathbf{Fin}(X)$ alors $u \sqsubseteq^\flat u \cup v$ et $v \sqsubseteq^\flat u \cup v$.

Pour $I \in$ Idéal$(\mathbf{Fin}(D), \sqsubseteq^\flat)$, on pose $c(I) = \bigcup\{u \,/\, u \in I\}$. Soit $x \in c(I)$ et $y \sqsubseteq x$; alors il existe $u \in I$ t.q. $x \in u$. Or $\{y\} \sqsubseteq^\flat u$. Donc $\{y\} \in I$ par définition d'idéal, d'où $y \in c(I)$.

Nous montrons maintenant que la fonction i est inverse à droite de c et que c est inverse à gauche de i. En effet, $i(c(I)) = I$: l'inclusion $\supseteq$ est évidente; pour prouver $\subseteq$ on procède comme suit. Soit $v \in i(c(I))$, c'est-à-dire $v = \{v_1, \ldots, v_n\}$ où $\forall i \exists u_i \in I \ v_i \in u_i$. Puisque $\{v_i\} \sqsubseteq^\flat u_i$, on a $\{v_i\} \in I$ par définition d'idéal. De plus, I a une borne supérieure $\bigsqcup I$ qui vérifie $\forall i \ \{v_i\} \sqsubseteq^\flat \bigsqcup I$, i.e. $\forall i \exists v'_i \in \bigsqcup I \ v_i \sqsubseteq v'_i$. Par conséquent, $\{v_1, \ldots, v_n\} \sqsubseteq^\flat \bigsqcup I$, ce qui implique $v \in I$ par définition d'idéal.

L'égalité $c(i(X)) = X$ conclut notre preuve : l'inclusion $\supseteq$ est évidente; soit $x \in c(i(X))$, alors il existe $u \subseteq X$ fini t.q. $x \in u$. Donc $x \in X$. $\square$

---

[2] lower powerposet ou Hoare's powerposet

[3] lower powerdomain ou Hoare's powerdomain

Étant donnés deux opcs D et E, l'espace des fonctions $D \to E$ est l'opc de toutes les fonctions continues de D dans E, avec l'ordre point par point :

$$f \leq g \;\text{ sii }\; \forall x \in D \; f(x) \sqsubseteq_E g(x)$$

Pour décrire les fonctions de $(D \to E)$ nous utilisons une syntaxe qui évoque le λ-calcul; $\lambda v \in D.e$ dénote la fonction f tel que pour tout $d \in D$, $f(d)$ est l'élément de E obtenu par remplacement de v par d dans e, où e est une expression mathématique quelconque. L'espace $D \to_\perp E$ est l'ensemble $\{f : D \to E \,/\, f(\perp) = \perp\}$ des fonctions continues strictes de D dans E, avec l'ordre point par point hérité de $D \to E$. L'*élévation*[4] $D_\perp$ d'un opc D est l'ensemble $\{< 0, d > \;/\, d \in D\} \cup \{\perp\}$ pourvu de l'ordre suivant :

$$x \leq y \;\text{ ssi }\; (x = \perp) \;\text{ ou }\; \exists d, d' \in D \; d \sqsubseteq d' \;\&\; x =< 0, d > \;\&\; y =< 0, d' >$$

Deux fonctions continues, *up* et *down*, sont associées à la construction de l'élévation. Leurs types et définitions sont les suivants :

$$\begin{array}{ll} up : D \to D_\perp & up(d) =< 0, d > \\ down : D_\perp \to_\perp D & down(< 0, d >) = d \;,\; down(\perp) = \perp_D \end{array}$$

On appelle *application élevée* la fonction continue $\_(\!(\_)\!)$ qui vérifie

$$\_(\!(\_)\!) : (D \to E)_\perp \times D \to E$$

$$f(\!(d)\!) = down(f)(d) \qquad \text{où } f \in (D \to E)_\perp \;\text{ et }\; d \in D$$

Étant donné un ensemble E, $E^{\odot}$ dénote l'ensemble des multi-ensembles finis d'éléments de E. L'union des multi-ensembles u et v s'écrit $u{\cdot}v$ et nous identifions les éléments $e \in E$ aux singletons $\{e\}$. Tout multi-ensemble est donc un produit $u = e_1{\cdot} \ldots {\cdot}e_n$ où $n \geq 0$; évidemment, les e_i ne sont pas nécessairement distincts deux à deux. Le produit vide est l'ensemble vide $\emptyset$. Dorénavant, $q_1{\cdot} \ldots {\cdot}q_n$ dénote le multi-ensemble vide quand $n = 0$. Afin de simplifier la notation (et dans le souci de rapprocher les constructions du calcul et du domaine), ces multi-ensembles s'écrivent comme des monômes $u = e_1^{m_1} \cdots e_k^{m_k}$ où $m_i > 0$. Soit $(P, \sqsubseteq$ $)$ un ensemble ordonné; l'ensemble ordonné des multi-ensembles finis de P est $(P^{\odot}, \lessdot)$, où $\emptyset$ est l'élément minimum et $\lessdot$ est la plus petite précongruence qui contient $\sqsubseteq$ et l'inclusion de multi-ensembles. C'est-à-dire, le plus petit préordre sur $P^{\odot}$ qui satisfait :

$$\begin{array}{c} p \sqsubseteq q \;\Rightarrow\; p \lessdot q \\ u \lessdot u{\cdot}v \\ u \lessdot u' \;\&\; v \lessdot v' \;\Rightarrow\; u{\cdot}v \lessdot u'{\cdot}v' \end{array}$$

Lemme 4.1.3

$$\begin{array}{lll} p_1{\cdot} \ldots {\cdot}p_k \lessdot u & \Leftrightarrow & \exists q_1, \ldots, q_k \; \forall i \; p_i \sqsubseteq q_i \;\& \\ & & (u = q_1{\cdot} \ldots {\cdot}q_k \;\; ou \;\; \exists v \; u = q_1{\cdot} \ldots {\cdot}q_k{\cdot}v) \end{array}$$

---

[4] lifting

Lemme 4.1.4 *Soit $(P, \sqsubseteq)$ un ensemble ordonné. La relation $\lessdot$ est un ordre sur $P^{\odot}$.*

Preuve. Soit $|u|$ la cardinalité du multi-ensemble u. Alors $u \lessdot v$ implique $|u| \leq |v|$. On montre $u \lessdot v \lessdot u \Rightarrow u = v$ par récurrence sur $|u|$. Si $|u| = 1$ on a $|v| = 1$, donc $u \sqsubseteq v \sqsubseteq u \Rightarrow u = v$. Sinon, soit p un élément maximal de u par rapport à l'ordre $\sqsubseteq$ et $u = p{\cdot}u'$. Il existe q et v' t.q. $v = q{\cdot}v'$ où $p \sqsubseteq q$ et $u' \lessdot v'$. Il existe aussi p', u'' t.q. $u = p'{\cdot}u''$ avec $q \sqsubseteq p'$ et $v' \lessdot u''$. Alors $p \sqsubseteq q \sqsubseteq p' \Rightarrow p = p' = q$ parce que p est maximal. D'où $u' = u'' \Rightarrow u' = v'$ par l'hypothèse de récurrence. $\square$

4.2 Équation $\mathcal{E}_1$

4.2.1 Le domaine des multi-ensembles $\mathcal{M}_1(D)$

La construction $\mathcal{M}_1(D)$ est fondée sur l'observation que ce domaine doit décrire les multi-ensembles, tout comme les domaines des parties décrivent les ensembles. Sa définition repose sur l'opération $(\,)^{\odot}$. Au lieu d'appliquer cette opération sur l'ordre partiel $(\mathcal{K}(D), \sqsubseteq)$ - choix usuel pour les domaines des parties - nous supposons que D est un treillis complet p-algébrique et utilisons l'ensemble des éléments compacts et premiers.

Définition 4.2.1 ***(Construction $\mathcal{M}_1$)***

Soit D un treillis complet p-algébrique. On définit

$$\mathcal{M}_1(D) = \textit{Idéal}(\textbf{\textit{Bag}}(D), \lessdot) \quad \textit{où} \quad \textbf{\textit{Bag}}(D) = \mathcal{KP}(D)^{\odot}$$

Dans cette formulation, $(\mathbf{Bag}(D), \lessdot)$ est l'analogue de l'ensemble de parties finies ordonnées par l'ordre bémol, $(\mathbf{Fin}(D), \sqsubseteq^{\flat})$, qui sert à définir le domaine des parties de Hoare.

Une présentation alternative de $(\mathbf{Bag}(D), \lessdot)$ résulte de l'association du multi-ensemble vide à l'élément minimum $\perp$ de D. Étant donné $(P, \sqsubseteq)$ avec minimum $\perp$, on note $P^{\oplus}$ l'ensemble des multi-ensembles finis non-vides de P et $\lessdot_{\perp}$ le préordre défini comme $\lessdot$, où $\perp$ est l'élément neutre, i.e.:

$$\perp{\cdot}u \lessdot_{\perp} u$$

Ici $\perp$ est le plus petit multi-ensemble non-vide; i.e. pour tout multi-ensemble u non-vide, $\perp \lessdot_{\perp} u$ est vérifié puisque $\perp \sqsubseteq p$ quelque soit $p \in P$.

Les ensembles ordonnés $(P^{\odot}, \lessdot)$ et $(P_{\perp}^{\oplus}, \lessdot_{\perp})$ sont isomorphes : les deux injections b, b' définies par

$$(P^{\odot}, \lessdot) \xrightarrow{b} (P_{\perp}^{\oplus}, \lessdot_{\perp}) \xrightarrow{b'} (P^{\odot}, \lessdot)$$

$$\begin{array}{ll} b(\emptyset) = \bot & b(p{\cdot}u) = p{\cdot}b(u) \end{array}$$

$$\begin{array}{ll} b'(\bot) = \emptyset & b'(p) = p \\ b'(\bot{\cdot}u) = b'(u) & b'(p{\cdot}u) = \begin{cases} p{\cdot}b'(u) & \text{si } b'(u) \neq \emptyset \\ p & \text{sinon} \end{cases} \end{array}$$

où $p \in P$ et $u \in P^{\odot}$ ou $u \in P^{\oplus}$, vérifient

$$u, v \in P^{\odot} \ \& \ u {<} v \ \Rightarrow \ b(u) {<_\bot} b(v)$$

$$u, v \in P^{\oplus}_{\bot} \ \& \ u {<_\bot} v \ \Rightarrow \ b'(u) {<} b'(v)$$

Par cet isomorphisme, le domaine $\mathcal{M}_1(D)$ devient Idéal$(\mathcal{KP}_\bot(D)^{\oplus}, <_\bot)$. La décision de mener la construction de $\mathcal{M}_1(D)$ sur la base des éléments de D à la fois compacts et premiers, au lieu d'en prendre simplement les compacts, est liée au besoin de l'existence de la fonction j qui doit transformer un idéal de $\mathcal{M}_1(D)$ dans un élément de D. Puisque tout treillis complet p-algébrique D est isomorphe à la complétion par cônes non-vides de $\mathcal{KP}_\bot(D)$, la définition de j apparaît naturellement : étant donné un idéal de multi-ensembles, j forme l'union des ensembles sous-jacents. Formellement, pour $a \in \mathcal{M}_1(D)$:

$$j(a) = \bigcup \{\, x \in \mathcal{KP}_\bot(D) \,/\, \exists u \in a \ \ x {<} u \,\}$$

Si l'on restreint le domaine de j à $\mathcal{K}(\mathcal{M}_1(D))$, la fonction prend la forme suivante :

$$j(a) = \bigcup \{x \in \mathcal{KP}_\bot(D) \,/\, \exists i \ x \sqsubseteq d_i\}$$

où $d_i \in \mathcal{KP}_\bot(D)$ et a est l'idéal principal $\downarrow(d_1{\cdot}\ldots{\cdot}d_n)$.

4.2.2 La fonction d'interprétation associée à $\mathcal{E}_1$

Les environnements sont des fonctions dans Env $= (Var \to \mathcal{M}_1(D))$ avec un domaine significatif fini; c'est-à-dire, pour tout environnement ξ, seulement un nombre fini de variables ont une image distincte du plus petit élément. Nous appelons *environnements compacts* les fonctions définies sur EnvC $= (Var \to \mathcal{K}(\mathcal{M}_1(D)))$. Pour tout V ensemble fini de variables, $\rho/_V$ dénote l'environnement tel que

$$\rho/_V(y) = \begin{cases} \rho(y) & \text{si } y \notin V \\ \text{indéfini} & \text{sinon} \end{cases}$$

La signification d'un terme avec ressources est le résultat de la composition des significations des *termes finis* qui lui sont associés. Par terme fini on entend les termes dont les paquets qui constituent les arguments et les substitutions ne contiennent qu'un nombre fini de composantes. La relation binaire $P \propto Q$ saisit l'idée que toute multiplicité présente dans le paquet P devient finie dans le paquet

Q : les multiplicités infinies de P sont remplacées par des multiplicités finies dans Q et les multiplicités finies de P peuvent voir leur valeur décroître dans Q. La définition de $\propto$ est :

$$M^n \propto \begin{cases} \mathbf{1} & \\ M^m & \text{pour tout } m \in \mathbb{N} \text{ si } n = \infty \\ M^m & \text{pour tout } m \leq n \text{ si } n \in \mathbb{N} \end{cases}$$

$$P \propto P' \text{ et } Q \propto Q' \ \Rightarrow \ (P \mid Q) \propto (P' \mid Q')$$

Le *produit d'environnements compacts* $\rho_m \cdot \ldots \cdot \rho_n$ est défini par:

$$m \leq n \ \&$$

$$(\forall i \ \rho_i \in \text{EnvC} \ \Rightarrow \ (\forall x \ \rho_i(x) = \downarrow d_{xi} \ \Rightarrow \ (\rho_m \cdot \ldots \cdot \rho_n)(x) = \downarrow (d_{xm} \cdot \ldots \cdot d_{xn})))$$

La fonction d'interprétation des termes (ouverts) de Λ_r dans D est définie par les clauses ci-dessous :

$$\mathcal{V}[\![\_]\!]_{\_} : \Lambda_r \times \text{Env} \rightarrow D$$

$$\begin{array}{rcl}
\mathcal{V}[\![x]\!]_\rho & = & j(\rho(x)) \\[2ex]
\mathcal{V}[\![\lambda x.M]\!]_\rho & = & up(\lambda v \in \mathcal{M}_1(D).\mathcal{V}[\![M]\!]_{\rho[x:=v]}) \\[2ex]
\mathcal{V}[\![MP]\!]_\rho & = & \bigsqcup \mathcal{V}[\![M]\!]_{\rho_0}((\downarrow d_1 \cdot \ldots \cdot d_n)) \\[2ex]
 & & \text{où } \begin{cases} \rho \supseteq \rho_0 \cdot \ldots \cdot \rho_n \ \& \ P \propto (M_1 \mid \cdots \mid M_n) \ \& \\ d_i \in \mathcal{V}[\![M_i]\!]_{\rho_i} \cap \mathcal{KP}_\perp(D) \end{cases} \\[4ex]
\mathcal{V}[\![M\langle P/x \rangle]\!]_\rho & = & \bigsqcup \mathcal{V}[\![M]\!]_{\rho_0[x:=\downarrow d_1 \cdot \ldots \cdot d_n]} \\[2ex]
 & & \text{où } \begin{cases} \rho \supseteq \rho'_0 \cdot \ldots \cdot \rho_n \ \& \ \rho'_0 = \rho_0/_x \ \& \\ P \propto (M_1 \mid \cdots \mid M_n) \ \& \\ d_i \in \mathcal{V}[\![M_i]\!]_{\rho_i} \cap \mathcal{KP}_\perp(D) \end{cases}
\end{array}$$

Les deux premières lignes de la définition sont évidentes. Dans la troisième, nous utilisons l'application élevée. Le trait distinctif de ces équations est l'utilisation de la borne supérieure pour donner à l'application sa signification et pour traiter les substitutions explicites. C'est la raison pour laquelle nous nous plaçons dans le cadre des treillis complets. La définition de la sémantique d'un terme en fonction des termes finis qui lui sont associés est motivée par le fait que les évaluations convergentes d'un terme ne font qu'un nombre fini de saisies. Nous avons souligné dans l'introduction du chapitre que la raison pour partager l'environnement ρ en autant de sous-environnements que nécessaire provient de l'observation que toute valeur d'un terme (i.e. toute évaluation convergente) peut être obtenue en distribuant d'abord les substitutions et en évaluant ensuite (c'est ce qui est fait dans λ_d). Remarquons que les environnements $\rho_0, \ldots, \rho_n$ utilisés dans la définition de la sémantique de l'application et des termes avec substitutions explicites sont des environnements compacts (condition exigée par la définition du produit). On pourrait certainement utiliser des environnements arbitraires, mais ce choix alourdirait les preuves.

4.2.3 Domaine $\mathcal{C}$ de cônes supérieurs

La solution canonique de l'équation aux domaines de la section précédente admet une description concrète dans le style des modèles de filtres du lambda calcul pur [9, 22, 29]. La logique sous-jacente, se compose d'un langage de types et d'une relation de conséquence entre les types.

La syntaxe des types (formules) est donnée par la grammaire suivante :

$$\begin{array}{llcl} (\texttt{Ft}\,) & \phi & ::= & \omega \,|\, \pi \to \phi \\ (\texttt{Fb}\,) & \pi & ::= & \phi \,|\, (\pi \times \pi) \end{array}$$

Sauf indication contraire, ϕ, σ, δ sont des types de `Ft` et π, ψ, θ, τ sont des types de `Fb` .

La constante ω joue le rôle de la vérité logique; autrement dit, ω est la plus petite unité d'information. Les types flèches sont utilisées dans la construction de la signification des fonctions. Le langage des types de `Fb` correspondant aux arguments est construit autour de l'opérateur ×, contrepartie logique de la composition parallèle du calcul. La fonction de ce *produit* de types est analogue à celle de la conjonction dans les théories de types avec intersection : il permet de regrouper les types des arguments d'une fonction. Par exemple, $((\pi \to \phi) \times \pi) \to \phi$ fera partie de la dénotation de $\lambda x.(xx)$. Cependant, à la différence de la conjonction, le produit n'est pas idempotent, i.e. $\pi \times \pi$ n'est pas équivalent à π en général (le produit est idempotent seulement pour ω.) Nous revenons sur cette particularité plus bas.

La relation de conséquence $\phi \leq \sigma$ est un préordre sur les formules qui peut s'interpréter comme "ϕ implique σ". C'est la plus petite relation reflexive et transitive qui vérifie les lois usuelles pour la flèche dans la théorie faible du lambda calcul, i.e. :

$$(1)\, \phi \leq \omega$$

$$(2)\, \pi \to \omega \leq \omega \to \omega$$

$$(3)\, \pi_1 \leq \pi_0 \,\&\, \phi_0 \leq \phi_1 \;\Rightarrow\; (\pi_0 \to \phi_0) \leq (\pi_1 \to \phi_1)$$

ainsi que les lois additionnelles suivantes :

$$(4)\,\omega \times \pi \;\leq\; \pi$$

$$(5)\,\pi \;\leq\; \omega \times \pi$$

$$(6)\,\pi_0 \times \pi_1 \;\leq\; \pi_1 \times \pi_0$$

$$(7)\,\pi_0 \times (\pi_1 \times \pi_2) \;\leq\; (\pi_0 \times \pi_1) \times \pi_2$$

$$(8)\,\pi_0 \;\leq\; \pi_0' \;\&\; \pi_1 \;\leq\; \pi_1' \;\Rightarrow\; \pi_0 \times \pi_1 \;\leq\; \pi_0' \times \pi_1'$$

On remarquera que le produit n'est pas idempotent, à la différence de la conjonction dans les théories de types avec intersection (modèles de filtres),

$$\pi \;\leq\; \pi_0 \;\&\; \pi \;\leq\; \pi_1 \;\not\Rightarrow\; \pi \;\leq\; (\pi_0 \times \pi_1)$$

En particulier, on n'a $\pi \;\leq\; \pi \times \pi$ que pour $\pi = \omega$. Néanmoins, le produit vérifie les autres propriétés usuelles de la conjonction :

$$\pi_0 \times \pi_1 \;\leq\; \pi_0 \text{ et } \pi_0 \times \pi_1 \;\leq\; \pi_1$$

On écrit $\phi \sim \sigma$ quand $\phi \;\leq\; \sigma$ et $\sigma \;\leq\; \phi$ sont démontrables. Par exemple, $\pi \to \omega \sim \omega \to \omega$, $\omega \times \pi \sim \pi$ et $(\pi_0 \times \pi_1) \times \pi_2 \sim \pi_0 \times (\pi_1 \times \pi_2)$. L'associativité du produit permet d'éviter l'utilisation de parenthèses dans les formules : la forme des types produit est $\sigma_1 \times \cdots \times \sigma_n$ où $n \geq 1$ et $\sigma_i \in \mathrm{Ft}$. La constante γ sera souvent utilisée comme abréviation du type $\omega \to \omega$.

La théorie des types que nous considérons possède deux propriétés fondamentales qui explicitent la forme des types liés par $\leq$.

Propriété du Produit - $\mathcal{E}_1$: Soit $\sigma_1, \ldots, \sigma_m \in \mathrm{Ft}$ et $J = \{j \,/\, \sigma_j \neq \omega\}$.

$$\phi_1 \times \cdots \times \phi_n \;\leq\; \sigma_1 \times \cdots \times \sigma_m \Leftrightarrow \exists \text{ injection } i : J \to [1, n] \; \forall j \in J \;\; \phi_{i(j)} \;\leq\; \sigma_j$$

De plus la taille de la preuve de $\phi_{i(j)} \;\leq\; \sigma_j$ est au plus égale à celle de $\pi \;\leq\; \psi$.

Preuve. On note $\psi = \sigma_1 \times \ldots \times \sigma_m$ et $\pi = \phi_1 \times \ldots \times \phi_n$.

La partie $\Leftarrow$ est simple. Si J est vide, alors $\psi = \overbrace{\omega \times \ldots \times \omega}^{m}$; pour tout π on a $\pi \;\leq\; \psi$ par les clauses 1,4,8 de la définition de $\leq$. Sinon, soit $J = \{j_1, \ldots, j_k\}$; on a $\phi_{i(j_1)} \times \ldots \times \phi_{i(j_k)} \;\leq\; \sigma_{j_1} \times \ldots \times \sigma_{j_k}$ par application de 8. De plus, $\pi \;\leq\; \phi_{i(j_1)} \times \ldots \times \phi_{i(j_k)}$ par commutativité, 4 et 8. Finalement, $\sigma_{j_1} \times \ldots \times \sigma_{j_k} \times \underbrace{\omega \times \ldots \times \omega}_{m-k} \;\leq\; \psi$ par commutativité. Ainsi, $\pi \;\leq\; \psi$ est vérifié par transitivité.

On démontre l'implication $\Rightarrow$ par récurrence sur la longueur l de la dérivation de $\pi \leq \psi$, définie comme le nombre de règles utilisées.

$l = 1$: Si $\pi \leq \psi$ est une instance de l'axiome de réflexivité, de 1 ou de 2, l'énoncé est immédiatement vérifié. Si l'axiome 4 est appliqué, on a $\phi_1 = \omega$, $m = n - 1$ et $\sigma_i = \phi_{i+1}$. Soit $J = \{j \,/\, \phi_{j+1} \neq \omega\}$ et l'injection $i : J \rightarrow \{1, \ldots, n\}$ définie par $i(j) = j + 1$. Alors $\phi_{i(j)} = \sigma_j$ implique $\phi_{i(j)} \leq \sigma_j$ par reflexivité. Le même argument sert dans le cas où c'est l'axiome 5 qui est utilisé : l'injection devient $i(j) = j - 1$. Si c'est l'axiome 6 qui prouve $\pi \leq \psi$, alors $m = n$ et il existe a t.q. $1 \leq a < n$ et $\psi = (\phi_{a+1} \times \cdots \times \phi_n) \times (\phi_1 \times \cdots \phi_a)$. L'injection est de la forme

$$i(j) = \begin{cases} a + j & \text{si } j \in \{1, \ldots, n - a\} \\ j - (a + 1) & \text{si } j \in \{n - a + 1, \ldots, n\} \end{cases}$$

Il faut noter que $\phi_{i(j)} = \sigma_j$ tant que σ_j est différent de ω; dans ce cas, $\phi_{i(j)} \leq \sigma_j$ aussi.
Nous omettons les détails pour une preuve de $\pi \leq \psi$ par l'axiome d'associativité, tout à fait similaire à celle donnée pour la commutativité.

$l > 1$: Trois possibilités sont à étudier, selon que la dérivation de $\pi \leq \psi$ termine par la règle 3, cas où il n'y a rien à prouver, par la règle 8 ou par transitivité. Dans le deuxième cas,

$$\exists a\, b \;\; 1 \leq a < n \;\;\&\;\; 1 \leq b < m \;\;\&\;\; \phi_1 \times \cdots \times \phi_a \leq \sigma_1 \times \cdots \times \sigma_b \quad \text{en } l_1 \text{ étapes et}$$

$$\phi_{a+1} \times \cdots \times \phi_n \leq \sigma_{b+1} \times \cdots \times \sigma_m \quad \text{en } l_2 \text{ étapes}$$

où $l_1 + l_2 + 1 = l$. En posant $J_1 = J \cap \{1, \ldots, b\}$ et $J_2 = J \cap \{b + 1, \ldots, m\}$, l'hypothèse de récurrence fournit deux injections $k_1 : J_1 \rightarrow \{1, \ldots, a\}$ et $k_2 : J_2 \rightarrow \{a + 1, \ldots, n\}$ t.q. $\omega \neq \phi_{k_h(j)} \leq \sigma_j$ en au plus l_h étapes pour tout $j \in J_h$, avec $h = 1, 2$. Si l'on définit $i : J \rightarrow \{1, \ldots, n\}$ par

$$i(j) = \begin{cases} k_1(j) & \text{si } j \in J_1 \\ k_2(j) & \text{si } j \in J_2 \end{cases}$$

on a $\omega \neq \phi_{i(j)} \leq \sigma_j$ en moins de l étapes, par h.r.
Le dernier cas est celui d'une preuve de $\pi \leq \psi$ par transitivité. C'est-à-dire, $\pi \leq \delta_1 \times \cdots \times \delta_r$ en l_1 étapes, $\delta_1 \times \cdots \times \delta_r \leq \psi$ en l_2 étapes et $l_1 + l_2 + 1 = l$. Soit $J_1 = \{j \,/\, \delta_j \neq \omega\}$ et $J_2 = \{j \,/\, \sigma_j \neq \omega\}$. Par h.r., il existe deux injections $k_1 : J_1 \rightarrow \{1, \ldots, n\}$ $k_2 : J_2 \rightarrow \{1, \ldots r\}$ qui vérifient $\omega \neq \phi_{k_h(j)} \leq \sigma_j$ pour tout $j \in J_h$, avec $h = 1, 2$. Il faut noter que J_2 est en réalité notre ensemble J. Du fait que $\delta_{k_2(j)} \neq \omega$ pour tout $j \in J$, on déduit $k_2(j) \in J_1$. Par conséquent, l'injection $i : J \rightarrow \{1, \ldots, n\}$ t.q. $i = k_1 \circ k_2$ est bien définie. L'hypothèse de récurrence donne $\phi_{k_1(k_2(j))} \leq \delta_{k_2(j)}$ et $\delta_{k_2(j)} \leq \sigma_j$ en au plus l_1 et l_2 étapes respectivement. Par transitivité, $\phi_{k_1(k_2(j))} \leq \sigma_j$ est prouvable en l étapes au plus. $\square$

Propriété de la Flèche - $\mathcal{E}_1$:

$$\pi \rightarrow \phi \leq \psi \rightarrow \sigma \Leftrightarrow (\sigma \neq \omega \Rightarrow \psi \leq \pi \ \& \ \phi \leq \sigma)$$

Preuve. La partie $\Leftarrow$ est prouvée par cas selon σ. Si $\sigma = \omega$, pour tout π, ϕ, ψ, on a

$$\pi \rightarrow \phi \leq \pi \rightarrow \omega \leq \omega \rightarrow \omega \leq \psi \rightarrow \omega$$

Dans le cas où $\sigma \neq \omega$, l'énoncé est vérifié par la troisième règle de $\leq$.

Nous montrons la partie $\Rightarrow$ par récurrence sur la taille l de l'inférence de $\pi \rightarrow \phi \leq \psi \rightarrow \sigma$. Soit $\sigma \neq \omega$. Si $l = 1$, alors $\pi \rightarrow \phi \leq \psi \rightarrow \sigma$, donc $\pi = \psi$ et $\phi = \sigma$.

Sinon $(l > 1)$, l'inférence est soit par application de la règle 3 de la définition de $\leq$, soit par transitivité. Le premier cas suppose $\psi \leq \pi$ et $\phi \leq \sigma$ directement. Dans le second cas, il existe θ t.q.

$$\pi \rightarrow \phi \leq \theta \text{ et } \theta \leq \psi \rightarrow \sigma$$

Soit l_1 et l_2 les longueurs respectives de ces morceaux de preuve. L'hypothèse de récurrence ne s'applique pas, car θ peut être un type produit. Si $\theta = \delta_1 \times \cdots \times \delta_n$ avec $\delta_i \in \mathtt{Ft}$, la propriété du produit implique l'existence d'un δ_k différent de ω, soit $\delta_k = \pi' \rightarrow \phi'$, qui satisfait $\delta_k \leq \psi \rightarrow \sigma$ en l_2 étapes au plus. De l'autre côté, par la propriété de la flèche, $\pi \rightarrow \phi \leq \theta$ implique qu'il y a au plus un δ_j différent de ω. Donc, $\theta = \pi' \rightarrow \phi'$ et $\pi \rightarrow \phi \leq \pi' \rightarrow \phi'$ est prouvable en l_1 étapes au plus. En résumant, la situation est la suivante :

$$\pi \rightarrow \phi \leq \pi' \rightarrow \phi' \leq \psi \rightarrow \sigma$$

Par récurrence on a $\phi' \neq \omega$, $\psi \leq \pi'$ et $\phi' \leq \sigma$. Une deuxième application de l'hypothèse de récurrence indique que $\pi' \leq \pi$ et $\phi \leq \phi'$ sont vérifiés. Donc, $\psi \leq \pi$ et $\phi \leq \sigma$. □

Pour satisfaire l'équation aux domaines $\mathcal{E}_1$ qui gouverne λ_r, le domaine concret doit être un treillis complet p-algébrique; c'est-à-dire, la complétion par cônes inférieurs non-vides d'un ensemble ordonné $(P, \sqsubseteq)$ avec $\perp$ (cf. proposition 4.1.1.) À cette fin, nous utilisons $(\mathtt{Ft}\ , \leq)$, où les éléments de $\mathtt{Ft}$ sont dans l'ordre inverse par rapport à la convention adoptée : ici, $\phi \leq \tau$ signifie que τ est moins précis que ϕ. Par dualité, nous sommes amenés à fonder la définition du modèle logique sur la notion de cône supérieur non-vide, appelé aussi cône (supérieur) quand il n'y a pas d'ambiguïté.

Définition 4.2.2 ***(Cône supérieur)***

Un ensemble $f \subseteq \mathtt{Ft}$ $(f \subseteq \mathtt{Fb}\,)$ *est un* cône supérieur *sur* $(\mathtt{Ft}\ , \leq)$ $((\mathtt{Fb}\ , \leq)$ *resp.) si et seulement si*

$$x \in f \ \& \ x \leq y \quad \Rightarrow \quad y \in f\,;$$

Soit $\mathcal{C}$ l'ensemble des cônes supérieurs non-vides sur $(\mathrm{Ft}\,, \leq)$*, ordonnés par inclusion. Étant donné un sous-ensemble non-vide A de* Ft *, on dénote par $\uparrow A$ le plus petit cône supérieur de $\mathcal{C}$ qui contient A. Pour les ensembles unitaires $\{a\}$ la notation se réduit à $\uparrow a$.*

Par construction, $(\mathcal{C}, \subseteq)$ est un treillis complet p-algébrique. Son minimum $\perp$ est $\uparrow\omega = \{\omega\}$, son maximum $\top$ est Ft , et $x \sqcup y = x \cup y$, $x \sqcap y = x \cap y$. Étant donné $A \subseteq \mathcal{C}$, $\uparrow A = \bigsqcup\{\uparrow a \,/\, a \in A\} = \bigcup\{\uparrow a \,/\, a \in A\}$.

Les éléments à la fois compacts et premiers de $\mathcal{C}$ sont ses cônes principaux $\uparrow\phi$, sauf $\perp = \uparrow\omega$; ses éléments compacts s'expriment comme des unions finies d'éléments de $\mathcal{KP}_\perp(\mathcal{C})$.

Nous affirmons non seulement que le domaine $\mathcal{C}$ résout l'équation sémantique $D = (\mathcal{M}_1(D) \to D)_\perp$ à isomorphisme près, mais qu'il constitue un modèle adéquat du calcul. Ce dernier point sera étudié dans le chapitre 5. Nous consacrons le reste de cette section au premier point; c'est-à-dire, à montrer l'isomorphisme suivant :

$$\mathcal{C} \sim (\mathcal{M}_1(\mathcal{C}) \to \mathcal{C})_\perp.$$

La preuve est basée sur l'observation que le domaine $\mathcal{M}_1(\mathcal{C})$ a une présentation logique concrète sur $(\mathrm{Fb}\,, \leq)$. Nous définissons ensuite la notion de filtre, contrepartie logique de celle d'idéal :

Définition 4.2.3 ***(Filtre)***

Un ensemble non-vide $f \subseteq \mathrm{Fb}$ est un filtre *sur $(\mathrm{Fb}\,, \leq)$ si et seulement si f est un cône supérieur co-dirigé sur $(\mathrm{Fb}\,, \leq)$. Un filtre f vérifie donc :*

$$\forall x, y \in f \,.\, \exists z \in f \,.\, z \leq x \;\&\; z \leq y$$

Soit $\mathcal{I}$ l'ensemble de filtres sur $(\mathrm{Fb}\,, \leq)$, ordonnés par inclusion. Étant donné $a \in \mathrm{Fb}$, on dénote par $\uparrow a$ le plus petit filtre contenant a.

Pour toute paire de formules π, ψ dans Fb , on a $\pi \leq \psi$ ssi $\uparrow\psi \subseteq \uparrow\pi$. En particulier, $\uparrow\omega \to \omega = \uparrow\pi \to \omega$ pour tout π. Autrement dit, si $\phi \neq \omega$ alors $\uparrow\phi$ contient tous les types $\pi \to \omega$.

L'ensemble ordonné $(\mathcal{I}, \subseteq)$ est ω-algébrique : l'opération de borne supérieure sur des ensembles dirigés coïncide avec l'union; ses éléments compacts sont les filtres principaux $\uparrow\pi$. Cet opc fournit une présentation concrète du domaine $\mathcal{M}_1(\mathcal{C})$.

Lemme 4.2.4 *$\mathcal{I}$ et $\mathcal{M}_1(\mathcal{C})$ sont isomorphes.*

Preuve. Rappelons que $\mathcal{M}_1(\mathcal{C}) = \text{Idéal}(\mathcal{KP}_\perp(\mathcal{C})^{\oplus}, <_\perp)$. Il suffit de montrer que les ensembles d'éléments compacts de $\mathcal{I}$ et de $\mathcal{M}_1(\mathcal{C})$ sont isomorphes. Ces ensembles sont :

$$\mathcal{K}(\mathcal{I}) = \{\uparrow\pi \,/\, \pi \in \mathrm{Fb}\,\}$$

$$\mathcal{K}(\mathcal{M}_1(\mathcal{C})) = \{\downarrow d_1 \cdot \ldots \cdot d_n \,/\, n \geq 1 \;\; \& \;\; d_i \in \mathcal{KP}_\perp(\mathcal{C})\}$$

$$\text{où } \mathcal{KP}_\perp(\mathcal{C}) = \{\uparrow\phi \,/\, \phi \in \texttt{Ft}\}$$

On définit $\kappa : \texttt{Fb} \rightarrow \mathcal{KP}(\mathcal{C})^\oplus$ par :

$$\begin{array}{rcll} \kappa(\phi) & = & \uparrow\phi & \text{si } \phi \in \texttt{Ft} \\ \kappa(\pi \times \psi) & = & \kappa(\pi)\cdot\kappa(\psi) & \end{array}$$

On vérifie facilement que $\uparrow\pi \in \mathcal{K}(\mathcal{I}) \Leftrightarrow \downarrow\kappa(\pi) \in \mathcal{K}(\mathcal{M}_1(\mathcal{C}))$. De plus, pour tout $\pi, \psi \in \texttt{Fb}$ on a

$$\pi \leq \psi \Leftrightarrow \kappa(\psi) <_\perp \kappa(\pi)$$

$\Rightarrow$: Soit $\pi = \phi_1 \times \ldots \times \phi_n$ et $\psi = \sigma_1 \times \ldots \times \sigma_m$. Donc $\kappa(\pi) = \uparrow\phi_1 \cdot \ldots \cdot \uparrow\phi_n$ et $\kappa(\psi) = \uparrow\sigma_1 \cdot \ldots \cdot \uparrow\sigma_m$. Soit $J = \{j \,/\, \sigma_j \neq \omega\} = \{j_1, \ldots, j_k\}$; par la propriété du produit, il existe une injection $i : J \rightarrow \{1, \ldots, n\}$ telle que $\phi_{i(j)} \leq \sigma_j$ pour tout $j \in J$. C'est-à-dire, $\forall j \in J \;\; \uparrow\sigma_j <_\perp \uparrow\phi_{i(j)}$. Si J n'est pas vide, $\uparrow\sigma_{j_1} \cdot \ldots \cdot \uparrow\sigma_{j_k} <_\perp \uparrow\phi_{i(j_1)} \cdot \ldots \uparrow\phi_{i(j_k)}$ est vérifié. Il faut noter que

$$\kappa(\psi) <_\perp (\uparrow\omega)^{n-k} \cdot \uparrow\sigma_{j_1} \cdot \ldots \cdot \uparrow\sigma_{j_k}$$

et que $\uparrow\phi_{i(j_1)} \cdot \ldots \uparrow\phi_{i(j_k)} <_\perp \kappa(\pi)$; par transitivité on a $\kappa(\psi) <_\perp \kappa(\pi)$. Dans le cas où $J = \emptyset$, on a $\kappa(\psi) = (\uparrow\omega)^m <_\perp \uparrow\omega <_\perp \kappa(\pi)$.

$\Leftarrow$: Par récurrence sur la preuve de $\kappa(\psi) <_\perp \kappa(\pi)$. Observons que $\kappa(\theta) = u \cdot v$ implique l'existence de θ_0, θ_1 t.q. $\theta = \theta_0 \times \theta_1$, $\kappa(\theta_0) = u$ et $\kappa(\theta_1) = v$.

- Si $\kappa(\psi) <_\perp \kappa(\pi)$ est obtenu par reflexivité, alors $\psi = \pi$ et $\pi \leq \psi$ par reflexivité de $\leq$.
- Si π et ψ sont tous les deux dans $\texttt{Ft}$ et $\uparrow\psi \subseteq \uparrow\pi$, alors $\pi \leq \psi$.
- Si $\kappa(\psi) = \kappa(\pi) \cdot \perp <_\perp \kappa(\pi)$, alors $\psi = \pi \times \omega$ implique $\pi \leq \psi$.
- Si $\kappa(\psi) <_\perp \kappa(\psi) \cdot \kappa(\pi') = \kappa(\pi)$, alors $\pi = \psi \times \pi' \leq \psi \times \omega \leq \psi$ est vérifié.
- Si $\kappa(\psi) = \kappa(\psi_0) \times \kappa(\psi_1) <_\perp \kappa(\pi_0) \times \kappa(\pi_1) = \kappa(\pi)$ découle de $\kappa(\psi_0) <_\perp \kappa(\pi_0)$ et $\kappa(\psi_1) <_\perp \kappa(\pi_1)$, on a $\pi_0 \leq \psi_0$ and $\pi_1 \leq \psi_1$ par hypothèse de récurrence. Donc $\pi = \pi_0 \times \pi_1 \leq \psi_0 \times \psi_1 = \psi$.
- Si $\psi = \psi_0 \times \psi_1$ et $\pi = \psi_1 \times \psi_0$, alors $\pi \leq \psi$ par commutativité de $\times$.
- Si $\kappa(\psi) <_\perp \kappa(\theta) <_\perp \kappa(\pi)$, l'hypothèse de récurrence donne $\theta \leq \psi$ et $\pi \leq \theta$. Par transitivité de $\leq$ on obtient $\pi \leq \psi$.

$\square$

La preuve de l'isomorphisme fondamental $\mathcal{C} \sim (\mathcal{I} \to \mathcal{C})_\perp$ nécessite la caractérisation des fonctions continues de $\mathcal{I}$ dans $\mathcal{C}$ comme des fonctions représentables. Nous introduisons ensuite les *fonctions seuil*, qui servent à déterminer les éléments compacts du domaine $\mathcal{I} \to \mathcal{C}$, l'opération d'application dans le modèle et les fonctions F et G qui réalisent l'isomorphisme.

Définition 4.2.5 ***(Fonctions seuil)***
Les fonctions seuil *de $\mathcal{I}$ dans $\mathcal{C}$, notées $f_{\pi\phi}$, sont :*

$$f_{\pi\phi}(c) = \begin{cases} \uparrow\phi & si \uparrow\pi \subseteq c \\ \uparrow\omega & sinon \end{cases}$$

Proposition 4.2.6 *Les éléments compacts de $(\mathcal{I} \to \mathcal{C})$ sont les bornes supérieures finies de la forme $\bigsqcup\limits_I f_{\pi_i\phi_i}$.*

Preuve. Soit $f \in (\mathcal{I} \to \mathcal{C})$, nous montrons les deux propriétés suivantes :

1. $f_{\pi\phi} \subseteq f \Leftrightarrow \uparrow\phi \subseteq f(\uparrow\pi)$
2. $f = \bigcup\{f_{\pi\phi} \,/\, f_{\pi\phi} \subseteq f\}$.

L'implication $\Rightarrow$ de (1) est facile à prouver ; il suffit de constater que $f_{\pi\phi} \subseteq f \Rightarrow f_{\pi\phi}(\uparrow\pi) \subseteq f(\uparrow\pi) \Leftrightarrow \uparrow\phi \subseteq f(\uparrow\pi)$. Quant à l'implication $\Leftarrow$ de (1), supposons $\uparrow\phi \subseteq f(\uparrow\pi)$. On peut se restreindre au cas où $\uparrow\pi \subseteq c$ puisque $f(c)$ contient toujours $\uparrow\omega$. Par la monotonie de f, on obtient $f_{\pi\phi}(c) = \uparrow\phi \subseteq f(\uparrow\pi) \subseteq f(c)$.

En utilisant (1), l'énoncé (2) équivaut à :

$$\forall x \in \mathcal{I} \;\; f(x) = \bigcup\{f_{\pi\phi}(x) \,/\, \uparrow\phi \subseteq f(\uparrow\pi)\}$$

Le membre droit de cette équation se récrit comme suit :

$$(*) \quad \bigcup\{\uparrow\phi \,/\, \exists\pi \;\uparrow\phi \subseteq f(\uparrow\pi) \;\&\; \uparrow\pi \subseteq x\}$$

Du fait que f est continue et que x est la borne supérieure de l'ensemble dirigé $\{\uparrow\pi \,/\, \uparrow\pi \subseteq x\}$, on déduit que $(*)$ est en fait l'union $\bigcup\{\uparrow\phi \,/\, \uparrow\phi \subseteq f(x)\}$, c'est-à-dire, $(*)$ n'est autre que $f(x)$, car $\mathcal{C}$ est un treillis p-algébrique. □

Définition 4.2.7 ***(Application dans le domaine des cônes)***
*L'*application $\_\cdot\_ : \mathcal{C} \times \mathcal{I} \to \mathcal{C}$ *dans le domaine $\mathcal{C}$ est la fonction continue définie par :*

$$x \cdot y = \begin{cases} \{\sigma \in \mathrm{F}t \;/\; \exists\pi \in y \,.\, (\pi \to \sigma) \in x\} & si\; x \neq \uparrow\omega \\ \{\omega\} & sinon \end{cases}$$

Corollaire 4.2.8 *Soit $x \neq \uparrow\omega$. Alors $\phi \in x \cdot \uparrow\pi$ ssi $\pi \to \phi \in x$.*

Preuve. Partie $\Rightarrow$: il découle de la définition d'application qu'il existe $\psi \in \uparrow\pi$ tel que $\psi \to \phi \in x$; alors $\pi \leq \psi \Rightarrow \psi \to \phi \leq \pi \to \phi \in x$, car x est fermé vers le haut par $\leq$. La partie $\Leftarrow$ est encore plus évidente. $\Box$

Définition 4.2.9 *On note F et G les fonctions qui vérifient $\mathcal{C} \xrightarrow{F} (\mathcal{I} \to \mathcal{C})_\perp \xrightarrow{G} \mathcal{C}$ et*

$$F(x) = \begin{cases} up(\lambda y \in \mathcal{I}.x\cdot y) & si\ x \neq \uparrow\omega \\ \perp & sinon \end{cases}$$

$$G(f) = \begin{cases} \uparrow\{\pi \to \phi \,/\, \phi \in f((\uparrow\pi))\} & si\ f \neq \perp \\ \uparrow\omega & sinon \end{cases}$$

Corollaire 4.2.10

$$F(x)((d)) = \left\{ \begin{array}{ll} x\cdot d & si\ x \neq \uparrow\omega \\ \uparrow\omega & sinon \end{array} \right\} = x\cdot d$$

Lemme 4.2.11 ***(Lemme de représentation pour $\mathcal{C}$)***
Pour tout $h \in (\mathcal{I} \to \mathcal{C})_\perp$ et tout $d \in \mathcal{I}$ on a $h((d)) = G(h)\cdot d$.

Preuve. Soit $h \in (\mathcal{I} \to \mathcal{C})_\perp$ et $d \in \mathcal{I}$. Si $h = \perp$, alors

$$G(h)\cdot d = \uparrow\omega\cdot d = \uparrow\omega = \perp((d))$$

Dans le cas où $h = up(g)$, en utilisant la continuité de h, de G et de l'application $\cdot$, plus l'agébricité des domaines, on a :

$$h((d)) = g(\bigsqcup_{\pi\in d}\uparrow\pi) = \bigcup_{f_{\psi\phi}\leq g}\ \bigcup_{\pi\in d} f_{\psi\phi}(\uparrow\pi)$$

$$G(h)\cdot d = \bigcup_{f_{\psi\phi}\leq g} G(up(f_{\psi\phi}))\cdot \bigsqcup_{\pi\in d}\uparrow\pi = \bigcup_{f_{\psi\phi}\leq g}\ \bigcup_{\pi\in d} G(up(f_{\psi\phi}))\cdot\uparrow\pi$$

Il est donc suffisant de prouver l'énoncé restreint à des fonctions seuil et à des filtres principaux comme arguments; i.e. pour tout ϕ , ψ, π

$$(*) \quad G(up(f_{\psi\phi}))\cdot\uparrow\pi = f_{\psi\phi}(\uparrow\pi)$$

Par définition,

$$G(up(f_{\psi\phi})) = \uparrow\{\theta \to \sigma \,/\, \sigma \in f_{\psi\phi}(\uparrow\theta)\} = \bigcup_{\sigma\in f_{\psi\phi}(\uparrow\theta)} \uparrow(\theta \to \sigma)$$

On prouve $G(up(f_{\psi\phi})) = \uparrow(\psi \to \phi)$. La partie $\supseteq$ est évidente. Pour l'inclusion $\subseteq$, prenons un $d \in G(up(f_{\psi\phi}))$; si $d = \omega$ alors $d \in \uparrow(\psi \to \phi)$ par définition de cône. Dans le cas où $d = \theta' \to \sigma'$, on a $d \in \theta \to \sigma$ donc $\theta \to \sigma \leq \theta' \to \sigma'$. Si $\sigma' = \omega$ alors $\psi \to \phi \leq d$ pour tout ψ, ϕ, i.e. $d \in \uparrow(\psi \to \phi)$. Si $\sigma' \neq \omega$, de la propriété de la flèche on tire $\theta' \leq \theta$ et $\sigma \leq \sigma'$. D'où $\sigma \neq \omega$ et $\phi \leq \sigma$ et par conséquent $\psi \in \uparrow\theta$. Donc $\theta' \leq \psi$ et aussi $\phi \leq \sigma'$. Une application de la règle 3 donne $\psi \to \phi \leq \theta' \to \sigma' = d$; ce qui implique $d \in \uparrow\psi \to \phi$.

Revenons à la preuve de $(*)$: par application de la propriété de la flèche on a $\uparrow\psi \to \phi\cdot\uparrow\pi = \{\sigma \,/\, \sigma = \omega \;\text{ ou }\; (\sigma \neq \omega \;\&\; \pi \leq \psi \;\&\; \phi \leq \sigma)\} = f_{\psi\phi}(\uparrow\pi)$, d'où $(*)$.
□

Théorème 4.2.12 ***(Solution de l'équation $\mathcal{E}_1$)***

1. $G \circ F = \boldsymbol{I}_{\mathcal{C}}$

2. $F \circ G = \boldsymbol{I}_{(\mathcal{I} \to \mathcal{C})_\perp}$

Preuve.

1. Soit $x \in \mathcal{C}$. Si $x = \uparrow\omega$, alors $G(F(\uparrow\omega)) = G(\perp) = \uparrow\omega$. Sinon, $G(F(x))$ s'écrit
$$\uparrow\{\pi \to \phi \,/\, \phi \in x\cdot\uparrow\pi\} = \{\pi \to \phi \in x\}$$
L'inclusion $x \subseteq G(F(x))$ en découle immédiatement. De l'autre côté, $\pi' \to \phi' \in G(F(x))$ signifie qu'il existe π, ϕ t.q. $\pi \to \phi \leq \pi' \to \phi' \;\&\; \pi \to \phi \in x$. Ce qui implique $\pi' \to \phi' \in x$, car x est fermé vers le haut par $\leq$.

2. Soit $f \in (\mathcal{I} \to \mathcal{C})_\perp$; pour tout $d \in \mathcal{I}$ on a
$$F(G(f))(\!(d)\!) = G(f)\cdot d = f(\!(d)\!)$$
par application du lemme de représentation des fonctions 4.2.11. Par conséquent, $F \circ G = \mathbf{I}_{(\mathcal{I} \to \mathcal{C})_\perp}$.

□

4.3 Équation $\mathcal{E}_2$

4.3.1 Le domaine des paquets de termes $\mathcal{M}_2(D)$

L'opération $\mathcal{M}_2$ sur les treillis complets p-algébriques est définie pour permettre l'interprétation de la composition parallèle. Plus précisément, $\mathcal{M}_2(D)$ est un treillis complet (donc avec des bornes supérieures arbitraires) avec un produit et des fonctions de transformation j pour interpréter $x\langle P/x\rangle$ et i pour interpréter les paquets unitaires.

Définition 4.3.1 ***(Construction $\mathcal{M}_2$)***

Le domaine des paquets est défini par:

$$\mathcal{M}_2(D) = \mathit{Id\acute{e}al}(\boldsymbol{Fin}(\mathcal{KP}(D)^{\odot}, <), \sqsubseteq^{\flat})$$

Les fonctions de transformation i_D et j_D sont

$$D \overset{i_D}{\rightarrow} \mathcal{M}_2(D) \overset{j_D}{\rightarrow} D$$

$$\begin{array}{ll} i_D(x) = \boldsymbol{Fin}(X) & \text{où } X = \{e \,/\, e \in \mathcal{KP}(D) \;\&\; e \sqsubseteq x\} \\ j_D(x) = \cup\{u \,/\, u \in x\} \cap \mathcal{KP}(D) & \end{array}$$

Soit $u, v \in \boldsymbol{Fin}(\mathcal{KP}(D)^{\odot})$; on note $u \bullet v$ l'élément de $\boldsymbol{Fin}(\mathcal{KP}(D)^{\odot})$ défini par

$$\{a{\cdot}b \,/\, a \in u \;\&\; b \in v\}$$

Le produit $\bullet : \mathcal{M}_2(D) \times \mathcal{M}_2(D) \rightarrow \mathcal{M}_2(D)$ est défini par

$$x \bullet y = \{w \,/\, w \sqsubseteq^{\flat} u \bullet v \;,\; u \in x \;\&\; v \in y\}$$

En faisant appel à l'isomorphisme de la proposition 4.1.2, on peut dire que $\mathcal{M}_2(D)$ est la complétion par cônes inférieurs de $(\mathcal{KP}(D)^{\odot}, <)$. Ce résultat permet de simplifier les définitions précédentes ainsi que les preuves qui viendront. On a,

$$\begin{array}{ll} i_D(x) = X & \text{où } X = \{e \,/\, e \in \mathcal{KP}(D) \;\&\; e \sqsubseteq x\} \\ j_D(x) = x \cap \mathcal{KP}(D) & \\ x \bullet y = x \cup y \cup \{a{\cdot}b \,/\, a \in x \;\&\; b \in y\} & \end{array}$$

Les transformations i_D et j_D seront souvent notés i et j. Remarquons que le produit est commutatif, associatif et que l'ensemble vide fonctionne comme élément neutre.

Proposition 4.3.2 *1. La fonction j est inverse à gauche de i. C'est-à-dire, pour tout $x \in D$ on a $(j \circ i)(x) = x$.*

2. Pour tout $x, y \in \mathcal{M}_2(D)$ on a $j(x \bullet y) = j(x) \sqcup j(y)$.

3. Le produit distribue sur la borne supérieure (union); i.e. pour tout $x, y, z \in \mathcal{M}_2(D)$ on a $(x \cup y) \bullet z = (x \bullet z) \cup (y \bullet z)$.

Preuve. On montre les énoncés pour la représentation de $\mathcal{M}_2$ comme domaine de cônes.

1. Par définition, $i(x)$ est un ensemble d'éléments premiers quelque soit $x \in D$. Donc $j(i(x)) = i(x) \cap \mathcal{KP}(D) = i(x)$.

2. Soit $x, y \in \mathcal{M}_2(D)$. On a $(x \bullet y) \cap \mathcal{KP}(D) = (x \cup y) \cap \mathcal{KP}(D)$, car les multi-ensembles de $x \bullet y$ ne sont pas dans D. Donc $j(x \bullet y) = (x \cap \mathcal{KP}(D)) \cup (y \cap \mathcal{KP}(D)) = j(x) \cup j(y)$.

3. Soit $x, y, z \in \mathcal{M}_2(D)$; on a $(x \cup y) \bullet z = x \cup y \cup z \cup \{a{\cdot}b \,/\, a \in x \;\&\; b \in z\} \cup \{a{\cdot}b \,/\, a \in y \;\&\; b \in z\} = (x \bullet z) \cup (y \bullet z)$.

□

4.3.2 Universalité de la construction $\mathcal{M}_2$

Définition 4.3.3 ***(Domaines et b-domaines)***
On note $\mathcal{D}$ la catégorie des treillis complets p-algébriques, désormais appelés domaines.

Un b-domaine *est une paire $(E, *)$ telle que*

- *E est un domaine avec élément minimum $\perp$.*
- *$*$ est une fonction binaire continue de $E \times E$ dans E qui satisfait :*

$$\begin{array}{ll} \textit{(assoc)} & (r{*}s){*}t = r{*}(s{*}t) \\ \\ \textit{(commut)} & r{*}s = s{*}r \\ \\ \textit{(neutre)} & s{*}\perp = s \end{array}$$

On note $\mathcal{BD}$ la catégorie des b-domaines et des homomorphismes continus.

Proposition 4.3.4 *Soit $D \in \mathcal{D}$. La construction $(\mathcal{M}_2(D), \bullet)$ est un b-domaine.*

Lemme 4.3.5 *L'opération $\mathcal{M}_2$ définit un foncteur de $\mathcal{D}$ dans $\mathcal{BD}$.*

Preuve. L'opération $\mathcal{M}_2$ transforme les objets de $\mathcal{D}$ en des treillis de cônes. Son effet sur les flèches $f : A \to B$ dans $\mathcal{D}$, i.e. $\mathcal{M}_2(f) : \mathcal{M}_2(A) \to \mathcal{M}_2(B)$, est défini de la façon suivante :

$$s \in \mathcal{M}_2(A) \;\Rightarrow\; \mathcal{M}_2(f)(s) = \bigcup_{e_1 \cdot \ldots \cdot e_n \in s} i_B(f(e_1)) \bullet \ldots \bullet i_B(f(e_n))$$

L'énoncé découle des deux propriétés suivantes :

- Pour tout $A \in \mathcal{D}$, on a $\mathcal{M}_2(\mathbf{I}_A) = \mathbf{I}_{\mathcal{M}_2(A)}$: soit $s \in \mathcal{M}_2(A)$; alors

  $$\begin{array}{rcl} \mathcal{M}_2(\mathbf{I}_A)(s) & = & \{d \,/\, d \in i_A(e_1) \bullet \ldots \bullet i_A(e_n) \;\&\; e_1 \cdot \ldots \cdot e_n \in s\} \\ & = & \{d \,/\, d \lessdot e_1 \cdot \ldots \cdot e_n \;\&\; e_1 \cdot \ldots \cdot e_n \in s\} \end{array}$$

  L'inclusion $s \subseteq \mathcal{M}_2(\mathbf{I}_A)(s)$ est évidente. D'autre part, $d \in \mathcal{M}_2(\mathbf{I}_A)(s)$ implique $d \in s$ car s est un ensemble clos vers le bas.

- Pour tout $g : A \to B\,,\ f : B \to C\ \in \mathcal{D}$, on a $\mathcal{M}_2(f \circ g) = \mathcal{M}_2(f) \circ \mathcal{M}_2(g)$: soit $s \in \mathcal{M}_2(A)$; alors $\forall a = a_1 \cdot \ldots \cdot a_l\ \exists e_1 \ldots e_n\ \exists d_1 \ldots d_k$

  $$\begin{array}{rcl} a \in (\mathcal{M}_2(f) \circ \mathcal{M}_2(g))(s) & \Leftrightarrow & a \in i_C(f(d_1)) \bullet \ldots \bullet i_C(f(d_k)) \\ & & \&\; d_1 \cdot \ldots \cdot d_k \in i_B(g(e_1)) \bullet \ldots \bullet i_B(g(e_n)) \\ & & \&\; e_1 \cdot \ldots \cdot e_n \in s \end{array}$$

  Autrement dit, $a \in (\mathcal{M}_2(f) \circ \mathcal{M}_2(g))(s)$ ssi $l \leq k$, $k \leq n$ et il existe deux injections $h : [1, l] \to [1, k]$ et $h' : [1, k] \to [1, n]$ qui vérifient :

  - $\forall 1 \leq j \leq l\ \ a_j \sqsubseteq f(d_{h(j)})\ \&\ a_j \in \mathcal{KP}(C)$ et

- $\forall 1 \leq p \leq k \;\; d_p \sqsubseteq g(e_{h'(p)})$ & $d_p \in \mathcal{KP}(B)$

On remarquera que $\forall c \in \mathcal{KP}(C)\ \forall e \in \mathcal{KP}(A)$ on a

$$\begin{array}{rcl} c \sqsubseteq f(g(e)) & \Leftrightarrow & c \sqsubseteq \bigsqcup\{f(d) \,/\, d \in \mathcal{KP}(B) \;\&\; d \sqsubseteq g(e)\} \\ & \Leftrightarrow & \exists d \in \mathcal{KP}(B)\;\; c \sqsubseteq f(d) \;\&\; d \sqsubseteq g(e) \end{array}$$

D'où

$$a_1 \cdot \ldots \cdot a_l \in (\mathcal{M}_2(f) \circ \mathcal{M}_2(g))(s) \Leftrightarrow$$

$$\forall 1 \leq j \leq l \;\; a_j \sqsubseteq f(g(e_{h'(h(j))})) \Leftrightarrow a \in \mathcal{M}_2(f \circ g)(s)$$

□

Lemme 4.3.6 *Soit $\mathcal{U} : \mathcal{BD} \to \mathcal{D}$ le foncteur d'oubli. La fonction continue i définit une transformation naturelle entre $\boldsymbol{I}_{\mathcal{D}}$ et $\mathcal{U} \circ \mathcal{M}_2$.*

Preuve. Il faut montrer que $\forall A, B \in \mathcal{D}\ \forall f : A \to B$ le diagramme suivant commute :

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "i_A"'] & B \arrow[d, "i_B"] \\ \mathcal{U}\circ\mathcal{M}_2(A) \arrow[r, "\mathcal{U}\circ\mathcal{M}_2(f)"] & \mathcal{U}\circ\mathcal{M}_2(B) \end{tikzcd}$$

Ce qui revient à prouver $\mathcal{M}_2(f)(i_A(x)) = i_B(f(x))$ pour tout $x \in A$. Nous observons que $e_1 \cdot \ldots e_n \in i_A(x)$ ssi $n = 1$. Alors

$$\mathcal{M}_2(f)(i_A(x)) = \{d \,/\, d \in i_B(f(e)) \;\&\; e \in i_A(x)\}$$

On a donc $i_B(f(e))$

$$\mathcal{M}_2(f)(i_A(x)) = \{d \in \mathcal{KP}(B) \,/\, d \sqsubseteq f(e) \;\&\; e \in \mathcal{KP}(A) \;\&\; e \sqsubseteq x\}$$

De plus $d \sqsubseteq f(x)$ ssi $\exists e \in \mathcal{KP}(A)\ e \sqsubseteq x$ & $d \sqsubseteq f(e)$, car A et B sont des treillis p-algébriques. Par conséquent,

$$i_B(f(x)) = \{d \in \mathcal{KP}(B) \,/\, d \sqsubseteq f(x)\} = \mathcal{M}_2(f)(i_A(x))$$

□

Théorème 4.3.7 ***(Universalité de la construction $\mathcal{M}_2$)***

La construction $\mathcal{M}_2$ est adjointe à gauche du foncteur d'oubli de $\mathcal{BD}$ dans $\mathcal{D}$, avec i comme unité. C'est-à-dire, étant donnés un domaine D et un b-domaine E, pour toute fonction continue $f : D \to \mathcal{U}(E)$ il existe un unique homomorphisme f' dans $\mathcal{BD}$ tel que le diagramme suivant commute :

$$\begin{tikzcd}
D \arrow[d, "i_D"'] \arrow[dr, "f"] & & \mathcal{M}_2(D) \arrow[r, "!f'"] & E \\
\mathcal{U}\circ\mathcal{M}_2(D) \arrow[r, "\mathcal{U}(f')"] & \mathcal{U}(E) & &
\end{tikzcd}$$

Preuve. Il suffit de trouver une fonction monotone de $\mathcal{K}(\mathcal{M}_2(D))$ dans E. Rappelons que les éléments compacts et premiers de $\mathcal{M}_2(D)$ sont les cônes principaux $\downarrow p$ où $p \in \mathcal{KP}(D)$, pour l'ordre $<$. Les éléments compacts de $\mathcal{M}_2(D)$ sont les unions finies de la forme $\bigcup \downarrow p_i$ avec $p_i \in \mathcal{KP}(D)$.

Soit $u = \{p_1, \dots, p_n\}$ où $p_i \in \mathcal{KP}_\perp(D)^\odot$. L'union $\bigcup\downarrow p_i$ est un élément compact de $\mathcal{M}_2(D)$. Si $p = e_1 \cdot \dots e_n$ où $e_i \in \mathcal{KP}(D)$, il est facile de voir que $\downarrow p = i_D(e_1) \bullet \dots \bullet i_D(e_n)$.

Puisque la fonction f' doit être un homomorphisme, sa restriction aux éléments compacts de $\mathcal{M}_2(D)$ doit valider l'équation suivante :

$$f'(\bigcup \downarrow p_i) = \bigsqcup f'(\downarrow p_i) = \bigsqcup f'(i_D(e^i_1)) * \dots * f'(i_D(e^i_{n_i})) = \bigsqcup f(e^i_1) * \dots * f(e^i_{n_i})$$

Si la fonction f' ainsi définie est monotone, son unique extension à tout le b-domaine $\mathcal{M}_2(D)$ est la fonction recherchée. Il faut donc prouver

$$\forall u, v \subseteq \mathcal{KP}(D)^\odot \ \downarrow u \subseteq \downarrow v \ \Rightarrow \ f'(\downarrow u) \sqsubseteq_E f'(\downarrow v)$$

On procède par récurrence sur la taille de u. Lorsque u est l'ensemble vide, l'énoncé est évident. Pour le cas de récurrence, soit $u = \{p, p_1, \dots, p_r\}$ et $v = \{q_1, \dots, q_s\}$ où $r, s \geq 0$. L'hypothèse $\downarrow u \subseteq \downarrow v$ est équivalente à $\bigcup\downarrow p_i \subseteq \bigcup\downarrow q_j$. Alors, $\downarrow p \subseteq \downarrow v$ et $\downarrow\{p_1, \dots, p_r\} \subseteq \downarrow v$.

L'hypothèse de récurrence donne $f'(\downarrow\{p_1, \dots, p_r\}) \sqsubseteq_E f'(\downarrow v)$.

Soit $p = e_1 \cdot \dots \cdot e_n$; la preuve de $f'(\downarrow p) \sqsubseteq f'(\downarrow v)$, d'où $f'(\downarrow u) \sqsubseteq_E f'(\downarrow v)$ par les propriétés des treillis, est par récurrence sur n, le nombre d'éléments de p. Il faut noter que $\downarrow p \subseteq \downarrow v$ ssi il existe q dans v tel que $p<q$. Supposons que ce q soit de la forme $d_1 \cdot \dots \cdot d_m$:

- si $n = 1$, alors $\exists d_k \ p \sqsubseteq d_k$ implique $f(p) \sqsubseteq_E f(d_k)$ par la monotonie de f. Comme

$$p < \underbrace{\perp_E \cdot \dots \cdot \perp_E}_{k-1 \text{ fois}} p \cdot \underbrace{\perp_E \cdot \dots \cdot \perp_E}_{m-k-1 \text{ fois}} \ \text{ et } \perp_E \sqsubseteq_E f(d_j) \ \forall j \in [1, m]$$

on obtient $f(p) \sqsubseteq_E f(d_1) * \ldots f(d_k) \ldots * f(d_m)$, ce qui implique $f(p) \sqsubseteq_E f'(\downarrow v)$.

- dans le cas où $n > 1$, supposons sans perte de généralité que

$$e_1 \sqsubseteq d_1 \quad \text{et} \quad e_2 \cdot \ldots \cdot e_n \lessdot d_2 \cdot \ldots \cdot d_m$$

  Par hypothèse de récurrence $f(e_2) * \ldots * f(e_n) \sqsubseteq_E f(d_2) * \ldots * f(d_m)$. De plus, $f(e_1) \sqsubseteq_E f(d_1)$ par la monotonie de f. Donc

$$f(e_1) * \ldots * f(e_n) \sqsubseteq_E f(d_1) * \ldots * f(d_m) \;\Rightarrow\; f'(\downarrow v)$$

□

Remarquons que dans la preuve du théorème 4.3.7, l'existence de l'élément neutre $\perp$ est essentielle. Son rôle est similaire à celui de l'idempotence du produit dans les domaines des parties classiques [37].

4.3.3 La fonction d'interprétation associée à $\mathcal{E}_2$

Nous gardons la notation utilisée dans la section précédente pour la fonction d'interprétation sur les termes, $\mathcal{V}[\![M]\!]_\rho$, et introduisons la notation $\mathcal{W}[\![P]\!]_\rho$ pour l'interprétation des paquets. Les environnements sont ici des fonctions dans Env $= [Var \to \mathcal{M}_2(D)]$. Par ailleurs, le produit $\bullet$ du domaine $\mathcal{M}_2(D)$ est étendu aux environnements (arbitraires) comme suit :

$$\forall \rho_0, \rho_1 \in \text{Env} \;\; (\rho_0 \bullet \rho_1)(x) = \rho_0(x) \bullet \rho_1(x)$$

Dans la définition de l'interprétation nous avons considéré une syntaxe où les substitutions explicites peuvent être associées à des paquets de termes, i.e. $Q\langle P/x\rangle$ appartient au langage des paquets. Cette modification est motivée par le modèle de l'équation $\mathcal{E}_2$ que nous allons construire. Nous conservons la relation d'évaluation et la sémantique opérationnelle inchangées.

$$\begin{array}{lcl}
\mathcal{V}[\![x]\!]_\rho & = & j(\rho(x)) \\
\mathcal{V}[\![\lambda x.M]\!]_\rho & = & up(\lambda v \in \mathcal{M}_2(D).\mathcal{V}[\![M]\!]_{\rho[x:=v]}) \\
\mathcal{V}[\![MP]\!]_\rho & = & \bigsqcup\limits_{\rho \sqsupseteq \rho_0 \cdot \rho_1} \mathcal{V}[\![M]\!]_{\rho_0}((\mathcal{W}[\![P]\!]_{\rho_1}))
\end{array}$$

$$\begin{array}{lcl}
\mathcal{W}[\![\mathbf{1}]\!]_\rho & = & \perp \\
\mathcal{W}[\![M]\!]_\rho & = & i(\mathcal{V}[\![M]\!]_\rho) \\
\mathcal{W}[\![Q\langle P/x\rangle]\!]_\rho & = & \bigsqcup\limits_{\rho \sqsupseteq \rho'_0 \cdot \rho_1} \mathcal{W}[\![Q]\!]_{\rho_0[x:=\mathcal{W}[\![P]\!]_{\rho_1}]} \qquad \rho_0 = \rho'_0[x := f]\;\&\; x \notin \rho'_0 \\
\mathcal{W}[\![(P|Q)]\!]_\rho & = & \bigsqcup\limits_{\rho \sqsupseteq \rho_0 \cdot \rho_1} \mathcal{W}[\![P]\!]_{\rho_0} \bullet \mathcal{W}[\![Q]\!]_{\rho_1} \\
\mathcal{W}[\![M^\infty]\!]_\rho & = & \bigsqcup\limits_{m \in \mathbb{N}} \mathcal{W}[\![M^m]\!]_\rho
\end{array}$$

L'interprétation des λ_r-termes, définie de façon compositionnelle, est une extension de l'interprétation habituelle du le λ-calcul. En effet, dans le λ-calcul tous les arguments ont des multiplicités infinies, ce qui implique $\rho = \rho\bullet\rho$ et aussi $\mathcal{W}[\![P]\!]_\rho = \mathcal{W}[\![P]\!]_\rho\bullet\mathcal{W}[\![P]\!]_\rho$. Par conséquent, les bornes supérieures utilisées, $\sqcup f$, sont simplement f.

4.3.4 Domaine $\mathcal{F}$ de filtres

La syntaxe des types incorpore la conjonction standard aux types de la théorie du modèle $\mathcal{C}$. Nous conservons le nom Ft pour les types des termes, et Fb pour les types des paquets. La théorie est définie par :

$$(\mathrm{Ft})\quad \phi \quad ::= \quad \omega \mid \pi \to \phi \mid (\phi\wedge\phi)$$

$$(\mathrm{Fb})\quad \pi \quad ::= \quad \phi \mid (\pi\wedge\pi) \mid (\pi\times\pi)$$

La relation de conséquence entre types $\leq$ est la plus petite relation reflexive et transitive qui satisfait les règles (1)-(8) de la théorie pour $\mathcal{C}$, définies dans la section 4.2.3, plus les lois usuelles de la conjonction et la distributivité du produit par rapport à la conjonction:

$$(9)\ (\phi\wedge\sigma) \leq \phi$$

$$(10)\ (\phi\wedge\sigma) \leq \sigma$$

$$(11)\ \phi \leq \sigma_0 \,\&\, \phi \leq \sigma_1 \;\Rightarrow\; \phi \leq (\sigma_0\wedge\sigma_1)$$

$$(12)\ (\pi \to \phi_0)\wedge(\pi \to \phi_1) \leq \pi \to (\phi_0\wedge\phi_1)$$

$$(13)\ (\pi_0 \times \pi)\wedge(\pi_1 \times \pi) \leq (\pi_0\wedge\pi_1) \times \pi$$

Définition 4.3.8 ***(Formules premières)***
Les formules premières *de la théorie* $(\mathrm{F}t, \leq)$ *(resp.* $(\mathrm{F}b, \leq)$*) sont les formules* $\phi \in \mathrm{F}\tilde{t}$ *(resp.* $\pi \in \mathrm{F}\tilde{b}$ *) où* $\mathrm{F}\tilde{t}$ *et* $\mathrm{F}\tilde{b}$ *sont définis par :*

$$\begin{array}{lcl} (\mathrm{F}\tilde{t})\quad \tilde{\zeta} & ::= & \gamma \mid \pi \to \tilde{\zeta} \\ (\mathrm{F}\tilde{b})\quad \tilde{\pi} & ::= & \tilde{\zeta} \mid (\tilde{\pi}\times\tilde{\pi}) \end{array}$$

où γ *signifie* $\omega \to \omega$.

Proposition 4.3.9 ***(Description normalisée des types)***

$$\forall \pi \in \mathrm{F}b \;\; \exists \psi_1 \ldots \psi_n \in \mathrm{F}\tilde{b} \quad \pi \sim \wedge_I \psi_i$$

Preuve. Par récurrence sur la taille du type π. L'énoncé est vérifié grâce aux règles (12) et (13) de distribution des constructeurs. $\square$

Nous montrons maintenant une propriété de la flèche pour la nouvelle théorie. Soit C^i_α le contexte de types à plusieurs trous, et t_α la séquence de types définis comme suit:

$$\begin{array}{lll}
C^i_\omega = \omega & & \\[1em]
C^i_{\pi\rightarrow\phi} = []_i & t_{\pi\rightarrow\phi} = \pi \rightarrow \phi & \\[1em]
C^i_{\pi\wedge\psi} = C^i_\pi \wedge C^{i+m}_\psi & t_{\pi\wedge\psi} = t_\pi, t_\psi & \text{si } t_\psi = t_1, \ldots, t_m \\[1em]
C^i_{\pi\times\psi} = C^i_\pi \times C^{i+m}_\psi & t_{\pi\times\psi} = t_\pi, t_\psi & \text{si } t_\psi = t_1, \ldots, t_m
\end{array}$$

On appelle C_α le contexte C^1_α. Il est clair que $\alpha = C_\alpha[t_\alpha]$.

Lemme 4.3.10 *Soit* $t_\alpha = (a_1 \rightarrow b_1), \ldots, (a_n \rightarrow b_n)$ *et* $t_\beta = (c_1 \rightarrow d_1), \ldots, (c_m \rightarrow d_m)$.

$$\begin{array}{ll}
\alpha \le \beta \Rightarrow & (\, m > 0 \Rightarrow n > 0 \,\&\, \forall l \in \{1, \ldots, m\}\, d_l \not\sim \omega \Rightarrow \\
& \exists J_l = \{i \,/\, c_l \le a_i\} \neq \emptyset \,\&\, \bigwedge_{J_l} b_i \le d_l \,)
\end{array}$$

Preuve. Par récurrence sur la dérivation de $\alpha \le \beta$, on examine la dernière règle utilisée.

(1) Ici $\alpha \le \omega = \beta$. Donc $m = 0$.
(2) Ici $\alpha = \pi \rightarrow \omega \le \omega \rightarrow \omega$. Donc $m > 0$ et $n > 0$.
(3) Ici $\alpha = a_1 \rightarrow b_1 \le c_1 \rightarrow d_1 = \beta$ avec $c_1 \le a_1$ et $b_1 \le d_1$. L'énoncé est donc évident.
(4) Ici $\alpha = \omega \wedge \beta \le \beta$. Alors, pour tout l t.q. $d_l \not\sim \omega$, il existe i vérifiant $(c_l \rightarrow d_l) = (a_i \rightarrow b_i)$.
(5)-(7),(9)-(10), (13) Même raisonnement que pour la règle numéro 4.
(8) Ici $\alpha = \alpha_0 \times \alpha_1 \le \beta_0 \times \beta_1$ où $\alpha_0 \le \beta_0$ et $\alpha_1 \le \beta_1$. Par conséquent, $\exists j, k$ $1 \le j \le n$ & $1 \le k \le m$

$$t_{\alpha_0} = (a_1 \rightarrow b_1), \ldots, (a_j \rightarrow b_j) \quad \text{et} \quad t_{\beta_0} = (c_1 \rightarrow d_1), \ldots, (c_k \rightarrow d_k)$$

$$t_{\alpha_1} = (a_{j+1} \rightarrow b_{j+1}), \ldots, (a_n \rightarrow b_n) \quad \text{et} \quad t_{\beta_1} = (c_{k+1} \rightarrow d_{k+1}), \ldots, (c_m \rightarrow d_m)$$

L'énoncé est vérifié directement par hypothèse de récurrence.
(11) Comme pour la règle (8).
(12) Ici $\alpha = (a \rightarrow b_1) \wedge (a \rightarrow b_2) \le a \rightarrow (b_1 \wedge b_2) = \beta$, l'énoncé découle de $\beta_1 \wedge \beta_2 \not\sim \omega \Rightarrow \beta_1 \not\sim \omega$ ou $\beta_2 \not\sim \omega$ et de $a \wedge a \le a$.

(transitivité) Dans le cas où $\alpha \leq \delta \leq \beta$ avec $\delta = C_\delta[(e_1 \to f_1), \ldots, (e_k \to f_k)]$ et $m > 0$ on a, par h.r., $k > 0$ et

$$\forall l \in [1, m]\ d_l \not\sim \omega \Rightarrow \exists K_l = \{j \,/\, c_l \leq e_j\} \neq \emptyset \,\&\, \bigwedge_{K_l} f_j \leq d_l$$

Soit $K'_l = \{j \,/ j \in K_l \;\&\; f_j \not\sim \omega\}$. Cet ensemble est non-vide parce que

$$\bigwedge_{K_l} f_j \leq d_l \not\sim \omega \Rightarrow \exists j \in K_l\ f_j \not\sim \omega$$

De l'hypothèse de récurrence appliquée sur $\alpha \leq \delta$ on déduit $n > 0$ et aussi

$$\forall j \in K'_l\ \exists I_j = \{i \,/\, e_j \leq a_i\} \neq \emptyset\ \bigwedge_{I_j} b_i \leq f_j$$

Par conséquent, $\forall l \in [1, m]\ \wedge_{K'_l}(\wedge_{I_j} b_i) \leq d_l$ est vérifié. Donc $J_l = \bigcup_{K'_l} I_j \neq \emptyset$ et $\wedge_{K'_l}(\wedge_{I_j} b_i) = \wedge_{J_l} b_i$. $\square$

La propriété usuelle de la flèche dans les théories de types avec intersection pour des calculs faibles est une conséquence immédiate du lemme 4.3.10. Pour une discussion sur l'importance de ce genre de loi dans la construction des modèles des filtres pour le lambda calcul fort cf. [22].

Corollaire 4.3.11 ***(Propriété de la flèche - $\mathcal{E}_2$)***
Soit I un ensemble fini.

$$\bigwedge_I (a_i \to b_i) \leq c \to d \Rightarrow I \neq \emptyset \;\&\; (d \not\sim \omega \Rightarrow \exists J = \{i \,/\, c \leq a_i\} \neq \emptyset \,\&\, \bigwedge_J b_i \leq d)$$

Définition 4.3.12 *Un ensemble $f \subseteq \mathrm{F}t$ (resp. $f \subseteq \mathrm{F}b$) est un* filtre *de* $(\mathrm{F}t\, , \leq)$ *(resp. $(\mathrm{F}b\, , \leq)$) si et seulement si f satisfait les trois clauses suivantes:*

- $\omega \in f$
- $\sigma, \tau \in f \Rightarrow (\sigma \wedge \tau) \in f$
- $\tau \leq \sigma \,\&\, \tau \in f \Rightarrow \sigma \in f$

Soit $\mathcal{F}$ l'ensemble de filtres de $\mathrm{F}t$ et $\mathcal{F}_b$ l'ensemble de filtres de $\mathrm{F}b$. Étant donné A un sous-ensemble non-vide de $\mathrm{F}t$ ou de $\mathrm{F}b$, on dénote par $\uparrow A$ le plus petit filtre de $\mathrm{F}t$ ou de $\mathrm{F}b$ qui contient A. Pour les singletons $\{a\}$ la notation devient $\uparrow a$.

Fait 4.3.13 *Les ensembles ordonnés $(\mathcal{F}, \subseteq)$ et $(\mathcal{F}_b\, , \subseteq)$ sont des treillis complets p-algébriques où $\uparrow\omega$ est le plus petit élément, $x \sqcup y = \uparrow(x \cup y)$ et $x \sqcap y = x \cap y$. Leurs plus grand élément sont $\mathrm{F}t$ et $\mathrm{F}b$ respectivement.*

Lemme 4.3.14

1. *Si* $A \subseteq \mathcal{F}$ *(*$A \subseteq \mathcal{F}_b$ *) est un ensemble co-dirigé, alors* $\bigsqcup A = \bigcup A$*.*

2. *Le filtre* $\uparrow(\phi \wedge \sigma)$ *est la borne supérieure de* $\uparrow\phi$ *et* $\uparrow\sigma$*; i.e.* $\uparrow\phi \sqcup \uparrow\sigma = \uparrow(\phi \wedge \sigma)$*.*

Preuve. Les énoncés qui composent la proposition sont standards. Nous montrons (2) à titre d'exemple. Il est clair que $\uparrow(\phi \wedge \sigma)$ est une borne supérieure de $\uparrow\phi$ et de $\uparrow\sigma$. Supposons qu'il existe un filtre d t.q.

$$\uparrow\phi \subseteq d \quad \uparrow\sigma \subseteq d \quad d \subseteq \uparrow(\phi \wedge \sigma)$$

On vérifie $\uparrow(\phi \wedge \sigma) \subseteq d$: soit $\delta \in \uparrow(\phi \wedge \sigma)$; alors $\phi \wedge \sigma \leq \delta$. D'autre part, $\phi \in d$ et $\sigma \in d$ impliquent $\phi \wedge \sigma \in d$ par définition de filtre, d'où $\delta \in d$. □

Proposition 4.3.15

1. *Les éléments compacts de* $\mathcal{F}$ *sont ses filtres principaux* $\uparrow\phi$ *sur* $\mathrm{F}t$ *. Les éléments compacts de* $\mathcal{F}_b$ *sont ses filtres principaux* $\uparrow\pi$ *sur* $\mathrm{F}b$ *.*

2. *Pour tout* $X \subseteq \mathcal{F}$*, si* $\uparrow\phi \subseteq \bigsqcup X$ *alors*

$$\exists x_1 \ldots x_n \in X \ \exists \phi_1 \ldots \phi_n \ (\forall i \ \phi_i \in x_i) \ \& \ \bigwedge_I \phi_i \leq \phi$$

3. *Les éléments à la fois compacts et premiers de* $\mathcal{F}$ *(resp.* $\mathcal{F}_b$ *) sont ses filtres principaux sur* $\mathrm{F}\tilde{t}$ *(resp.* $\mathrm{F}\tilde{b}$ *).*

Preuve.

1. Il est facile de voir que pour tout ensemble X de filtres, $\uparrow\phi \sqsubseteq \bigsqcup X \Rightarrow \exists x \in X \uparrow\phi \sqsubseteq x$. Nous montrons ensuite l'implication inverse: c compact $\Rightarrow \exists\phi \ c = \uparrow\phi$.

   Soit $X = \{\uparrow\sigma \,/\, \sigma \in c\}$. Puisque $c \sqsubseteq \bigsqcup X$, par hypothèse il existe $Y = \{\uparrow\sigma_1, \ldots, \uparrow\sigma_n\} \subseteq X$ t.q. $c \subseteq Y$. C'est-à-dire, $c \subseteq \uparrow\wedge_I \sigma_i$. D'autre part, si $\phi \in \uparrow\wedge_I \sigma_i$ alors $\wedge_I \sigma_i \leq \phi$, d'où $\phi \in c$, car $\forall i \ \sigma_i \in c \Rightarrow \wedge_I \sigma_i \in c$ par la définition de filtre.

2. Chaque élément x de X s'écrit $x = \bigsqcup\{c \,/\, c \text{ est compact } \& \ c \subseteq x\}$. Donc $\uparrow\phi \subseteq \bigsqcup Z$ où $Z = \{c \,/\, c \text{ est compact } \& \ \exists x \in X \ c \subseteq x\}$. Puisque $\uparrow\phi$ est compact, par le lemme 4.3.14(2) on a $\exists Y = \{c_1, \ldots, c_n\} \subseteq Z$ t.q.

$$c_i = \uparrow\phi_i \Rightarrow \uparrow\phi \subseteq \uparrow\bigwedge_I \phi_i$$

   D'où $\wedge_I \phi_i \leq \phi$.

3. Soit $\uparrow\psi \in \mathcal{KP}(\mathrm{Fb})$. Supposons $\psi \sim \wedge_I \psi_i$ avec $\psi_i \in \mathrm{F}\tilde{\mathrm{b}}$. Alors $\uparrow\psi \subseteq \sqcup\uparrow\psi_i$. Puisque $\uparrow\psi$ est premier, il existe $i \in I$ t.q. $\uparrow\psi \subseteq \uparrow\psi_i$. Donc $\psi_i \leq \psi \sim \wedge_I \psi_i \leq \psi_i$ implique $\psi \sim \psi_i$. On conclut $\psi \in \mathrm{F}\tilde{\mathrm{b}}$.

   Supposons maintenant $\psi \in \mathrm{F}\tilde{\mathrm{b}}$ et $\uparrow\psi \notin \mathcal{KP}(\mathcal{F}_{\mathrm{b}})$. Puisque de $\uparrow\psi$ est compact, $\uparrow\psi = \sqcup\uparrow\psi_i$ où $\psi_i \in cP\mathcal{F}_{\mathrm{b}}$ et le nombre d'éléments de Idépasse 1. Alors $\uparrow\psi = \uparrow\wedge_I\psi_i$. C'est-à-dire, $\psi \sim \wedge_I\psi_i$, ce qui contredit l'hypothèse $\psi \in \mathrm{F}\tilde{\mathrm{b}}$.

□

Proposition 4.3.16 ***(Propriété du produit - $\mathcal{E}_2$)***
Soit $\psi, \pi, \pi_1, \ldots, \pi_n \in \mathrm{F}\tilde{b}$ et $\phi_1, \ldots \phi_m \in \mathrm{F}\tilde{t}$. Alors

1. $\bigwedge\limits_I \pi_i \leq \psi \Rightarrow \exists i \in I \ \pi_i \leq \psi$

2. $\pi \sim \sigma_1 \times \ldots \times \sigma_p \ \& \ \pi \leq \phi_1 \times \ldots \times \phi_m \Rightarrow$

$$\exists \textit{ injection } g : [1, m] \to [1, p] \ \forall i \in [1, m] \ \ \sigma_{g(i)} \leq \phi_i$$

Preuve. La partie (1) de l'énoncé est conséquence de la proposition 4.3.15(3), car $\uparrow\psi \in \mathcal{KP}(\mathcal{F}_{\mathrm{b}})$. La preuve de la partie (2) est conséquence de la propriété de la flèche 4.3.11 et de l'observation suivante: si $\sigma \times \pi' \leq \phi \times \psi'$ et $\sigma \leq \phi$, alors $\pi' \leq \psi'$. □

On procède comme pour l'équation $\mathcal{E}_1$, lemme 4.2.4, pour prouver l'isomorphisme

$$\mathcal{F} \sim (\mathcal{M}_2(\mathcal{F}) \to \mathcal{F})_\perp$$

Lemme 4.3.17 *$\mathcal{F}_b$ et $\mathcal{M}_2(\mathcal{F})$ sont des treillis algébriques isomorphes.*

Preuve. On se limite à montrer qu'il y a un isomorphisme entre les éléments compacts de ces treillis. Par définition, $\mathcal{M}_2(\mathcal{F}) = \mathcal{Q}(\mathcal{KP}(\mathcal{F})^{\odot}, <)$. Nous utilisons ici sa présentation comme domaine de cônes sur l'ensemble ordonné $(\mathcal{KP}(\mathcal{F})^{\odot}, <)$. Les ensembles qui nous intéressent sont

$$\begin{array}{rcl} \mathcal{K}(\mathcal{F}_{\mathrm{b}}) & = & \{\uparrow\pi \,/\, \pi \in \mathrm{Fb}\} \\ \mathcal{K}(\mathcal{M}_2(\mathcal{F})) & = & \{\bigcup\limits_I \downarrow u \,/\, I \text{ est fini et } u \in \mathcal{KP}(\mathcal{F})^{\odot}\} \\ & \text{où} & \mathcal{KP}(\mathcal{F}) = \{\uparrow\phi \,/\, \phi \in \mathrm{F}\tilde{\mathrm{t}}\} \end{array}$$

On définit $\kappa : \mathrm{F}\tilde{\mathrm{b}} \to \mathbf{Fin}(\mathcal{KP}(\mathcal{F})^{\odot})$ et $\kappa' : \mathrm{Fb} \to \mathbf{Fin}(\mathcal{KP}(\mathcal{F})^{\odot})$ comme suit:

$$\begin{array}{ll} \kappa(\phi) = \{\uparrow\phi\} & \text{si } \phi \in \text{F}\tilde{\text{t}} \\ \kappa(\pi \times \psi) = \kappa(\pi)\bullet\kappa(\psi) & \\ & \\ \kappa'(\omega) = \emptyset & \\ \kappa'(\pi \to \phi) = \bigcup_I \uparrow\pi \to \phi_i & \text{si } \phi \sim \bigwedge_I \phi_i \ \text{ et } \ \forall i\, \phi_i \in \text{F}\tilde{\text{t}} \\ \kappa'(\pi \times \pi') = \kappa'(\pi)\bullet\kappa'(\pi') & \\ \kappa'(\pi\wedge\pi') = \kappa'(\pi) \cup \kappa'(\pi') & \end{array}$$

Nous montrons quatre énoncés, à savoir :

1. $\forall\psi\,\exists\psi_1,\ldots,\psi_n \in \text{F}\tilde{\text{b}}\ \ \psi \sim \bigwedge\psi_i \ \Rightarrow\ \kappa'(\psi) = \bigcup_I \kappa(\psi_i)$
2. $\forall\pi,\psi \in \text{F}\tilde{\text{b}}\ \ \kappa(\psi)\sqsubseteq^\flat\kappa(\pi) \ \Rightarrow\ \pi \ \leq\ \psi$
3. $\uparrow\pi \in \mathcal{K}(\mathcal{F}_{\text{b}}\ )\Leftrightarrow\kappa'(\pi) \in \mathcal{K}(\mathcal{M}_2(\mathcal{F}))$
4. $\pi\ \leq\ \psi \Leftrightarrow \kappa'(\psi)\sqsubseteq^\flat\kappa'(\pi)$

L'énoncé du théorème découle des points (3) et (4).

1. On procède par récurrence sur ψ. Si $\psi = \omega$, alors $\kappa'(\psi)$ est une union vide. Si $\psi = \pi \to \phi$ et $\phi\wedge_I\phi_i$ où $\phi_i \in \text{F}\tilde{\text{t}}$, l'énoncé est vérifié par définition de κ. Si $\psi = \pi\wedge\pi'$ alors on utilise l'hypothèse de récurrence. Finalement, si $\psi = \pi \times \pi'$ alors par récurence
$$\kappa'(\psi) = \bigcup_L \kappa(\pi_l)\bullet\bigcup_J \kappa(\pi'_j)$$
avec $\pi \sim \bigwedge_L \pi_l$ et $\pi' \sim \bigwedge_J \pi'_j$. On définit I comme le produit cartésien de L et J, i.e. $I = L \times J$, et $\psi_i = \pi_l \times \pi'_j$ pour $i = (l,j)$. Alors $\psi \sim \wedge_I\psi_i$ et aussi $\kappa'(\psi) = \cup_I\psi_i$.

2. La démonstration de ce point est basée sur l'observation suivante : soit $\pi = \phi_1\times\ldots\times\phi_n \in \text{F}\tilde{\text{b}}$ (c'est-à-dire, les ϕ_i n'ont pas des conjonctions à droite de la flèche) et $d_\pi\ =\ \uparrow\phi_1\cdot\ldots\cdot\uparrow\phi_n$; alors une récurrence sur n permet de montrer $d_\pi \in \kappa(\pi)$ et aussi $e\lessdot d_\pi$ quelque soit $e \in \kappa(\pi)$. C'est-à-dire, d_π est le multi-ensemble maximal de $\kappa(\pi)$. Par conséquent, $\kappa(\psi)\sqsubseteq^\flat\kappa(\pi)\ \Rightarrow\ d_\psi\lessdot d_\pi$. D'où, $\pi\ \leq\ \psi$.

3. Il est facile de montrer l'énoncé restreint à des formules premières, i.e. $\uparrow\pi\ \in\ \mathcal{KP}(\mathcal{F}_{\text{b}}\ )\Leftrightarrow\kappa(\pi)\ \in\ \mathcal{KP}(\mathcal{M}_2(\mathcal{F}))$, en sachant que $\uparrow\phi_1\bullet\ldots\bullet\uparrow\phi_n\ =\ \downarrow\uparrow\phi_1\cdot\ldots\cdot\uparrow\phi_n$, si $\phi_i \in \text{F}\tilde{\text{t}}$.

La partie $\Rightarrow$ de l'énoncé général est comme suit : soit $\pi \in \mathrm{F\tilde{b}}$; alors $\pi \sim \wedge_I \psi_i$ et, par le point (1), on a $\kappa'(\pi) = \cup \kappa(\psi_i)$. Puisque $\kappa(\psi_i)$ est un filtre premier de $\mathcal{M}_2(\mathcal{F})$, leur union $\kappa'(\pi)$ est un élément compact.

La partie $\Leftarrow$ est immédiate.

4. La partie $\Rightarrow$ est prouvée par récurrence sur la dérivation de $\pi \leq \psi$: dans les cas de base, l'énoncé est vérifié immédiatement. On analyse le cas de récurrence pour quelques règles :

    - Supposons que $\pi \leq \psi$ est montré par la règle 3, i.e. $\pi = \pi_0 \rightarrow \phi_0 \leq \pi_1 \rightarrow \phi_1$ où $\pi_1 \leq \pi_0$ et $\phi_0 \leq \phi_1$. Soit $\phi_0 \sim \wedge_I \sigma_i$ et $\phi_1 \sim \wedge_J \delta_j$ avec σ_i, δ_j des formules premières. Par définition,

      $$\kappa'(\pi) = \bigcup_J \uparrow(\pi_1 \rightarrow \delta_j) \qquad \text{et} \qquad \kappa'(\psi) = \bigcup_I \uparrow(\pi_0 \rightarrow \sigma_i)$$

      On doit montrer $\forall j \in J \ \exists i \in I \ \pi_0 \rightarrow \sigma_i \leq \pi_1 \rightarrow \delta_j$. Or $\wedge_I \sigma_i \sim \phi_0 \leq \phi_1 \leq \delta_j$ et $\delta_j \in \mathrm{F\tilde{t}}$ impliquent $\exists i \ \sigma_i \leq \delta_j$ par la propriété de la flèche 4.3.11.

    - Dans le cas où $\pi = (\theta \rightarrow \phi_0) \wedge (\theta \rightarrow \phi_1) \leq \theta \rightarrow (\phi_0 \wedge \phi_1) = \psi$, on a $\kappa'(\pi) = \kappa'(\theta \rightarrow \phi_0) \cup \kappa'(\theta \rightarrow \phi_1)$. Supposons $\phi_0 \sim \wedge \sigma_i$ et $\phi_1 \sim \wedge \delta_j$; alors

      $$\kappa'(\pi) = \bigcup_I \uparrow \theta \rightarrow \sigma_i \ \cup \ \bigcup_J \uparrow \theta \rightarrow \delta_j$$

      C'est-à-dire, en posant $L = (I \times J)$ (le produit cartésien) et $\varphi_(i, j) = \sigma_i \wedge \delta_j$, on a $\kappa'(\pi) = \cup_L \varphi_l = \kappa'(\psi)$.

    - Dans le cas où $\pi = (\pi_0 \times \theta) \wedge (\pi_1 \times \theta) \leq (\pi_0 \wedge \pi_1) \times \theta = \psi$, on obtient $\kappa'(\pi) = \kappa'(\psi)$, puisque le produit $\bullet$ est distribué par la conjonction.

    Les cas restants sont conséquence directe de l'hypothèse de récurrence.

Nous prouvons maintenant l'implication $\Leftarrow$ de l'énoncé. Soit $\kappa'(\psi) \sqsubseteq^\flat \kappa'(\pi)$. En utilisant la partie (1), si $\pi \sim \wedge_J \pi_j$ et $\psi \sim \wedge_I \psi_i$ avec $\pi_j \in \mathrm{F\tilde{b}}$ et $\psi_i \in \mathrm{F\tilde{b}}$, alors $\cup_I \kappa(\psi_i) \sqsubseteq^\flat \cup_J \kappa(\pi_j)$. On montre

$$\forall i \in I \ \exists j \in J \ \kappa(\psi_i) \sqsubseteq^\flat \kappa(\pi_j)$$

Par construction, $\forall i \in I \ d_{\psi_i} \in \kappa(\psi_i)$. Donc $\exists e \in \cup_J \kappa(\pi_j) \ d_{\psi-i} \lessdot e$ est vérifié en utilisant la propriété (1). C'est-à-dire, il y a un j dans J tel que $e \in \kappa(\pi_j)$. Donc $d_{\psi_i} \lessdot d_{\pi_j}$ aussi. En conclusion, $\forall d \in \kappa(\psi_i) \ \exists j \in J \ d \lessdot d_{\psi_i} \lessdot d_{\pi_j}$. Par le point (2), $\forall i \in I \ \exists j \in J \pi_j \leq \psi_i$. Donc $\wedge_J \pi_j \leq \wedge_I \psi_i$.

□

Définition 4.3.18 ***(Application dans le domaine des filtres $\mathcal{F}$)***

L'application dans le domaine des filtres est la fonction continue $\_\cdot\_ : \mathcal{F} \times \mathcal{F}_b \rightarrow \mathcal{F}$ *définie par*

$$x{\cdot}y = \begin{cases} \uparrow\{\phi \in \mathrm{Ft} \;/\; \exists \pi \in y \,.\, (\pi \rightarrow \phi) \in x\} & si\ x \neq \uparrow\omega \\ \uparrow\omega & sinon \end{cases}$$

Il faut noter que la seule différence avec l'application du domaine de cônes est la nécessité de forcer la construction d'un filtre ici, par l'opération $\uparrow$*.*

Les fonctions F et G qui réalisent l'isomorphisme sont celles de la définition 4.2.9, pourvu que l'on entende le symbole $\uparrow$ comme la construction d'un filtre. De même pour la définition des fonctions seuil. Le domaine qui nous occupe vérifie une proposition analogue à 4.2.6 où les éléments compacts de $(\mathcal{F}_{\mathrm{b}} \rightarrow \mathcal{F})$ sont caractérisés comme des bornes finies de fonctions seuil. Le résultat de représentation des fonctions continues en découle.

Lemme 4.3.19 ***(Lemme de représentation pour $\mathcal{F}$)***

$$\forall h \in (\mathcal{F}_b \rightarrow \mathcal{F})_\perp \; \forall d \in \mathcal{F}_b \quad h(\!(d)\!) = G(h){\cdot}d$$

Preuve. Comme dans le cas du lemme 4.2.11, l'énoncé découle de l'implication

$$G(up(f_{\psi\phi})) = \uparrow(\psi \rightarrow \phi) \;\Rightarrow\; G(up(f_{\psi\phi})){\cdot}\uparrow\pi = f_{\psi\phi}(\uparrow\pi)$$

Nous montrons l'inclusion $\subseteq$ de la prémisse comme suit (la partie $\supseteq$ est trivial) : soit $A = \{\theta \rightarrow \sigma \,/\, \sigma \in f_{\psi\phi}(\uparrow\theta)\}$ et $\tau \in G(up(f_{\psi\phi}))$, i.e. $\tau \in \sqcup_A \uparrow\varphi$. Il existe donc $\delta_1, \ldots, \delta_n$ t.q. $\wedge_I \delta_i \leq \tau$ et $\forall i \; \exists \varphi_i \in A \; \varphi_i \leq \delta_i$. Puisque $\psi \rightarrow \phi \leq \varphi$ pour tout $\varphi \in A$, on a $\psi \rightarrow \phi \leq \wedge_I \delta_i \leq \tau$. D'où $\tau \in \uparrow\psi \rightarrow \phi$. □

Théorème 4.3.20 ***(Solution de l'équation $\mathcal{E}_2$)***

1. *$G \circ F = \boldsymbol{I}_{\mathcal{F}}$*
2. *$F \circ G = \boldsymbol{I}_{(\mathcal{F}_b \rightarrow \mathcal{F})_\perp}$*

Preuve.

1. Soit $x \in \mathcal{F}$; si $x = \uparrow\omega$ alors $G(F(\uparrow\omega)) = G(\perp) = \uparrow\omega$. Si $x \supset \uparrow\omega$, $G(F(x))$ dénote le filtre $\uparrow\{\pi \rightarrow \phi \,/\, \phi \in x{\cdot}\uparrow\pi\}$, ou encore $\uparrow\{\pi \rightarrow \phi \in x\}$. L'inclusion $x \subseteq G(F(x))$ en découle. D'autre part, si $\tau \in G(F(x))$, alors $\wedge_I \tau_i \leq \tau$ pour des types τ_i dans x. Ce qui implique $\tau \in x$ par définition de filtre.

2. En utilisant le résultat de représentation des fonctions, comme dans la preuve du lemme 4.2.11(2).

□

Chapitre 5

Modèle de cônes

Le modèle $\mathcal{C}$ que nous avons construit pour λ_r s'avère adéquat par rapport à la sémantique observationnelle standard du calcul. Cependant, ce n'est pas une adéquation complète; le modèle discrimine strictement plus de termes que les contextes de λ_r (cf. [81, 45] pour le problème d'adéquation complète du modèle D_∞ de Scott, [51, 66, 12] pour PCF, [2, 3, 15, 29] pour des résultats similaires dans le cas du lambda calcul faible, et aussi [78].) Plus exactement, nous montrons dans ce chapitre que le modèle discrimine autant que le calcul de ressources avec test de convergence (opérateur qui manque au calcul λ_r, comme l'indique le contre-exemple 3.6.13.) La preuve de l'adéquation complète de $\mathcal{C}$ pour le langage λ_r^c repose sur la description des dénotations de termes dans $\mathcal{C}$ à travers un système d'affectation de types $\mathcal{P}$ défini sur la théorie ($\texttt{Ft}$, $\leq$). L'ordre sur Λ_{rc} engendré par l'interprétation des termes dans $\mathcal{P}$ coïncide avec le préordre observationnel $\sqsubseteq_{rc}$. Nous utilisons une technique de preuve par *réalisabilité* et *définissabilité des points compacts* (cf. [3, 15, 29]) pour montrer cette correspondance.

Dans la section 5.1 nous donnons une formulation plus explicite de la fonction d'interprétation des termes dans le modèle de cônes. Dans la section 5.2 nous définissons le système d'affectation de types $\mathcal{P}$, raffinement des systèmes de types avec intersection à la Coppo qu'on utilise pour construire des modèles de filtres [9, 17, 29]. La conjonction usuelle est remplacée par un *produit* $\times$ non-idempotent et la contraction d'hypothèses n'est pas autorisée. La définition du système $\mathcal{P}$ repose sur le système d'affectation de types présenté dans [17]. Les nouvelles règles permettent de typer le test de convergence et incorporent la relation de conséquence $\leq$ entre types[1].

La section 5.3 est consacrée à la preuve de la stabilité par expansion du typage dans $\mathcal{P}$: on introduit une relation de réduction $\triangleright$ dans λ_r^c dont la stratégie normalisante est l'évaluation $\rightarrow_{rc}$ et on montre que si $M \triangleright N$ et si N a un type ϕ dans

---

[1] La théorie de types étudiée dans [17] fait intervenir uniquement l'équivalence $\sim$ entre types; la relation de conséquence n'est pas nécessaire lorsqu'il s'agit de montrer l'adéquation. Le rôle de $\leq$ dans la preuve de la complétude de l'adéquation est mis en évidence dans la preuve du lemme de convergence 5.6.3.

un contexte Γ, alors M a aussi ce type, dans le même contexte. La définition de $\triangleright$ rappelle la réduction du calcul λ_d. On montre aussi un résultat d'extensionnalité du système de typage, nécessaire dans la preuve de complétude du système de types.

L'interprétation des termes dans $\mathcal{P}$ est définie dans la section 5.4, où l'on montre que celle-ci coïncide avec l'interprétation dans le modèle de cônes. Les sections suivantes sont consacrées à la preuve d'adéquation complète de l'ordre engendré par l'interprétation dans $\mathcal{P}$, par rapport à $\sqsubseteq_{rc}$. Dans la section 5.5 on montre la caractérisation de la convergence dans le système de types par la technique de réalisabilité dans le style de [2] : on donne une interprétation des types par des ensembles de termes et on montre sa correction par rapport à la notion d'affectation de types. La section 5.6 contient la preuve de l'adéquation complète qui découle de la complétude de l'interprétation des types, et de l'existence de termes caractéristiques pour chaque type. Comme dans les lambda calculs avec fonctions parallèles standards, ces termes ne pourraient pas être définis sans le test de convergence.

Dans la section 5.7 nous montrons que les contextes de λ_r^c (donc de λ_r), sur le λ-calcul, sont aussi puissants que les contextes avec multiplicités. Ce résultat avait été annoncé dans la section 3.8.1.

5.1 Interprétation de λ_r^c dans $\mathcal{C}$

Nous sommes concernés ici par le calcul λ_r avec test de convergence. L'extension de la fonction d'interprétation associée à l'équation aux domaines $\mathcal{E}_1$ est basée sur l'observation que l'équation a été conçue pour que, suivant la conception de Scott, les fonctions divergentes ne pourvoient aucune information, c'est-à-dire, aient $\perp$ comme dénotation. Ainsi,

$$\mathcal{V}[\![\mathrm{c}P]\!]_\rho = \begin{cases} \mathcal{V}[\![\mathbf{I}]\!]_\rho & \text{si } \exists M \;\; P \equiv (M \mid Q) \;\&\; \mathcal{V}[\![M]\!]_\rho \neq \perp \\ \perp & \text{sinon} \end{cases}$$

L'ensemble des équations sémantiques présentées ci-dessous correspond à l'interprétation du calcul dans le modèle logique $\mathcal{C}$. Dans ce cadre, un environnement ξ est compact si et seulement si pour toute variable $x \in \mathcal{D}(\xi)$ il existe $\pi \in \mathtt{Fb}$ tel que $\xi(x) = \uparrow\pi$. Par ailleurs, le produit d'environnements compacts devient

$$\forall i \;\; \rho_i = \uparrow\pi_i \;\Rightarrow\; \rho_1 \cdot \ldots \cdot \rho_n = \uparrow(\pi_1 \times \ldots \times \pi_n)$$

comme corollaire de l'isomorphisme entre $\mathcal{I}$ et $\mathcal{M}_1(\mathcal{C})$ (cf. lemme 4.2.4.) Puisque $\mathcal{C}$ est une solution de l'équation aux domaines à isomorphisme près, les fonctions F et G sont utilisées de la façon usuelle pour définir la sémantique du calcul.

$$\begin{aligned}
\mathcal{V}\llbracket x\rrbracket_\rho &= \bigcup\{\phi\in \textsc{Ft}\ /\ \phi\in\rho(x))\\
\mathcal{V}\llbracket \lambda x.M\rrbracket_\rho &= \uparrow\{\pi\to\phi\,/\,\phi\in\mathcal{V}\llbracket M\rrbracket_{\rho[x:=\uparrow\pi]}\}\\
\mathcal{V}\llbracket MP\rrbracket_\rho &= \bigcup_{\substack{\rho\supseteq\rho_0\cdot\ldots\cdot\rho_n\\ P\propto(M_1\mid\cdots\mid M_n)\\ \phi_i\in\mathcal{V}\llbracket M_i\rrbracket_{\rho_i}}} (\{\sigma\,/\,\exists\pi\in\uparrow\phi_1\times\cdots\times\phi_n\ (\pi\to\sigma)\in\mathcal{V}\llbracket M\rrbracket_{\rho_0}\}\cup\{\omega\})\\
\mathcal{V}\llbracket M\langle P/x\rangle\rrbracket_\rho &= \bigcup_{\substack{\rho\supseteq\rho_0'\cdot\ldots\cdot\rho_n\\ \rho_0'=\rho_0/_x\\ P\propto(M_1\mid\cdots\mid M_n)\\ \phi_i\in\mathcal{V}\llbracket M_i\rrbracket_{\rho_i}}} \mathcal{V}\llbracket M\rrbracket_{\rho_0[x:=\uparrow\phi_1\times\cdots\times\phi_n]}\\
\mathcal{V}\llbracket \mathrm{c}P\rrbracket_\rho &= \begin{cases}\mathcal{V}\llbracket \mathbf{I}\rrbracket_\rho & \text{si } \exists M\ \ P\equiv(M\mid Q)\ \&\ \gamma\in\mathcal{V}\llbracket M\rrbracket_\rho\\ \bot & \text{sinon}\end{cases}
\end{aligned}$$

Lemme 5.1.1 *Soit $M\in\Lambda_r$ et $\rho,\rho'\in Env$;*

(Monotonie) $\quad \rho\subseteq\rho' \ \Rightarrow\ \mathcal{V}\llbracket M\rrbracket_\rho \sqsubseteq \mathcal{V}\llbracket M\rrbracket_{\rho'}$

(Continuité) $\quad \tau\in\mathcal{V}\llbracket M\rrbracket_\rho \Leftrightarrow \exists\xi\in EnvC\ \xi\subseteq\rho\ \tau\in\mathcal{V}\llbracket M\rrbracket_\xi$

Preuve. Par récurrence sur la définition de la fonction sémantique. C'est ici que l'on a besoin de $\rho\supseteq\rho_0\cdot\ldots\cdot\rho_n$ à la place de $\rho=\rho_0\cdot\ldots\cdot\rho_n$. Standard. □

Définition 5.1.2 ***(Sémantique engendrée par le modèle $\mathcal{C}$)***
Pour tout $M,N\in\Lambda_{rc}$,

$$M\sqsubseteq_{\mathcal{C}}N \overset{def}{\Leftrightarrow} \forall\rho\in Env\ \mathcal{V}\llbracket M\rrbracket_\rho\subseteq\mathcal{V}\llbracket N\rrbracket_\rho$$

Nous consacrons une grande partie de ce chapitre à montrer l'adéquation complète du modèle $\mathcal{C}$ par rapport au calcul λ_r^c, c'est-à-dire,

$$\forall M,N\in\Lambda_{rc}\ \ M\sqsubseteq_{\mathcal{C}}N \Leftrightarrow M\sqsubseteq_{rc}N$$

5.2 Système de types $\mathcal{P}$ associé à $\mathcal{C}$

Les jugements prouvables dans le système de types $\mathcal{P}$ ont la forme $\Gamma\vdash T:\tau$ où Γ est un multi-ensemble fini d'hypothèses $x_1:\pi_1,\ldots x_n:\pi_n$, T est un terme (resp. paquet de termes) et τ est un type dans $\textsc{Ft}$ (resp. dans $\textsc{Fb}$.) On utilise la notation $x \in \Gamma$ pour $\exists \pi \; x : \pi \in \Gamma$. Les règles structurelles usuelles sont regroupées dans une seule règle qui autorise un nombre fini d'échanges, affaiblissements, et de produits d'hypothèses. Cette manipulation des hypothèses est formalisée par la relation $\gg$.

Définition 5.2.1 *On définit $\gg$ comme la plus petite relation reflexive et transitive qui contient les paires*

$$\begin{array}{lrcl} \textit{(échange)} & \Gamma, x : \pi, y : \psi, \Delta & \gg & \Gamma, y : \psi, x : \pi, \Delta \\ \textit{(affaiblissement)} & \Gamma & \gg & x : \pi, \Gamma \\ \textit{(produit)} & \Gamma, x : \pi, x : \psi, \Delta & \gg & \Gamma, x : \pi \times \psi, \Delta \end{array}$$

La figure 5.1 contient les règles du système $\mathcal{P}$.

$$\text{L1: } x : \phi \vdash x : \phi \qquad\qquad \text{L6: } \frac{\Gamma \vdash (M|M^{\infty}) : \pi}{\Gamma \vdash M^{\infty} : \pi}$$

$$\text{L2: } \frac{x : \pi, \Gamma \vdash N : \phi}{\Gamma \vdash \lambda x.N : \pi \to \phi} \; (x \notin \Gamma) \qquad\qquad \text{L7: } \Gamma \vdash T : \omega$$

$$\text{L3: } \frac{\Gamma \vdash M : \pi \to \phi \quad \Delta \vdash P : \pi}{\Gamma, \Delta \vdash (MP) : \phi} \qquad\qquad \text{L8: } \frac{\Delta \vdash T : \tau}{\Gamma \vdash T : \tau} \; (\Delta \gg \Gamma)$$

$$\text{L4: } \frac{\Gamma \vdash P : \pi \quad x : \pi, \Delta \vdash M : \phi}{\Gamma, \Delta \vdash M\langle P/x \rangle : \phi} \; (x \notin \Delta) \qquad\qquad \text{L9: } \frac{\Gamma \vdash T : \tau}{\Gamma \vdash T : \sigma} \; (\tau \leq \sigma)$$

$$\text{L5: } \frac{\Gamma \vdash P : \pi \quad \Delta \vdash Q : \psi}{\Gamma, \Delta \vdash (P|Q) : \pi \times \psi} \qquad\qquad \text{L10: } \frac{\Gamma \vdash P : \omega \to \omega}{\Gamma \vdash \mathrm{c}P : \phi \to \phi}$$

FIG. 5.1 - *Système de types $\mathcal{P}$*

Remarquons que les règles d'affectation de types diffèrent essentiellement des règles des systèmes de types avec intersection classiques [21, 70]. Dans le système $\mathcal{P}$, la manipulation des hypothèses est "multiplicative"; les contractions usuelles ne sont pas autorisées. Ceci permet de traiter chaque occurrence de variable de façon indépendante du reste et d'associer une ressource à chacune. Les types produit sont de la forme $\sigma_1 \times \cdots \times \sigma_n$ où $n \geq 1$ et $\sigma_i \in$ Ft . La non-idempotence du produit est fondamentale; ici un argument n'a un type produit $\sigma_1 \times \cdots \times \sigma_n$ que s'il contient au moins autant de ressources que de types σ_i différents de ω.

Lemme 5.2.2 ***(Règle dérivée)***

La règle suivante est démontrable dans le système $\mathcal{P}$:

$$\frac{x : \pi, \Gamma \vdash T : \tau}{x : \psi, \Gamma \vdash T : \tau}\ (\psi \leq \pi)$$

Preuve. L'énoncé est conséquence directe de la généralisation de l'axiome (L1) : $\psi \leq \phi \Rightarrow x : \psi \vdash x : \phi$. On prouve cette implication comme suit : soit $\psi = \sigma_1 \times \ldots \times \sigma_n \leq \phi$; par la propriété du produit il existe i t.q. $\sigma_i \leq \phi$. Donc

$$\frac{\dfrac{\dfrac{x : \sigma_i \vdash x : \sigma_i}{x : \sigma_i \vdash x : \phi}\,(L9)}{x : \sigma_1, \ldots, x : \sigma_n \vdash x : \phi}\,(L8)}{x : \psi \vdash x : \phi}\,(L8)$$

□

Convention 5.2.3 *Le lemme 5.2.2 autorise des affaiblissements additionnels pendant les preuves. Par la suite, on considère la relation $\gg$ enrichie avec le nouvel axiome d'affaiblissement*

$$\psi \leq \pi \ \Rightarrow\ x : \pi, \Gamma \gg x : \psi, \Gamma$$

La règle dérivée est ainsi incorporée au système $\mathcal{P}$ comme cas particulier de la règle (L8).

Nous introduisons maintenant quelques définitions.

Définition 5.2.4 – *La notation $\pi \triangleright \phi_1, \ldots, \phi_m$ rend compte des facteurs significatifs ϕ_i dans π, i.e. ceux différents de ω. La relation est définie par:*

  - $\omega \triangleright \varepsilon$ *(la séquence vide)*
  - $\psi \to \delta \triangleright \psi \to \delta$
  - $\pi \triangleright \phi_1, \ldots, \phi_m \ \&\ \pi' \triangleright \phi'_1, \ldots, \phi'_k \ \Rightarrow\ \pi \times \pi' \triangleright \phi_1, \ldots, \phi_m, \phi'_1, \ldots, \phi'_k$

- *On note $\{\Gamma\}$ tout contexte obtenu à partir de Γ par permutation d'une partie de ses hypothèses. C'est-à-dire, $\{\Gamma\}$ représente les Δ t.q. $\Gamma \gg \Delta$ est prouvable en utilisant seulement l'axiome d'échange.*

- *Soit T un terme ou un paquet de termes et y une variable; on définit Γ_T et $\Gamma/_y$ par récurrence sur Γ comme suit :*

$$\Gamma \ \textit{vide} \quad \Rightarrow \quad \left[\begin{array}{l} \Gamma_T \ \ \textit{vide} \\ \Gamma/_y \ \ \textit{vide} \end{array}\right.$$

$$\Gamma = x : \pi, \Delta \quad \Rightarrow \quad \left[\begin{array}{l} \Gamma_T = \begin{cases} x : \pi, \Delta_T & \textit{si } x \in fv(T) \\ \Delta_T & \textit{sinon} \end{cases} \\ \\ \Gamma/_y = \begin{cases} x : \pi, \Delta/_y & \textit{si } x \neq y \\ \Delta/_y & \textit{sinon} \end{cases} \end{array}\right.$$

- *Si* $\Gamma_x = x : \pi$ *(il n'y a qu'une seule occurrence de* x *dans* Γ*) alors on dit que* $\Gamma(x) = \pi$.

- *On définit* $\Gamma^{\times}$ *comme le contexte vérifiant : si* $\Gamma_x = x : \pi_1, \ldots, x : \pi_n$ *alors* $\Gamma^{\times}(x) \sim \pi_1 \times \ldots \times \pi_n$.

- *On note* $\tilde{\Gamma}$ *l'ensemble d'hypothèses t.q. pour tout* $x : \pi \in \Gamma$*, si* $\pi = \phi_1 \times \ldots \times \phi_n$ *où* $\phi_i \in \mathrm{Ft}$ *, alors* $x : \phi_1, \ldots, x : \phi_n \in \tilde{\Gamma}$.

Proposition 5.2.5

1. $\Gamma \vdash T : \tau \;\Rightarrow\; \tilde{\Gamma} \vdash T : \tau$.

2. $\Gamma^{\times} \vdash T : \tau \;\Rightarrow\; \Gamma \vdash T : \tau$.

3. $\Gamma \vdash T : \tau \;\&\; y \notin fv(T) \;\Rightarrow\; \Gamma/_y \vdash T : \tau$*, donc* $\Gamma_T \vdash T : \tau$.

4. $x : \pi_1, \ldots, x : \pi_n \gg x : \psi_1, \ldots, x : \psi_m \;\Rightarrow\; \psi_1 \times \cdots \times \psi_m \leq \pi_1 \times \cdots \times \pi_n$.

Preuve.

1. Nous montrons $\forall \Gamma\; \Gamma \vdash T : \tau \;\Rightarrow\; \tilde{\Gamma} \vdash T : \tau$ par récurrence sur la taille de la dérivation de $\Gamma \vdash T : \tau$, en examinant la dernière règle utilisée. La taille 1 correspond à l'application de (L1) ou (L7); dans ces cas, l'énoncé est immédiatement vérifié. Supposons que la taille est au moins 2 :

   (L2) Soit $T = \lambda x.M$, $\tau = \pi \rightarrow \sigma$ et $x : \pi, \Gamma \vdash M : \sigma$ avec $x \notin \Gamma$. Par h.r., on a $x \tilde{:} \pi, \tilde{\Gamma} \vdash M : \sigma$. Par (L8), qui permet de recomposer l'hypothèse $x : \pi$, on arrive à prouver $x : \pi, \tilde{\Gamma} \vdash M : \sigma$. Alors $\tilde{\Gamma} \vdash \lambda x.M : \pi \rightarrow \sigma$ par (L2).

   (L3) Soit $T = MP$, $\Gamma_1 \vdash M : \pi \rightarrow \tau$ et $\Gamma_2 \vdash P : \pi$ avec $\Gamma = \Gamma_1, \Gamma_2$. Les deux séquents $\tilde{\Gamma} \vdash M : \pi \rightarrow \tau$ et $\tilde{\Gamma_2} \vdash P : \pi$ sont prouvables par hypothèse de récurrence. Comme $\tilde{\Gamma} = \tilde{\Gamma_1}, \tilde{\Gamma_2}$, une application de (L3) donne $\tilde{\Gamma} \vdash MP : \tau$.

   (L4) et (L5) Comme pour les règles (L2) et (L3), respectivement.

   (L6), (L9) et (L10) Par hypothèse de récurrence.

   (L8) Soit $\Delta \vdash T : \tau$ où $\Delta \gg \Gamma$. Il est facile de voir que $\tilde{\Delta} \gg \tilde{\Gamma}$. Par ailleurs, l'hypothèse de récurrence donne $\tilde{\Delta} \vdash T : \tau$. Donc, $\tilde{\Gamma} \vdash T : \tau$ par (L8).

2. Soit $\Gamma^{\times} \vdash M : \phi$. De la partie (1), on déduit $\tilde{\Gamma^{\times}} \vdash M : \phi$. Or, $\tilde{\Gamma^{\times}} \gg \tilde{\Gamma} \gg \Gamma$. On obtient $\Gamma \vdash M : \phi$ par application de (L8) .

3. Par récurrence sur la dérivation de $\Gamma \vdash T : \tau$. Évident.

4. Cette partie de la proposition est corollaire de l'énoncé suivant : Soit $\Gamma \gg \Delta$; si $\Gamma_x = x : \pi_1, \ldots, x : \pi_n$ et $\Delta_x = x : \psi_1, \ldots, x : \psi_m$, alors $\psi_1 \times \ldots \times \psi_m \leq \pi_1 \times \ldots \times \pi_n$. La preuve est par récurrence sur la longueur de l'inférence de $\Gamma \gg \Delta$. Nous examinons chacune des règles de la définition de $\gg$.

   Échange : $\Gamma = \Gamma_1, y_1 : \theta_1, y_2 : \theta_2, \Gamma_2 \gg \Gamma_1, y_2 : \theta_2, y_1 : \theta_1, \Gamma_2$. Si $y_1 = y_2 = x$ alors $\pi_1 \times \ldots \theta_1 \times \theta_2 \ldots \times \pi_n \leq \pi_1 \times \ldots \theta_2 \times \theta_1 \ldots \times \pi_n$ par commutativité de $\leq$. Sinon, $\psi_i = \pi_i$ pour tout i.

   Affaiblissement : $\Gamma \gg y : \theta, \Gamma = \Delta$. Si $y \neq x$ alors $\forall i \; \psi_i = \pi_i$. Si $y = x$ alors $\psi_1 = \theta$ et $\theta \times \pi_1 \ldots \times \pi_n \leq \omega \times \pi_1 \ldots \times \pi_n \leq \pi_1 \times \ldots \times \pi_n$.

   Dans le cas où $\Gamma = \Gamma_1, y : \theta, \Gamma_2 \gg \Gamma_1, y : \theta', \Gamma_2$ avec $\theta' \leq \theta$, $y = x$ et $\theta = \pi_i$, on a $(\pi_1 \times \ldots \times \pi_i) \times \theta' \leq (\pi_1 \times \ldots \times \pi_i) \times \theta$ et $(\pi_{i+2} \times \ldots \times \pi_n) \leq (\pi_{i+2} \times \ldots \times \pi_n)$ impliquent $\pi_1 \times \ldots \times \theta' \ldots \pi_n \leq \pi_1 \times \ldots \times \theta \times \ldots \times \pi_n$.

   Produit : $\Gamma = \Gamma_1, y : \theta_1, y : \theta_2, \Gamma_2 \gg \Gamma_1, y : \theta_1 \times \theta_2, \Gamma_2 = \Delta$. Si $y \neq x$ il n'y a rien à prouver. Sinon, $\exists i \; \pi_i = \theta_1 \;\&\; \pi_{i+1} = \theta_2$; donc, par associativité et commutativité de $\leq$:

   $$\pi_1 \times \ldots \pi_{i-1} \times (\theta_1 \times \theta_2) \times \pi_{i+2} \ldots \times \pi_n \leq \pi_1 \times \ldots \pi_{i-1} \times \theta_1 \times \theta_2 \times \ldots \times \pi_n$$

   Transitivité : Par transitivité de $\leq$.

□

L'affectation de types dans le système $\mathcal{P}$ est dirigée par la syntaxe, à utilisation de (L8) et (L9) près.

Proposition 5.2.6

1. $\Gamma \vdash P : \pi \;\&\; \pi \triangleright \phi_1, \ldots, \phi_n \Rightarrow (n \geq 1 \Rightarrow \exists M_1, \ldots, M_n, Q \;\exists \Gamma_1, \ldots \Gamma_n$

$$P \equiv (M_1 \mid \ldots \mid M_n \mid Q) \quad \textit{et} \quad \Gamma_1, \ldots, \Gamma_n \gg \Gamma \quad \textit{et} \quad \forall 1 \leq i \leq n \;\Gamma_i \vdash M_i : \phi_i)$$

2. $\Gamma \vdash M\langle P/x \rangle : \phi \Rightarrow \exists \psi, \Gamma_1, \Gamma_2$

$$\Gamma_1 \vdash P : \psi \quad \textit{et} \quad x : \psi, \Gamma_2 \vdash M : \phi \quad \textit{où} \quad x \notin \Gamma_2 \quad \textit{et} \quad \Gamma_1, \Gamma_2 \gg \Gamma$$

3. $\Gamma \vdash MP : \phi \Rightarrow \exists \pi, \Gamma_1, \Gamma_2$

$$\Gamma_1 \vdash M : \pi \rightarrow \phi \quad \textit{et} \quad \Gamma_2 \vdash P : \pi \quad \textit{où} \quad \Gamma_1, \Gamma_2 \gg \Gamma$$

4. $\Gamma \vdash \lambda x.M : \phi \Rightarrow (\phi \neq \omega \Rightarrow \exists \pi, \sigma, \Delta$

$$x : \pi, \Delta \vdash M : \sigma \quad \textit{où} \quad \phi = \pi \rightarrow \sigma \,, \, x \notin \Delta \quad \textit{et} \quad \Delta \gg \Gamma)$$

Preuve.

1. Par récurrence sur la taille de la dérivation de $\Gamma \vdash P : \pi$.

   (L1) à (L4) et (L10) Immédiat car P est nécessairement un terme ($\pi \in \mathtt{Ft}$).

   (L5) Ici $P = (P_1 \mid P_2)$, $\pi = \pi_1 \times \pi_2$ et $\Delta_i \vdash P_i : \pi_i$, $i = 1, 2$ où $\Gamma = \Delta_1, \Delta_2$. Par hypothèse, il existe $1 \leq i \leq n$ tel que $\pi_1 \triangleright \phi_1, \ldots, \phi_i$ et $\pi_2 \triangleright \phi_{i+1}, \ldots, \phi_n$. Quand ces deux séquences de facteurs significatifs ne sont pas vides, par h.r. , $\exists M_1, \ldots, M_n, Q_1, Q_2$ $\exists \Gamma_1, \ldots, \Gamma_n$ qui vérifient

   $$P_1 \equiv (M_1 \mid \ldots \mid M_i \mid Q_1) \;\; \& \;\; P_2 \equiv (M_{i+1} \mid \ldots \mid M_n \mid Q_2)$$

   où $\Gamma_i \vdash M_i : \phi_i$, $\Gamma_1, \ldots, \Gamma_i \gg \Delta_1$ et $\Gamma_{i+1}, \ldots, \Gamma_n \gg \Delta_2$.

   Par conséquent, $P \equiv (M_1 \mid \ldots \mid M_n \mid Q_1 \mid Q_2)$ et $\Gamma_1, \ldots, \Gamma_n \gg \Delta_1, \Delta_2 \gg \Gamma$.

   Si π_1 (resp. π_2) n'avait pas de facteur significatif, le terme Q serait P_1 (resp. P_2) et le reste de l'énoncé découlerait de l'hypothèse de récurrence sur $\Delta_2 \vdash P_2 : \pi_2$ (resp. $\Delta_1 \vdash P_1 : \pi_1$).

   (L6) Dans ce cas $P = M^\infty$ et $\Gamma \vdash (M \mid M^\infty) : \pi$. Par h.r., il existe $M_1, \ldots, M_n, Q$ et $\Gamma_1, \ldots, \Gamma_n \gg \Gamma$ t.q. $(M \mid M^\infty) \equiv (M_1 \mid \ldots \mid M_n \mid Q)$ avec $\Gamma_i \vdash M_i : \phi_i$. En effet, $P = M^\infty \equiv (M \mid M^\infty)$. On en conclut $P \equiv (M_1 \mid \ldots \mid M_n \mid Q)$ par transitivité de $\equiv$.

   (L7) L'application de (L7) est impossible sous nos hypothèses, qui imposent $n \geq 1$.

   (L8) Immédiat par hypothèse de récurrence.

   (L9) Ici $\Gamma \vdash P : \psi$ où $\psi \leq \pi$. Soit $\psi = \sigma_1 \times \ldots \times \sigma_m$ et $\psi \triangleright \delta_1, \ldots, \delta_r$. Par h.r., $P \equiv (N_1 \mid \ldots \mid N_r \mid Q')$, $\Gamma_i \vdash M_i : \delta_i$ et aussi $\Gamma_1, \ldots, \Gamma_k \gg \Gamma$.

   Or, $\delta_1 \times \ldots \times \delta_r \leq \pi$ résulte de l'application de la propriété du produit. Il existe donc une injection $b : [1, n] \to [1, r]$ t.q. $\delta_{b(j)} \leq \phi_j$ pour tout $j = 1, \ldots, n$. Alors, $n \leq r$ et $\Gamma_{b(j)} \vdash N_{b(j)} : \phi_j$, $j = 1, \ldots, n$. C'est une évidence que $(N_1 \mid \ldots \mid N_r) \equiv (N_{b(1)} \mid \ldots \mid N_{b(n)} \mid N'_1 \mid \ldots \mid N'_{r-n})$ où chaque N'_k est en fait N_l t.q. $l \neq b(j)$ pour tout $j = 1, \ldots, n$.

   Si l'on pose $Q = (N'_1 \mid \ldots \mid N'_{r-n} \mid Q')$, on conclut $P \equiv (N_{b(1)} \mid \ldots \mid N_{b(n)} \mid Q)$; nos termes M_j sont alors les termes $N_{b(j)}$.

2. Par récurrence sur la taille de la dérivation de $\Gamma \vdash M\langle P/x \rangle : \phi$:

   (L4) Ici $x : \pi, \Gamma_1 \vdash M : \phi$ et $\Gamma_2 \vdash P : \pi$ avec $x \notin \Gamma_2$ et $\Gamma = \Gamma_1, \Gamma_2$. On a $\Gamma_1, \Gamma_2 \gg \Gamma$ par reflexivité de la relation $\gg$.

   (L7) Ici $\phi = \omega$. Soit $\psi = \omega$ et Γ_1, Γ_2 des contextes vides. Par application de (L7), on a $\vdash P : \psi$ and $x : \psi \vdash M : \omega$. Nous passons de Γ_1, Γ_2 à Γ par affaiblissement.

(L8) Ici $\Sigma \vdash M\langle P/x\rangle : \phi$ où $\Sigma \gg \Gamma$. Par h.r., $\exists\psi\ \exists\Sigma_1, \Sigma_2$, pour lesquels $\Sigma_1, \Sigma_2 \gg \Sigma$, $x \notin \Sigma_2$, $\Sigma_1 \vdash P : \psi$ et aussi $x : \psi, \Sigma_2 \vdash M : \phi$. Par transitivité, $\Sigma_1, \Sigma_2 \gg \Gamma$ est vérifié.

(L9) Par hypothèse de récurrence et la même règle (L9).

3. Par récurrence sur l'inférence de $\Gamma \vdash MP : \phi$. Deux cas sont significatifs : Si la dérivation termine par une application de (L3), l'énoncé est immédiatement vérifié. Si celle-ci termine par (L9), alors $\Gamma \vdash MP : \sigma$ avec $\sigma \leq \phi$. Par h.r.,

$$\Gamma_1 \vdash M : \pi \rightarrow \sigma \quad \text{et} \quad \Gamma_2 \vdash P : \pi \quad \text{où} \quad \Gamma_1, \Gamma_2 \gg \Gamma$$

Du fait que $\sigma \leq \phi \Rightarrow \pi \rightarrow \sigma \leq \pi \rightarrow \phi$, on déduit $\Gamma_1 \vdash M : \pi \rightarrow : \phi$ par règle (L9).

4. Par récurrence, comme dans les cas précédents.

□

Proposition 5.2.7

1. $\Gamma \vdash x : \phi \ \Rightarrow\ (\phi \neq \omega \ \Rightarrow\ \exists x : \pi \in \Gamma\ \pi \leq \phi)$

2. $\Gamma \vdash \boldsymbol{I} : \pi \rightarrow \phi \ \Rightarrow\ \pi \leq \phi$

3. $\Gamma \vdash cx : \psi \rightarrow \sigma \ \Rightarrow\ \exists x : \pi \in \Gamma\ \pi \leq \gamma\ \&\ \psi \leq \sigma$

4. $\Gamma \vdash x^{\infty} : \pi \ \Rightarrow\ (\pi \nsim \omega \ \Rightarrow\ \exists n > 0\ \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma\ \ \psi_1 \times \ldots \times \psi_n \leq \pi)$

Preuve.

1. Par récurrence sur la taille de la dérivation de $\Gamma \vdash x : \phi$. Soit $\phi \neq \omega$. Dans le cas de base (taille 1), le séquent est une instance de l'axiome (L1), et $\pi = \phi$. Si la taille est au moins 2, la preuve du séquent termine soit par (L8) soit par (L9). Dans le premier cas, nous avons $\Delta \vdash x : \phi$ où $\Delta \gg \Gamma$. Par h.r., il existe ψ qui vérifie $x : \psi \in \Delta$ et $\psi \leq \phi$. On observe que les règles pour transformer Δ en Γ sont conservatrices : Γ contient des hypothèses $x : \pi$ tel que $\pi \leq \psi$ (on ne peut que renforcer les hypothèses). Ceci implique $\pi \leq \phi$ par transitivité. Dans le second cas, soit $\Gamma \vdash x : \sigma$ où $\sigma \leq \phi$; par h.r. il existe π t.q. $x : \pi \in \Gamma$ et $\pi \leq \sigma$. On obtient $\pi \leq \phi$ par transitivité.

2. Sous l'hypothèse $\phi \neq \omega$ ($\phi = \omega \Rightarrow \pi \leq \phi$ pour tout π), on prouve la proposition par récurrence sur la taille de l'inférence de $\Gamma \vdash \mathbf{I} : \pi \rightarrow \phi$. Le cas de base est immédiat. Supposons que la dernière règle utilisée est (L2). Alors, on a $x : \pi, \Gamma \vdash x : \phi$ avec $x \notin \Gamma$. La partie (1) de cette

proposition implique $\pi \leq \phi$. Si la dérivation termine par (L8), par h.r. on a $\Delta \vdash \mathbf{I} : \pi \to \phi$ et $\pi \leq \phi$, où $\Delta \gg \Gamma$. Le dernier cas concerne la règle (L9); c'est-à-dire, $\Gamma \vdash \mathbf{I} : \delta$ avec $\delta \leq \pi \to \phi$. Le type δ, dans `Ft` , doit être de la forme $\pi' \to \phi'$. La propriété de la flèche plus $\phi \neq \omega$ impliquent $\phi' \neq \omega$, $\pi \leq \pi'$ et $\phi' \leq \phi$. L'hypothèse de récurrence donne $\pi' \leq \phi'$, alors $\pi \leq \phi$ par transitivité.

3. Par récurrence sur la dérivation de $\Gamma \vdash \mathrm{c}x : \psi \to \sigma$. Si celle-ci termine par application de (L10), i.e. $\psi = \sigma$ (donc $\psi \leq \sigma$) et $\Gamma \vdash x : \gamma$, la partie (1) garantit l'existence de $x : \pi$ dans Γ t.q. $\pi \leq \gamma$. Si la dérivation termine par (L8), en utilisant l'hypothèse de récurrence,

$$\Delta \vdash \mathrm{c}P : \psi \to \sigma \ \ \& \ \ x : \theta \in \Delta \ \ \& \ \ \theta \leq \gamma \ \ \& \ \ \psi \leq \sigma$$

   où $\Delta \gg \Gamma$. D'après la définition de $\gg$, l'hypothèse $x : \theta$ de Δ a dû se transformer en $x : \pi$ dans Γ, vérifiant $\pi \leq \theta$. Par transitivité, $\pi \leq \gamma$. Finalement, quand la dérivation termine par (L9), nous avons $\Gamma \vdash \mathrm{c}x : \delta$ où $\delta \leq \psi \to \sigma$. Remarquons que $\delta \in$ `Ft` implique $\delta = \psi' \to \sigma'$. Par h.r., $\exists x : \pi \in \Gamma \ \pi \leq \gamma \ \& \ \psi' \leq \sigma'$. Si $\sigma = \omega$, alors $\psi \leq \sigma$ immédiatement. Sinon, $\psi \leq \psi'$ et $\sigma' \leq \sigma$ par la propriété de la flèche; alors $\psi \leq \sigma$ par transitivité.

4. Cette propriété est corollaire de la propriété suivante: $\forall p \geq 0, \forall m \geq 0 \ \forall P$

$$(P \equiv (x^p \mid \overbrace{x^\infty \mid \cdots \mid x^\infty}^{m}) \ \ \& \ \ p + m \geq 1 \ \ \& \ \ \Gamma \vdash P : \pi \ \ \Rightarrow$$

$$\pi \sim \omega \quad \text{ou} \quad \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma \ \ \psi_1 \times \ldots \times \psi_n \leq \pi)$$

   que l'on montre par récurrence sur la dérivation de $\Gamma \vdash P : \pi$.

   Le cas de base correspond à une application de (L7), i.e. $\pi = \omega$, ou de (L1), auquel cas on a $p = 1$, $m = 0$, $\pi \in$ `Ft` et $\Gamma = x : \pi$ donc $\pi \leq \pi$ par reflexivité.

   Pour le cas de récurrence, nous examinons quatre situations, selon la dernière règle appliquée :

   (L5) Soit $P = (P_0 \mid P_1)$, $\pi = \pi_0 \times \pi_1$ et $\Gamma = \Gamma_0, \Gamma_1$ avec $\Gamma_0 \vdash P_0 : \pi_0$ et $\Gamma_1 \vdash P_1 : \pi_1$. Il est facile de prouver que $\exists p_0, m_0, p_1, m_1 \ p = p_0 + p_1 \ \& \ m = m_0 + m_1$ et aussi

$$P_0 \equiv (x^{p_0} \mid \underbrace{x^\infty \mid \cdots \mid x^\infty}_{m_0}) \quad \text{et} \quad P_1 \equiv (x^{p_1} \mid \underbrace{x^\infty \mid \cdots \mid x^\infty}_{m_1})$$

   avec $p_0 + m_0 \geq 1$ et $p_1 + m_1 \geq 1$. Par h.r., si $\pi_0 \sim \omega$ et $\pi_1 \sim \omega$ alors $\pi \sim \omega$. Dans le cas où $\pi_0 \nsim \omega$ et $\pi_1 \nsim \omega$ alors $\exists x : \psi_1, \ldots, x : \psi_k \in \Gamma_1 \ \ \psi_1 \times \ldots \times \psi_k \leq \pi_0 \ \exists x : \psi_{k+1}, \ldots, x : \psi_{k+q} \in \Gamma_2 \ \ \psi_{k+1} \times \ldots \times \psi_{k+q} \leq \pi_1$. Par conséquent, tous ces ψ_i sont dans Γ et $\psi_1 \times \ldots \times \psi_{k+q} \leq \pi_0 \times \pi_1 = \pi$. Si, par exemple, $\pi_0 \sim \omega$ mais $\pi_1 \nsim \omega$, on vérifie $\psi_{k+1} \times \ldots \times \psi_{k+q} \leq \omega \times \psi_{k+1} \times \ldots \times \psi_{k+q} \leq \pi_0 \times \pi_1 = \pi$.

(L6) Dans ce cas, on a $p = 0$, $m = 1$ et $\Gamma \vdash (x \mid x^{\infty}) : \pi$. L'énoncé s'ensuit par hypothèse de récurrence.

(L8) Ici $\Delta \vdash P : \pi$ avec $\Delta \gg \Gamma$. Par h.r., $\exists x : \psi_1, \ldots, x : \psi_n \in \Delta \;\; \psi_1 \times \ldots \times \psi_n \leq \pi$. L'énoncé est vérifié car les règles définissant $\gg$ sont conservatrices.

(L9) Ici $\Gamma \vdash P : \psi$ avec $\psi \leq \pi$. L'énoncé est vérifié par hypothèse de récurrence et transitivité de $\leq$.

□

5.3 Stabilité du typage par expansion. Extensionnalité

L'affectation de types dans le système $\mathcal{P}$ est *stable par expansion*: les types des termes trouvés par évaluation de $M \in \Lambda_{rc}$ sont aussi des types de M. Dans le but de montrer cette propriété, on définit une relation de *réduction* $\triangleright$ dont la stratégie faible coïncide avec l'évaluation $\rightarrow_{rc}$. La réduction opère à tous les niveaux d'un terme: (β), (*saisie*), (c1), (c2) et (v) sont réalisables sous les abstractions aussi bien que dans les arguments et substitutions explicites. De plus, la réduction d'un terme permet de distribuer arbitrairement les entrées de substitution aux sous-termes. L'ensemble de règles qui définissent $\triangleright$ comprend les règles pour $\rightarrow_{rc}$, définies dans la figure 3.6, dans lesquelles le symbole $\rightarrow_{rc}$ est remplacé par $\triangleright$ et le symbole de la saisie $\succ$ par $\succcurlyeq$, plus les règles structurelles de la figure 5.2.

Proposition 5.3.1 *Soit* $\Gamma \vdash M : \phi$.

1. $M =_\alpha M' \;\Rightarrow\; \Gamma \vdash M' : \phi$

2. $M \equiv M' \;\Rightarrow\; \Gamma \vdash M' : \phi$

Preuve. Nous nous limitons à prouver la partie (1); l'énoncé (2) découle essentiellement de la proposition 5.2.6(1). Soit $\Gamma[z/x] = \Gamma/_x \cup \{z : \pi \,/\, x : \pi \in \Gamma\}$. Il est facile de montrer que $\Gamma \vdash N : \phi$ implique $\Gamma[z/x] \vdash N[z/x] : \phi$ pourvu que $z \notin var(N)$ et $z \notin \Gamma$. Supposons $\phi \neq \omega$ et $z \notin var(x, N)$. On examine deux situations:

- dans le cas où $\lambda x.N =_\alpha \lambda z.N[z/x]$ et $\Gamma \vdash \lambda x.N : \phi$, les propositions 5.2.5 et 5.2.6(4) impliquent $\phi = \pi \rightarrow \sigma$ et

$$x : \pi, \Delta \vdash N : \sigma \quad \text{où } x \notin \Delta \quad \text{et } \Delta \gg \Gamma_N$$

$$\frac{M \triangleright M'}{\lambda x.M \triangleright \lambda x.M'} \qquad \frac{M \triangleright M'}{(M \mid P) \triangleright (M' \mid P)} \qquad \frac{P \triangleright P'}{(MP) \triangleright (MP')}$$

$$\frac{P \triangleright P'}{M\langle P/x\rangle \triangleright M\langle P'/x\rangle} \qquad \frac{P \triangleright P'}{\mathrm{c}P \triangleright \mathrm{c}P'} \qquad \frac{P \triangleright P' \quad Q \equiv P}{Q \triangleright P'}$$

$$\frac{M\langle N/x\rangle \succcurlyeq M'}{(L(M \mid P))\langle N/x\rangle \succcurlyeq L(M' \mid P)} \qquad \frac{M\langle N/x\rangle \succcurlyeq M'}{(L\langle (M \mid P)/z\rangle)\langle N/x\rangle \succcurlyeq L\langle (M' \mid P)/z\rangle}$$

FIG. 5.2 - *Réduction dans* λ_r^c

Comme $z \notin \Delta \cup fv(N)$, la remarque précédente entraîne $z : \pi, \Delta \vdash N[z/x] : \sigma$. D'où $\Delta \vdash \lambda z.N[z/x] : \pi \to \sigma$. On a aussi $\Delta \gg \Gamma_N \gg \Gamma$, donc $\Gamma \vdash \lambda z.N[z/x] : \pi \to \sigma$ est démontrable par la règle (L8).

– si $N\langle P/x\rangle =_\alpha N[z/x]\langle P/z\rangle$ et $\Gamma \vdash N\langle P/x\rangle : \phi$, on applique les propositions 5.2.5 et 5.2.6(2) et on procède comme dans le cas précédent.

□

Lemme 5.3.2 *1. Si* $M\langle N/x\rangle \succcurlyeq M'$ *et* $\Gamma \vdash M' : \phi$ *alors* $\Gamma \vdash M\langle N/x\rangle : \phi$.

2. Si $M \triangleright M'$ *et* $\Gamma \vdash M' : \phi$ *alors* $\Gamma \vdash M : \phi$.

Preuve.

1. Par récurrence sur la preuve de $M\langle N/x\rangle \succcurlyeq M'$, puis sur la taille de la dérivation de $\Gamma \vdash M' : \phi$. Nous nous limitons aux cas où $\phi \neq \omega$; sinon, $\Gamma \vdash M' : \omega$ peut être dérivé directement par (L7).

   (a) Si $M = x$ et $M' = N$, alors on prouve $\Gamma \vdash x\langle N/x\rangle$ par (L4) à partir de $x : \phi \vdash x : \phi$ et de l'hypothèse $\Gamma \vdash N : \phi$.

   (b) Si $M = LP$ et $M' = L'P$ avec $L\langle N/x\rangle \succcurlyeq L'$, alors on sait, par la proposition 5.2.6(3),

   $$\Gamma_1 \vdash L' : \pi \to \phi \quad \text{et} \quad \Gamma_2 \vdash P : \pi \quad \text{avec } \Gamma_1, \Gamma_2 \gg \Gamma$$

   Par h.r., on a $\Gamma_1 \vdash L\langle N/x\rangle : \pi \to \phi$; c'est-à-dire, par la proposition 5.2.6(2),

   $$x : \psi, \Delta_1 \vdash L : \pi \to \phi \quad \text{et} \quad \Delta_2 \vdash N : \psi$$

où $x \notin \Delta_1$ et $\Delta_1, \Delta_2 \gg \Gamma_1$. Donc, $x : \psi, \Delta_1, \Gamma_2 \vdash LP : \phi$ par (L3) et ensuite $\Delta_1, \Gamma_2, \Delta_2 \vdash (LP)\langle N/x\rangle : \phi$ par (L4) (on observe que $x \notin \Delta_1, \Gamma_2$). Comme $\gg$ est une relation transitive, on a $\Delta_1, \Gamma_2, \Delta_2 \gg \Gamma$; le lemme en découle par (L8).

(c) Si $M = L\langle P/z\rangle$ et $M' = L'\langle P/z\rangle$ avec $L\langle N/x\rangle \succcurlyeq L'$ et $z \notin fv(N, x)$, alors

$$z : \psi, \Gamma_1 \vdash L' : \phi \quad \text{et} \quad \Gamma_2 \vdash P : \psi \quad \text{où} \quad z \notin \Gamma_1 \quad \text{et} \quad \Gamma_1, \Gamma_2 \gg \Gamma$$

par la proposition 5.2.6(2). De l'h.r. on obtient $z : \psi, \Gamma_1 \vdash L\langle N/x\rangle : \phi$. par la proposition 5.2.6(2), ce séquent peut être décomposé en

$$x : \pi, \Sigma \vdash L : \phi \quad \text{et} \quad \Delta \vdash N : \pi \quad \text{où} \quad x \notin \Sigma \quad \text{et} \quad \Sigma, \Delta \gg z : \psi, \Gamma_1$$

Par ailleurs, $\Delta_N \vdash N : \pi$ et $\Delta_N \gg \Delta$ impliquent $\Sigma, \Delta_N \gg z : \psi, \Gamma_1$. Puisque $z \notin \Delta_N$, on a $\Sigma \gg z : \psi, \Sigma/_{z}$ et $\Sigma/_{z}, \Delta_N \gg \Gamma_1$; donc

$$x : \pi, z : \psi, \Sigma/_{z} \vdash L : \phi$$

En appliquant (L4) deux fois on arrive à

$$\Sigma/_{z}, \Gamma_2, \Delta_N \vdash (L\langle P/z\rangle)\langle N/x\rangle : \phi$$

d'où $\Gamma \vdash (L\langle P/z\rangle)\langle N/x\rangle : \phi$ par (L8).

(d) Si $M = (\mathrm{c}L)\langle N/x\rangle$ et $M' = (\mathrm{c}L')$, où $L\langle N/x\rangle \succcurlyeq L'$, on examine l'inférence de $\Gamma \vdash \mathrm{c}L' : \phi$. Dans le cas où celle-ci termine par (L8) ou (L9), le lemme est vérifié par h.r. et la même règle. Sinon, le séquent est dérivé par (L10); i.e. $\Gamma \vdash L' : \gamma$ et $\phi = \phi' \to \phi'$. Par h.r., on a $\Gamma \vdash L\langle N/x\rangle : \gamma$. C'est-à-dire, par la proposition 5.2.6(2), il existe $\Gamma_1, \Gamma_2, \sigma$ t.q.

$$x : \sigma, \Gamma_1 \vdash L : \gamma \quad \text{et} \quad \Gamma_2 \vdash N : \sigma \quad \text{avec} \quad x \notin \Gamma_1 \text{ and } \Gamma_1, \Gamma_2 \gg \Gamma$$

En utilisant la même règle (L10), on obtient $x : \sigma, \Gamma_1 \vdash \mathrm{c}L : \phi$ et alors, par (L4) puis (L8), on a $\Gamma \vdash (\mathrm{c}L)\langle N/x\rangle : \phi$.

(e) Si $M = L_0(L_1 \mid P)$ et $M' = L_0(L_1' \mid P)$ avec $L_1\langle N/x\rangle \succcurlyeq L_1'$, alors par la proposition 5.2.6(1)(3),

$$\Gamma_0 \vdash L_1' : \sigma \quad \Gamma_1 \vdash P : \pi' \quad \Gamma_2 \vdash L_0 : \pi \to \phi$$

où $\pi \sim \sigma \times \pi'$ et $\Gamma_1, \Gamma_2, \Gamma_3 \gg \Gamma$. Par h.r., $\Gamma_0 \vdash L_1\langle N/x\rangle : \sigma$. Une nouvelle application de la proposition 5.2.6 donne:

$$x : \psi, \Delta_1 \vdash L_1 : \sigma \quad \text{et} \quad \Delta_2 \vdash N : \psi \quad \text{avec} \quad x \notin \Delta_1 \Delta_1, \Delta_2 \gg \Gamma_0$$

Donc, $x : \psi, \Delta_1, \Gamma_1 \vdash (L_1 \mid P) : \sigma \times \pi'$ par (L5) et

$$x : \psi, \Delta_1, \Gamma_1, \Gamma_2 \vdash L_0(L_1 \mid P) : \phi$$

par (L9) puis (L3). Finalement, $\Delta_1, \Gamma_1, \Gamma_2, \Delta_2 \vdash (L_0(L_1 \mid P))\langle N/x\rangle : \phi$ par (L4). On en conclut $\Gamma \vdash M : \phi$ par (L8) car $\Delta_1, \Gamma_1, \Gamma_2, \Delta_2 \gg \Gamma_0, \Gamma_1, \Gamma_2 \gg \Gamma$.

(f) Si $M = L_0\langle L_1 \mid P/z\rangle$ et $M' = L_0\langle L_1' \mid P/z\rangle$ avec $L_1\langle N/x\rangle \succcurlyeq L_1'$, le raisonnement est similaire au précédent.

2. Cette partie de la proposition est conséquence immédiate de l'énoncé suivant où T est un terme simple ou un paquet de termes :

$$T \triangleright T' \ \& \ \Gamma \vdash T' : \phi \ \Rightarrow \ \Gamma \vdash T : \phi$$

La preuve est par récurrence sur les dérivations de $T \triangleright T'$ et de $\Gamma \vdash T' : \phi$. Tout au long de la démonstration nous appellerons $\mathcal{R}$ la dernière règle utilisée dans la dérivation de ce séquent. Il faut observer que si $\mathcal{R}$ est (L7), (L8) ou (L9) alors on montre $\Gamma \vdash M : \phi$ par hypothèse de récurrence et la même règle, indépendamment de la réduction. Si $T \triangleright T'$ est dérivé par application d'un axiome les cas à étudier sont:

(β) Ici $T = (\lambda x.M)P \triangleright M\langle P/x\rangle = T'$. Supposons que $\mathcal{R}$ est (L4); c'est-à-dire, $\Gamma_1 \vdash P : \psi$ et $x : \psi, \Gamma_2 \vdash M : \phi$ où $\Gamma_1, \Gamma_2 \gg \Gamma$ et $x \notin \Gamma_2$. En utilisant (L2) et (L3), dans l'ordre, on déduit $\Gamma \vdash (\lambda x.M)P : \phi$.

(c1) Ici $T = \mathrm{c}V \triangleright \mathbf{I} = T'$. Par la proposition 5.2.7(2), $\phi = \pi \to \sigma$ et $\pi \leq \sigma$. Or, pour toute valeur V, on a $\Gamma \vdash \mathrm{c}V : \sigma \to \sigma$. Donc $\Gamma \vdash T : \phi$ est dérivé par (L9), puisque $\sigma \to \sigma \leq \phi$.

(c2) Dans ce cas $T = \mathrm{c}P \triangleright \mathrm{c}M = T'$ et $P \equiv (M \mid Q)$. Si $\mathcal{R}$ est (L10), alors $\Gamma \vdash M : \omega \to \omega = \phi$. On en déduit $\Gamma \vdash P : (\omega \to \omega) \times \omega$ par application de (L7) et (L5), d'où $\Gamma \vdash T : \phi$ par (L10).

(v) Ici $T = (\lambda x.M)\langle P/y\rangle \triangleright \lambda x.(M\langle P/y\rangle)$, avec $x \notin fv(P, y)$. Dans ce cas, $\mathcal{R}$ est (L2), i.e.

$$\phi = \pi \to \sigma \ \& \ x : \pi, \Gamma \vdash M\langle P/y\rangle : \sigma \ \& \ x \notin \Gamma$$

Par application de la proposition 5.2.6(2),

$$\Sigma \vdash P : \psi \ \& \ y : \psi, \Delta \vdash M : \sigma \ \& \ \Sigma, \Delta \gg \Gamma$$

D'une part, $\Sigma_P \vdash P : \psi$ et alors $\Sigma_P, \Gamma_2 \gg \Gamma$, par affaiblissement et transitivité. D'autre part, soit $\Delta' = \Delta - \Delta_x$; on en déduit facilement $\Delta \gg x : \pi, \Delta'$. En utilisant (L8) deux fois on prouve

$$x : \pi \times \psi, \Delta' \vdash M : \phi$$

Donc, par (L2) d'abord et (L4) ensuite, on a $y : \psi, \Delta' \vdash \lambda x.M : \pi \to \sigma$ et $\Sigma_P, \Delta' \vdash (\lambda x.M)\langle P/y\rangle : \pi \to \sigma$, car $y \notin \Gamma$ implique $y \notin \Delta'$. Finalement, en appliquant (L8), on obtient $\Gamma \vdash T; \phi$.

Pour le cas où la longueur de $T \triangleright T'$ est au moins 2, la propriété est vérifiée essentiellement par hypothèse de récurrence et la même règle $\mathcal{R}$. Nous examinons en détail deux règles:

- Soit $T = M\langle P/x\rangle \triangleright M'\langle Q/x\rangle = T'$, où $P \equiv (N \mid Q)$, $M\langle N/x\rangle \succcurlyeq M'$ et $x \notin fv(N)$. On peut supposer que $\mathcal{R}$ est (L4), c'est-à-dire:

  $$\Gamma_1 \vdash Q : \psi \quad \text{et} \quad x : \psi, \Gamma_2 \vdash M' : \phi$$

  avec $x \notin \Gamma_2$ et $\Gamma = \Gamma_1, \Gamma_2$. Donc, par la partie (1) du lemme 5.3.2, $x : \psi, \Gamma_2 \vdash M\langle N/x\rangle : \phi$. D'après la proposition 5.2.6, il existe π, Σ', Σ'' t.q.

  $$\Sigma' \vdash N : \pi \quad \text{et} \quad x : \pi, \Sigma'' \vdash M : \phi \quad \text{avec} \quad \Sigma', \Sigma'' \gg x : \psi, \Gamma_2$$

  Soit $\Sigma^* = \Sigma'' - \Sigma''_x$; alors $x : \pi, \Sigma'' \gg x : \pi, x : \psi, \Sigma^*$. En utilisant (L8) deux fois on prouve

  $$x : \pi \times \psi, \Sigma^* \vdash M : \phi$$

  Par ailleurs, $\Sigma'_N \vdash N : \pi$ et $\Sigma'_N, \Sigma^* \gg \Gamma_2$. Donc,

  $$\Sigma'_N, \Gamma_1 \vdash (N \mid Q) : \pi \times \psi \quad \text{et} \quad \Sigma'_N, \Gamma_1, \Sigma^* \vdash M\langle P/ :\rangle \phi$$

  par les règles (L5) et (L4) appliquées successivement, d'où on dérive $\Gamma \vdash T : \phi$ par (L8).

- Soit $T \equiv T''$ et $T'' \triangleright T'$. Par h.r. $\Gamma \vdash T'' : \phi$, d'où $\Gamma \vdash T : \phi$ par la proposition 5.3.1.

□

Théorème 5.3.3 ***(Stabilité du typage par expansion)*** *Pour tout $M \in \Lambda_{rc}$,*

$$M \rightarrow_{rc} M' \ \& \ \Gamma \vdash M' : \phi \ \Rightarrow \ \Gamma \vdash M : \phi$$

Preuve. Le théorème est conséquence directe du lemme 5.3.2, car $M \rightarrow_{rc} M'$ implique $M \triangleright M'$. □

Nous démontrons ensuite une propriété d'extensionnalité (cf. Hindley [42, 43]) pour le système de types $\mathcal{P}$, qui correspond à la règle $\Gamma \vdash M\phi\wedge(\omega \rightarrow \omega)$ & $x \notin fv(M) \Rightarrow \Gamma \vdash (\lambda x.(Mx))$ nécessaire pour la complétude (cf. [32]).

Lemme 5.3.4 ***(Lemme d'extensionnalité)***
Soit $x \notin fv(M)$, $x \notin \Gamma$. Alors,

$$\Gamma \vdash M : \pi \rightarrow \sigma \Leftrightarrow \Gamma \vdash M : \gamma \ \& \ x : \pi, \Gamma \vdash Mx^{\infty} : \sigma$$

Preuve. L'implication $\Rightarrow$ est simple à démontrer : si $\Gamma \vdash M : \pi \to \sigma$, par (L9) on a $\Gamma \vdash M : \gamma$, car $\pi \to \sigma \leq \pi \to \omega \leq \omega \to \omega = \gamma$ est prouvable en appliquant la troisième et la deuxième clause de la définition de $\leq$ (dans cet ordre). Or, $x : \pi \vdash x^\infty : \pi$, en particulier pour tout $x \notin \Gamma$. On déduit $x : \pi, \Gamma \vdash Mx^\infty : \sigma$ en utilisant (L3).

En ce qui concerne l'implication $\Leftarrow$, supposons $\Gamma \vdash M : \gamma$ et $\{x : \pi, \Gamma\} \vdash Mx^\infty : \sigma$, où $x \notin \Gamma$. Dans le cas où $\sigma = \omega$, on sait que $\omega \to \omega \leq \pi \to \omega \leq \pi \to \sigma$, alors $\Gamma \vdash M : \pi \to \sigma$ par (L9). Soit $\sigma \neq \omega$. On prouve $\Gamma \vdash M : \pi \to \sigma$ par récurrence sur la taille l de la dérivation $\{x : \pi, \Gamma\} \vdash Mx^\infty : \sigma$, sans compter les applications de (L8). On examine deux cas possibles selon la dernière règle différente de (L8) utilisée :

(L3) Dans ce cas, $\Sigma \vdash M : \psi \to \sigma$ et $\Delta \vdash x^\infty : \psi$, où $\Sigma, \Delta \gg \{x : \pi, \Gamma\}$. Même si Σ contient éventuellement des hypothèses sur x, du fait que $x \notin fv(M)$, on déduit $\Sigma_M \vdash M : \psi \to \sigma$. Comme $\Sigma_M \gg \Gamma$, on a $\Gamma \vdash M : \psi \to \sigma$ par (L8). Si $\psi \sim \omega$, alors $\psi \to \sigma \leq \pi \to \sigma$ pour tout π; donc, $\Gamma \vdash M : \pi \to \sigma$ par (L9).

Sinon, i.e. $\psi \nsim \omega$, la proposition 5.2.7(4) garantit

$$\exists\, x : \psi_1, \ldots, x : \psi_m \in \Delta \;\; \psi_1 \times \cdots \times \psi_m \leq \psi$$

D'autre part, $\Delta_x \gg x : \pi$ implique $\pi \leq \psi_1 \times \ldots \psi_m$. Donc $\pi \leq \psi$ et $\psi \to \sigma \leq \pi \to \sigma$, d'où $\Gamma \vdash M : \pi \to \sigma$ par (L9).

(L9) On a $\{x : \pi, \Gamma\} \vdash Mx^\infty : \sigma'$ avec $\sigma' \leq \sigma$. Par h.r., $\Gamma \vdash M : \pi \to \sigma'$; donc, $\Gamma \vdash M : \pi \to \sigma$ par application de (L9). $\square$

5.4 Interprétation de λ_r^c dans $\mathcal{P}$

L'interprétation logique d'un terme, dans le modèle $\mathcal{C}$, coïncide avec l'ensemble des types que le système $\mathcal{P}$ permet d'affecter à ce terme. Plus précisément, la sémantique de λ_r^c engendrée par le modèle $\mathcal{C}$, $\sqsubseteq_{\mathcal{C}}$, coïncide avec la sémantique de λ_r^c dans le système de types $\mathcal{P}$.

Définition 5.4.1 ***(Sémantique de λ_r^c dans $\mathcal{P}$)***
Pour tout $M, N \in \Lambda_{rc}$,

$$M \sqsubseteq_{\mathcal{P}} N \overset{def}{\Leftrightarrow} \forall \Gamma\, \forall \phi \;\; (\Gamma \vdash M : \phi \;\Rightarrow\; \Gamma \vdash N : \phi)$$

L'équivalence associée est $\simeq_{\mathcal{P}}$.

Le préordre $\sqsubseteq_{\mathcal{P}}$ est compatible avec les constructions du calcul et clos par $=_\alpha$ et $\equiv$.

Lemme 5.4.2 – *$M \sqsubseteq_{\mathcal{P}} N \Rightarrow \forall C\; C[M] \sqsubseteq_{\mathcal{P}} C[N]$.*

- $(M =_\alpha N$ *ou* $M \equiv N) \Rightarrow M \simeq_{\mathcal{P}} N$.

Preuve. Le premier point est vérifié par récurrence sur le contexte C, en utilisant la proposition 5.2.6. Le deuxième est immédiat par définition du système $\mathcal{P}$ (voir figure 5.1). □

Notation 5.4.3 *On note* ξ_Γ *l'environnement compact qui vérifie* $\xi_\Gamma(x) = \uparrow\Gamma^\times(x)$ *pour toute variable* x *dans* Γ.

Soit $\xi \in EnvC$*; on note* Γ_ξ *le contexte* $\Gamma_\xi = \{x : \pi \,/\, \xi(x) = \uparrow\pi\}$.

Lemme 5.4.4 *Pour tout* Δ, Γ *ensembles d'hypothèses, on vérifie*

1. $\Delta \gg \Gamma \;\Rightarrow\; \xi_\Delta \subseteq \xi_\Gamma$
2. $\xi_{\Delta,\Gamma} = \xi_\Delta \cdot \xi_\Gamma$

Preuve.

1. On considère séparément les règles définissant $\gg$: l'énoncé est immédiatement vérifié pour les règles du produit, de l'affaiblissement et de l'échange lorsque les variables échangées sont différentes - dans les cas où elles sont la même variable, on utilise la commutativité de $\times$. En ce qui concerne l'affaiblissement qui transforme les hypothèses $x : \pi$ en $x : \psi$ avec $\psi \leq \pi$ (qu'on a prouvé dans le système $\mathcal{P}$, cf. lemme 5.2.2), on a $\Gamma^\times(x) \leq \Delta^\times(x)$. Donc $\uparrow\Delta^\times(x) \subseteq \uparrow\Gamma^\times(x)$.
2. Par définition, $\xi_{\Delta,\Gamma}(x) = \uparrow(\Delta, \Gamma)^\times(x) = \uparrow(\Delta^\times(x) \times \Gamma^\times(x)) = \xi_\Delta \cdot \xi_\Gamma$.

□

Lemme 5.4.5 *Soit* $\xi \in EnvC$*; l'égalité suivante est vérifiée:*

$$\mathcal{V}[\![N]\!]_\xi = \{\tau \,/\, \Gamma_\xi \vdash N : \tau\}$$

Preuve. L'inclusion de droite à gauche $\supseteq$, est conséquence de l'énoncé plus général suivant:

$$(*) \quad \Gamma \vdash N : \tau \;\Rightarrow\; \tau \in \mathcal{V}[\![N]\!]_{\xi_\Gamma}$$

En effet, $\Gamma_\xi \vdash N : \tau$ et $\xi_{\Gamma_\xi} = \xi$ impliquent $\tau \in \mathcal{V}[\![N]\!]_{\xi_\Gamma}$ en utilisant $(*)$. On prouve $(*)$ par récurrence sur la taille de la dérivation de $\Gamma \vdash N : \tau$.

(L1) Dans ce cas, $\Gamma = x : \tau$ et $N = x$. Comme $\mathcal{V}[\![x]\!]_{\xi_\Gamma} = j(\xi_\Gamma(x)) = \uparrow\tau$, on a $\tau \in \mathcal{V}[\![x]\!]_{\xi_\Gamma}$.

(L2) Dans ce cas, $N = \lambda x.M$ et $\tau = \pi \rightarrow \phi$. Soit $\Delta = x : \pi, \Gamma$; par récurrence on a $\phi \in \mathcal{V}[\![M]\!]_{\xi_\Delta}$. De plus, $\xi_\Delta = \xi_\Gamma[x := \uparrow\pi]$; alors $\pi \rightarrow \phi \in \mathcal{V}[\![N]\!]_{\xi_\Gamma}$ par définition de la fonction sémantique.

(L3) Ici $N = (MP)$. Supposons $\Delta \vdash P : \pi$, $\Sigma \vdash M : \pi \to \phi$ et $\Gamma = \Sigma, \Delta$. Par récurrence, $\pi \to \phi \in \mathcal{V}[\![M]\!]_{\xi_\Sigma}$. Considérons le cas où $\pi = \phi_1 \times \cdots \times \phi_n$ et $\pi \not\sim \omega$. Il existe $M_1, \ldots, M_n, R$ et $\Delta_1, \ldots, \Delta_n$ vérifiant $P \equiv (M_1 \mid \cdots \mid M_n \mid R)$, $\Delta_i \vdash M_i : \phi_i$ et $\Delta_1, \ldots, \Delta_n \gg \Delta$. Par application de l'hypothèse de récurrence, $\phi_i \in \mathcal{V}[\![M_i]\!]_{\xi_{\Delta_i}}$ pour tout i. On remarquera que $\xi_\Sigma, \xi_{\Delta_1}, \ldots, \xi_{\Delta_n}$ sont des environnements compacts; alors $\Sigma, \Delta_1, \ldots, \Delta_n \gg \Gamma$ implique $\xi_\Gamma \supseteq \xi_\Sigma \cdot \xi_{\Delta_1} \cdot \ldots \cdot \xi_{\Delta_n}$ par le lemme 5.4.4. On vérifie donc $\phi \in \mathcal{V}[\![MP]\!]_{\xi_\Gamma}$ puisque $P \propto (M_1 \mid \cdots \mid M_n)$.

Le cas où $\pi \sim \omega$ est immédiat.

(L4) Dans ce cas $N = M\langle P/x\rangle$. Supposons $\Gamma = \Sigma, \Delta$, $x \notin \Sigma$ et $x : \pi, \Sigma \vdash M : \tau$, $\Delta \vdash P : \pi$. On a $\tau \in \mathcal{V}[\![M]\!]_{\xi_{\Sigma[x:=\uparrow\pi]}}$ par récurrence. Soit $\pi = \phi_1 \times \cdots \times \phi_n$; il existe $M_1, \cdots, M_n$ et $\Delta_1, \cdots, \Delta_n$ tels que $P \propto (M_1 \mid \cdots \mid M_n)$ et $\Delta_i \vdash M_i : \phi_i$. Alors $\phi_i \in \mathcal{V}[\![M_i]\!]_{\xi_{\Delta_i}}$ par récurrence. On a $\tau \in \mathcal{V}[\![M\langle P/x\rangle]\!]_{\xi_\Gamma}$, car $\xi_\Gamma \supseteq \xi_\Sigma \cdot \xi_{\Delta_1} \cdot \ldots \cdot \xi_{\Delta_n}$ par le lemme 5.4.4.

(L8) Ici, $\Delta \gg \Gamma$ et $\tau \in \mathcal{V}[\![M]\!]_{\xi_\Delta}$. Donc $\xi_\Delta \subseteq \xi_\Gamma$ par le lemme 5.4.4, d'où le théorème découle par monotonie.

(L10) Ici, $N = \mathrm{c}P$, $\tau = \phi \to \phi$ et $\Gamma \vdash P : \gamma$. Par la proposition 5.2.6 , il existe M, Q, Γ' t.q. $P \equiv (M \mid Q)$ et $\Gamma \vdash M : \gamma$, où $\Gamma' \gg \Gamma$. Par récurrence (la dérivation du type de M est au plus de la taille de la dérivation du type de P), $\gamma \in \mathcal{V}[\![M]\!]_{\xi_{\Gamma'}}$. Par le lemme 5.4.4 et par monotonie, $\gamma \in \mathcal{V}[\![M]\!]_{\xi_\Gamma}$. Donc, $\phi \to \phi \in \mathcal{V}[\![\mathbf{I}]\!] = \mathcal{V}[\![\mathrm{c}P]\!]_{\xi_\Gamma}$.

(L7) et (L9) ont des preuves évidentes puisque, par définition, l'interprétation est un cône supérieur.

Nous démontrons ensuite l'inclusion $\mathcal{V}[\![N]\!]_\xi \subseteq \{\tau \,/\, \Gamma_\xi \vdash N : \tau\}$ par récurrence sur N. Pour ω l'énoncé est vérifié par (L7). Soit $\tau \not\sim \omega$ et $\tau \in \mathcal{V}[\![N]\!]_\xi$.

$N = x$: soit $\xi(x) = \uparrow\pi$; par hypothèse τ appartient à $j(\uparrow\pi) = \cup\{\uparrow\sigma \,/\, \sigma \in$ Ft $\& \uparrow\sigma \subseteq \uparrow\pi\}$; ceci implique $\pi \leq \tau$. Comme $x : \tau \vdash x : \tau$ est une instance de l'axiome (L1) et comme $x : \tau \gg x : \pi \gg \Gamma_\xi$, on a $\Gamma_\xi \vdash x : \tau$ par la règle (L8).

$N = \lambda x.M$: soit $\tau = \pi \to \phi \in \mathcal{V}[\![\lambda x.M]\!]_\xi$. C'est-à-dire,

$$\exists \psi \to \sigma \;\; \sigma \in \mathcal{V}[\![M]\!]_{\xi[x:=\uparrow\psi]} \;\&\; \psi \to \sigma \leq \pi \to \phi$$

De la propriété de la flèche, on déduit, soit $\phi = \omega$, auquel cas $\phi \in \mathcal{V}[\![M]\!]_{\xi[x:=\uparrow\psi]}$ car les dénotations sont non-vides, soit $\sigma \neq \omega \;\&\; \phi \neq \omega \;\&\; \pi \leq \psi \;\&\; \sigma \leq \phi$; dans le dernier cas, $\phi \in \mathcal{V}[\![M]\!]_{\xi[x:=\uparrow\psi]}$. Par monotonie, $\uparrow\psi \subseteq \uparrow\pi$ implique $\phi \in \mathcal{V}[\![M]\!]_{\xi[x:=\uparrow\psi]}$. D'où $\Gamma_{\xi[x:=\uparrow\pi]} \vdash M : \phi$ par récurrence.

On constate que $\Gamma_{\xi[x:=\uparrow\pi]} = \Gamma_{\xi/x}, x : \pi$; par conséquent $\Gamma_{\xi/x} \vdash \lambda x.M : \pi \to \phi$ est démontrable par application de (L2). De plus, $\Gamma_{\xi/x} \gg \Gamma_\xi$ par affaiblissements successifs; l'énoncé s'ensuit par application de (L8).

$N = (MP)$: $\tau \in \mathcal{V}[\![MP]\!]_\xi$ signifie qu'il existe $\pi, \psi_1, \ldots, \psi_n$, $M_1, \cdots, M_n$ et des environnements compacts $\xi_0, \xi_1, \cdots, \xi_n$ t.q.

- $P \propto (M_1 \mid \cdots \mid M_n)$
- $\xi \supseteq \xi_0 \cdot \ldots \cdot \xi_n$
- $\forall i \ \psi_i \in \mathcal{V}[\![M_i]\!]_{\xi_i}$
- $\pi \in \uparrow\psi_1 \times \cdots \times \psi_n$
- $\pi \to \tau \in \mathcal{V}[\![M]\!]_{\xi_0}$.

L'hypothèse de récurrence donne $\Gamma_{\xi_i} \vdash M_i : \psi_i$ et $\Gamma_{\xi_0} \vdash M : \pi \to \tau$. En utilisant la règle (L5) n fois on a

$$\Gamma_{\xi_1}, \cdots, \Gamma_{\xi_n} \vdash (M_1 \mid \cdots \mid M_n) : \pi$$

On prouve $\Gamma_{\xi_1}, \cdots, \Gamma_{\xi_n} \vdash P : \pi$ en affectant le type ω à chaque composante de P qui ne figure pas parmi les M_i. Par application de (L3) on obtient $\Gamma_{\xi_0}, \Gamma_{\xi_n} \vdash MP : \tau$. Finalement, $\xi \supseteq \xi_0 \cdot \ldots \cdot \xi_n$ implique $\Gamma_{\xi_0}, \cdots, \Gamma_{\xi_n} \gg \Gamma_\xi$. On conclut $\Gamma_\xi \vdash MP : \phi$ par (L8).

$N = M\langle P/x\rangle$: en faisant les mêmes hypothèses que pour l'application, sauf que ici $\xi \supseteq \xi_0' \cdot \ldots \cdot \xi_n$ où $\xi_0' = \xi_0/x$, on obtient $\tau \in \mathcal{V}[\![M]\!]_{\xi_0[x:=\uparrow\phi_1\times\cdots\times\phi_n]}$. Soit $\pi = \phi_1 \times \cdots \times \phi_n$; l'hypothèse de récurrence donne $\Gamma_{\xi_0[x:=\uparrow\pi]} \vdash M : \tau$. D'où $\Gamma_{\xi_i} \vdash M_i : \phi_i$ par h.r. sur l'hypothèse $\phi_i \in \mathcal{V}[\![M_i]\!]_{\xi_i}$. On a vu que le séquent $\Gamma_{\xi_1}, \cdots, \Gamma_{\xi_n} \vdash P : \pi$ est alors prouvable. Remarquer que $\Gamma_{\xi_0[x:=\uparrow\pi]} = \Gamma_{\xi_0'}, x : \pi$; par application de (L4) $\Gamma_{\xi_0'}, \cdots, \Gamma_{\xi_n} \vdash M\langle P/x\rangle : \tau$ est vérifié. Puisque $\Gamma_{\xi_0'}, \Gamma_{\xi_1}, \cdots, \Gamma_{\xi_k} \gg \Gamma_\xi$, la règle (L8) nous permet de déduire $\Gamma_\xi \vdash N : \tau$. la preuve est complètement similare dans le cas où $N = Q\langle P/x\rangle$.

$N = \mathrm{c}P$: par définition, $\tau \in \mathcal{V}[\![\mathbf{I}]\!]$ et $\gamma \in \mathcal{V}[\![M]\!]_\xi$, où $P \equiv (M \mid Q)$. Puisque $\tau = \pi \to \sigma$, et $\sigma \in \mathcal{V}[\![x]\!]_{[x:\uparrow\pi]}$, on a $\pi \leq \sigma$. D'autre part, par h.r. $\Gamma_\xi \vdash M : \gamma$. En donnant le type ω à Q, on a $\Gamma_\xi \vdash P : \gamma$. Donc, $\Gamma_\xi \vdash \mathrm{c}P : \sigma \to \sigma$ par (L10), d'où $\Gamma_\xi \vdash \mathrm{c}P : \pi \to \sigma$ par (L9). $\Box$

Ce lemme nous permet de relier la signification des termes dans des environnements arbitraires au système de types :

Théorème 5.4.6 $\mathcal{V}[\![M]\!]_\rho = \{\sigma \ / \ \exists\Gamma \ \Gamma \vdash M : \sigma \ \& \ \xi_\Gamma \subseteq \rho\}$

Preuve. Soit $\Gamma \vdash M : \sigma$ et $\xi_\Gamma \subseteq \rho$. Alors $\sigma \in \mathcal{V}[\![M]\!]_{\xi_\Gamma}$ par le lemme 5.4.5 précédent. Grâce à la monotonie de l'interprétation, nous avons $\sigma \in \mathcal{V}[\![M]\!]_\rho$.

Supposons maintenant que $\sigma \in \mathcal{V}[\![M]\!]_\rho$. C'est-à-dire, par continuité, $\sigma \in \mathcal{V}[\![M]\!]_\xi$ pour un certain environnement compact ξ tel que $\xi \subseteq \rho$. Une application du lemme 5.4.5 donne $\Gamma_\xi \vdash M : \sigma$. En conclusion, $\xi_{\Gamma_\xi} \subseteq \rho$ par transitivité parce que $\xi_{\Gamma_\xi} = \xi$. □

Théorème 5.4.7

$$\forall M, N \in \Lambda_{rc}\ M \sqsubseteq_{\mathcal{C}} N \Leftrightarrow M \sqsubseteq_{\mathcal{P}} N$$

Preuve. Soit M et N deux termes de Λ_{rc} tel que $M \sqsubseteq_{\mathcal{C}} N$. Supposons qu'il existe Γ et ϕ t.q. $\Gamma \vdash M : \phi$. Par application du théorème 5.4.6, $\phi \in \mathcal{V}[\![M]\!]_{\xi_\Gamma}$. De plus, $\Gamma_{\xi_\Gamma} \vdash N : \phi$ est conséquence du lemme 5.4.5. Par la proposition 5.2.5(1), $\tilde{\Gamma}_{\xi_\Gamma} \vdash N : \phi$; d'où $\Gamma \vdash N : \phi$, car on vérifie $\tilde{\Gamma}_{\xi_\Gamma} = \tilde{\Gamma} \gg \Gamma$.

L'autre implication est vérifiée par un raisonnement similaire. □

5.5 Adéquation de la convergence

La sémantique abstraite de λ_r^c engendrée par le système de types $\mathcal{P}$ est adéquate par rapport à la convergence : il est possible de caractériser le prédicat $\Downarrow_{rc}$ dans $\mathcal{P}$. Dans le cadre de la sémantique standard, il y a une seule forme de valeur : les abstractions, toutes confondues. Ce qui caractérise les abstractions du point de vue de l'affectation de types est qu'elles partagent le type γ, le plus grand type flèche de la théorie (il est clair que $\pi \to \sigma \leq \omega \to \omega = \gamma$ quelque soit π et σ.) Nous consacrons la section à montrer

$$\forall M \in \Lambda_{rc}\ \ M{\Downarrow_{rc}} \Leftrightarrow \vdash M : \gamma$$

Cette propriété d'adéquation étend le résultat obtenu pour λ_r (cf. [17]). La technique de preuve consiste à montrer la correction de la notion de typage dans $\mathcal{P}$ par rapport à un *prédicat de réalisabilité*[2] , $\Gamma \vDash T : \tau$, clos par $\sqsubseteq_{rc}$.

5.5.1 Réalisabilité dans $\mathcal{P}$

Définition 5.5.1 *Pour des termes clos de Λ_{rc}, la définition de $\vDash$ est :*

$$\begin{array}{lcl}
\vDash M : \omega & \overset{def}{\Leftrightarrow} & vrai \\[2ex]
\vDash M : \pi \to \phi & \overset{def}{\Leftrightarrow} & M{\Downarrow_{rc}}\ \&\ \forall P\ (\ \vDash P : \pi\ \Rightarrow\ \vDash MP : \phi) \\[2ex]
\vDash P : \pi & \overset{def}{\Leftrightarrow} & \pi \triangleright \phi_1, \ldots, \phi_n\ \ \&\ \ n > 0\ \ \& \\
 & & \exists R\ \forall i\ \exists M_i\ \ \vDash M_i : \phi_i\ \ \&\ \ P \equiv (M_1 \mid \cdots \mid M_n \mid R)
\end{array}$$

---

[2] Le prédicat $\vDash$ dont nous nous servons apparaît déjà dans la formulation de l'adéquation donnée dans [17].

Son extension à des termes ouverts de Λ_{rc} *est définie comme suit :*

$$\Gamma \vDash M : \phi \overset{def}{\Leftrightarrow} \forall P_1, \ldots, P_n \ (\forall i \ \vDash P_i : \pi_i \Rightarrow \ \vDash M\langle P_1/x_1\rangle \cdots \langle P_n/x_n\rangle : \phi)$$

où $\Gamma = \{x_1 : \pi_1, \ldots, x_n : \pi_n\}$ *est un ensemble fini d'hypothèses sans variables répétées t.q.* $fv(M) \subseteq \{x_1, \ldots, x_n\}$*.*

Dorénavant, étant donné $\Gamma \vDash M : \phi$, nous posons $\Gamma(x) = \pi$ lorsque $x : \pi \in \Gamma$, et $\Gamma(y) = \omega$ pour toute variable y absente de Γ. Le préordre engendré par $\vDash$ est défini par :

Définition 5.5.2 *Pour tout* $M, N \in \Lambda_{rc}$*,*

$$M \sqsubseteq_{\mathcal{R}} N \overset{def}{\Leftrightarrow} \forall \Gamma, \phi \ \ (\Gamma \vDash M : \phi \Rightarrow \Gamma \vDash N : \phi)$$

Proposition 5.5.3 *Pour tout* $M, N \in \Lambda_{rc}$*,* $M \sqsubseteq_{\mathcal{A}} N \ \Rightarrow \ M \sqsubseteq_{\mathcal{R}} N$*.*

Preuve. Soit $\Gamma = \{x_1 : \pi_1, \ldots, x_n : \pi_n\}$ et $\Gamma \vDash M : \phi$. On prouve $\Gamma \vDash N : \phi$ par récurrence structurelle sur ϕ. Si $\phi = \omega$, l'énoncé découle de la définition de $\vDash$. Dans le cas où $\phi = \pi \to \sigma$: soit $\vDash P_i : \pi_i$ pour $i = 1, \ldots, n$ et $\vDash Q : \pi$. Puisque $M \sqsubseteq_{\mathcal{A}} N$, on vérifie $M\langle \tilde{P}/\tilde{x}\rangle Q \sqsubseteq_{\mathcal{A}} N\langle \tilde{P}/\tilde{x}\rangle Q$ par la proposition 3.5.15. De $M\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc}$ on déduit $N\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc}$. Par ailleurs, $\vDash N\langle \tilde{P}/\tilde{x}\rangle Q : \sigma$ est vérifié par récurrence. Donc $\Gamma \vDash N : \pi \to \sigma$. □

Corollaire 5.5.4 *Pour tout* $M, N \in \Lambda_{rc}$*,* $M \asymp N \Rightarrow M \sqsubseteq_{\mathcal{R}} N$*.*

Preuve. Par le lemme 3.5.11 et la proposition 5.5.3. □

Proposition 5.5.5 *Soit* $M, P, Q \in \Lambda_{rc}$*. Si* $y \notin fv(M)$ *et* $\tilde{x} \notin fv(Q)$ *alors* $M\langle \tilde{P}/\tilde{x}\rangle Q \sim_{\mathcal{R}} (My^{\infty})\langle Q/y\rangle\langle \tilde{P}/\tilde{x}\rangle$

Preuve. Par les propositions 3.5.15(1) et 3.5.15(3), puis par la proposition 5.5.3. □

5.5.2 Correction de la réalisabilité

La preuve de la correction de l'interprétation des types $\vDash$ par rapport aux système de types nécessite le lemme suivant :

Lemme 5.5.6 $(\Gamma \vDash M : \phi \ \& \ \phi \leq \sigma) \Rightarrow \Gamma \vDash M : \sigma$

Preuve. Par récurrence sur σ, on prouve d'abord l'énoncé pour des termes clos, i.e. avec Γ vide. Si $\sigma = \omega$, pour tout M on a $\vDash M : \sigma$. Si $\sigma = \psi \to \sigma'$, le type ϕ est $\pi \to \phi'$, pour certains π, ϕ'. Par hypothèse, on a $M{\Downarrow}_{rc}$; ceci prouve l'énoncé pour $\sigma' = \omega$. Sinon, étant donné $\vDash P : \psi$, on montre $\vDash MP : \sigma'$. D'une part, en supposant $\psi \triangleright \phi_1, \ldots, \phi_n$, on constate que

$$P \equiv (M_1 \mid \cdots \mid M_n \mid R) \quad \text{et} \quad \vDash M_i : \phi_i \quad \text{pour tout } i$$

D'autre part, en appliquant la propriété de la flèche, on a $\psi \leq \pi$ et $\phi' \leq \sigma'$. Alors, par la propriété du produit, si $\pi \sim \delta_1 \times \ldots \times \delta_m$ et $J = \{j \,/\, \delta_j \neq \omega\}$ est de taille k, il existe une injection $i : J \to \{1, \ldots, n\}$ t.q. $\forall j \in J \;\; \phi_{i(j)} \leq \delta_j$. Par h.r., on a $\vDash M_{i(j)} : \delta_j$. Puisque les types différents des δ_i qui composent π sont nécessairement ω et que tout terme satisfait cette formule, $\vDash (M_{i(j_1)} \mid \cdots \mid M_{i(j_k)}) : \delta_{j_1} \times \ldots \times \delta_{j_k}$ implique $\vDash P : \pi$. Par hypothèse, $\vDash MP : \phi'$. L'hypothèse de récurrence donne $\vDash MP : \sigma'$.

Ceci termine la preuve de l'énoncé restreint à des termes clos. Pour des termes ouverts, si l'on pose $\Gamma = x_1 : \pi_1, \ldots, x : \pi_n$, l'hypothèse s'écrit $\vDash M\langle \tilde{P}/\tilde{x}\rangle : \phi$, pour tout $\tilde{P}$ t.q. $\vDash P_i : \pi_i$. Alors $\vDash M\langle \tilde{P}/\tilde{x}\rangle : \sigma$, d'où l'on tire $\Gamma \vDash M : \sigma$ par définition de $\vDash$. $\Box$

Théorème 5.5.7 ***(Correction de $\vDash$)***

$$\Gamma \vdash M : \phi \;\Rightarrow\; \Gamma^{\times} \vDash M : \phi$$

Preuve. Par récurrence sur la longueur de l'inférence de $\Gamma \vdash M : \phi$. On suppose $\Gamma^{\times} = x_1 : \pi_1, \ldots, x_k : \pi_k$ pour $k \geq 0$.

(L1) On prouve $x : \phi \vDash x : \phi$, en montrant que $\vDash x\langle P/x\rangle : \phi$ est vrai pour tout P qui satisfait $P \equiv (N \mid Q)$ et $\vDash N : \phi$.

Du fait que N est clos, on déduit $x\langle P/x\rangle \to_{rc} N\langle Q/x\rangle \asymp N$. Alors $N \sqsubseteq_{\mathcal{A}} x\langle P/x\rangle$ et, aussi, $\vDash x\langle P/x\rangle : \phi$ par la proposition 5.5.3.

(L2) L'hypothèse de récurrence donne $x : \pi, \Gamma^{\times} \vDash N : \sigma$ (donc $\phi = \pi \to \sigma$ et $M = \lambda x.N$), avec $x \notin \Gamma^{\times}$. Soit $P_1, \ldots, P_k, P$ des paquets de termes clos t.q. $\vDash P_i : \pi_i$ pour tout i et $\vDash P : \pi$. D'un côté on a $\vDash N\langle P/x\rangle\langle P_1/x_1\rangle \ldots \langle P_k/x_k\rangle : \sigma$, de l'autre on a

$$(\lambda x.N)\langle \tilde{P}/\tilde{x}\rangle P \overset{\star}{\to}_{rc} (\lambda x.(N\langle \tilde{P}/\tilde{x}\rangle))P \overset{\beta}{\to}_{rc} N\langle \tilde{P}/\tilde{x}\rangle\langle P/x\rangle \asymp N\langle P/x\rangle\langle \tilde{P}/\tilde{x}\rangle$$

car $\tilde{P}, P$ sont clos et $x \neq x_i$ pour tout i. Le lemme 3.5.6 et la proposition 5.5.3 donnent $\vDash N\langle \tilde{P}/\tilde{x}\rangle\langle P/x\rangle : \sigma$ et $N\langle P/x\rangle\langle \tilde{P}/\tilde{x}\rangle \sqsubseteq_{\mathcal{R}} (\lambda x.N)\langle \tilde{P}/\tilde{x}\rangle P$. Par conséquent, $\vDash (\lambda x.N)\langle \tilde{P}/\tilde{x}\rangle P : \sigma$. C'est-à-dire, $\Gamma \vDash (\lambda x.N) : \pi \to \sigma$.

(L3) Soit $\Sigma^{\times} \vDash N : \pi \to \phi$ et $\Delta \vdash P : \pi$ (donc $M = (NP)$) avec $\pi \triangleright \phi_1, \ldots, \phi_n$.

D'après la proposition 5.2.6(1), $\exists M_1, \ldots, M_n, R \; \exists \Delta_1, \ldots, \Delta_n$ t.q. $P \equiv (M_1 \mid \ldots \mid M_n \mid Q)$, $\Delta_i, \ldots, \Delta_n \gg \Delta$ et $\Delta_i \vdash M_i$ où les preuves de ces séquents sont plus courtes que celle de $\Delta \vdash P : \pi$. Ceci implique $\Delta_i^\times \vDash M_i : \phi_i$, $i = 1, \ldots, n$, par hypothèse de récurrence.

Soit $P_1, \ldots, P_k$ des paquets clos vérifiant $\vDash P_i : \pi_i$. Du fait que

$$\exists \delta_i \; \pi_i \sim \Sigma^\times(x_i) \times \Delta^\times(x_i) \sim \Sigma^\times(x_i) \times \Delta_1^\times(x_i) \times \ldots \times \Delta_n^\times(x_i) \times \delta_i$$

on peut trouver des paquets Q_i, R_j^i et S_i t.q. $P_i \equiv (Q_i \mid R_1^i \mid \cdots \mid R_n^i \mid S_i)$, avec $\vDash Q_i : \Sigma^\times(x_i)$ et $\vDash R_j^i : \Delta_j^\times(x_i)$. Il est clair que

$$\vDash N\langle \tilde{Q}/\tilde{x} \rangle : \pi \rightarrow \phi \text{ et } \vDash N_j : \phi_j \text{ où } N_j = M_j\langle \tilde{R}_j/\tilde{x} \rangle$$

et aussi que $\vDash (N_1 \mid \ldots \mid N_n | Q) : \pi$ par définition de $\vDash$. Donc,

$$\vDash (N\langle \tilde{Q}/\tilde{x} \rangle)(N_1 \mid \ldots \mid N_n | Q) : \sigma$$

Essentiellement par le lemme 3.5.14(3),

$$(N\langle \tilde{Q}/\tilde{x} \rangle)(N_1 \mid \ldots \mid N_n | Q) \sqsubseteq_{\mathcal{A}} N(M_1 \mid \ldots \mid M_n | Q)\langle \tilde{P}/\tilde{x} \rangle$$

En utilisant la proposition 5.5.3, on arrive à $\Gamma^\times \vDash (NP) : \phi$.

(L4) Comme dans les cas précédents.

(L7) Immédiat.

(L8) Soit $\Delta^\times \vDash M : \phi$ avec $\Delta \gg \Gamma$ et $\vDash P_i : \pi_i$ pour $i = 1, \ldots, k$. La proposition 5.2.6 donne $\pi_i \leq \Delta^\times(x_i)$ pour chaque x_i. Le prédicat $\vDash$ est clos par $\leq$ (cf. lemme 5.5.6); aussi $\vDash P_i : \Delta^\times(x_i)$ est-il vérifié. On en déduit $\vDash M\langle \tilde{P}/\tilde{x} \rangle : \phi$.

(L9) Par le lemme 5.5.6.

(L10) Soit $\Gamma \vdash P : \omega \rightarrow \omega$, $M = \mathrm{c}P$ et $\phi = \sigma \rightarrow \sigma$. Par la proposition 5.2.6, il existe N, Q, Δ t.q. $P \equiv (N \mid Q)$, $\Delta \gg \Gamma$ et $\Delta \vdash N : \omega \rightarrow \omega$, où l'inférence de ce séquent est plus courte que celle de $\Gamma \vdash P : \omega \rightarrow \omega$. L'hypothèse de récurrence donne $\Delta^\times \vDash N : \omega \rightarrow \omega$. Comme $\Delta^\times \gg \Gamma^\times$, on a $\Gamma^\times \vDash N : \omega \rightarrow \omega$ en appliquant le lemme 5.5.6.

Pour $\vDash Q_i : \pi_i$ et $\vDash R : \sigma$, on doit montrer $(\mathrm{c}P)\langle \tilde{Q}/\tilde{x} \rangle {\Downarrow_{rc}}$ et $\vDash (\mathrm{c}P)\langle \tilde{Q}/\tilde{x} \rangle R : \sigma$. On remarque que $\vDash N\langle \tilde{Q}/\tilde{x} \rangle : \gamma$, i.e. $N\langle \tilde{Q}/\tilde{x} \rangle {\Downarrow_{rc}}$. De plus, $\mathrm{c}(N\langle \tilde{Q}/\tilde{x} \rangle) \asymp (\mathrm{c}N)\langle \tilde{Q}/\tilde{x} \rangle$ implique $(\mathrm{c}N)\langle \tilde{Q}/\tilde{x} \rangle) {\Downarrow_{rc}}$ par le lemme 3.5.11. On constate que ce terme converge vers l'identité et que $(\mathrm{c}P)\langle \tilde{Q}/\tilde{x} \rangle {\Downarrow_{rc}}$. Par ailleurs,

$$(\mathrm{c}P)\langle \tilde{Q}/\tilde{x} \rangle) R \overset{\star}{\rightarrow}_{rc} \mathbf{I} R \overset{\star}{\rightarrow}_{rc} R$$

On conclut $\vDash (\mathrm{c}P)\langle \tilde{Q}/\tilde{x} \rangle R : \sigma$ par la proposition 5.5.3. □

Le théorème suivant décrit la convergence en fonction de la notion d'affectation de types dans le système $\mathcal{P}$ (c'est la dernière pièce nécessaire à la démonstration de l'adéquation du modèle $\mathcal{C}$, présentée dans la section 5.6.2.)

Théorème 5.5.8 ***(Adéquation de la convergence)***
Pour tout M terme clos de Λ_{rc}, $M{\Downarrow_{rc}} \Leftrightarrow\ \vdash M : \gamma$.

Preuve. Soit $M\overset{\star}{\rightarrow}_{rc}V$; comme pour toute abstraction, $\vdash V : \gamma$. Alors $\vdash M : \gamma$ par le théorème 5.3.3. Ceci prouve l'implication $\Rightarrow$. Supposons maintenant que $\vdash M : \gamma$. Par le théorème de correction, $\vDash M : \gamma$. C'est-à-dire, $M{\Downarrow_{rc}}$. $\square$

Une des conséquences du théorème 5.5.8 est que ω est le seul type du terme Ω. Supposons $\vdash \Omega : \phi$ où $\phi \neq \omega$. Alors, $\phi = \pi \rightarrow \sigma \leq \omega \rightarrow \omega$ implique $\vdash \Omega : \gamma$ par (L9). Une application du théorème 5.5.8 donne $\Omega{\Downarrow_{rc}}$. Puisque toute évaluation finie de Ω est susceptible de continuation, n'atteignant jamais une valeur, l'hypothèse amène à contradiction.

5.6 Adéquation complète du modèle $\mathcal{C}$ sur λ_r^c

Dans cette section nous montrons l'équivalence :

$$\forall M, N \in \Lambda_{rc}\ \ M \sqsubseteq_{\mathcal{P}} N \Leftrightarrow M \sqsubseteq_{rc} N$$

d'où découle l'adéquation complète du modèle $\mathcal{C}$ par rapport au langage λ_r^c. La partie $\Rightarrow$ est prouvée essentiellement par le théorème 5.5.8. Pour la partie $\Leftarrow$, nous montrons la complétude de la réalisabilité : $\Gamma \vDash M : \phi \Rightarrow \Gamma \vdash M : \phi$, dont la preuve repose sur l'existence de *termes et paquets caractéristiques* pour chaque type.

5.6.1 Termes caractéristiques

On définit des *paquets caractéristiques* P_τ qui vérifient $\vdash P_\tau : \tau$ pour chaque type τ. Lorsque $\tau \in$ Ft , on écrit M_τ. En même temps que le paquet caractéristique P_τ, on construit une abstraction T_τ supposée tester si un terme appartient à l'interprétation de τ, i.e.

$$\vDash P : \tau \ \Rightarrow\ (T_\tau P){\Downarrow_{rc}}\mathbf{I}$$

ceci implique $\vdash P : \tau \Rightarrow (T_\tau P){\Downarrow_{rc}}\mathbf{I}$ par le théorème de correction 5.5.7.

La définition est par récurrence mutuelle sur τ. Pour $\tau = \omega$, le choix est clair :

$$M_\omega = \Omega \qquad \text{et} \qquad T_\omega = \lambda x.\mathbf{I}$$

Le terme caractéristique de type flèche $\pi \rightarrow \sigma$ est la fonction (l'abstraction) qui prend un argument de type π et rend le terme caractéristique de type σ. Dans un calcul typé, $M_{\pi\to\sigma}$ serait $\lambda x\!:\!\pi.M_\sigma$. Nous nous servons du terme T_π pour contrôler que l'argument appliqué à l'abstraction ait le type π. C'est-à-dire,

$$M_{\pi\to\sigma} = \lambda x.(T_\pi x^\infty)M_\sigma$$

Remarquons que $M_{\pi\to\omega}$ est essentiellement $\lambda x.\Omega$ quelque soit π. Aucun contrôle n'est effectué sur l'argument parce que $\pi \to \omega \sim \omega \to \omega$. La définition de $T_{\pi\to\sigma}$ mérite une explication. Rappelons que $\vDash N : \pi \to \sigma$ signifie $N{\Downarrow}_{rc}$ et $\vDash NQ : \sigma$ pour tout Q t.q. $\vDash Q : \pi$. En particulier, $\vDash NP_\pi : \sigma$ est vérifié par récurrence. L'abstraction $T_{\pi\to\sigma}$ est supposée tester la convergence de l'argument, puis vérifier si l'application de cet argument à P_π réussit le test associé à σ. Dans les calculs faibles standards étendus avec test de convergence, $T_{\pi\to\sigma}$ est $\lambda x.(\mathrm{c}x)(T_\sigma(xP_\pi))$. Cette définition ne convient pas au paradigme des ressources, car elle repose sur l'hypothèse que l'argument N peut être utilisé deux fois pendant l'évaluation. Pour résoudre ce problème, nous divisons la définition de $T_{\pi\to\sigma}$ en deux cas, comme suit :

$$T_{\pi\to\sigma} = \begin{cases} \lambda x.(\mathrm{c}x) & \text{si } \sigma = \omega \\ \lambda x.T_\sigma(xP_\pi) & \text{sinon} \end{cases}$$

Il faut souligner que cette définition est appropriée car

$$MQ_1 \ldots Q_n{\Downarrow}_{rc} \Rightarrow (M{\Downarrow}_{rc} \;\&\; \forall i \in [1,n]\; MQ_1 \ldots Q_i{\Downarrow}_{rc})$$

Autrement dit, on pourrait définir $\vDash M : \pi \to \sigma$ par

$$\vDash M : \pi \to \omega \;\overset{def}{\Leftrightarrow}\; M{\Downarrow}_{rc}$$

$$\vDash M : \pi \to \sigma \;\overset{def}{\Leftrightarrow}\; \sigma \neq \omega \Rightarrow (\forall P \;\vDash P : \pi \Rightarrow \;\vDash MP : \sigma)$$

Les paquets caractéristiques pour les types produit sont construits en utilisant la composition parallèle :

$$P_{\pi\times\psi} = (P_\pi \mid P_\psi)$$

Intuitivement, le test pour $\pi\times\psi$ doit vérifier, successivement, que l'argument a les deux types π et ψ; i.e. $T_{\pi\times\psi} = \lambda x.(T_\pi x^\infty)(T_\psi x^\infty)$. Or ce terme peut mener à une conclusion erronée; par exemple lorsque $\pi = \omega$ et l'argument est P_ψ. Dans ce cas, $T_{\omega\times\psi}P_\psi$ diverge. Le terme approprié n'effectue pas de test sur les composantes ω du type :

$$\pi \sim \phi_1 \times \ldots \times \phi_n \;\&\; \forall i\; \phi_i \neq \omega \;\Rightarrow\; T_\pi = \lambda x.(T_{\phi_1}x^\infty)\ldots(T_{\phi_n}x^\infty)$$

Pour terminer, nous examinons l'utilisation des multiplicités dans la construction des termes caractéristiques. Des multiplicités infinies ont été employées systématiquement lorsque, en réalité, des exposants finis sont suffisants. La première observation à faire est que le test associé aux types dans Ft n'utilise son argument qu'une fois; le terme $T_{\pi\to\sigma}$ est linéaire et la variable abstraite apparaît en position fonctionnelle. La deuxième remarque est que le test associé aux types produit est composé d'un nombre fini de tests "simples" (tests des types flèche). Donc, les multiplicités infinies peuvent être remplacées par la multiplicité 1 dans T_π. Par conséquent, le terme caractéristique $M_{\pi\to\sigma}$ n'a besoin que de permettre autant d'accès à son argument que de types différents de ω dans π. On pourrait définir $M_{\pi\to\sigma} = \lambda x.(T_\pi x^n)M_\sigma$ si $\pi \triangleright \phi_1, \ldots, \phi_n$.

Proposition 5.6.1

1. $\vdash P_\tau : \tau$

2. $\vdash T_\tau : \tau \to (\sigma \to \sigma)$ *pour tout* $\sigma \in \mathrm{Ft}$

Preuve. Immédiate, par construction des termes. □

Lemme 5.6.2 ***(Lemme de caractérisation)***

1. $\Gamma \vdash P_\tau : \tau' \Leftrightarrow \tau \leq \tau'$ *et*

2. $\Gamma \vdash T_\tau : \tau' \to (\xi \to \sigma) \Leftrightarrow \tau' \leq \tau \ \& \ \xi \leq \sigma$

Preuve. Les parties $\Leftarrow$ des points (1) et (2) sont des corollaires de la proposition précédente et de la règle (L9) du système de typage.

Les démonstrations des parties $\Rightarrow$ sont par récurrence sur le type τ et sur les tailles des dérivations associées à P_τ pour l'énoncé (1) et à T_τ pour l'énoncé (2). Nous détaillons les cas les plus significatifs et décorons souvent les références aux énoncés (1), (2) du type τ correspondant.

Pour tout τ, si $\Gamma \vdash P_\tau : \tau'$ est dérivé par (L7), $\tau' = \omega$ implique $\tau \leq \tau'$. Quand les dérivations associées à P_τ et à T_τ terminent par application d'une des règles (L8) ou (L9), on vérifie les énoncés (1) et (2) par hypothèse de récurrence et transitivité de $\leq$.

Toute inférence de types pour $M_{\pi\to\sigma}$ et $T_{\pi\times\psi}$ entraîne le typage de leurs sous-termes $T_\pi x^\infty$ et $T_\psi x^\infty$. La proposition suivante simplifiera notre preuve :

$$(3)\ \Gamma \vdash T_\tau x^\infty : \xi \to \sigma \ \Rightarrow \ \xi \leq \sigma \ \&$$

$$(\tau \nsim \omega \ \Rightarrow \ \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma \ \ \psi_1 \times \ldots \times \psi_n \leq \tau\,)$$

On montre facilement que $(2\tau) \Rightarrow (3\tau)$ pour tout τ : Si l'inférence du type de $T_\tau x^\infty$ termine par (L3), on a $\Gamma_1 \vdash T_\tau : \psi \to (\xi \to \sigma)$ et $\Gamma_2 \vdash x^\infty : \psi$ avec $\Gamma = \Gamma_1, \Gamma_2$. En utilisant (2τ), on vérifie $\psi \leq \tau$ et $\xi \leq \sigma$. Deux cas sont à examiner : pour $\psi \sim \omega$, on a $\tau \sim \omega$ directement. Sinon, par la proposition 5.2.7(4),

$$\exists x : \psi_1, \ldots, x : \psi_n \in \Gamma_2 \ \ \psi_1 \times \ldots \times \psi_n \leq \psi$$

Donc, $\psi_1 \times \ldots \times \psi_n \leq \tau$ par transitivité.

Dans le cas où l'inférence de $\Gamma \vdash T_\tau x^\infty : \xi \to \sigma$ terminerait par une application de (L8), c'est-à-dire, où

$$\Delta \vdash T_\tau x^\infty : \xi \to \sigma \quad \text{avec } \Delta \gg \Gamma\,,$$

l'hypothèse de récurrence donnerait $\xi \le \sigma$ ainsi que

$$\tau \nsim \omega \;\Rightarrow\; \exists x : \psi'_1, \ldots, x : \psi'_m \in \Delta \;\; \psi'_1 \times \ldots \times \psi'_m \le \tau$$

Puisque l'ensemble de règles définissant $\gg$ n'autorise qu'à affaiblir les hypothèses, on a $\exists x : \psi_1, \ldots, x : \psi_n \in \Gamma \;\; \psi_1 \times \ldots \times \psi_n \le \psi'_1 \times \ldots \times \psi'_m \le \tau$.

Pour une dérivation terminant par application de (L9), i.e. $\Gamma \vdash T_\tau x^\infty : \delta$ où $\delta \le \xi \to \sigma$, on déduit $\delta = \xi' \to \sigma'$ car δ est dans $\mathtt{Ft}$. Par h.r. $\xi' \le \sigma'$ et

$$\tau \nsim \omega \;\Rightarrow\; \exists x : \psi_1, \ldots, x : \psi_n \in \Gamma \;\; \psi_1 \times \ldots \times \psi_n \le \tau$$

Par la propriété de la flèche, si $\sigma = \omega$ alors $\xi \le \sigma$ est corroboré immédiatement. Sinon, $\sigma' \neq \omega$ et aussi $\xi \le \xi'$ et $\sigma' \le \sigma$. Il en découle que $\xi \le \sigma$, par transitivité.

$\tau = \omega$: (1) Le seul type du terme Ω est ω alors $\tau \le \tau' = \omega$ par reflexivité.
(2) Si l'inférence de $\Gamma \vdash \lambda x.\mathbf{I} : \tau' \to (\xi \to \sigma)$ termine par (L2), $x : \tau', \Gamma \vdash \mathbf{I} : \xi \to \sigma$ implique $\xi \le \sigma$; $\tau' \le \omega$ est vérifié immédiatement.

$\tau = \pi \to \omega$: (1) Si la dérivation termine par une application de (L2), i.e. $\tau' = \psi \to \sigma$ et $x : \psi, \Gamma \vdash \Omega : \sigma$, on déduit $\pi \to \omega \le \gamma \le \psi \to \omega$ du fait que ω est le seul type de Ω.
(2) Quand la dernière règle utilisée dans la preuve de $\Gamma \vdash \lambda x.(\mathrm{c}x) : \tau' \to (\xi \to \sigma)$ est (L2), à partir de $x : \tau', \Gamma \vdash \mathrm{c}x : \xi \to \sigma$ et $x \notin \Gamma$ on a $\tau' \le \gamma$, puis $\xi \le \sigma$ par la proposition 5.2.7(3).

$\tau = \pi \to \sigma$ et $\sigma \neq \omega$: (1) Soit $\Gamma \vdash \lambda x.(T_\pi x^\infty)M_\sigma : \tau'$ dont la dérivation termine par (L2); donc, on a $\tau' = \pi' \to \sigma'$ et $x : \pi', \Gamma \vdash (T_\pi x^\infty)M_\sigma : \sigma'$, avec $x \notin \Gamma$. Si la dernière règle est (L3), alors $\Gamma_1 \vdash T_\pi x^\infty : \xi' \to \sigma'$ et $\Gamma_2 \vdash M_\sigma : \xi'$, où $x : \pi', \Gamma = \Gamma_1, \Gamma_2$. Puisque (2 π) est vérifié par h.r., la proposition (3π) donne $\xi' \le \sigma'$ et $\pi' \le \pi$. De plus, $\sigma \le \xi'$ est vrai par h.r. , ce qui implique $\sigma \le \sigma'$ par transitivité. En conjonction avec $\pi' \le \pi$, la propriété de la flèche donne $\pi \to \sigma \le \pi' \to \sigma'$.
(2) Sous l'hypothèse $\sigma \neq \omega$, nous montrons $\tau' \le \pi \to \sigma$ et $\xi \le \sigma$ pour une dérivation typique de $\Gamma \vdash \lambda x.T_\sigma(xP_\pi) : \tau' \to (\xi \to \sigma)$, c'est-à-dire, à utilisation de (L8) et (L9) près :

$$
(\mathrm{L2})\ \dfrac{(\mathrm{L3})\ \dfrac{\overset{(D1)}{\vdots}\atop{\Gamma_1 \ \vdash\ T_\sigma : \psi \rightarrow (\xi \rightarrow \sigma)} \qquad (\mathrm{L3})\ \dfrac{\overset{(D2)}{\vdots}\atop{x : \tau' \ \vdash\ x : \theta \rightarrow \psi} \quad \overset{(D3)}{\vdots}\atop{\Gamma_2 \ \vdash\ P_\pi : \theta}}{x : \tau', \Gamma_2 \vdash xP_\pi : \psi}}{x : \tau', \Gamma_1, \Gamma_2 \vdash T_\sigma(xP_\pi) : \xi \rightarrow \sigma}}{\Gamma = \Gamma_1, \Gamma_2 \vdash \lambda x.T_\sigma(xP_\pi) : \tau' \rightarrow (\xi \rightarrow \sigma)}
$$

On remarque que :

- $\psi \leq \sigma$ et $\xi \leq \sigma$, par h.r. sur $(D1)$. Alors il suffit de prouver $\tau' \leq \tau$.

- $\tau' \leq \theta \rightarrow \psi$ à partir de $(D2)$ et en appliquant la propriété 5.2.7(1); ainsi, il existe π', σ' tel que $\tau' = \pi' \rightarrow \sigma'$. La propriété de la flèche donne $\psi \neq \omega$, car $\sigma \neq \omega$. Par conséquent, $\theta \leq \pi'$ et $\sigma' \leq \psi$.

- $\pi \leq \theta$, par h.r. sur $(D3)$.

Ces résultats réunis impliquent $\pi \leq \pi'$ et aussi $(\sigma' \leq \sigma \Rightarrow \tau' = \pi' \rightarrow \sigma' \leq \pi \rightarrow \sigma = \tau)$.

$\tau = \pi \times \psi$: (1) D'après la proposition 5.2.6, les composantes de P_τ ont les types suivants :

$$\Gamma_1 \vdash P_\pi : \pi' \quad \text{et} \quad \Gamma_2 \vdash P_\psi : \psi'$$

où $\tau' \sim \pi' \times \psi'$ and $\Gamma_1, \Gamma_2 \gg \Gamma$. Le résultat découle de l'hypothèse de récurrence sur ces séquents, dont les dérivations sont plus courtes que celle de $\Gamma \vdash : P_\tau : \tau'$.
(2) Comme pour le cas de la flèche, nous omettons les détails de la récurrence nécessaire à la preuve, et montrons le résultat sur une dérivation typique de $\Gamma \vdash T_{\phi_0 \times \phi_1} : \tau' \rightarrow (\xi \rightarrow \sigma)$, où $\phi_i \not\sim \omega$:

$$
(\mathrm{L2})\ \dfrac{\overset{(\mathrm{L8})}{\tau' = \psi_1 \times \psi_2}\ \dfrac{\overset{(D1)}{\vdots}\atop{x : \psi_1, \Gamma_1 \ \vdash\ T_{\phi_0} x^\infty : \theta \rightarrow (\xi \rightarrow \sigma)} \qquad \overset{(D2)}{\vdots}\atop{x : \psi_2, \Gamma_2 \ \vdash\ T_{\phi_1} x^\infty : \theta}}{x : \tau', \Gamma_1, \Gamma_2 \vdash (T_{\phi_0} x^\infty)(T_{\phi_1} x^\infty) : \tau' \rightarrow (\xi \rightarrow \sigma)}}{\Gamma = \Gamma_1, \Gamma_2 \vdash \lambda x.(T_{\phi_0} x^\infty)(T_{\phi_1} x^\infty) : \tau' \rightarrow (\xi \rightarrow \sigma)}
$$

Quelques conséquences de cette dérivation sont :

- à partir de $(D1)$ et $(3\phi_0)$, on a $\psi_1 \leq \phi_0$ et $\theta \leq \xi \rightarrow \sigma$. Donc, il existe ξ', σ' vérifiant $\theta = \xi' \rightarrow \sigma'$.

- à partir de $(D2)$ et $(3\phi_1)$, on a $\psi_2 \leq \phi_1$ et $\xi' \leq \sigma'$. Alors, $\tau' = \psi_1 \times \psi_2 \leq \phi_0 \times \phi_1 \sim \tau$. D'autre part, $\sigma = \omega$ implique $\xi \leq \sigma$. Dans le cas où $\sigma \neq \omega$, par la propriété de la flèche, σ' est aussi différent aussi de ω; de plus, $\xi \leq \xi'$ and $\sigma' \leq \sigma$. En utilisant le fait que $\xi' \leq \sigma'$, on conclut $\xi \leq \sigma$ par transitivité. □

5.6.2 Complétude de la réalisabilité

Le lemme suivant est une instance importante du théorème de complétude :

Lemme 5.6.3 ***(Lemme de Convergence)***

$$\Gamma \vDash M : \gamma \;\Rightarrow\; \Gamma \vdash M : \gamma$$

Preuve. Soit $\Gamma \vDash M : \gamma$ où $fv(M) \subseteq \{x_1, \ldots, x_n\}$ et $\Gamma(x_i) = \pi_i$ pour tout $1 \leq i \leq n$. Alors $M\langle P_{\pi_1}/x_1\rangle \cdots \langle P_{\pi_n}/x_n\rangle \Downarrow_{rc}$ par définition puisque $\vDash P_{\pi_i} : \pi_i$. Le théorème d'adéquation de la convergence 5.5.8 donne

$$\vdash M\langle P_{\pi_1}/x_1\rangle \cdots \langle P_{\pi_n}/x_n\rangle : \gamma$$

Par la proposition 5.2.6, il existe $\psi_1, \ldots, \psi_n$ t.q. $\vdash P_{\pi_i} : \psi_i$ pour tout i et $x_1 : \psi_1, \ldots, x_n : \psi_n \vdash M : \gamma$. On obtient, par le lemme de caractérisation 5.6.2, $\pi_i \leq \psi_i$, donc $x_1 : \psi_1, \ldots, x_n : \psi_n \gg x_1 : \pi_1, \ldots, x_n : \pi_n \gg \Gamma$. On conclut $\Gamma \vdash M : \gamma$ par application de (L8). □

Théorème 5.6.4 ***(Complétude de $\vDash$)***

$$\Gamma \vDash M : \phi \Rightarrow \Gamma \vdash M : \phi$$

Preuve. Étant donné $\Gamma \vDash M : \phi$, on montre $\Gamma \vdash M : \phi$ par récurrence sur ϕ. Si $\phi = \omega$ alors $\Gamma \vdash M : \omega$ par (L7). Si $\phi = \pi \rightarrow \omega$, alors $\Gamma \vDash M : \gamma$ et l'énoncé est vérifié par le lemme de la convergence 5.6.3. Soit $\phi = \pi \rightarrow \sigma$ avec $\sigma \neq \omega$ et $\Gamma = x_1 : \pi_1, \ldots, x_n : \pi_n$. L'hypothèse $\Gamma \vDash M : \phi$ implique, pour tout $P_1, \ldots, P_n, Q$ t.q. $\vDash P_i : \pi_i$ et $\vDash Q : \pi$,

$$\begin{array}{ll} (*) & M\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc} \quad \text{et} \\ (**) & \vDash M\langle \tilde{P}/\tilde{x}\rangle Q : \sigma \end{array}$$

La conclusion $\Gamma \vdash M : \phi$ est vérifié par le lemme d'extensionnalité 5.3.4, car $(*)$ implique $\Gamma : M : \gamma$, et $(**)$ implique $y : \pi, \Gamma \vdash My^\infty : \sigma$, où $y \notin \Gamma$, $y \notin fv(M)$. La séquence suivante d'implications est vérifiée par les théorèmes 5.5.8 et 5.5.7, la définition de $\vDash$ et le lemme 5.6.3 dans cet ordre :

$$M\langle \tilde{P}/\tilde{x}\rangle \Downarrow_{rc} \Rightarrow \vdash M\langle \tilde{P}/\tilde{x}\rangle : \gamma \;\Rightarrow\; \vDash M\langle \tilde{P}/\tilde{x}\rangle : \gamma \;\Rightarrow\; \Gamma \vDash M : \gamma \;\Rightarrow\; \Gamma \vdash M : \gamma$$

D'autre part, par la proposition 5.5.5,

$$M\langle \tilde{P}/\tilde{x}\rangle Q \sim_{\mathcal{R}} (My^\infty)\langle Q/y\rangle\langle \tilde{P}/\tilde{x}\rangle$$

Cette équivalence, plus $(**)$ impliquent $\vDash (My^\infty)\langle Q/y\rangle\langle \tilde{P}/\tilde{x}\rangle : \sigma$. Par définition, $y : \pi, \Gamma \vDash My^\infty : \sigma$. L'hypothèse de récurrence donne $y : \pi, \Gamma \vdash My^\infty : \sigma$. □

Théorème 5.6.5 ***(Adéquation)***

Pour tout $M, N \in \Lambda_{rc}$,

$$M \sqsubseteq_{\mathcal{P}} N \;\Rightarrow\; M \sqsubseteq_{rc} N$$

Preuve. Soit $M, N \in \Lambda_{rc}$ et C un λ_r^c-contexte. Alors,

$$\begin{tikzcd}
M \sqsubseteq_{\mathcal{P}} N \arrow[d, "\text{lemme 5.4.2}"] & \& & C[M]\Downarrow_{rc} \arrow[d, "\text{th. 5.5.8}"] \\
C[M] \sqsubseteq_{\mathcal{P}} C[N] \arrow[dr] & & \vdash C[M] : \gamma \arrow[dl] \\
& \vdash C[N] : \gamma \arrow[d, "\text{th. 5.5.8}"] & \\
& C[N]\Downarrow_{rc} &
\end{tikzcd}$$

□

Théorème 5.6.6 ***(Adéquation Complète)***
Pour tout $M, N \in \Lambda_{rc}$,

$$M \sqsubseteq_{rc} N \;\Rightarrow\; M \sqsubseteq_{\mathcal{P}} N$$

Preuve. Soit $M, N \in \Lambda_{rc}$, Γ un ensemble d'hypothèses et $\phi \in \mathrm{Ft}$. Alors,

$$\begin{tikzcd}
M \sqsubseteq_{rc} N \arrow[d, "\text{p. 5.5.3 + l. 3.5.5}"] & \& & \Gamma \vdash M : \phi \arrow[d, "\text{th. 5.5.7}"] \\
A[M] \sqsubseteq_{\mathcal{R}} A[N] \arrow[dr] & & \Gamma^{\times} \vDash M : \phi \arrow[dl] \\
& \Gamma^{\times} \vDash N : \phi \arrow[d, "\text{th. 5.6.4}"] & \\
& \Gamma^{\times} \vdash N : \phi \arrow[d, "\text{p. 5.2.5}"] & \\
& \Gamma \vdash N : \phi &
\end{tikzcd}$$

□

Le modèle $\mathcal{C}$ est complètement adéquat pour λ_r^c: on a

$$M \sqsubseteq_{\mathcal{C}} N \overset{\text{th.5.4.7}}{\Longleftrightarrow} M \sqsubseteq_{\mathcal{P}} N \overset{\text{th.5.6.6 + th. 5.6.5}}{\Longleftrightarrow} M \sqsubseteq_{rc} N$$

5.6.3 Adéquation et incomplétude du modèle $\mathcal{C}$ sur λ_r

L'adéquation du modèle $\mathcal{C}$ par rapport au calcul de ressources λ_r est conséquence directe du théorème 5.6.5, appliqué sur des termes de Λ_r :

$$\forall M, N \in Lr \ \ M \sqsubseteq_{\mathcal{C}} N \Leftrightarrow M \sqsubseteq_{rc} N \ \Rightarrow \ M \sqsubseteq_r N$$

Comme conséquence de $\neg(M \sqsubseteq_r N \ \Rightarrow \ M \sqsubseteq_{rc} N)$ (voir contre-exemple 3.6.13), le modèle $\mathcal{C}$ n'est pas complètement adéquat pour λ_r. Ce contre-exemple peut être formulé facilement en termes du système de types. Soit $P = \lambda x_1 \dots x_n.\Omega$ et $P' = (\lambda x_1 \dots x_n.\Omega \mid \lambda x_1 \dots x_m.\Omega)$. Puisque $xP \lesssim xP'$, on a $xP' \sqsubseteq_r xP$. Par contre, $\neg(xP' \sqsubseteq_{\mathcal{P}} xP)$: set $\phi \neq \omega$ et γ_i le type flèche d'arité i défini par

$$\underbrace{\omega \to (\omega \to \dots (\omega \to \omega)}_{i \text{ flèches}}$$

Il est clair que $\vdash \lambda x_1 \dots x_i.\Omega : \gamma_i$ est vérifié quelque soit i; alors $\vdash P' : \gamma_n \times \gamma_m$ implique $x : \gamma_n \times \gamma_m \to \phi \vdash xP' : \phi$. Mais $x : \gamma_n \times \gamma_m \to \phi \not\vdash xP : \phi$ car P n'a pas de types de la forme $\psi \sim \gamma_n \times \gamma_m$ (il suffit de remarquer que P est un terme, donc ne peut avoir des types produit.)

5.7 Le calcul avec ressources sur les λ-termes

Le pouvoir de séparation des contextes avec multiplicités sur les termes du lambda calcul pur a été étudié par Boudol et Laneve dans [18]. Le résultat fondamental est une caractérisation intensionnelle de la sémantique observationnelle du calcul avec multiplicités sur des λ-termes basée sur l'analogue des arbres de Böhm pour le calcul faible, appelés arbres de Lévy-Longo. L'extension de ce résultat au calcul avec ressources (avec test de convergence) s'ensuit. C'est ce que nous allons montrer ici. La situation est donc la suivante : pour tout $M, N \in \Lambda$,

$$M \sqsubseteq_{rc} N \ \Rightarrow \ M \sqsubseteq_m N \ \Leftrightarrow \ M \leq^{\eta}_{\mathcal{L}} N \ \overset{?}{\Rightarrow} \ M \sqsubseteq_{rc} N$$

La preuve de l'implication décorée avec le signe? utilise les lemmes auxiliaires suivants:

Lemme 5.7.1 *La β-conversion préserve les types : pour tout $M, N \in \Lambda$,*

$$M =_{\beta} N \ \Rightarrow \ M \simeq_{\mathcal{P}} N$$

Preuve. L'énoncé est une extension (triviale) du lemme A.1 [18], montré pour le système d'affectation de types associé à λ_r (donc sans test de convergence), sans relation de conséquence $\leq$. En fait, seulement l'adéquation du modèle $\mathcal{C}$ est nécessaire. □

Lemme 5.7.2 *(Proposition 5.4 [18])*

$$y \leq^{\eta}_{\mathcal{L}} Y \;\Rightarrow\; \begin{cases} Y =_{\beta} \lambda z_1 \ldots z_k . y Z_1^{\infty} \ldots Z_k^{\infty} & où\ \forall i\ y \neq z_i \leq^{\eta}_{\mathcal{L}} Z_i \\ Y \in PO_{\infty} \end{cases}$$

Lemme 5.7.3 *Si* $\Gamma \vdash M : \phi$ *alors* $\Gamma \vdash \lambda y . M y^{\infty} : \phi$ *pour tout* $y \notin var(M)$.

Preuve. Par récurrence sur le type ϕ. Si $\phi = \omega$, l'énoncé est immédiat par (L7). Si $\phi = \pi \to \sigma$ alors $y : \pi \vdash y : \pi$ et $\vdash y^{\infty} : \omega$ impliquent $y : \pi \vdash y^{\infty} : \pi$ par (L5) et (L9). Donc, par (L3) on a

$$y : \pi, \Gamma \vdash M y^{\infty} : \sigma$$

d'où $\Gamma \vdash \lambda y . M y^{\infty} : \phi$ par (L2). □

Lemme 5.7.4 *Pour tout* ϕ*, si* $M \in PO_{\infty}$ *alors* $\vdash M : \phi$.

Preuve. Si $\phi = \omega$, l'énoncé est vérifié par application de (L7). Soit $\phi = \pi_1 \to \ldots \pi_n \to \omega$ où $n \geq 1$. Par définition de PO$_{\infty}$, quelque soit m il existe M' t.q. $M =_{\beta} \lambda x_1 \ldots x_m . M'$, où $x_i \neq x_j$ pour $i \neq j$. Ceci est vrai en particulier pour $m = n$. Comme $x_n : \pi_n, \ldots, x_1 : \pi_1 \vdash M' : \omega$ vaut par la règle (L7), nous avons $\vdash \lambda x_1 \ldots x_n . M' : \phi$ en appliquant (L2) n fois. Il en découle que $\vdash M : \phi$ par le lemme 5.7.1. □

Lemme 5.7.5

$$y \leq^{\eta}_{\mathcal{L}} Y \;\Rightarrow\; \forall \Gamma \forall \tau \begin{cases} (1) & \Gamma \vdash y : \tau \;\Rightarrow\; \Gamma \vdash Y : \tau \\ (2) & \Gamma \vdash y^{\infty} : \tau \;\Rightarrow\; \Gamma \vdash Y^{\infty} : \tau \end{cases}$$

Preuve. Soit $y \leq^{\eta}_{\mathcal{L}} Y$. Nous montrons (1) et (2) simultanément par récurrence sur τ. Si $\tau = \omega$, l'énoncé est trivial; sinon, τ est, soit un type flèche, soit un type produit, et on a $Y =_{\beta} \lambda z_1 \ldots z_k . y Z_1^{\infty} \ldots Z_k^{\infty}$, où $\forall i\ z_i \leq^{\eta}_{\mathcal{L}} Z_i$, ou $Y \in$ PO$_{\infty}$ par le lemme 5.7.2. Si $Y \in$ PO$_{\infty}$, l'énoncé est vérifié par application du lemme 5.7.4. Sinon, puisque les types sont préservés par β-conversion, on se limite à montrer que les termes $\lambda z_1 \ldots z_k . y Z_1^{\infty} \ldots Z_k^{\infty}$ et $(\lambda z_1 \ldots z_k . y Z_1^{\infty} \ldots Z_k^{\infty})^{\infty}$ possèdent le type τ sous le contexte Γ.

Si $\tau = (\pi_1 \to (\ldots (\pi_h \to \omega) \ldots))$, pour montrer (1) on examine deux cas:

- $h < k$: Par application de (L7) on a

  $$\Gamma \vdash \lambda z_{h+1} \ldots z_k . y Z_1^{\infty} \ldots Z_k^{\infty} : \omega$$

  alors $\Gamma \vdash \lambda z_1 \ldots z_k . y Z_1^{\infty} \ldots Z_k^{\infty} : (\pi_1 \to (\ldots (\pi_h \to \omega) \ldots))$ est prouvable en utilisant h fois (L8) et h fois (L2).

- $h \geq k$: On montre facilement que $z_i : \pi_i \vdash z_i^\infty : \pi_i$ pour tout i; donc

  $$z_i : \pi_i \vdash Z_i^\infty : \pi_i$$

  par hypothèse de récurrence (2), car π_i est un sous-type propre de τ. À partir de $\Gamma \vdash y : \tau$, on déduit

  $$z_k : \pi_k, \ldots, z_1 : \pi_1, \Gamma \vdash y Z_1^\infty \ldots Z_k^\infty : (\pi_{k+1} \rightarrow (\ldots (\pi_h \rightarrow \omega) \ldots))$$

  en utilisant k fois la règle (L3). Pour obtenir $\Gamma \vdash \lambda z_1 \ldots z_k . y {Z^\infty}_1 \ldots {Z^\infty}_k : \tau$, on termine par k applications de (L2).

Quant à (2), $\Gamma \vdash y^\infty : \tau$ implique $\tilde{\Gamma} \vdash y^\infty : \tau$; donc

$$\exists y : \phi_1, \ldots y : \phi_n \in \Gamma \;\; \phi_1 \times \ldots \times \phi_n \leq \tau$$

Comme $\tau \in \mathrm{Ft}$ est différent de ω, la propriété du produit garantit l'existence d'un indice j tel que $\phi_j \leq \tau$. Alors, $\tilde{\Gamma} \vdash y : \phi_i$ implique $\tilde{\Gamma} \vdash y : \tau$ par application de la règle (L9). D'une part, l'hypothèse de récurrence (1) donne $\tilde{\Gamma} \vdash Y : \tau$. D'autre part, comme $\vdash Y^\infty : \omega$ par application de (L7), on dérive $\tilde{\Gamma} \vdash (Y \mid Y^\infty) : \tau \times \omega$ par (L5), puis $\Gamma \vdash Y^\infty : \tau$ par (L9) et (L8), car $\tau \times \omega \leq \tau$ et $\tilde{\Gamma} \gg \Gamma$.

Si $\tau = \tau_0 \times \tau_1$, la partie (1) est vérifiée par vacuité. Soit $\Gamma \vdash y^\infty : \tau$; on obtient facilement

$$\Gamma_0 \vdash y^\infty : \tau_0 \quad \text{et} \quad \Gamma_1 \vdash y^\infty : \tau_1 \quad \text{avec } \Gamma_0, \Gamma_1 \gg \Gamma$$

L'hypothèse de récurrence (2) donne $\Gamma_0 \vdash Y^\infty : \tau_0$ et $\Gamma_1 \vdash Y^\infty : \tau_1$, d'où l'on dérive $\Gamma \vdash (Y^\infty \mid Y^\infty) : \tau_0 \times \tau_1$ par (L5) et (L8). Or, $Y^\infty \equiv (Y^\infty \mid Y^\infty)$. La proposition 5.3.1 nous permet donc de conclure $\Gamma \vdash Y^\infty : \tau$. □

Lemme 5.7.6

$$A \leq_{\mathcal{L}}^{\eta} M \;\Rightarrow\; A \sqsubseteq_{rc} M$$

Preuve. Par le résultat d'adéquation qui lie la sémantique opérationnelle au système de types $\mathcal{P}$, il suffit de montrer l'énoncé suivant :

$$A \leq_{\mathcal{L}}^{\eta} M \;\&\; \Gamma \vdash A : \phi \Rightarrow \Gamma \vdash M : \phi$$

On procède par cas sur A.

- Si $A = \lambda x_1 \ldots x_m . \Omega$ alors $M {=_\beta} \lambda x_1 \ldots x_n . M'$ avec $n \geq m$. Par la proposition 5.2.6 et du fait que ω est le seul type de Ω, tout type de A est de la forme $\pi_1 \rightarrow \ldots \rightarrow \pi_k \rightarrow \omega$ avec $k \leq m$, en particulier ϕ. On déduit $x_m : \pi_m, \ldots, x_1 : \pi_1 \vdash \lambda x_{m+1} \ldots x_n . M' : \omega$ par (L7), d'où $\vdash \lambda x_1 \ldots x_n . M' : \phi$ en appliquant (L2) m fois. Comme conséquence du lemme 5.7.1, $\vdash M : \phi$ est vérifié. Donc $\Gamma \vdash M : \phi$ par (L8).

- Si $A = \lambda x_1 \ldots x_n.xA_1 \ldots A_m$, on a, soit $M \in \mathrm{PO}_\infty$, soit

  $$M {=_\beta} \lambda y_1 \ldots y_t \lambda x_1 \ldots x_n.xM_1 \ldots M_m Y_1^\infty \ldots Y_t^\infty$$

  avec $A_i {\leq^\eta_\mathcal{L}} M_i$ et $y_i {\leq^\eta_\mathcal{L}} Y_i$ pour tout i. Dans le premier cas, $\Gamma \vdash M : \tau$ pour tout τ, par le lemme 5.7.4. Dans le second cas, par le lemme 5.7.3 on a

  $$\Gamma \vdash \lambda y_1 \ldots y_y \lambda x_1 \ldots x_n.xA_1 \ldots A_m y_1^\infty \ldots y_t^\infty : \phi$$

  En appliquant m fois l'hypothèse de récurrence, on obtient

  $$\Gamma \vdash : \lambda y_1 \ldots y_y \lambda x_1 \ldots x_n.xM_1 \ldots M_m y_1^\infty \ldots y_t^\infty : \phi$$

  De plus, tout typage de y_i^∞ dans cette preuve peut-être remplacé par un typage de Y_i^∞ (cf. lemme 5.7.5(2)). Donc, $\Gamma \vdash M : \phi$ est conséquence du lemme de préservation du typage 5.7.1.

□

Nous avons maintenant les outils pour prouver

$$\forall M, N \;\; M {\leq^\eta_\mathcal{L}} N \;\Rightarrow\; M {\sqsubseteq_{rc}} N$$

Soit M et N deux termes du lambda calcul et C un contexte de λ_r^c t.q. $M {\leq^\eta_\mathcal{L}} N$ et $C[M]{\Downarrow_{rc}}$. Le lemme d'approximation, restreint à des lambda termes purs, dit qu'il existe un approximant A de M vérifiant $C[A]{\Downarrow_{rc}}$. Il est clair que $A {\leq^\eta_\mathcal{L}} N$, par conséquent $C[N]{\Downarrow_{rc}}$ par application du lemme 5.7.6.

Un corollaire immédiat de ce résultat est que les multiplicités séparent autant que les ressources quand il s'agit de lambda termes. Autrement dit:

$$\forall M, N \in \Lambda \;\; M \sqsubseteq_m N \;\Leftrightarrow\; M {\sqsubseteq_{rc}} N$$

Chapitre 6

Modèle de filtres

6.1 Adéquation du modèle $\mathcal{F}$ sur λ_r^c

L'adéquation du modèle de filtres $\mathcal{F}$ par rapport au calcul λ_r^c découle par un raisonnement similaire à celui utilisé dans la preuve d'adéquation du modèle $\mathcal{C}$. Certaines preuves du chapitre 5 étant adaptables au nouveau modèle, nous serons ici beaucoup plus brefs.

Remarquons que l'interprétation de (cP) dans $\mathcal{F}$ admet une écriture plus concise que dans le modèle $\mathcal{C}$, qui est :

$$\mathcal{V}[\![\mathrm{c}P]\!]_\rho = \begin{cases} \mathcal{V}[\![\mathbf{I}]\!]_\rho & \text{si } \mathcal{W}[\![P]\!]_\rho \neq \perp \\ \perp & \text{sinon} \end{cases}$$

Remarque 6.1.1 *La fonction d'interprétation* $\mathcal{V}[\![-]\!] : \Lambda_{rc} \times Env \to \mathcal{F}$ *est continue.*

Définition 6.1.2 ***(Sémantique engendrée par le modèle $\mathcal{F}$)***
Pour tout $M, N \in \Lambda_{rc}$,

$$M \sqsubseteq_{\mathcal{F}} N \overset{def}{\Leftrightarrow} \forall \rho \in Env\ \mathcal{V}[\![M]\!]_\rho \subseteq \mathcal{V}[\![N]\!]_\rho$$

Dans le reste de la section on introduit un système de types $\mathcal{P}_\wedge$ sur la théorie $(\textsc{Fb}\ , \leq)$[1] et le préordre $\sqsubseteq_{\mathcal{P}_\wedge}$ sur Λ_{rc}, engendré par l'interprétation des termes dans $\mathcal{P}_\wedge$. On montre ensuite que $\sqsubseteq_{\mathcal{F}}$ coïncide avec $\sqsubseteq_{\mathcal{P}_\wedge}$ et on termine par un schéma de la preuve d'adéquation $M \sqsubseteq_{\mathcal{P}_\wedge} N \Rightarrow M \sqsubseteq_{rc} N$.

Le système d'affectation de types $\mathcal{P}_\wedge$ est défini par les règles du système $\mathcal{P}$ (figure 5.1) plus les règles dans la figure 6.1, qui manipulent la conjonction.

Il faut souligner que l'introduction de la conjonction n'a pas modifié le traitement "multiplicatif" des hypothèses. En effet, ni dans $\mathcal{P}$ ni dans $\mathcal{P}_\wedge$ on ne peut

---

[1] Pour le modèle de filtres cette théorie inclut un opérateur de conjonction $\wedge$ en plus du produit $\times$.

$$\text{L11: } \frac{\Gamma \vdash T : \pi \quad \Gamma \vdash T : \psi}{\Gamma \vdash T : \pi \wedge \psi} \qquad \text{L12: } \frac{\Gamma \vdash T : \pi \wedge \psi}{\Gamma \vdash T : \pi\,(\psi)}$$

FIG. 6.1 - *Système de types* $\mathcal{P}_\wedge$

montrer, par exemple, que $(\lambda x.xx)\mathbf{I}^1$ a un type différent de ω. Pourtant, nous ne pouvons pas affirmer que $\mathcal{P}$ et $\mathcal{P}_\wedge$ ont le même pouvoir discriminant.

Proposition 6.1.3 *Si* $\tau \in \mathcal{W}\llbracket M^\infty \rrbracket_\rho$, *alors* $\exists m \in \ N \ \ \tau \in \mathcal{W}\llbracket M^m \rrbracket_\rho$.

Preuve. Soit $\tau \in \sqcup \mathcal{W}\llbracket M^i \rrbracket_\rho$. Par la proposition 4.3.15, il existe $\tau_1, \ldots, \tau_n$ t.q. $\forall i \ \exists m_i \ \ \tau_i \in \mathcal{W}\llbracket M^{m_i} \rrbracket_\rho$. Il est clair que $\tau_i \in \mathcal{W}\llbracket M^{m_j} \rrbracket_\rho$ pour tout j t.q. $m_i \leq m_j$. Soit $k = max\{m_j\}$. Donc $\forall i \ \tau_i \in \mathcal{W}\llbracket M^k \rrbracket_\rho$. Par la définition de filtre, $\tau \in \mathcal{W}\llbracket M^k \rrbracket_\rho$. $\square$

Lemme 6.1.4 *(Comparable au lemme 5.4.5, partie* $\Leftarrow$*)*

$$\begin{array}{lcl} \Gamma \vdash M : \tau & \Rightarrow & \tau \in \mathcal{V}\llbracket M \rrbracket_{\xi_\Gamma} \\ \Gamma \vdash P : \tau & \Rightarrow & \tau \in \mathcal{W}\llbracket P \rrbracket_{\xi_\Gamma} \end{array}$$

Preuve. Par récurrence sur la dérivation du séquent $\Gamma \vdash T : \tau$ (où T désigne soit un terme soit un paquet).
(L1) Ici, $\Gamma = x : \tau$ et $T = x$. Comme $\mathcal{V}\llbracket x \rrbracket_{\xi_\Gamma} = j(\xi_\Gamma(x)) = \uparrow\tau$, $\tau \in \mathcal{V}\llbracket x \rrbracket_{\xi_\Gamma}$ est vérifié.

(L2) Dans ce cas $T = \lambda x.M$ et $\tau = \pi \rightarrow \phi$. Soit $\Delta = x : \pi, \Gamma$; par récurrence on a $\phi \in \mathcal{V}\llbracket M \rrbracket_{\xi_\Delta}$. De plus $\xi_\Delta = \xi_\Gamma[x := \uparrow\pi]$; alors $\phi \in \mathcal{V}\llbracket T \rrbracket \cdot \uparrow\pi$. D'où $\pi \rightarrow \phi \in \mathcal{V}\llbracket T \rrbracket_{\xi_\Gamma}$.

(L3) Dans ce cas $\Gamma = \Sigma, \Delta$ et $T = (MP)$. Supposons $\pi \rightarrow \tau \in \mathcal{V}\llbracket M \rrbracket_{\xi_\Sigma}$ et $\pi \in \mathcal{W}\llbracket P \rrbracket_{\xi_\Delta}$. Alors $\xi_\Gamma = \xi_\Sigma \cdot \xi_\Delta$ et $\tau \in \mathcal{V}\llbracket MP \rrbracket_{\xi_\Gamma}$ par la définition d'application dans le domaine des filtres (cf. 4.3.18).

(L4) Ici $\Gamma = \Sigma, \Delta$ et $T = M < P/x >$. Supposons $x \notin \Delta$, $\Delta' = x : \pi, \Delta$ et aussi $\tau \in \mathcal{V}\llbracket M \rrbracket_{\xi_{\Delta'}}$ et $\pi \in \mathcal{W}\llbracket P \rrbracket_{\xi_\Sigma}$. Par monotonie de la fonction d'interprétation, $\tau \in \mathcal{V}\llbracket M \rrbracket_{\xi_{\Delta[x := \mathcal{W}\llbracket P \rrbracket_{\xi_\Sigma}]}}$. D'où $\tau \in \mathcal{V}\llbracket M < P/x > \rrbracket_{\xi_\Gamma}$, car $\xi_\Gamma = \xi_\Delta \cdot \xi_\Sigma$.

(L5) Ici $\Gamma = \Sigma, \Delta$, $\tau = \pi \times \psi$ and $T = (P|Q)$. L'hypothèse de récurrence est : $\pi \in \mathcal{W}\llbracket P \rrbracket_{\xi_\Sigma}$ et $\psi \in \mathcal{W}\llbracket Q \rrbracket_{\xi_\Delta}$. Alors $\xi_\Gamma = \xi_\Sigma \cdot \xi_\Delta$. Il est clair que le type $\pi \times \psi$ appartient au produit (dans le sens de l'opération dans le modèle) $\mathcal{W}\llbracket P \rrbracket_{\xi_\Sigma} \cdot \mathcal{W}\llbracket Q \rrbracket_{\xi_\Delta}$. Donc $\tau \in \mathcal{W}\llbracket P \mid Q \rrbracket_{\xi_\Gamma}$.

(L6) Ici $T = M^\infty$ et $\tau = \pi$. Soit $\pi \in \mathcal{W}[\![M|M^\infty]\!]_{\xi_\Gamma}$; par la proposition 4.3.15, $\wedge_I \psi_i \leq \pi$, pour des ψ_i dans $\mathcal{W}[\![M]\!]_{\xi_{i_0}} \cdot \mathcal{W}[\![M^\infty]\!]_{\xi_{i_1}}$ et $\xi_\Gamma \supseteq \xi_{i_0} \cdot \xi_{i_1}$.

Pour chaque $i \in I$ on prouve $\psi_i \in \mathcal{W}[\![M^\infty]\!]_{\xi_\Gamma}$. Il y a trois cas à considérer : si $\psi_i \in \mathcal{W}[\![M]\!]_{\xi_{i_0}}$ alors $\psi_i \in \mathcal{W}[\![M^\infty]\!]_{\xi_{i_0}}$ par définition. Donc $\psi_i \in \mathcal{W}[\![M^\infty]\!]_{\xi_\Gamma}$ par monotonie, car $\xi_{i_0} \subseteq \xi_\Gamma$. Si $\psi_i \in \mathcal{W}[\![M^\infty]\!]_{\xi_{i_1}}$ l'énoncé est vérifié directement par monotonie. Finalement, lorsque $\psi_i = \psi_{i_0} \times \psi_{i_1}$ avec $\psi_{i_0} \in \mathcal{W}[\![M]\!]_{\xi_{i_0}}$ et $\psi_{i_1} \in \mathcal{W}[\![M^\infty]\!]_{\xi_{i_1}}$, par la propriété 6.1.3, il existe $m \in \mathbb{N}$ t.q. $\psi_{i_1} \in \mathcal{W}[\![M^m]\!]_{\xi_{i_1}}$. Par conséquent, $\psi_{i_0} \times \psi_{i_1} \in \mathcal{W}[\![M^{m+1}]\!]_{\xi_\Gamma}$ est vérifié par monotonie, ce qui implique $\psi_i \in \mathcal{W}[\![M^\infty]\!]_{\xi_\Gamma}$. D'où $\wedge_I \psi_i \in \mathcal{W}[\![M^\infty]\!]_{\xi_\Gamma}$ et aussi $\pi \in \mathcal{W}[\![M^\infty]\!]_{\xi_\Gamma}$, par définition de filtre.

(L8) Par le lemme 5.4.4 et la monotonie de l'interprétation.

(L7), (L9), (L11) et (L12) sont des cas immédiats parce que les interprétations sont des filtres. La preuve pour (L10) est similaire à celle donnée pour le lemme 5.4.5. $\square$

Lemme 6.1.5 *(Comparable au lemme 5.4.5, partie ⇒)*
Soit $M, P \in \Delta^m$, *et* $\xi \in EnvC$.

$$\mathcal{V}[\![M]\!]_\xi \subseteq \{\phi \,/\, \Gamma_\xi \vdash M : \phi\}$$

$$\mathcal{W}[\![P]\!]_\xi \subseteq \{\pi \,/\, \Gamma_\xi \vdash P : \pi\}$$

Preuve. Par récurrence sur T, un terme ou un paquet. Pour ω l'énoncé est vérifié par (L7).

$T = x$: soit $\xi(x) = \uparrow\pi$; par la définition de j, $\pi \leq \phi$. Donc, $\Gamma_\xi \vdash x : \phi$ par l'axiome (L1) et la règle (L8), car $x : \phi \gg \Gamma_\xi$.

Dans tous les autres cas de termes, $\phi \in \mathcal{V}[\![T]\!]_\rho$ implique $\phi \sim \wedge\phi_i$ où $\phi_i \in \tilde{\mathrm{Ft}}$ par la proposition 4.3.9 et $\phi_i \in \mathcal{V}[\![T]\!]_\rho$. Il suffit donc de montrer l'énoncé pour ces ϕ_i premiers, i.e. montrer $\Gamma_\xi \vdash M : \phi_i$. On conclut $\Gamma_\xi \vdash M : \phi$ par application des règles (L11) et (L9).

$T = \lambda x.M$: soit $\phi = \pi \to \sigma \in \tilde{\mathrm{Ft}}$, alors $\sigma \in \mathcal{V}[\![M]\!]_{\xi[x:=\uparrow\pi]}$. Par h.r.

$$\Gamma_{\xi[x:=\uparrow\pi]} \vdash M : \sigma$$

C'est-à-dire, $\Gamma_\xi/x, x : \uparrow\pi \vdash M : \sigma$. Donc $\Gamma_\xi \vdash \lambda x.M : \pi \to \phi$ est vérifié par (L2) et (L8) puisque $\Gamma_\xi/x \gg \Gamma_\xi$.

$T = (MP)$: soit $\phi \in \tilde{\mathrm{Ft}}$; alors $\phi \in \mathcal{V}[\![M]\!]_{\xi_0} \cdot \mathcal{W}[\![P]\!]_{\xi_1}$ pour des ξ_0, ξ_1 t.q. $\xi \supseteq \xi_0 \cdot \xi_1$. C'est-à-dire, il existe $\pi \in \mathcal{W}[\![P]\!]_{\xi_1}$ qui vérifie $\pi \to \phi \in \mathcal{V}[\![M]\!]_{\xi_0}$. Sans perte de généralité, on peut supposer que ξ_0 et ξ_1 sont des environnements compacts par la continuité de l'interprétation[2]. Par hypothèse de récurrence,

$$\Gamma_{\xi_0} \vdash M : \pi \to \phi \qquad \text{et} \qquad \Gamma_{\xi_1} \vdash P : \pi$$

De plus, $\Gamma_{\xi_0}, \Gamma_{\xi_1} \gg \Gamma_\xi$; donc $\Gamma_\xi \vdash (MP) : \phi$ par (L8) et (L3).

$T = M\langle P/x\rangle$: considérons $\phi \in \widetilde{\mathrm{Ft}}$; alors $\phi \in \mathcal{V}[\![M]\!]_{\xi_0[x:=\mathcal{W}[\![P]\!]_{\xi_1}]}$ pour des environnements ξ_0, ξ_1 t.q. $\xi \supseteq \xi_0/x{\cdot}\xi_1$.

Par continuité, $\phi \in \bigsqcup\{\mathcal{V}[\![M]\!]_{\xi_0[x:=\uparrow\pi]} \,/\, \pi \in \mathcal{W}[\![P]\!]_{\xi_1}\}$, donc il existe $\pi \in \mathcal{W}[\![P]\!]_{\xi_1}$ t.q. $\phi \in \mathcal{V}[\![M]\!]_{\xi_o[x:=\uparrow\pi]}$ parce que l'ensemble est co-dirigé. De l'hypothèse de récurrence on obtient $\Gamma_{\xi_1} \vdash P : \pi$ et $\Gamma_{\xi_0[x:=\uparrow\pi]} \vdash M : \phi$. De plus, $\Gamma_{\xi_0[x:=\uparrow\pi]} = \Gamma_{\xi_0/x}, x : \pi$. D'où $\Gamma_{\xi_0/x}, \Gamma_{\xi_1} \vdash M\langle P/x\rangle : \phi$ par (L4). On conclut par (L8) puisque $\Gamma_{\xi_0/x}, \Gamma_{\xi_1} \gg \Gamma_\xi$.

$T = \mathrm{c}P$: puisque $\phi \nsim \omega$, on a $\phi = \pi \to \sigma \in \mathcal{V}[\![\mathbf{I}]\!]_\xi$ et $\gamma \in \mathcal{W}[\![P]\!]_\xi$. On montre facilement que $\pi \leq \sigma$. par h.r., $\Gamma_\xi \vdash P : \gamma$ donc $\Gamma_\xi \vdash \mathrm{c}P : \sigma \to \sigma$ par (L10) , d'où $\Gamma_\xi \vdash \mathrm{c}P : \pi \to \sigma$ par (L9).

Dans les cas des paquets on peut aussi se limiter à étudier le cas de $\pi \in \widetilde{\mathrm{Fb}}$.

$T = \mathbf{1}$: $\tau \in \mathcal{W}[\![T]\!]_\xi$ ssi $\tau \sim \omega$; l'énoncé est vérifié par (L7).

$T = (Q|P)$: soit $\pi \in \mathcal{W}[\![T]\!]_\xi$ t.q. $\pi \in \mathcal{W}[\![Q]\!]_{\xi_0}{\cdot}\mathcal{W}[\![P]\!]_{\xi_1}$ avec ξ_0, ξ_1 with $\xi \supseteq \xi_0{\cdot}\xi_1$. Deux situations sont à analyser : ou bien $\pi \in \mathcal{W}[\![Q]\!]_{\xi_0}$ (ou $\pi \in \mathcal{W}[\![P]\!]_{\xi_1}$) ou bien $\pi \in \{\pi_0 \times \pi_1 \,/\, \pi_0 \in \mathcal{W}[\![Q]\!]_{\xi_0} \,\&\, \pi_1 \in \mathcal{W}[\![P]\!]_{\xi_1}\}$. Dans le premier cas,

$$(\mathrm{L8})\ \dfrac{(\mathrm{L9})\ \dfrac{(\mathrm{L5})\ \dfrac{(\text{h.r.})\ \Gamma_{\xi_0} \vdash Q : \pi \quad \Gamma_{\xi_1} \vdash P : \omega\ (\mathrm{L7})}{\Gamma_{\xi_0}, \Gamma_{\xi_1} \vdash (Q|P) : \pi \times \omega}}{\Gamma_{\xi_0}, \Gamma_{\xi_1} \vdash (Q|P) : \pi}}{\Gamma_\xi \vdash (Q|P) : \pi}$$

Quant à la deuxième possibilité, $\pi = \pi_0 \times \pi_1$ implique $\Gamma_{\xi_0} \vdash Q : \pi_0$ et $\Gamma_{\xi_1} \vdash P : \pi_1$. On obtient le résultat par (L5) et (L8).

$T = M^\infty$: supposons $\pi \in \mathcal{W}[\![M^\infty]\!]_\xi$; alors il existe $m \in \mathbb{N}$ t.q. $\pi \in \mathcal{W}[\![M^m]\!]_\xi$ par la propriété 6.1.3. On prouve l'énoncé par récurrence sur m. Si $m = 0$, alors $\pi \sim \omega$ et on utilise (L9). Si $m > 0$, alors $\pi \in \bigsqcup\{\mathcal{W}[\![M]\!]_{\xi_0}{\cdot}\mathcal{W}[\![M^{m-1}]\!]_{\xi_1}$ avec $\xi \supseteq \xi_0{\cdot}\xi_1\}$.

---

[2] Par continuité, il existe ξ_0', ξ_1' compacts t.q. $\xi_i' \subseteq \xi_i$ et $\pi \in \mathcal{W}[\![P]\!]_{\xi_1'}$ tandis que $\pi \to \phi \in \mathcal{V}[\![M]\!]_{\xi_0'}$. On peut donc appliquer l'hypothèse de récurrence et conclure par monotonie.

On analyse trois cas : si $\pi \in \mathcal{W}[\![M]\!]_{\xi_0}$, on dérive

$$(\text{L8})\ \dfrac{(\text{L6})\ \dfrac{(\text{L9})\ \dfrac{(\text{L5})\ \dfrac{(\text{h.r.})\ \Gamma_{\xi_0} \vdash M : \pi \quad \vdash M^\infty : \omega\ (\text{L7})}{\Gamma_{\xi_0} \vdash (M|M^\infty) : \pi \times \omega}}{\Gamma_{\xi_0} \vdash (M|M^\infty) : \pi}}{\Gamma_{\xi_0} \vdash M^\infty : \pi}}{\Gamma_\xi \vdash M^\infty : \pi}$$

Dans le cas où $\pi \in \mathcal{W}[\![M^{m-1}]\!]_{\xi_1}$, par h.r. $\Gamma_{\xi_1} \vdash M^\infty : \pi$. Donc $\Gamma_\xi \vdash M^\infty : \pi$ est vérifié par (L8). Si $\pi = \pi_0 \times \pi_1$ avec $\pi_0 \in \mathcal{W}[\![M]\!]_{\xi_0}$ et $\pi_1 \in \mathcal{W}[\![M^\infty]\!]_{\xi_1}$, on dérive

$$(\text{L8})\ \dfrac{(\text{L6})\ \dfrac{(\text{L5})\ \dfrac{(\text{h.r.})\ \Gamma_{\xi_0} \vdash M : \pi_0 \quad \Gamma_{\xi_1} \vdash M^\infty : \pi_1\ (\text{h.r..})}{\Gamma_{\xi_0}, \Gamma_{\xi_1} \vdash (M|M^\infty) : \pi_0 \times \pi_1}}{\Gamma_{\xi_0}, \Gamma_{\xi_1} \vdash M^\infty : \pi_0 \times \pi_1}}{\Gamma_\xi \vdash M^\infty : \pi_0 \times \pi_1}$$

□

Théorème 6.1.6 *Pour tout* $M, N \in \Lambda_{rc}$, $M \sqsubseteq_{\mathcal{F}} N \Leftrightarrow M \sqsubseteq_{\mathcal{P}_\wedge} N$.

Preuve. L'énoncé est corollaire des lemmes 6.1.4 et 6.1.5 ; la preuve est similaire à celle du théorème 5.4.7. □

Tout comme $\mathcal{P}$, le nouveau système $\mathcal{P}_\wedge$ est stable par expansion, i.e.

$$M \overset{\star}{\to}_{rc} M' \ \& \ \Gamma \vdash M' : \phi \ \Rightarrow \ \Gamma \vdash M : \phi$$

La preuve de cette propriété ne diffère pas beaucoup de celle du théorème 5.3.3; les règles (L11) et (L12) ne posent pas de problème. Il suffit de constater que l'on peut reformuler les propriétés structurelles énoncées dans la proposition 5.2.6 : par exemple, pour l'application on a

$$\Gamma \vdash MP : \phi \Leftrightarrow \exists n > 0\ \forall i \in [1, n]\ \exists \Delta_i, \Sigma_i, \pi_i, \phi_i$$

$$\Delta_i \vdash M : \pi_i \to \phi_i \ \& \ \Sigma_i \vdash P : \pi_i \ \& \ \Delta_i, \Sigma_i \gg \Gamma \ \& \ \phi_1 \wedge \ldots \wedge \phi_n \leq \phi$$

En ce qui concerne le prédicat de réalisabilité $\Gamma \vDash T : \tau$ pour le système $\mathcal{P}_\wedge$, la définition est basée sur la description normalisée de τ :

Définition 6.1.7 *On définit deux prédicats de réalisabilité, l'un pour des types de termes, $\Gamma \vDash M : \phi$, l'autre pour des types de paquets, $\Gamma \Vdash P : \pi$. Pour des termes et des paquets clos de Λ_{rc}, et des formules premières, ils sont définis par :*

$$\vDash M : \omega \quad \overset{def}{\Leftrightarrow} \quad vrai$$

$$\vDash M : \pi \to \phi \quad \overset{def}{\Leftrightarrow} \quad (\pi \to \phi \in \widetilde{\mathrm{F}t} \;\Rightarrow\; M{\Downarrow_{rc}} \;\&\; (\forall P \Vdash P : \pi \;\Rightarrow\; \vDash MP : \phi))$$

$$\Vdash P : \omega \quad \overset{def}{\Leftrightarrow} \quad vrai$$

$$\Vdash P : \pi \quad \overset{def}{\Leftrightarrow} \quad \begin{array}{l}(\pi \in \widetilde{\mathrm{F}b} \;\&\; \pi \triangleright \phi_1, \ldots, \phi_n \;\Rightarrow \\ P \equiv (M_1 \mid \ldots \mid M_n \mid R) \;\&\; \forall i \vDash M_i \; \phi_i)\end{array}$$

L'extension aux formules non-premières est :

$$\vDash M : \phi \quad \overset{def}{\Leftrightarrow} \quad \phi \sim \wedge_I \sigma_i \;\&\; \sigma_i \in \widetilde{\mathrm{F}t} \;\&\; (\forall i \in I \vDash M : \sigma_i)$$

$$\Vdash P : \pi \quad \overset{def}{\Leftrightarrow} \quad \pi \sim \wedge_I \pi_i \;\&\; \pi_i \in \widetilde{\mathrm{F}b} \;\&\; (\forall i \in I \Vdash P : \pi_i)$$

L'extension aux termes et paquets ouverts de Λ_{rc} est :

$$\Gamma \vDash M : \phi \quad \overset{def}{\Leftrightarrow} \quad (\forall i\, \forall P_i \Vdash P_i : \pi_i \;\Rightarrow\; \vDash M\langle \tilde{P}/\tilde{x} \rangle : \phi)$$

$$\Gamma \Vdash P : \pi \quad \overset{def}{\Leftrightarrow} \quad (\forall i\, \forall P_i \Vdash P_i : \pi_i \;\Rightarrow\; \Vdash P\langle \tilde{P}/\tilde{x} \rangle : \pi)$$

où $\Gamma = x_1 : \pi_1, \ldots, x_n : \pi_n$ et $fv(P), fv(M) \subseteq \{x_1, \ldots, x_n\}$.

Le préordre sur Λ_{rc} engendré par l'interprétation est comme pour le modèle de cônes :

$$M \sqsubseteq_{\mathcal{R}} N \;\overset{def}{\Leftrightarrow}\; \forall \Gamma \;\phi\;\; \Gamma \vDash M : \phi \;\Rightarrow\; \Gamma \vDash N : \phi$$

On suppose que $P\langle \tilde{P}/\tilde{x} \rangle$ est équivalent à un ensemble de paquets, où les substitutions explicites ont été affectées aux termes composantes, comme on fait dans le calcul λ_d.

Proposition 6.1.8 *Pour tout $M, N \in \Lambda_{rc}$, si $M \sqsubseteq_{\mathcal{A}} N$ alors $M \sqsubseteq_{\mathcal{R}} N$.*

Preuve. Supposons $\vDash M : \phi$ et $\phi \sim \wedge_I \phi_i$ où $\phi_i \in \widetilde{\mathrm{Ft}}$. On montre l'énoncé par récurrence sur I; le cas de base ($\phi \in \widetilde{\mathrm{Ft}}$) est vérifié par une preuve complètement similaire à celle de la proposition 5.5.3. Dans le cas de récurrence, on a $\vDash M : \phi_i$ pour tout i. Donc $\vDash N : \phi_i$ par h.r. Donc $\vDash N : \phi$. $\Box$

Lemme 6.1.9

$$\Gamma \vDash M : \tau \ \ \& \ \ \tau \leq \tau' \ \Rightarrow \ \Gamma \vDash M : \tau'$$
$$\Gamma \Vdash P : \tau \ \ \& \ \ \tau \leq \tau' \ \Rightarrow \ \Gamma \Vdash P : \tau'$$

Preuve. Il suffit de montrer l'énoncé pour Γ vide. Soit $\tau \sim \wedge_I \tau_i$, et $\vDash T : \tau$. On procède par inspection de la définition de $\leq$.

(1) Ici, $\tau' = \omega$. Donc $\vDash M : \tau'$ et $\Vdash P : \tau'$ pour tout T par définition.

(2) Ici, $\tau = \pi \to \omega \ \leq \ \omega \to \omega = \tau'$. Par hypothèse $M\Downarrow_{rc}$. Il est clair que pour tout P, on a $\vDash MP : \omega$. D'autre part, $\Vdash P : \tau$ implique $P \equiv (N \mid Q)$ et $\vDash N : \tau$. Donc $\vDash N : \tau'$, ce qui implique $\Vdash P : \tau'$.

(3) Dans ce cas, $\tau = \pi_0 \to \phi_0 \ \leq \ \pi_1 \to \phi_1 = \tau'$, où $\pi_1 \ \leq \ \pi_0$ et $\phi_0 \ \leq \ \phi_1$. Soit $\phi_0 \sim \wedge_I \sigma_i$ et $\phi_1 \sim \wedge_J \varphi_j$.

- Supposons $\vDash M : \tau$. Alors $M\Downarrow_{rc}$. On doit montrer $\vDash M : \wedge_J \pi_1 \to \varphi_j$, i.e. $\vDash M : \pi_1 \to \varphi_j$. Soit P t.q. $\Vdash P : \pi_1$. Par hypothèse de récurrence $\Vdash P : \pi_0$; alors $\vDash MP : \sigma_i$ pour tout $i \in I$. Donc $\vDash MP : \phi_0$. Par hypothèse de récurrence $\vDash MP : \phi_1$; c'est-à-dire, $\vDash MP : \varphi_j$ pour tout $j \in J$. Par définition, $\vDash M : \pi_1 \to \varphi_j$, car $M\Downarrow_{rc}$.

- Supposons $\Vdash P : \tau$. Alors $\Vdash P : \pi_0 \to \sigma_i$ pour tout $i \in I$. I.e.

  $$\forall i \in I \ \exists N_i, Q_i \ \ P \equiv (N_i \mid Q_i) \ \& \ \vDash N_i : \pi_0 \to \sigma_i$$

  Remarquons que pour tout $j \in J$ il existe $i_j \in I$ t.q. $\pi_0 \to \sigma_{i_j} \ \leq \ \pi_1 \to \varphi_j$. Par hypothèse de récurrence, $\vDash N_{i_j} : \pi_1 \to \varphi_j$. Donc $\Vdash P : \wedge_J \pi_1 \to \varphi_j$ implique $\Vdash P : \tau'$.

(4)-(7) Dans ces cas, $\tau \sim \tau'$. L'énoncé est vérifié par définition.

(8) Ici, $\tau = \pi_0 \times \pi_1 \ \leq \ \pi_0' \times \pi_1' = \tau'$ où $\pi_i \ \leq \ \pi_i'$, $i = 0, 1$. Soit

$$\pi_0 \sim \wedge_I \psi_i \ \ \ \pi_1 \sim \wedge_J \theta_j \ \ \ \pi_0 \sim \wedge_K \psi_k' \ \ \ \pi_1 \sim \wedge_H \theta_h'$$

Par hypothèse on a $\Vdash P : \psi_i \times \theta_j$ pour tous $i \in I$ et $j \in J$. Supposons $\psi_i \triangleright a_1, \ldots, a_n$ et $\theta_j \triangleright a_{n+1}, \ldots, a_p$. Alors, $P \equiv (M_1 \mid \ldots \mid M_n \mid R)$ et $\vDash M_i : a_i$.

On doit montrer $\Vdash P : \wedge_{K \times H}(\psi_k' \times \theta_h')$; c'est-à-dire, $\Vdash P : \psi_k' \times \theta_h'$ pour tous k, h. Supposons $\psi_k' \triangleright b_1, \ldots, b_m$ et $\theta_h' \triangleright b_{m+1}, \ldots, b_q$. Par hypothèse, $\wedge_I \psi_i \ \leq \ \psi_k' \sim b_1 \times \ldots \times b_m$ et $\wedge_J \theta_j \ \leq \ \theta_h' \sim b_{m+1} \times \ldots \times b_q$. En appliquant la proposition 4.3.16, il existe i, j t.q. $\psi_i \ \leq \ b_1 \times \ldots \times b_m$ et $\theta_j \ \leq \ b_{m+1} \times \ldots \times b_q$. C'est-à-dire,

$$a_1 \times \ldots \times a_n \ \leq \ b_1 \times \ldots \times b_m \quad \text{et} \quad a_{n+1} \times \ldots \times a_p \ \leq \ b_{m+1} \times \ldots \times b_q$$

Rappelons que ces a_i, b_j sont des flèches premières; par la proposition 4.3.16 on peut construire deux injections, $g : [1,m] \rightarrow [1,n]$ et $g' : [m+1,q] \rightarrow [n+1,p]$, t.q.

$$(\forall i \in [1,m]\ \ a_{g(i)} \leq b_i) \ \ \& \ \ (\forall i \in [m+1,q]\ \ a_{g'(i)} \leq b_i)$$

Par hypothèse de récurrence, $\forall i \in [1,m] \ \vDash M_{g(i)} : b_i$ et $\forall i \in [m+1,q] \ \vDash M_{g'(i)} : b_i$. C'est-à-dire,

$$\Vdash (M_{g(1)} \mid \ldots \mid M_{g(m)} \mid M_{g'(m+1)} \mid \ldots \mid M_{g'(q)}) : \psi'_k \times \theta'_h$$

Donc $\Vdash P : \psi'_k \times \theta'_h$. Les indices k, h étant arbitraires, on conclut $\Vdash T : \wedge_{K \times H}(\psi'_k \times \theta'_h)$. Donc $\Vdash P : \tau'$ par définition.

(9)-(10) Ici $\tau = \pi \wedge \psi \leq \pi = \tau'$. Puisque la normalisation de τ est la conjonction des normalisations de π et de ψ, on a soit $\vDash M : \tau'$ soit $\Vdash P : \pi$ par hypothèse.

(11) -(13) Dans ces cas $\tau \sim \tau'$; l'énoncé est vérifié par définition.
□

On remarquera que le prédicat $\Vdash$, sur des termes et des types de Ft est équivalent au prédicat $\vDash$.

Théorème 6.1.10 ***(Correction de $\vDash$ pour $\mathcal{P}_\wedge$)***

$$\Gamma \vdash T : \tau \ \Rightarrow \ \Gamma^\times \Vdash P : \tau$$

Preuve. La preuve est par récurrence sur la dérivation de $\Gamma \vdash T : \tau$. On analyse la dernière règle de typage utilisée: pour (L1) et (L2) la preuve est similaire à celle du théorème 5.5.7. Pour (L3) on a $\Delta^\times \vDash M : \pi \rightarrow \tau$ et $\Sigma^\times \Vdash P : \pi$. Alors, il existe deux séquences ρ et ρ' de substitutions explicite t.q. $\vDash M\rho : \pi \rightarrow \tau$ et $\Vdash P\rho' : \pi$. Donc $\vDash (M\rho)(P\rho') : \tau$. Soit ξ la séquence de substituions qui résulte de la réunion de ρ et ρ' (ces séquences constituent un partage de ξ). Puisque $(M\rho)(P\rho') \sqsubseteq_{\mathcal{A}} (MP)\xi$, par la proposition 6.1.8 on a $\vDash (MP)\xi : \tau$. C'est-à-dire, $\Gamma \vDash MP : \tau$. Dans le cas de (L4) on utilise l'hypothèse de récurrence sur les deux branches de la dérivation. Dans le cas de (L5), on suppose $\Sigma^\times \vDash P : \pi$ et $\Delta^\times \vDash Q : \psi$. alors $(\Sigma, \Delta)^\times \vDash (P \mid Q) : \pi \times \psi$ par un raisonnement similaire à celui utilisé dans la preuve du lemme 6.1.9, cas (8). Si la dernière règle est (L6), l'hypothèse de récurrence donne $\Gamma^\times \vDash (M \mid M^\infty) : \tau$. Le paquet M^∞ a assez de termes pour permettre l'interprétation de τ, puisqu'il est inépuisable. Le cas de (L7) est évident et celui de (L8) est comme dans la preuve du théorème 5.5.7. Si la dernière règle utilisée est (L9), l'énoncé est vérifié par le lemme 6.1.9. Les cas de (L11) et (L12) découlent facilement de l'hypothèse de récurrence et de la définition de $\vDash$. □

La stabilité du typage par expansion et la correction de la nouvelle interprétation $\vDash$ dans le système $\mathcal{P}_\wedge$ impliquent un résultat d'adéquation de la convergence similaire à celui pour le système $\mathcal{P}$, énoncé dans le théorème 5.5.8. L'adéquation du préordre $\sqsubseteq_{\mathcal{P}_\wedge}$ par rapport à la sémantique opérationnelle en est un corollaire :

$$\forall M, N \in \Lambda_{rc} \ (M \sqsubseteq_{\mathcal{P}_\wedge} N \ \Rightarrow \ M \sqsubseteq_{rc} N)$$

Finalement, la correspondance entre l'interprétation des termes dans $\mathcal{P}_\wedge$ et dans le modèle de filtres $\mathcal{F}$, énoncée dans les lemmes 6.1.4 et 6.1.5, permet de conclure l'adéquation du modèle :

$$\forall M, N \in \Lambda_{rc} \ (M \sqsubseteq_{\mathcal{F}} N \Leftrightarrow M \sqsubseteq_{\mathcal{P}_\wedge} N \ \Rightarrow \ M \sqsubseteq_{rc} N)$$

6.2 À propos de la complétude du modèle

Le but de cette section est d'apporter quelques éléments de réponse au problème de complète adéquation du modèle $\mathcal{F}$. Le point de départ dans l'analyse du problème est l'impossibilité de prouver la complétude par la technique de définissabilité des points compacts du domaine, pourtant appropriée dans le cas du modèle $\mathcal{C}$. La condition d'application d'une telle approche est l'existence de termes caractéristiques pour chaque type. Mais le calcul de ressources ne fournit pas de moyens pour décrire la conjonction de la théorie $\mathcal{P}_\wedge$.

La conjonction des formules de Ft (types des termes du langage) ne pose pas de problèmes. Le terme caractéristique de $\phi \wedge \sigma$ est : $M_{\phi\wedge\sigma} = M_\phi \oplus M_\sigma$ où $\oplus$ est l'opérateur de choix non-déterministe définissable dans λ_r; les types de $M_{\phi\wedge\sigma}$ forment le filtre principal $\uparrow(\phi\wedge\sigma)$. Pourtant, on ne sait pas définir de test pour les formules avec conjonctions. En ce qui concerne la théorie de types sur Ft , la conjonction peut être éliminée : on peut restreindre le système de types à des formules premières. Dans ces formules, la conjonction apparaît toutefois à gauche de la flèche. Soit $\pi \to \sigma \in \tilde{\mathrm{Ft}}$; nous avons vu que le terme caractéristique de ce type est de la forme $M_{\pi\to\sigma} = \lambda x.(T_\pi x^\infty) M_\sigma$, ce qui suppose l'existence d'un test pour π. Sur la représentation normalisée de π, $\wedge_I \psi_i$, T_π doit être une abstraction qui contrôle que son argument a tous les types ψ_i. Ceci implique que l'argument doit être disponible, en son entier, autant de fois que des types ψ_i. Dans le calcul non-déterministe $\lambda_{||}$ [17] (avec sémantique observationnelle "may testing"), la définition du test consiste en une séquentialisation des contrôles :

$$T_{\phi\wedge\sigma} = \lambda x.(T_\phi x)(T_\sigma x)$$

ce qui ne pose aucun problème car les arguments appliqués au test sont inépuisables. Ce n'est pas le cas de λ_r. Si, par exemple, $\vdash M : \phi\wedge\sigma$, alors $T_{\phi\wedge\sigma} M$ se réduit vers le terme bloqué $T_\sigma x\langle \mathbf{1}/x \rangle$, puisque l'argument n'a que multiplicité 1. Cette difficulté semble insurmontable; le calcul λ_r n'a pas la capacité de "multiplier" les occurrences des ressources.

Pourtant, notre conjecture est que le modèle de filtres est complètement adéquat par rapport à λ_r^c. Autrement dit, que les types avec conjonction et produit discriminent autant que les types avec produit seulement. Notre argument est que la manipulation des hypothèses est toujours multiplicative. Les conjonctions qui apparaissent dans les hypothèses ont été introduites par le moyen de l'axiome; elles ne correspondent pas a un besoin de typer de façon différente deux occurrences de la même variable (comme c'est le cas dans les théories de types avec intersection usuelles). Comment le démontrer? Nous n'avons pas réussi à construire une preuve directe, sur les dérivations dans les systèmes de types $\mathcal{P}$ et $\mathcal{P}_\wedge$. Une approche différente consisterait à démontrer un lemme d'approximation pour le système de types avec conjonction, dans le style de [69], et de prouver ensuite que les contextes de λ_r^c permettent de séparer deux approximants qui ne sont pas compatibles. Notre conjecture à ce propos est que les outils développés dans le chapitre 3 pour λ_r devraient être adaptables à λ_r^c.

Nous ouvrons maintenant une parenthèse, pour discuter sommairement comment étendre le calcul λ_r^c, afin de lui donner les moyens de définir le test pour la conjonction. Nous introduisons un *produit synchrone* ou *opérateur de duplication* entre termes, $M\diamond N$, dont la propriété essentielle est de faire partager ses arguments à M et à N. C'est-à-dire, $(M\diamond N)P\rightarrow^{\diamond}_{rc}(MP\diamond NP)$ Le nouveau calcul est noté $\lambda^{\diamond}_{rc}$. Évidemment, il ne s'agit pas de construire les fonctions parallèles, que l'on décrit par exemple avec $\|$, puisqu'elles sont déjà dans λ_r (rappelons que $\oplus$ y est définissable). Bien au contraire, $\diamond$ serait le "meet" dans le domaine.

Un tel opérateur permet la définition du test pour la conjonction : il suffit de poser

$$M_{\pi\rightarrow\sigma} = \lambda x.((T_{\pi_1}\diamond\ldots\diamond T_{\pi_n})x^\infty)M_\sigma$$

si $\pi \simeq \pi_1\wedge\ldots\wedge\pi_n$[3].

Les termes de $\lambda^{\diamond}_{rc}$ sont :

$$\begin{array}{ll}(\Lambda^{\diamond}_{rc}) & M ::= x \mid \lambda x.M \mid (MP) \mid M\langle P/x\rangle \mid \mathrm{c}P \mid (M\diamond M)\\ & P ::= \mathbf{1} \mid M \mid (P \mid P) \mid M^\infty\end{array}$$

La relation d'évaluation $\rightarrow^{\diamond}_{rc}$ de ce calcul est un raffinement de $\rightarrow_{rc}$ (cf. figure 3.6), dont les nouvelles règles sont écrites dans la figure 6.2.

Remarquons que la règle de saisie n'a pas été modifiée. En effet, la distribution des arguments et des entrées de substitution à M et à N dans le produit synchrone $(M\diamond N)$ garantit la disponibilité des ressources sur les variables de tête de M et de N. Par exemple, si $P \equiv (L \mid P')$, alors

$$(x\diamond N)Q_1\ldots Q_k\langle P/x\rangle \overset{\star}{\rightarrow}{}^{\diamond}_{rc}(x\tilde{Q}\langle P/x\rangle\diamond N\tilde{Q}\langle P/x\rangle)\rightarrow^{\diamond}_{rc}(L\tilde{Q}\langle P'/x\rangle\diamond N\tilde{Q}\langle P/x\rangle)$$

---

[3] Le produit synchrone rappelle l'opérateur + de [29], équipé d'une sémantique "must testing", utilisé dans la définition du test associé aux types conjonction : dans ce papier, $T_{\phi\wedge\sigma} = \lambda x.(T_\phi + T_\sigma)$.

$$(M\diamond N)P\rightarrow^{\diamond}_{rc}(MP\diamond NP) \quad (M\diamond N)\langle P/x\rangle\rightarrow^{\diamond}_{rc}(M\langle P/x\rangle\diamond N\langle P/x\rangle)$$

$$\frac{M\rightarrow^{\diamond}_{rc}M'}{M\diamond N\rightarrow^{\diamond}_{rc}M'\diamond N} \qquad \frac{N\rightarrow^{\diamond}_{rc}N'}{M\diamond N\rightarrow^{\diamond}_{rc}M\diamond N'}$$

$$\mathrm{c}(M\diamond N)\rightarrow^{\diamond}_{rc}(\mathrm{c}M)\diamond(\mathrm{c}N)$$

FIG. 6.2 - *Évaluation dans* $\lambda^{\diamond}_{rc}$

Les termes irréductibles du calcul sont de trois types, à savoir : termes bloqués, abstractions $\lambda x.M$, et produits synchrones de termes irréductibles $(M_1\diamond\ldots\diamond M_n)$. Les valeurs du calcul sont un sous-ensemble des termes irréductibles. Dans l'approche standard, les abstractions sont des valeurs de $\lambda^{\diamond}_{rc}$, tandis que les termes bloqués sont identifiés aux termes divergents. En ce qui concerne les produits synchrones, seuls les produits d'abstractions sont observables. La syntaxe des valeurs est donc :

$$(\mathbb{V}^{\diamond}_{rc}) \quad V ::= \lambda x.M \,|\, (V\diamond V)$$

Ce calcul, comme les autres versions du calcul de ressources, vérifie un lemme des contextes, ce qui indique qu'il n'est pas complètement arbitraire. Rappelons que le préordre $\sqsubseteq^{\diamond}_{rc}$ identifie termes bloqués, divergence et produits synchrones dont les composantes ne sont pas toutes des abstractions.

Soit $\mathcal{A}$ l'ensemble des contextes applicatifs définis par

$$A ::= [] \mid AP \mid A\langle P/x\rangle \mid \mathrm{c}A$$

où P est un paquet de $\lambda^{\diamond}_{rc}$. Remarquer que l'opérateur de duplication n'apparaît pas en tête dans la définition de $\mathcal{A}$. Cependant, les paquets P peuvent avoir des produits synchrones comme sous-termes.

Lemme 6.2.1 ***(Lemme des contextes pour*** $\lambda^{\diamond}_{rc}$***)***

$$\forall M,N\in\Lambda^{\diamond}_{rc} \;\; M\sqsubseteq^{\diamond}_{rc}N \Leftrightarrow M\sqsubseteq_{\mathcal{A}}N$$

Preuve. La structure des contextes applicatifs pour $\lambda^{\diamond}_{rc}$ étant la même que ceux de λ^{c}_{r}, la démonstration de l'énoncé est une extension de la preuve du lemme 3.5.5. Nous indiquons les modifications à apporter. La partie $\Leftarrow$ est triviale. La partie $\Rightarrow$ est un cas particulier de la proposition suivante:

$$\forall M_1,\ldots,M_p\;\forall N_1,\ldots,N_p\;(\forall i\;M_i\sqsubseteq_{\mathcal{A}}N_i)\;\Rightarrow\;(\forall C\;C[\tilde{M}]\Downarrow^{\diamond}_{rc}\;\Rightarrow\;C[\tilde{N}]\Downarrow^{\diamond}_{rc})$$

où C est un contexte de $\lambda^{\diamond}_{rc}$ à plusieurs trous. Soit C t.q. $C[\tilde{M}]$ et $C[\tilde{N}]$ sont des termes clos et il existe une valeur V vérifiant $C[\tilde{M}]\overset{\star}{\rightarrow}_{rc}V$ en un nombre fini de pas l. Nous montrons $C[\tilde{N}]\Downarrow^{\diamond}_{rc}$ par récurrence sur (l, t, d) (selon l'ordre lexicographique), où t est le nombre de trous de C et d est le nombre de produits synchrones dans C. (Pour λ^c_r la récurrence sur (l, t) était suffisante.)

Le cas intéressant est $l > 0, t > 0$. Soit $C = \mathrm{c}^m W\tilde{B}^1 \ldots \tilde{B^m}$; nous établissons la proposition par cas sur W,

1. $W = C_0\tilde{B}^0$ où
   (a) $C_0 = x$ ou
   (b) $C_0 = \lambda x.D$ ou
   (c) $C_0 = []_i$ ou encore
   (d) $C_0 = D\diamond D'$
2. $W = (W_0 \mid W_1)$
3. $W = D^\infty$.

Pas de modifications à faire pour (1a)-(1c), (2) et (3). Dans ces cas la preuve se fait essentiellement par récurrence sur l (la première étape d'évaluation peut être reproduite à partir de $C[\tilde{N}]$), et en plus, dans le cas (1c) sur t.

Il reste à examiner l'évaluation de $C[\tilde{M}]$ pour $C_0 = D\diamond D'$ (cas (1d)) : les premières étapes sont de distribution des arguments, des substitutions et des tests de convergence

$$C[\tilde{M}]\overset{\star}{\rightarrow}{}^{\diamond}_{rc}(\mathrm{c}^m D\tilde{B}^1 \ldots \tilde{B^m})[\tilde{M}]\diamond(\mathrm{c}^m D'\tilde{B}^1 \ldots \tilde{B^m})[\tilde{M}]$$

Alors $C[\tilde{M}]\Downarrow^{\diamond}_{rc}$ implique $(\mathrm{c}^m D\tilde{B}^1 \ldots \tilde{B^m})[\tilde{M}]\Downarrow^{\diamond}_{rc}$ et aussi $(\mathrm{c}^m D'\tilde{B}^1 \ldots \tilde{B^m}[\tilde{M}]\Downarrow^{\diamond}_{rc}$ en au plus l étapes d'évaluation. Comme chacun de ces deux derniers termes a un produit synchrone de moins que $C[\tilde{M}]$, l'hypothèse de récurrence donne

$$(\mathrm{c}^m D\tilde{B}^1 \ldots \tilde{B^m})[\tilde{N}]\Downarrow^{\diamond}_{rc} \quad \text{et} \quad (\mathrm{c}^m D'\tilde{B}^1 \ldots \tilde{B^m}[\tilde{N}]\Downarrow^{\diamond}_{rc}$$

Alors $C[\tilde{N}]\Downarrow^{\diamond}_{rc}$. □

Pour ce qui est du modèle, l'équation sémantique associée à $\diamond$ est donnée par le "meet" dans le domaine des filtres, i.e. l'intersection. C'est-à-dire,

$$\mathcal{V}[\![M\diamond N]\!]_\rho = \mathcal{V}[\![M]\!]_\rho \sqcap \mathcal{V}[\![N]\!]_\rho$$

où l'environnement est dupliqué. La description dans le système de types utilise un type union $\vee$ [7] : le type de $M\diamond N$ est $\sigma \vee \phi$, lorsque σ est le type de M et ϕ est celui de N. La propriété essentielle de la théorie de types avec union est que

$$(\phi_1 \rightarrow \sigma) \vee (\phi_2 \rightarrow \sigma) \sim (\phi_1 \wedge \phi_2) \rightarrow \sigma$$

permettant ainsi d'éliminer la conjonction à gauche de la flèche (notre souci majeur pour la définition des termes caractéristiques).

Chapitre 7

Conclusion

Pour conclure, nous présentons une synthèse des problèmes ouverts en indiquant parfois des pistes qu'il conviendrait, à notre avis, explorer, ainsi que quelques lignes de recherche que nous jugeons intéressantes.

Plusieurs problèmes restent ouverts en ce qui concerne l'étude du pouvoir de discrimination de λ_r et aussi de λ_r^c, que nous avons entamé. Pour les termes du lambda calcul pur, il a été montré que le calcul avec ressources (non-déterministe) a le même pouvoir de séparation que λ_m (déterministe). Reste à savoir si λ_r (ou λ_r^c) peut discriminer plus de λ_m-termes que λ_m (ou λ_m plus test de convergence) lui-même. Autrement dit, si les termes avec multiplicités sont sensibles ou pas au non-déterminisme. Pour montrer que λ_r et λ_m diffèrent sur Λ_m il suffirait de construire un exemple, ce qui n'est pas, a priori, une tâche simple. De l'autre coté, puisque l'on dispose du lemme d'approximation, on pourrait imaginer une preuve classique de $\forall M, N \in \Lambda_m \; (M \sqsubseteq_m N \;\Rightarrow\; M \sqsubseteq_r N)$, c'est-à-dire, par un résultat de séparation du genre $A \not\sqsubseteq_r N \Rightarrow A \not\sqsubseteq_m N$, où A est un approximant qui provient d'un terme dans Λ_m, et $N \in \Lambda_m$. Or, les approximants des λ_m-termes ne sont pas nécessairement des termes de Λ_m, ce qui empêche de parler de $A \not\sqsubseteq_m N$. Une approche qui semble plus adaptée pour résoudre cette question consisterait à construire un préordre algébrique $\lll$ sur Λ_r et montrer qu'il définit une sémantique adéquate pour $\sqsubseteq_r$. S'il s'avère que $\lll$ est plus fin que $\sqsubseteq_m$ sur Λ_m, on aura prouvé que λ_r et λ_m ne diffèrent pas sur Λ_m, c'est-à-dire,

$$\forall M, N \in \Lambda_m \;\; M \lll N \;\;\Rightarrow\;\; M \sqsubseteq_r N \;\;\Rightarrow\;\; M \sqsubseteq_m N \;\;\Rightarrow\;\; M \lll N$$

Si, au contraire, $M \sqsubseteq_m N \;\not\Rightarrow\; M \lll N$, on devrait pouvoir se servir de l'information recueillie pendant la tentative de preuve pour construire un exemple.

Le préordre algébrique dont on parle doit être plus fin que $\leq_{\mathcal{L}_r}$. En particulier, il devrait rendre compte de l'η-conversion. Tel préordre $\lll$ jouerait le rôle de $\leq_{\mathcal{L}}^{\eta}$ pour le λ-calcul, et on pourrait se demander s'il ne constitue pas une sémantique algébrique complètement adéquate pour λ_r. En étendant convenablement la notion d'approximant à λ_r^c, ce qui implique que $\leq$ doit être sensible au nombre de ressources des paquets, un résultat de complète adéquation pour $\sqsubseteq_{rc}$ impliquerait une deuxième caractérisation de la structure locale (dans la terminologie de Barendregt [8]) du modèle $\mathcal{C}$.

La question se pose aussi sur la caractérisation de la sémantique observationnelle engendrée par le codage du calcul avec ressources dans le π-calcul. En effet, λ_r a été défini par Boudol à partir d'une analyse fine du codage du λ-calcul faible dans π. Boudol et Laneve [20] montrent que λ_m peut être codé de façon adéquate dans π. Le fait que λ_m (et λ_r) manque du test de convergence met en évidence qu'il n'est pas aussi discriminant que π. On s'attend à pouvoir étendre le codage et le résultat d'adéquation à λ_r et λ_r^c. Qu'en est-t-il de la complétude pour λ_r^c ?

Dans la section 2.2.3, nous avons posé le problème de l'adéquation complète des traductions $(-)_v^c : \lambda_v \to \lambda_c$ et $(-)_c^v : \lambda_c \to \lambda_v$, c'est-à-dire, de la vérification de l'équation (2) de Mitchell (cf. section 2.2). Les traductions considérées permettent la simulation de la convergence du calcul source dans le calcul cible. Notre conjecture est que ces propriétés passent aux contextes. Pour le montrer il faudrait prouver que le terme obtenu par l'application successive des deux traductions (dans le bon ordre) est équivalent au terme d'origine : $((M)_v^c)_c^v \simeq_c M$ et $((M)_c^v)_v^c \simeq_v M$.

Comme nous l'avons dit dans le chapitre 2, aucune indication n'est donnée dans [56] sur la façon de prouver l'adéquation complète d'une traduction compositionnelle. Il serait intéressant de chercher des conditions nécessaires et suffisantes générales. Autre question de sur ce sujet concerne la modularité des extensions du λ-calcul. Nous avons vu que le choix non-déterministe ne peut pas être ajouté au lambda calcul faible avec appel par valeur λ_v, mais si à λ_c, malgré l'inter-simulation de λ_v et λ_c. La question qui se pose naturellement est : sous quelles conditions l'ajout d'opérateurs préserve une traduction (complètement) adéquate ?

L'étude des modèles du calcul avec ressources n'est certainement pas clos. En particulier, la question se pose à propos du genre de modèle qu'on peut associer au sous-calcul déterministe de λ_r, le calcul avec multiplicités λ_m. Du point de vue des systèmes de types, dans l'état actuel des travaux, il ne semble pas que l'on puisse restreindre $\mathcal{P}$ pour rendre compte de l'homogéneité des ressources dans λ_m. Remarquons qu'une condition nécessaire pour que le système de types associé à λ_m soit correct est que sa restriction aux λ-termes soit équivalente au système de types avec intersection standard. C'est-à-dire, les types π dans $\pi \to \sigma$, ne peuvent être, a priori, que des produits quelconques de types (associés tous à la même ressource).

La comparaison entre le pouvoir de discrimination du système de types $\mathcal{P}$ et du système $\mathcal{P}_\wedge$, évoqué dans le chapitre 6, reste un problème ouvert.

On devrait aussi étudier la possibilité de modifier le système de types actuel pour construire une sémantique complètement adéquate de λ_r (le résultat de complète adéquation de la thèse est sur λ_r^c). Il s'agit de reproduire l'effet du test de convergence dans la théorie de types, c'est-à-dire, de reproduire la capacité de compter le nombre de ressources d'un argument. Avec cette modification, le résultat de continuité syntaxique pour λ_r devrait pouvoir aider à montrer la conjecture sur l'adéquation complète du modèle de filtres (associé au système de types $\mathcal{P}_\wedge$ avec à la fois le produit et l'intersection).

Les résultats syntaxiques de Boudol et Laneve sur des sémantiques observationnelles non-standards du calcul avec ressources (où blocage et divergence sont différents), énoncés dans la section 3.8.1, constituent une forte motivation pour l'étude de la contrepartie algébrique et dénotationnelle de ces scénarios observationnels. Motivés par le résultat concernant le λ-calcul et λ_m, on peut se demander si, par exemple, l'interprétation algébrique de λ_r n'engendre pas une sémantique complètement adéquate de λ_r par rapport au préordre plat ${\sqsubseteq^\flat}_r$.

Nous avons montré comment donner une sémantique dénotationnelle à un langage qui ne satisfait pas les postulats classiques des modèles à environnements pour le lambda calcul, à savoir, la distribution de l'environnements aux sous-termes. Dans cette thèse, l'étude des modèles porte exclusivement sur la sémantique standard du calcul de ressources. Peut-on développer des modèles pour les sémantiques non-standards, verticale et plate, suivant cette approche? Est-il possible de caractériser le blocage, comme on caractérise la convergence, dans un système raffiné de types avec intersection?

Il serait intéressant de chercher des liens entre le calcul avec ressources et le fragment multiplicatif de la logique linéaire. En effet, une relation entre ces calculs n'est pas à exclure; rappelons que le système de types $\mathcal{P}$ distingue les occurrences des variables et manipule les hypothèses de façon multiplicative. Toujours en ce qui concerne $\mathcal{P}$, il faut souligner que il est essentiellement le même système de types utilisé par Gardner dans [34], où elle aborde des questions d'optimalité de la réduction de λ-termes.

Pour conclure, le domaine d'application du calcul avec ressources se trouve entre la programmation fonctionnelle, avec des langages à la ML, et la programmation distribué, avec des langages comme FACILE [35, 36] (basé sur SML, incorporant des primitives de communication) et PICT [64, 63] (typé, basé sur un fragment du π-calcul). Il est assez raisonnable de penser, par exemple, que les propriétés du système de types associé au calcul avec ressources peuvent être utiles en compilation.

Bibliographie

[1] M. Abadi, L. Cardelli, P.-L. Curien, J.-J. Lévy. *Explicit Substitutions.* Journal of Functional Programming, 1. 1991.

[2] S. Abramsky . *The Lazy Lambda Calculus.* Dans D. Turner, editeur, Research Topics in Functional Programming, Addison Wesley. 1989.

[3] S. Abramsky, L. Ong. *Full Abstraction in the lazy lambda calculus.* Information and Computation, 105(2). 1993.

[4] F. Alessi, M. Dezani-Ciancaglini, U. de Liguoro. *May and Must Convergency in Concurrent λ-calculus.* LNCS 841. 1994.

[5] A. Asperti, G. Longo. *Categories, Types and Structures.* MIT Press. 1991.

[6] E. Astesiano, G. Costa. *Non-Determinism and Fully Abstract Models.* R.A.I.R.O. Theor. Inf. 14. 1980.

[7] F. Barbanera, M. Dezani-Ciancaglini. *Intersection and Union types.* Information and Computation 119(2). 1995.

[8] H.P. Barendregt. *The Lambda Calculus.* North-Holland. 1985.

[9] H.P. Barendregt, M. Coppo, M. Dezani-Ciancaglini. *A Filter Lambda Model and the Completeness of Type Assignment.* Journal of Symbolic Logic 48. 1983.

[10] G. Berry. *Some Syntactic and Categorical Constructions of Lambda Calculus Models.* Rapport de Recherche 80, INRIA Sophia-Antipolis. 1981.

[11] G. Berry, G. Boudol. *The Chemical Abstract Machine.* Theoretical Computer Science 96(1). 1992.

[12] G. Berry, P.-L. Curien, J.-J. Lévy. *Full Abstraction for Sequential Languages : the State of the Art.* M. Nivat et J. Reynolds, editeurs, Algebraic Methods in Semantics. Cambridge University Press. 1985.

[13] B. Bloom, J. Riecke. *LCF should be lifted.* Proc. Conf. Algebraic Methodology and Software Technology. Departement of Computer Science, University of Iowa. 1989.

[14] G. Boudol. *A Lambda-Calculus for Parallel Functions.* Rapport de Recherche 1231, INRIA Sophia-Antipolis. 1990.

[15] G. Boudol. *Lambda-Calculi for (strict) Parallel Functions.* Information and Computation, 108(1). 1994.

[16] G. Boudol. *Asynchrony and the π-calculus.* Rapport de Recherche 1702, INRIA Sophia-Antipolis. 1992.

[17] G. Boudol. *The lambda calculus with multiplicities.* Rapport de Recherche 2025, INRIA Sophia-Antipolis. 1993.

[18] G. Boudol, C. Laneve. *The discriminating power of multiplicities in the λ-calculus.* Information and Computation 126 (1). 1996.

[19] G. Boudol, C. Laneve. *Termination, deadlock and divergence in the λ-calculus with multiplicities.* 11th Mathematical Foundations of Programming Semantics Conference. Electronic Notes in Computer Science. 1995.

[20] G. Boudol, C. Laneve. *λ-Calculus, Multiplicities and the π-Calculus.* Rapport de Recherche 2581, INRIA Sophia-Antipolis. 1995.

[21] M. Coppo, M. Dezani-Ciancaglini. *An extension of the basic functionality theory for the λ-calculus.* Notre Dame Journal of Formal Logic 21. 1980.

[22] M. Coppo, M. Dezani-Ciancaglini, F. Honsell, G. Longo. *Extended Type Structures and Filter Lambda Models.* Dans G. Lolli, G. Longo, and A. Marcja, editeurs, Logic Colloquium 82. Elsevier Science Publishers B.V. (North-Holland). 1984.

[23] M. Coppo, M. Dezani-Ciancaglini, G. Longo. *Applicative Information Systems.* LNCS 159. Springer-Verlag. 1983.

[24] M. Coppo, M. Dezani-Ciancaglini, B. Venneri. *Principal types schemes and lambda calculus semantics.* Dans J.R. Hindley and J.P. Seldin, editeurs, To H.B. Curry : Essays on Combinatory Logic, Lambda Calculus and Formalism. Academic Press. 1980.

[25] M. Coppo, M. Dezani-Ciancaglini, B. Venneri. *Functional characters of solvable terms.* Zeit. Math. Logik Grund, 27. 1981.

[26] P.-L. Curien, T.Hardin, J.-J. Lévy. *Confluence Properties of Weak and Strong Calculi of Explicit Substitutions.* Journal of ACM 43(2). 1996.

[27] B.A. Davey, H.A. Priestley. *Introduction to Lattices and Order.* Cambridge University Press. 1990.

[28] M. Dezani-Ciancaglini, U. de Liguoro, A. Piperno. *Filter Models for a Parallel and Non-deterministic λ-calculus.* LNCS 711. 1993.

[29] M. Dezani-Ciancaglini, U. de Liguoro, A. Piperno. *Fully Abstract Semantics for Concurrent λ-calculus.* LNCS 789. 1994.

[30] M. Dezani-Ciancaglini, U. de Liguoro, A. Piperno. *A Filter Model for Concurrent λ-calculus.* À paraître dans SIAM Journal.

[31] M. Dezani-Ciancaglini, U. de Liguoro, A. Piperno. *Filter Models for Conjunctive-Disjunctive λ-calculi.* À paraître dans TCS.

[32] M. Dezani-Ciancaglini, I. Margaria. *A Characterization of F-complete type assignments.* Theoretical Computer Science 45. 1986.

[33] M. Felleisen. *On the Expressive Power of Programming Languages.* LNCS 432. 1990.

[34] P. Gardner. *Discovering Needed Reductions Using Type Theory.* LNCS 789. 1994.

[35] A. Giacalone, P. Mishra, S. Prasad. *Facile : A Symmetric Integration of Concurrent and Functional Programming.* Journal of Parallel Programming 18(2). 1989.

[36] A. Giacalone, P. Mishra, S. Prasad. *Operational and Algebraic Semantics for FACILE : A Symmetric Integration of Concurrent and Functional Programming.* LNCS 443. 1990.

[37] C.A. Gunter. *Semantics of Programming Languages : Structures and Techniques.* MIT Press. 1190.

[38] C.A. Gunter, D.S. Scott. *Semantic Domains.* Dans J. van Leeuwen, editeur, Handbook of Theoretical Computer Science, volume B. Elseiver. 1990.

[39] M. Hennessy. *The semantics of call-by-value and call-by-name in a nondeterministic environment.* SIAM J. Computing 9(1). 1980.

[40] M. Hennessy, G. Plotkin. *Full Abstraction for a Simple Parallel Programming Language.* LNCS 74. Springer-Verlag. 1979.

[41] J.R. Hindley, G. Longo. *Lambda-calculus Models and Extensionality.* Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 26. 1980.

[42] J.R. Hindley. *The Simple Semantics for Coppo-Dezani-Sallé Types.* International Symposium on Programming. LNCS 137. 1982.

[43] J.R. Hindley. *The Completeness Theorem for Typing λ-terms.* Theoretical Computer Science 22. 1983.

[44] J.R. Hindley. *Types with Intersection : An Introduction.* Formal Aspects of Computing 4. 1992.

[45] J.M.E. Hyland. *A Syntactic Characterization of the Equality in Some Models for the Lambda Calculus.* Journal of the London Mathematical Society 12. 1976.

[46] J.-L. Krivine. *Lambda-calcul, types et modèles.* Masson. 1991.

[47] J.-J.Lévy. *An algebraic interpretation of the $\lambda\beta K$-calculus; and an application of a labelled λ-calculus.* Theoretical Computer Science, 2(1). 1976.

[48] J.-J. Lévy. *Réductions Correctes et Optimales dans le Lambda Calcul.* Thèse de doctorat. Université de Paris VII. 1978.

[49] U. de Liguoro, A. Piperno. *Non-Deterministic Extensions of Untyped λ-calculus.* Information and Computation 122(2). 1995.

[50] G. Longo. *Set Theoretical Models of Lambda Calculus : Theories, expansions and isomorphisms.* Annals of Pure and Applied Logic, 24. 1983.

[51] R. Milner. *Fully Abstract Models of Typed λ-calculi.* Theoretical Computer Science, 4. 1977.

[52] R. Milner. *Communication and Concurrency.* Prentice Hall. 1989.

[53] R. Milner, J. Parrow, D. Walker. *A Calculus of Mobile Processes, Parts I and II.* Information and Computation, 100(1). 1992.

[54] R. Milner. *Functions as Processes.* Mathematical Structures in Computer Science, 2. 1992.

[55] R. Milner. *The Polyadic π-Calculus: A tutorial.* Dans F. Bauer, W. Brauer et H. Schwichtenberg, editeurs, Logic and Algebra of Specification. Springer-Verlag. 1993.

[56] J.C. Mitchell. *On Abstraction and the Expressive Power of Programming Languages.* LNCS 526. 1991.

[57] J.H. Morris. *Lambda Calculus Models of Programming Languages.* PhD M.I.T. 1968.

[58] M. Nielsen, G. Plotkin, G. Winskel. *Petri Nets, Event Structures and Domains.* Theoretical Computer Science 13. 1981.

[59] C.-H. Luke Ong. *Fully Abstract Models of the Lazy Lambda Calculus.* In Proceedings of the 29th Conference on Foundations of Computer Science. The Computer Science Press. 1988.

[60] C.-H. L. Ong. *The Lazy Lambda Calculus/ An Investigation into the Foundations of Functional Programming.* PhD Thesis, Imperial College. 1988.

[61] C.-H. L. Ong. *Concurrent Lambda Calculus and a General Precongruence Theorem for Applicative Bisimulations.* Non publié, Cambridge University. 1992.

[62] C.-H. L. Ong. *Non-determinism in a functional setting.* LICS'93, IEEE Comp. Soc. Press, Los alamitos. 1993.

[63] B. Pierce. *Programming in the Pi-Calculus.* Tutorial Notes. Computer Laboratory, University of Cambridge. 1995.

[64] B. Pierce, D. Turner. *Concurret Objects in a Process Calculus.* LNCS 907. 1995.

[65] G. Plotkin. *Call-by-name, call-by-value and the λ-calculus.* Theoretical Computer Science, 1. 1975.

[66] G. Plotkin. *LCF Considered as a Programming Language.* Theoretical Computer Science, 5. 1975.

[67] G. Plotkin. *Post-Graduate Lecture Notes on Advanced Domain Theory.* Deptartment of Computer Science, University of Edinburgh. 1981.

[68] J. Riecke. *Fully Abstract Translations between Functional Languages.* Mathematical structures in Compuetr Science 11. 1992.

[69] S. Ronchi della Rocca. *Proof Theory and Foundations of Programming.* International Summer School in Logic for Computer Science. Université de Savoie. 1993.

[70] P. Sallé. *Une extension de la théorie des types en λ-calcul.* LNCS 62. (1978)

[71] D. Sangiorgi. *Expressing Mobility in Process Algebras: First Order and Higher Order Paradigms.* PhD. Thesis, Department of Computer Science, Edinburgh University. 1993.

[72] D. Sangiorgi. *The Lazy Lambda Calculus in a Concurrency Scenario.* Information and Computation, 120(1). 1994.

[73] D. Sangiorgi. *Lazy Functions and Mobile Processes.* Rapport de Recherche INRIA 2515. 1995.

[74] D. Scott. *Continuous Lattices.* Lecture Notes in Mathematics 274. Springer. 1972.

[75] D. Scott. *Domains for Denotational Semantics.* LNCS 140. Springer-Verlag. 1982.

[76] K. Sieber. *Relating Full Abstraction Results for Different Programming Languages.* LNCS 472. 1990.

[77] K. Sieber. *Call-by-Value and Non-Determinism.* LNCS 664. 1993.

[78] A. Stoughton. *Fully Abstract Models of Programming Languages.* Research Notes in Theoretical Computer Science. PITMAN. 1988.

[79] B. Thomsen. *A Calculus of Higher-order Communicating Systems.* Actes du 16th Annual Symposium on Principles of Programming Languages. 1989.

[80] B. Thomsen. *Calculi of Higher-order Communicating Systems.* PhD Thesis, Department of Computing, Imperial College, London. 1990.

[81] C. Wadsworth. *The relation between computational and denotational properties for Scott D_∞-model of the lambda-calculus.* SIAM Journal on Computing, 5. 1976.

[82] C. Wadsworth. *Approximate Reduction and Lambda-Calculus Models.* SIAM Journal on Computing, 7. 1978.

Glossaire

Dans le cadre d'un lambda calcul hypothétique λ_h, les conventions de notation sont les suivantes :

symbole	**signification**
λ_h	extension du λ calcul faible
Λ_h	ensemble des termes
Λ_h^o	ensemble des termes clos
$\mathbb{V}_h$	ensemble des valeurs
$\rightarrow_h$	évaluation
$\not\rightarrow_h$	impossibilité d'évaluer
$\Downarrow_h$	prédicat de convergence
$\Uparrow_h$	prédicat de divergence
$\sqsubseteq_h$	préordre observationnel
$\simeq_h$	équivalence observationnelle

c	test de convergence unaire, 100
c	combinateur de test de convergence, 50
p	combinateur de test de convergence parallèle, 50
λ^s	constructeur des abstractions strictes, 54
λ^v	constructeur des abstractions par valeur, 51
$\|\|$	opérateur de composition parallèle, 48
$\oplus$	opérateur de choix non-déterministe, 49
$\uplus$	opérateur unaire non-déterministe dans le lambda calcul, 81
	union multi-ensembliste de solutions, 43
$\diamond$	produit synchrone, 224
$\perp$	élément minimum d'un domaine, 152
$\top$	élément maximum d'un domaine, 152
$\bigsqcup$	borne supérieure, 152