THÈSE

pour obtenir le titre de

DOCTEUR DE L'ÉCOLE POLYTECHNIQUE
spécialité Informatique

Propriétés de sécurité dans le λ-calcul

soutenue par

Tomasz Blanc

présentée le 7 novembre 2006, à l'École Polytechnique, devant le jury composé de :

Messieurs	Pierre-Louis Curien	*Président*
	Jean-Jacques Lévy	*Directeur de thèse*
	Jan Willem Klop	*Rapporteur*
	Cosimo Laneve	*Rapporteur*
	Cédric Fournet	*Examinateur*
	Didier Rémy	*Examinateur*

Remerciements

Au moment d'écrire ces lignes qui seront, je n'en doute pas, les plus lues de cette dissertation, je me rends compte du chemin parcouru au cours de ces trois dernières années. Tout ce travail n'aurait bien sûr pas pu s'accomplir sans l'aide et le soutien de mon entourage que je tiens à remercier ici.

Une thèse peut être vue comme un chemin de l'arbre de syntaxe d'un terme du λ-calcul. Rencontrer un noeud d'application requiert de faire un choix. Tomber sur une variable liée signifie un retour en arrière, vers l'abstraction qui la lie. L'expérience de Jean-Jacques Lévy dans ce domaine m'a été d'un grand secours pour trouver le bon chemin. Je tiens ici à le remercier de m'avoir fait l'honneur de diriger ma thèse. Sa bonne humeur et son indéfectible optimisme ont été de précieux alliés dans les quelques moments de doute ou de découragement qui étiquettent parfois le chemin d'une thèse.

Une thèse peut aussi être vue comme un périple. En acceptant la lourde tâche de rapporteur de ma thèse, Jan Willem Klop et Cosimo Laneve m'ont aidé à mettre en perspective mon travail et à gommer certaines imprécisions. Je les remercie chaleureusement pour l'intérêt qu'ils ont porté à mes travaux. Plus fondamentalement, ces discussions m'ont aussi fourni l'occasion de visiter les villes d'Amsterdam et de Bologne. Avant de commencer cette thèse, j'avais aussi pu découvrir Cambridge au cours de mon stage de DEA, au sein du laboratoire de Microsoft Research. Je suis reconnaissant envers Cédric Fournet de m'avoir encadré au cours de cette première expérience de recherche, particulièrement formatrice, qui a jeté les premières bases de mon travail de thèse. Je remercie Pierre-Louis Curien pour avoir accepté de présider le jury de cette thèse et Didier Rémy pour son implication dans ce jury.

Le cadre verdoyant et bucolique de l'INRIA en fait un lieu particulièrement propice à la méditation. Dans ce cadre studieux, l'ambiance chaleureuse qui règne entre les murs du bâtiment 8 autorise quelques moments de détente entre deux sous-sous-cas d'une preuve. Je souhaite remercier les membres des projets Moscova, Cristal, Contrainte, Atoll et Arles pour leur simplicité, leur humour et leur disponibilité.

Enfin, je voudrais remercier ceux qui, dans l'ombre, ont contribué de façon indirecte mais fondamentale à mener à bien ce travail. Je remercie bien sûr toute ma famille pour son soutien au quotidien. Et je remercie Flora qui a été ma première Phan (fan ?) et ses parents qui m'ont fait découvrir la cuisine vietnamienne et en particulier le thịt quây et le bò bún chã giò qui ont été mes fournisseurs officiels d'énergie au cours de cette thèse.

Table des matières

Introduction **1**

1 Syntaxe et propriétés du λ-calcul et du λ-calcul étiqueté **5**
1.1 Le λ-calcul 5
1.2 Le λ-calcul étiqueté 13
1.3 Irréversibilité des contextes 19

2 λ-calcul par valeur et étiquettes **25**
2.1 Le λ-calcul par valeur 26
2.1.1 Confluence 27
2.1.2 Développements finis 30
2.1.3 Standardisation 31
2.1.4 Stabilité 39
2.2 Le λ-calcul par valeur étiqueté 44
2.2.1 Confluence 48
2.2.2 Développements finis 53
2.2.3 Standardisation 62
2.2.4 Stabilité 64

3 λ-calcul faible et étiquettes **69**
3.1 Le λ-calcul faible 71
3.2 Le λ-calcul faible étiqueté 77
3.3 Partage de sous-termes 84

4 Inspection de pile **89**
4.1 Le λ_{sec}-calcul : l'inspection de pile formalisée 90
4.2 Le λ_t-calcul : un λ-calcul avec un contexte 92
4.3 Du λ_t-calcul à l'inspection de pile 96

5 Non-interférence **107**
5.1 Non-interférence dans le λ-calcul et le λ-calcul par valeur 109
5.2 Le λ_m-calcul 113
5.3 Le λ_m-calcul étiqueté 119
5.3.1 Propriétés de la réduction étiquetée 126
5.3.2 Réduction respectant une utilisation-mémoire 129
5.3.3 Non-interférence 132

6 Indépendance **143**
6.1 Le λ_n-calcul 145
6.2 Le λ_n-calcul étiqueté 148

Conclusion **163**

Liste des figures **168**

Bibliographie **171**

Introduction

Avec le développement d'internet, les codes mobiles, c'est-à-dire pouvant se déplacer sur le réseau, sont devenus de plus en plus populaires. Le téléchargement de programmes depuis le réseau est maintenant une opération banale. Ainsi, des applications Java ou *C#* sont couramment exécutées dans les navigateurs internet, le plus souvent de façon transparente. La programmation et l'exécution d'applications mobiles posent des questions de sécurité. Comment se protéger des applications malveillantes qui pourraient profiter de leur mobilité pour s'emparer de données personnelles ou bien perturber les systèmes hôtes ? Restreindre les droits des programmes mobiles ne suffit pas : comme l'a montré Hardy [21], un programme mobile pourrait tirer profit des privilèges d'autres applications présentes sur le système hôte.

La question de la sécurité en présence de programmes mobiles a été examinée sous des angles variés. Les réponses apportées ont été multiples. Dans la pratique, des langages tels que Java [17, 31] ou *C#* [9] intègrent un mécanisme de sécurité, l'inspection de pile, qui attribue des droits à un programme en fonction (1) de son origine et (2) de la chaîne d'appels conduisant à son exécution. Avant une opération sensible, telle que l'effacement d'un fichier, on teste si le programme possède bien le droit d'effectuer cette opération. Cependant, les travaux de Fournet et Gordon [15] ont montré les limites de ce système purement empirique : aucune réelle garantie de sécurité n'est assurée par l'inspection de pile.

Les analyses de flot d'information constituent une approche dont les fondations théoriques sont plus solides. Ces analyses, initiées par Denning et Denning [14, 13] puis développées par Volpano et Smith [41], Heintze et Riecke [22], Abadi et Banerjee [1], consistent à classer les données dans des catégories en fonction de leur niveau de sécurité. De façon générale, on distingue les données secrètes des données publiques. L'analyse en elle-même suit la propagation des données secrètes au cours de l'exécution du programme et permet de garantir que les sorties publiques du programme ne donnent aucune information sur les entrées secrètes. De cette façon, on peut s'assurer que les programmes mobiles ne s'emparent pas de données que l'on souhaite garder secrètes. Du point de vue formel, il s'avère que cette garantie de sécurité correspond à une propriété de non-interférence. Cette notion avait été introduite indépendamment par Goguen et Meseguer [16]. Simonet et Pottier d'une part et Myers, Nystrom, Zdancewic et Zheng d'autre part, ont poursuivi cet effort et sont parvenus à intégrer ces analyses statiques de flot d'information dans des langages complets : Flow Caml [39] et Jif [32] sont respectivement issus d'Objective Caml [28] et de Java [31]. Cependant, la propriété de non-interférence peut parfois être trop restrictive : la réponse négative à un test de mot de passe donne ainsi une information (certes minime) sur le mot de passe. De ce fait, dans le cadre d'une analyse de flot, cette réponse est considérée comme aussi secrète que le mot de passe. Certains travaux comme ceux de Chong et Myers [12] ou d'Almeida Matos et Boudol [4] proposent des propriétés de non-interférence plus souples.

Une approche plus récente consiste à envisager un contrôle dynamique des programmes afin d'obtenir plus de finesse et de flexibilité dans les propriétés de sécurité obtenues. Chong et Myers ont étudié dans [12] un protocole de sécurité inspiré d'une vente par enchères secrètes. Au cours

de cette vente, chaque enchérisseur soumet son offre (secrète) dans une enveloppe scellée. Après scellement de son enveloppe, il ne peut plus modifier son offre. Les enveloppes ne sont descellées qu'au moment où *toutes* les enveloppes ont été scellées. Le contenu des enveloppes est alors rendu public afin de pouvoir désigner le vainqueur des enchères. Le but de ce protocole de sécurité est d'assurer l'indépendance entre les différentes enchères. En considérant ce protocole sous l'angle des analyses de flot d'information, on remarque que les offres sont initialement secrètes, puis deviennent publiques. Ce changement du niveau de sécurité, appelé déclassification, n'est pas correct du point de vue d'une analyse de flot d'information. Chong et Myers ont abordé ce problème en assouplissant leur définition de la non-interférence. La politique de sécurité de la Muraille de Chine constitue un autre exemple de contrôle dynamique de l'exécution d'un programme. Cette politique s'inspire de la gestion des conflits d'intérêts économiques et a été introduite par Brewer et Nash dans [10]. Dans ce système, deux individus, Alice et Bob, sont intuitivement des acteurs économiques en concurrence. Charlie est un partenaire économique potentiel pour Alice et Bob : il peut s'agir d'un consultant qui pourrait aider à fixer les prix des produits vendus par Alice ou Bob. Initialement, Charlie peut choisir d'interagir avec Alice ou Bob. Mais la politique de la Muraille de Chine impose que, une fois que l'interaction avec Alice (respectivement Bob) a eu lieu, Charlie n'a plus le droit d'interagir avec Bob (resp. Alice). En effet, Charlie pourrait faire profiter à Bob (resp. Alice) des informations secrètes dont il aurait eu connaissance au moment de l'interaction avec Alice (resp. Bob). L'objectif visé est l'indépendance des actions d'Alice et Bob. De même que dans l'exemple des Enchères scellées, on observe que certains événements entraînent un changement dans la politique de sécurité. Les enchères deviennent publiques *après* que le scellement de toutes les enveloppes. Dans la Muraille de Chine, Charlie n'a plus le droit d'interagir avec Alice *après* avoir interagi avec Bob. Ces politiques de sécurité dynamiques dépendent crucialement de la notion d'histoire du calcul.

Cette dissertation a pour objet d'étudier ces trois approches dans un cadre formel minimal et commun. Nous commençons d'abord par observer que l'inspection de pile, les analyses de flot d'information et la Muraille de Chine utilisent de façon fondamentale la notion d'histoire du calcul. Plus précisément, l'inspection de pile utilise l'origine des fonctions et la succession des appels qui aboutit à l'appel d'une fonction. D'un point de vue synthétique, une analyse de flot d'information est une analyse de dépendance des termes présents vis-à-vis des termes originaux. Enfin, la politique de Muraille de Chine définit les actions autorisées en fonction des actions passées. Dans cette dissertation, nous prenons le parti d'étudier ces différentes approches dans le λ-calcul étiqueté [29]. En effet, les étiquettes du λ-calcul expriment la dépendance des termes présents vis-à-vis des réductions passées. Ces étiquettes donnent donc un accès à l'histoire du calcul qui permet d'exprimer et d'étudier dans le cadre du λ-calcul les trois approches mentionnées précédemment.

En exprimant dans le λ-calcul étiqueté l'inspection de pile, la non-interférence et la Muraille de Chine, nous avons obtenu au passage certains développements significatifs qui portent, notamment, sur des variantes du λ-calcul étiqueté. Ces résultats, qui sont indépendants des approches de sécurité étudiées ultérieurement, sont exposés dans les premiers chapitres de la dissertation. Les trois derniers chapitres concernent les trois approches de sécurité mentionnées précédemment.

Dans le premier chapitre, nous rappelons la syntaxe du λ-calcul et λ-calcul étiqueté et nous mentionnons les propriétés fondamentales de ces langages : ils sont confluents et ils vérifient les théorèmes des développements finis et de standardisation. De plus, les étiquettes du λ-calcul permettent d'exprimer simplement la propriété de stabilité. Nous ajoutons un nouveau résultat, inspiré du constat suivant. Un avantage appréciable du λ-calcul étiqueté est qu'il permet d'éviter les coïncidences syntaxiques qui se produisent, par exemple, dans le terme $M = I(Ix)$. Ce terme contient deux radicaux, qui, une fois contractés, aboutissent au même terme Ix. Cependant cette confluence est accidentelle. La meilleure preuve est que la notion de radical-résidu n'est pas la

même dans les deux cas. Dans le λ-calcul étiqueté, la contraction des deux radicaux ne donne pas le même résultat. Nous prouvons dans la section 1.3 un résultat voisin. Nous montrons qu'on a la réduction $C[M] \twoheadrightarrow_e C[M']$ si et seulement si on a $M \twoheadrightarrow_e M'$. Ce théorème d'irréversibilité des contextes (p. 20) est faux dans le λ-calcul sans étiquette. Ce résultat signifie intuitivement qu'un contexte ne peut pas être présent puis disparaître et enfin réapparaître au cours d'une réduction : la disparition d'un contexte est *irréversible*.

Dans le chapitre 2, nous introduisons formellement une variante du λ-calcul : le λ-calcul par valeur. Les radicaux de ce calcul sont les radicaux $(\lambda x.M)V$ du λ-calcul dont l'argument V est une valeur. Ce calcul s'inspire de la stratégie d'évaluation en appel par valeur des langages tels que Objective Caml [28] ou SML/NJ [40]. Cependant, par contraste avec ces langages, l'ordre d'évaluation des radicaux n'est pas imposé. Nous montrons que le λ-calcul par valeur est confluent. Nous prouvons aussi qu'il existe une réduction standard propre au λ-calcul par valeur et qui ne coïncide pas avec la réduction standard du λ-calcul. Ce raisonnement aboutit au théorème de standardisation (p. 34). Par ailleurs, les étiquettes du λ-calcul n'expriment pas la notion de stabilité dans le λ-calcul par valeur. Nous adaptons les étiquettes et la réduction étiquetée au λ-calcul par valeur. Nous prouvons que le calcul obtenu est confluent, qu'il conserve le théorème de standardisation et que ses étiquettes permettent, comme dans le cadre du λ-calcul étiqueté, d'exprimer simplement la stabilité. On montre également le théorème des développements finis (p. 60) à l'aide d'une preuve intuitive fondée sur une notion étendue d'imbrication des radicaux. Cette preuve peut s'adapter de façon élémentaire au λ-calcul.

Le chapitre 3 est consacré au λ-calcul faible. Dans un tel calcul, les sous-termes des abstractions ne sont pas réductibles. Un des avantages d'un calcul faible est qu'il permet de simplifier la notion de partage. Comme l'a montré Lamping dans [25], pour exprimer le partage dans le λ-calcul, on représente les termes par des graphes dont les sous-contextes peuvent être partagés. Par exemple, dans le terme $(\lambda x.xa(xb))(\lambda y.Iy)$ où $I = \lambda x.x$, il est nécessaire de partager le radical Iy indépendamment de la valeur de y. Ainsi, on doit partager le sous-contexte $I[\ ]$. Après la réduction des radicaux externes, on obtient $(\lambda y.Iy)a((\lambda y.Iy)b)$ puis $Ia((\lambda y.Iy)b)$ où le sous-contexte (partagé) $I[\ ]$ est instancié avec deux sous-termes différents : a et y. Dans le calcul faible, les contractions des radicaux ne peuvent se produire sous une abstraction. Ainsi, dans le précédent exemple, le sous-terme Iy de $(\lambda x.xa(xb))(\lambda y.Iy)$ ne peut être réduit puisqu'il est contenu par l'abstraction $\lambda y.Iy$. Plus généralement, dans le λ-calcul faible, on peut représenter les termes avec partage simplement avec des graphes acycliques orientés. Dans [43], Wadsworth décrit deux algorithmes pour la réduction de termes dans le λ-calcul faible avec partage. Nous étudions une variante confluente du λ-calcul faible, introduite par Maranget et Lévy dans [30]. Comme le λ-calcul, cette variante vérifie les théorèmes des développements finis et de standardisation. Nous introduisons ensuite le λ-calcul faible étiqueté qui conserve les propriétés fondamentales du λ-calcul faible. La réduction étiquetée du λ-calcul faible est inspirée du deuxième algorithme de Wadsworth. Le théorème de partage (p. 87) montre que les étiquettes expriment le partage : les sous-termes d'un terme qui ont la même étiquette sont égaux. Ils peuvent donc être partagés.

Dans le chapitre 4, nous rappelons la syntaxe du λ_{sec}-calcul, le langage introduit par Fournet et Gordon dans [15] pour décrire formellement le mécanisme d'inspection de pile. Nous nous inspirons de ce mécanisme pour introduire le λ_t-calcul. L'inspection de pile permet de contrôler l'exécution d'une fonction en exploitant une information locale (l'origine de la fonction) et une information de contexte (l'enchaînement d'appel ayant conduit à son exécution). De la même manière, le λ_t-calcul conditionne la contraction des radicaux en fonction d'une information locale (l'étiquette du radical) et d'une information de contexte (le chemin menant de la racine au radical). En adaptant la stratégie d'évaluation et les conditions de contraction des radicaux, on obtient un langage, le λ_{ts}-calcul, qui permet de faire le lien avec une variante du λ_{sec}-calcul nommée λ_{secW}-calcul. Plus

précisément, le théorème de correction (p. 102) montre qu'il existe une traduction du λ_{secW}-calcul dans le λ_{ts}-calcul pour laquelle une réduction dans le λ_{secW}-calcul correspond à une réduction dans le λ_{ts}-calcul.

Dans le chapitre 5, nous nous penchons sur la notion de non-interférence qui a été introduite par Goguen et Meseguer dans [16]. Cette propriété se définit intuitivement de la façon suivante : un principal, qui dispose d'un certain ensemble de commandes, n'interfère pas avec un autre principal si les actions du premier n'ont aucun effet visible pour le second. Si on reprend le point du vue des analyses de flot d'information [14, 13, 41, 22, 1, 39, 32], il s'agit de déterminer si les sorties publiques d'un programme permettent d'obtenir des informations sur les entrées secrètes de ce programme. Par contraste avec ces analyses de flot statiques, on adopte une approche plus dynamique, inspirée par la technique employée par Abadi, Lampson et Lévy dans [3]. Ces derniers proposent une analyse de dépendances dans le contexte du λ-calcul. A l'aide du λ-calcul étiqueté, ils déterminent quels sous-termes du terme initial contribuent au résultat de la réduction. Dans un premier temps, on adapte cette approche à la question de la non-interférence dans le λ-calcul et le λ-calcul par valeur. On établit en particulier la relation entre non-interférence et stabilité dans ces deux langages. Dans un second temps, on examine les difficultés engendrées par la présence d'effets de bord. Pour cela, on introduit le λ_m-calcul qui étend le λ-calcul par valeur avec les entiers, un branchement et des références. Comme le souligne Simonet [39], l'exemple $\texttt{ifz}\, x\, \texttt{then}\, ()\, \texttt{else}\, y{:=}0$ montre que si, après la réduction de ce terme, la valeur associée à y est non nulle, alors on peut en déduire $x = 0$. En d'autres termes, la non-réduction du sous-terme $y{:=}0$ donne une information sur x. De ce fait, l'expression d'une propriété d'interférence est plus délicate dans ce contexte. Pour résoudre cette difficulté, on exploite le *théorème d'irréversibilité des chemins* (p. 24). Cette propriété permet de nommer les adresses mémoire de façon structurelle et de dater les réductions d'un terme. La datation de la réduction d'un terme permet ensuite d'identifier les adresses mémoire qui interfèrent et les intervalles de temps pendant lesquels elles interfèrent. Ces informations permettent de définir formellement la propriété de non-interférence (p. 133) puis de montrer que les étiquettes utilisées pour le λ_m-calcul expriment bien cette propriété (p. 140).

Dans le chapitre 6, nous examinons en détail une politique de sécurité dynamique : la Muraille de Chine. Cette politique a pour but de contrôler les interactions entre plusieurs individus. Pour exprimer ces interactions au sein du λ-calcul, on ajoute à ce langage la notion de *principal* : un principal peut représenter un individu, une organisation ou tout autre entité dont l'identité importe. En annotant les termes du λ-calcul par des principaux, nous obtenons le λ_n-calcul. Ces annotations sont par nature bien différentes des étiquettes du λ-calcul. Alors que les principaux sont des annotations statiques, les étiquettes du λ-calcul sont dynamiques dans la mesure où elles se composent au cours du calcul. L'introduction du concept de principal au cœur du λ-calcul permet de définir la notion de réduction indépendante de l'interaction entre deux principaux A et B : il s'agit d'une réduction où toutes les contractions impliquant A peuvent être faites, indifféremment, avant ou après celles qui impliquent B. Comme nous l'avons vu précédemment, la politique de sécurité de la Muraille de Chine s'appuie sur l'histoire des événements passés. Pour exprimer cette politique dans le langage, cette histoire doit être accessible. Pour cela, nous ajoutons au λ_n-calcul les étiquettes du λ-calcul. Ces dernières nous permettent d'exprimer formellement la Muraille de Chine dans le λ_n-calcul étiqueté. Dans ce langage, on prouve que les étiquettes expriment simplement la propriété d'indépendance (p. 154 et 158). On montre également qu'une réduction qui respecte la politique de sécurité de la Muraille de Chine pour les principaux Alice et Bob est indépendante de l'interaction entre Alice et Bob (p. 160).

Chapitre 1

Syntaxe et propriétés du λ-calcul et du λ-calcul étiqueté

1.1 Le λ-calcul

Le λ-calcul est un langage de programmation introduit par Church dans les années 30. Ce langage minimaliste est d'un grand intérêt théorique puisque Church et Turing ont montré que la notion de calculabilité dans ce langage est la même que celle de la machine de Turing. Mais ce langage est également d'un grand intérêt pratique car il est à l'origine de la famille des langages de programmation fonctionnels. Certains de ces langages, comme Lisp, Objective Caml, SML/NJ ou Haskell [20, 28, 40, 34] sont largement utilisés en pratique. Ces attraits théoriques et pratiques expliquent que le λ-calcul a été très largement étudié [7]. Dans cette section, nous rappelons succinctement la syntaxe et les propriétés du λ-calcul qui seront exploitées dans cette dissertation.

Soit $\mathbf{X} = \{x_1, x_2, x_3, \ldots\}$ un ensemble infini dénombrable de variables de programme, aussi notées x,y,z. La syntaxe des termes et des contextes du λ-calcul est définie de la façon suivante.

Termes	$M,N \in \mathbf{\Lambda}$	$::=$	x	Variable
		$\mid$	$\lambda x.M$	Abstraction
		$\mid$	MN	Application
Contextes	$C \in \mathbf{\Lambda}[\,]$	$::=$	$[\,]$	Vide
		$\mid$	$\lambda x.C$	
		$\mid$	CN	
		$\mid$	MC	

Un terme (aussi appelé λ-expression) peut être une variable, une abstraction ou une application. L'abstraction $\lambda x.M$, dont le sous-terme M est appelé *corps*, peut être intuitivement comprise comme le terme M vu comme fonction de x. Cette abstraction sera parfois notée $\lambda\_.M$ lorsque la variable x n'apparaît pas dans le corps M de la fonction. Et l'application MN consiste à fournir un argument N à une fonction M. Cette intuition sera formalisée par la règle de réduction de base : la β-réduction. Pour alléger les notations, on supposera que l'application associe à gauche : l'écriture $\lambda x.MN$ sera utilisée pour $\lambda x.(MN)$. Un contexte est un terme dont un unique sous-terme est un vide $[\,]$. On note $C[M]$ (respectivement $C[C']$) le terme (resp. le contexte) obtenu en remplaçant le vide de C par le terme M (resp. le contexte C'). Cette opération est définie formellement de la façon suivante.

$$\begin{array}{rcl@{\qquad}rcl}
[\,][M] &=& M & (\lambda x.C)[M] &=& \lambda x.C[M] \\
(CN)[M] &=& C[M]N & (NC)[M] &=& NC[M]
\end{array}$$

$$\begin{array}{rcl@{\qquad}rcl}
[\,][C'] &=& C' & (\lambda x.C)[C'] &=& \lambda x.C[C'] \\
(CN)[C'] &=& C[C']N & (NC)[C'] &=& NC[C']
\end{array}$$

Pour clarifier les notations, nous écrirons $[M]$ pour $[\,][M]$ et on mettra, si nécessaire, en exergue la nature du contexte C est le notant $C[\,]$.

En toute rigueur, un sous-terme d'un terme M est caractérisé par un couple (C,N) constitué d'un contexte C et d'un terme N qui vérifient $M = C[N]$. Lorsqu'il n'y aura pas d'ambiguïté, on se contentera de noter N le sous-terme (C,N).

Une variable x peut avoir plusieurs occurrences dans un même terme, par exemple dans le terme $M = (\lambda x.x)x$. Dans cet exemple, les occurrences de x jouent des rôles différents. L'occurrence contenue par l'abstraction est, informellement, une variable de fonction et joue donc le rôle de variable muette. Cette distinction est formalisée par la notion de variable *libre* et *liée*. Une occurrence de la variable x dans un terme M est dite *liée* si et seulement s'il existe deux contextes $C[\,]$ et $C'[\,]$ tels que $M = C[\lambda x.C'[x]]$ et tous les sous-contextes de $C'[\,]$ qui sont de la forme $\lambda y.C''[\,]$ vérifient $x \neq y$. On dira que l'abstraction $\lambda x.C'[x]$ lie l'occurrence x. Si une occurrence d'une variable n'est pas liée, cette occurrence est dite *libre*. Dans l'exemple mentionné plus haut, l'occurrence contenue dans l'abstraction $\lambda x.x$ est liée par cette abstraction alors que l'autre occurrence est libre dans M. On définit formellement l'ensemble $\mathtt{BV}(M)$ des variables liées de M et l'ensemble $\mathtt{FV}(M)$ des variables libres de M. Ces ensembles s'étendent naturellement aux contextes.

$$\begin{array}{rcl@{\qquad}rcl}
\mathtt{BV}(x) &=& \emptyset & \mathtt{FV}(x) &=& \{x\} \\
\mathtt{BV}(MN) &=& \mathtt{BV}(M) \cup \mathtt{BV}(N) & \mathtt{FV}(MN) &=& \mathtt{FV}(M) \cup \mathtt{FV}(N) \\
\mathtt{BV}(\lambda x.M) &=& \{x\} \cup \mathtt{BV}(M) & \mathtt{FV}(\lambda x.M) &=& \mathtt{FV}(M) - \{x\}
\end{array}$$

$$\begin{array}{rcl@{\qquad}rcl}
\mathtt{BV}([\,]) &=& \emptyset & \mathtt{FV}([\,]) &=& \emptyset \\
\mathtt{BV}(C[\,]\,N) &=& \mathtt{BV}(C[\,]) \cup \mathtt{BV}(M) & \mathtt{FV}(C[\,]\,N) &=& \mathtt{FV}(C[\,]) \cup \mathtt{FV}(N) \\
\mathtt{BV}(MC[\,]) &=& \mathtt{BV}(M) \cup \mathtt{BV}(C[\,]) & \mathtt{FV}(MC[\,]) &=& \mathtt{FV}(M) \cup \mathtt{FV}(C[\,]) \\
\mathtt{BV}(\lambda x.C[\,]) &=& \{x\} \cup \mathtt{BV}(C[\,]) & \mathtt{FV}(\lambda x.C[\,]) &=& \mathtt{FV}(C[\,]) - \{x\}
\end{array}$$

L'opération de substitution $M\{x\backslash N\}$ définie ci-dessous remplace les occurrences libres de la variable x dans M ou $C[\,]$ par le terme N.

$$\begin{array}{rcll}
x\{x\backslash N\} &=& N & \\
y\{x\backslash N\} &=& y & \text{si } x \neq y \\
(MM')\{x\backslash N\} &=& M\{x\backslash N\}M'\{x\backslash N\} & \\
(\lambda x.M)\{x\backslash N\} &=& \lambda x.M & \\
(\lambda y.M)\{x\backslash N\} &=& \lambda z.(M\{y \leftarrow z\}\{x\backslash N\}) & \text{où } z = \mathsf{Conv}_\alpha(x,y,M,N)
\end{array}$$

$$\begin{array}{rcll}
[\,]\{x\backslash N\} &=& [\,] & \\
(CM)\{x\backslash N\} &=& C\{x\backslash N\}N\{x\backslash M\} & \\
(MC)\{x\backslash N\} &=& M\{x\backslash N\}C\{x\backslash N\} & \\
(\lambda x.C)\{x\backslash N\} &=& \lambda x.C & \\
(\lambda y.C)\{x\backslash N\} &=& \lambda z.C\{y \leftarrow z\}\{x\backslash N\} & \text{où } z = \mathsf{Conv}_\alpha(x,y,M,N)
\end{array}$$

Cette opération est définie de sorte que les variables libres de N ne soient pas *capturées*, c'est-à-dire liées par une abstraction de M. Cette contrainte apparaît notamment dans le cas où la substitution opère dans une abstraction $\lambda y.M$. Si y est une variable libre de N (et que x est libre dans M), alors on renomme la variable liée par l'abstraction pour ne par capturer les occurrences libres de y dans

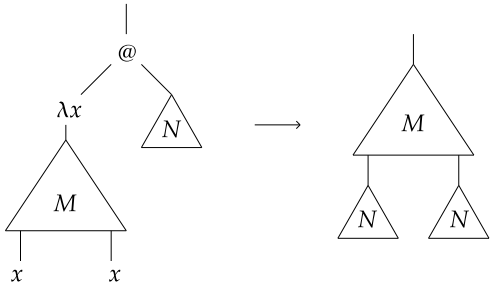


FIG. 1.1 – *β-réduction*

N. Cette opération, que nous appellerons plus tard α-conversion, est un remplacement syntaxique $\{y \leftarrow z\}$ des variables liées y par z. La variable z choisie pour le renommage est issue de la fonction Conv_α : cette dernière fournit un nom de variable qui ne cause pas de capture. Cette fonction est formellement définie de la façon suivante.

Si $x \notin \mathtt{FV}(M)$ **ou** $y \notin \mathtt{FV}(N)$, alors $\mathsf{Conv}_\alpha(x,y,M,N) = y$.
Sinon, $\mathsf{Conv}_\alpha(x,y,M,N)$ est la première variable de l'énumération $x_1,x_2,x_3,\ldots$ de $\mathbf{X}$ telle que $z \notin \mathtt{FV}(M) \cup \mathtt{FV}(N)$.

Si M contient une occurrence libre de x, et $M = C[x]$, alors les termes $M\{x\backslash N\}$ et $C[N]$ ne sont donc pas toujours égaux. Par exemple si $M = \lambda y.yx$ et $N = y$, on a $M\{x\backslash N\} = \lambda z.zy$ alors que $C[N] = \lambda y.yy$.

La définition de la substitution, que nous utiliserons plus tard pour définir les réductions, montre que le nom d'une variable liée n'a, intuitivement pas d'importance. De ce fait, la relation d'égalité de terme pertinente est la notion d'égalité modulo renommage d'occurrence liée (ou α-conversion). On définit cette relation $\equiv$ de la façon suivante.

$$\begin{array}{rcll} x & \equiv & x & \\ \lambda x.M & \equiv & \lambda y.M'\{x\backslash y\} & \text{si } y \notin \mathtt{FV}(\lambda x.M) \text{ et } M \equiv M' \\ MN & \equiv & M'N' & \text{si } M \equiv M' \text{ et } N \equiv N' \end{array}$$

La relation $\equiv$ est une relation d'équivalence. Par la suite, pour alléger les notations, nous écrirons $M = N$ en lieu et place de $M \equiv N$.

La réduction de base du λ-calcul est la β-réduction. Cette réduction formalise les descriptions intuitives que nous avions données pour l'abstraction et l'application. Ainsi, lorsqu'un terme N est appliqué en argument à la fonction (de x) $\lambda x.M$, on obtient un terme où les occurrences *formelles* de x dans M sont remplacées par l'argument *réel* N. Cette réduction est illustrée sur la figure 1.1. Les règles de contexte permettent de contracter n'importe quel sous-terme.

$$(\beta)\ (\lambda x.M)N \to M\{x\backslash N\}$$

$$(\nu)\ \frac{M \to M'}{MN \to M'N} \qquad (\mu)\ \frac{N \to N'}{MN \to MN'} \qquad (\xi)\ \frac{M \to M'}{\lambda x.M \to \lambda x.M'}$$

On utilise la notation habituelle $\twoheadrightarrow$ pour la fermeture réflexive et transitive de $\to$. On remarque que $M \to M'$ si et seulement s'il existe un contexte C tel que $M = C[(\lambda x.N)P]$ et $M' = C[N\{x\backslash P\}]$. De façon plus générale, les sous-termes $r = (C,R)$ d'un terme M où R est de la forme $R = (\lambda x.N)P$ sont appelés *β-radicaux*. La notation $M \xrightarrow{r} M'$ permet de préciser le radical impliqué dans la réduction. Dans ce cas, on dit que cette réduction élémentaire contracte le radical (C,R). Si

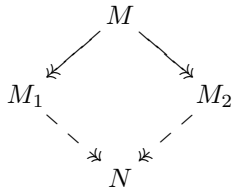


FIG. 1.2 – *Confluence*

$R = (\lambda x.N)P$, on a $M' = C[N\{x\backslash P\}]$. Dans ce cas, le sous-terme $(C,N\{x\backslash P\})$ de M' est appelé *contractum* de (C,R). Par la suite, en l'absence d'ambiguïté, le radical (C,R) sera noté R. Un terme qui ne contient pas de radical est dit en *forme normale*. On dit que M a une forme normale s'il existe une réduction $M \twoheadrightarrow M'$ telle que M' est en forme normale. Dans ce cas, M est dit *normalisable*. Si toutes les réductions issues de M aboutissent à une forme normale, M est dit *fortement normalisable*. La notation $\mathcal{R} : M \to M'$ (respectivement $\mathcal{R} : M \twoheadrightarrow M'$) permet de donner un nom $\mathcal{R}$ à une réduction élémentaire (resp. une réduction). Lorsque $\mathcal{R} : M \twoheadrightarrow M'$ et $\mathcal{R}' : M' \twoheadrightarrow M''$, on peut composer ces réductions : $(\mathcal{R};\mathcal{R}') : M \twoheadrightarrow M' \twoheadrightarrow M''$. La composition de réduction est une opération associative ayant comme élément neutre la réduction qui ne réduit aucun radical, appelée réduction vide et notée $\emptyset$.

On énumère les propriétés fondamentales du λ-calcul : la confluence (aussi appelé théorème de Church-Rosser) et les théorèmes des développements finis et de standardisation. Les preuves complètes de ces résultats peuvent être trouvées dans [7, 24].

On note qu'un terme peut contenir plusieurs β-radicaux distincts. Par exemple, le terme $M = (\lambda y.Iu)z$ (où on utilise la notation habituelle $I = \lambda x.x$) contient deux radicaux : M et Iu. Plusieurs réductions issues d'un même terme sont donc possibles. Par exemple, on a $M \to Iu$ et $M \to (\lambda y.u)z$. Cependant, il s'avère que le λ-calcul vérifie une propriété, fondamentale, de confluence, illustrée sur la figure 1.2.

Résultat 1.1 (Confluence) *Si $M \twoheadrightarrow M_1$ et $M \twoheadrightarrow M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow N$ et $M_2 \twoheadrightarrow N$.*

Si deux réductions issues d'un même terme M aboutissent à des termes M_1 et M_2 différents, il existe deux réductions issues de ces deux termes qui aboutissent à un même terme N.

Si $\mathcal{R} : M \xrightarrow{r} N$ et que s est un radical de N, il est intéressant de savoir ce qu'il est advenu de s dans N. On introduit pour cela la notion de *résidu* de radical qui est illustrée sur la figure 1.3 et qui est définie de la façon suivante. On pose $r = (C[\,],R)$ et $s = (C'[\,],S)$. Le contractum de r est noté $(C[\,],R')$.

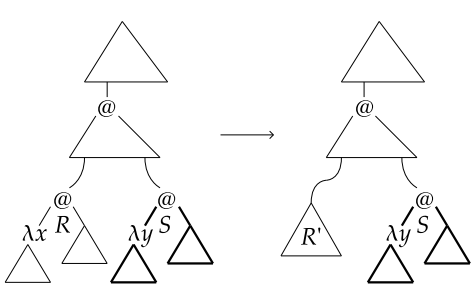
Si $r = s$, alors s n'a pas de résidu dans N.

Si r **et** s **sont deux sous-expressions disjointes**, alors, en supposant, par exemple, que r est à gauche de s, on a $M = C_0[PQ]$ avec $C[\,] = C_0[C_1[\,]\,Q]$ et $C'[\,] = C_0[PC_2[\,]]$. Le résidu de s dans N est $(C_0[C_1[R']\,C_2[\,]],S)$.

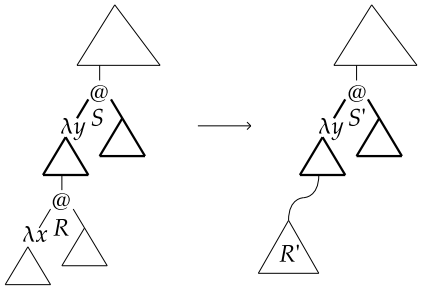
Si s **contient** r, alors on a $S = C_1[R]$ où $C_1[\,] \neq [\,]$. Par conséquent, $s' = (C'[\,],C_1[R'])$ est un radical de N et s' est le radical résidu de s dans N.

Si r **contient** s, on pose $R = (\lambda x.P)Q$ et $R' = P\{x\backslash Q\}$.

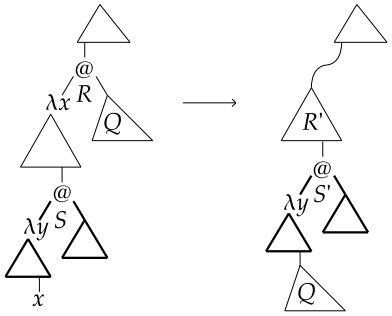
1. Si s est dans P, alors $C'[\,] = C[(\lambda x.C_1[\,])Q]$. On pose $C''[\,] = C[C_1\{x\backslash Q\}[\,]]$. Le radical $(C''[\,],S\{x\backslash Q\})$ est le résidu de s dans N.
2. Si s est dans Q, alors $C'[\,] = C[(\lambda x.P)C_1[\,]]$. Si la variable x n'a pas d'occurrence libre dans P, s n'a pas de résidu dans N. Si la variable x a une occurrence libre dans P, on a $P = C_2[x]$. On peut supposer, en renommant éventuellement des variables



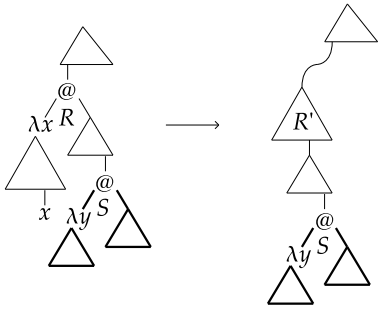
R et S sont disjoints



S contient R



S est dans l'abstraction de R



S est dans l'argument de R

FIG. 1.3 – *Résidus d'un radical S après la contraction du radical R*

liées, qu'on a $\mathtt{FV}(Q) \cap \mathtt{BV}(P) = \emptyset$. De là, on a $P\{x\backslash Q\} = C_1\{x\backslash Q\}[Q]$ et on pose $C''[\,] = C[C_2\{x\backslash Q\}[C_1[\,]]]$. Le radical $(C''[\,],S)$ est un résidu de s dans N.

Si $\mathcal{R} : M \to M'$, on note $r/\mathcal{R}$ l'ensemble des résidus de r par $\mathcal{R}$. De même, si $\mathcal{F}$ est un ensemble de radicaux de M, on note $\mathcal{F}/\mathcal{R}$ l'ensemble des résidus des radicaux de $\mathcal{F}$. Dans M', les radicaux qui ne sont pas un résidu d'un radical de M sont dits *créés* par la réduction $\mathcal{R}$. Par exemple, si on considère le terme $R = \Delta I$ (où on utilise la notation habituelle $\Delta = \lambda x.xx$) et la réduction $\mathcal{R} : R \to II = R'$, le radical R n'a pas de résidu dans R' : le radical R' est donc créé par $\mathcal{R}$. On note que la contraction d'un radical peut créer un radical de deux façons différentes. Un radical peut être créé *par le haut* comme dans la réduction $(\lambda x.I)u \to Iu$: dans ce cas la contraction du radical *relie* l'application située au-dessus du radical avec l'abstraction située sous l'abstraction du radical. Ou bien, un radical peut être créé *par le bas* comme dans la réduction $(\lambda x.xu)I$: la contraction du radical relie une application contenue dans l'abstraction du radical à l'abstraction en argument. Un radical peut être à la fois créé par le haut et par le bas lorsque l'abstraction du radical est I, comme dans l'exemple $(I\Delta)\Delta \to \Delta\Delta$.

Une réduction *relative à un ensemble de radicaux* $\mathcal{F}$ ne contracte que des radicaux ou des résidus de $\mathcal{F}$. Une réduction $\mathcal{R} : M \twoheadrightarrow M'$ relative à $\mathcal{F}$ est dite *complète* si N ne contient aucun résidu de radicaux de $\mathcal{F}$, ce qui peut aussi s'écrire $\mathcal{F}/\mathcal{R} = \emptyset$. Les réductions complètes sont aussi appelées *développements*. Les développements du λ-calcul vérifient la propriété fondamentale suivante.

Résultat 1.2 (Développements finis) *Soit $\mathcal{F}$ est un ensemble de radicaux de M.*

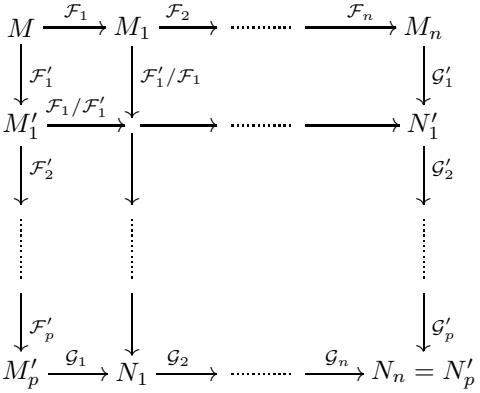
1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Les développements de $\mathcal{F}$ finissent tous sur un même terme N.*
3. *L'ensemble des résidus d'un radical R de M dans N est indépendant du développement considéré.*

Cette propriété exprime le fait qu'un ensemble de radicaux ne peut être responsable d'une réduction infinie. C'est la création de nouveaux radicaux qui peut créer une divergence. Ainsi, après une réduction, le terme $\Delta\Delta$ ne contient plus aucun résidu d'un radical du terme initial. Mais à chaque réduction, un nouveau radical est créé, ce qui explique la divergence. Par ailleurs, on observe que les développements sont équivalents du point de vue de la notion de résidu. Par conséquent, par abus, on notera indifféremment $M \xrightarrow{\mathcal{F}} M'$ tous les développements de $\mathcal{F}$. La réduction $\emptyset$ est le développement de l'ensemble vide.

Ce résultat permet de définir une nouvelle opération sur les réductions. On considère deux réduction $\mathcal{R} : M \xrightarrow{\mathcal{F}_1} M_1 \xrightarrow{\mathcal{F}_2} \ldots \xrightarrow{\mathcal{F}_n} M_n$ et $\mathcal{R}' : M \xrightarrow{\mathcal{F}'_1} M'_1 \xrightarrow{\mathcal{F}'_2} \ldots \xrightarrow{\mathcal{F}'_p} M'_n$ issues de M. La **réduction-résidu** de $\mathcal{R}'$ par $\mathcal{R}$ et, respectivement, la réduction-résidu de $\mathcal{R}$ par $\mathcal{R}'$, notées respectivement $\mathcal{R}'/\mathcal{R}$ et $\mathcal{R}/\mathcal{R}'$, sont définies de la façon suivante : pour $1 \leq i \leq p$ et $1 \leq j \leq n$, on définit récursivement les réductions $\mathcal{R}_j$ et $\mathcal{R}'_i$ et les ensembles de radicaux $\mathcal{G}_j$ et $\mathcal{G}'_i$ de la façon suivante.

$$\begin{array}{ll} \text{Pour } j \in \{1 \ldots n\} & \mathcal{G}_1 = \mathcal{F}_1/\mathcal{R}' \text{ et pour } j > 1 : \mathcal{G}_j = \mathcal{F}_j/\mathcal{R}_{j-1} \\ & \mathcal{R}_j : M_n \xrightarrow{\mathcal{G}_1} N_1 \xrightarrow{\mathcal{G}_2} \ldots \xrightarrow{\mathcal{G}_j} N_j \\ \text{Pour } i \in \{1 \ldots p\} & \mathcal{G}'_1 = \mathcal{F}'_1/\mathcal{R} \text{ et pour } i > 1 : \mathcal{G}'_i = \mathcal{F}'_i/\mathcal{R}'_{i-1} \\ & \mathcal{R}'_i : M_n \xrightarrow{\mathcal{G}'_1} N'_1 \xrightarrow{\mathcal{G}'_2} \ldots \xrightarrow{\mathcal{G}'_i} N'_i \end{array}$$

La réduction $\mathcal{R}'_p$ est $\mathcal{R}'/\mathcal{R}$ et $\mathcal{R}_n$ est $\mathcal{R}/\mathcal{R}'$. Le théorème des développements finis implique que ces deux réductions-résidus finissent sur un même terme.



Cette notion de réduction-résidu nous permet de définir une relation entre réductions issues d'un même terme : on a $\mathcal{R} \leq \mathcal{R}'$ si et seulement si $\mathcal{R}/\mathcal{R}' = \emptyset^n$. A l'aide du théorème des développements finis, on obtient les propriétés suivantes.

Résultat 1.3

1. *(a)* $(\mathcal{R}_1; \mathcal{R}_2)/\mathcal{R} = (\mathcal{R}_1/\mathcal{R}); (\mathcal{R}_2/(\mathcal{R}/\mathcal{R}_1))$
   (b) $\mathcal{R}/(\mathcal{R}_1; \mathcal{R}_2) = (\mathcal{R}/\mathcal{R}_1)/\mathcal{R}_2$
   (c) $\emptyset/\mathcal{R} = \emptyset$
   (d) $\mathcal{R}/\emptyset = \mathcal{R}$
2. *La relation $\leq$ est un préordre. L'équivalence associée $\sim$ vérifie :*
   (a) Si $\mathcal{R}_1 \sim \mathcal{R}'_1$ et $\mathcal{R}_2 \sim \mathcal{R}'_2$, alors $(\mathcal{R}_1; \mathcal{R}_2) \sim (\mathcal{R}'_1; \mathcal{R}'_2)$.
   (b) Si $\mathcal{R}_1 \sim \mathcal{R}'_1$ et $\mathcal{R}_2 \sim \mathcal{R}'_2$, alors $(\mathcal{R}_1/\mathcal{R}_2) \sim (\mathcal{R}'_1/\mathcal{R}'_2)$.
3. *On considère l'ensemble quotient $(\mathcal{R}(M)/\sim)$ où $\mathcal{R}(M)$ est l'ensemble des réductions issues de M. On note $[\mathcal{R}]$ la classe de $\mathcal{R}$.*
   (a) $[\emptyset]$ est l'élément minimal

(b) Pour tout couple $(\mathcal{R}_1,\mathcal{R}_2)$*, il existe une réduction* $\mathcal{R}$ *telle que* $[\mathcal{R}]$ *est le plus petit majorant de* $[\mathcal{R}_1]$ *et* $[\mathcal{R}_2]$*. On la note* $[\mathcal{R}_1] \sqcup [\mathcal{R}_2]$*.*

L'équivalence $\sim$ est une congruence vis-à-vis de la composition et de l'opération de résidu. L'ensemble quotient $(\mathcal{R}(M)/\leq)$ a une structure de sup-treillis.

On définit un ordre entre les radicaux d'un même terme. Si r et s sont deux radicaux du terme M, on dit que r **est à gauche de** s (ce que l'on note $r \leq_g s$) si et seulement si r contient s ou si r et s sont disjoints et r se trouve à gauche de s. Cet ordre vérifie les propriétés suivantes.

Résultat 1.4 *On suppose* $\mathcal{R} : M \xrightarrow{s} N$*. Soit* r *un radical de* M *qui vérifie* $r <_g s$*.*

1. *Le radical* r *a un unique résidu* r' *dans* M'*.*
2. *Le radical* r' *est strictement à gauche des radicaux créés par* $\mathcal{R}$*.*
3. *Si* t *est un radical de* M *à droite de* r*, alors les résidus de* t *dans* M' *sont à droite de* r'*.*

Ces propriétés impliquent, en particulier, qu'un radical r ne peut être dupliqué par la contraction de radicaux plus à droite que r ou ses résidus. La réduction $M_0 \xrightarrow{r_1} M_1 \xrightarrow{r_2} M_2 \xrightarrow{r_3} \cdots \xrightarrow{r_n} M_n$ est **standard** si et seulement si pour tout i,j tels que $1 \leq i < j \leq n$ le radical r_j n'est pas un résidu d'un radical r'_j de X_{i-1} situé à gauche de r_i. En d'autres termes, une réduction standard réduit les radicaux de l'extérieur vers l'intérieur et de la gauche vers la droite. Les propriétés mentionnées plus haut sur l'ordre des radicaux impliquent que dans le λ-calcul, tout terme atteignable par une réduction quelconque est atteignable par une réduction standard : c'est le théorème de standardisation.

Résultat 1.5 (Standardisation) *Si* $M \twoheadrightarrow N$*, il existe une réduction standard* $\mathcal{R} : M \twoheadrightarrow N$*.*

Pour obtenir la syntaxe des préfixes des termes du λ-calcul, on étend la syntaxe des termes du λ-calcul en ajoutant une nouvelle construction, Ω, qui marque la limite du préfixe.

$$\begin{array}{ll} \text{Préfixes} & X, Y ::= x \mid \lambda x.X \mid XY \mid \Omega \\ \texttt{FV}(\Omega) = \emptyset & \texttt{BV}(\Omega) = \emptyset \qquad \Omega\{x\backslash X\} = \Omega \end{array}$$

Les définitions des variables libres et liées et de la substitution sont étendues de façon naturelle. Les préfixes permettent d'introduire une relation d'ordre sur les termes. Cette relation est définie formellement de la façon suivante. Un préfixe X est préfixe d'un autre préfixe (ou d'un autre terme) Y, ce que l'on note $X \preccurlyeq Y$, si Y coïncide avec X dans les limites de X.

$$\begin{array}{rcl} x & \preccurlyeq & x \\ \Omega & \preccurlyeq & X \\ \lambda x.X & \preccurlyeq & \lambda x.X' \quad \text{si et seulement si } X \preccurlyeq X' \\ XY & \preccurlyeq & X'Y' \quad \text{si et seulement si } X \preccurlyeq X' \text{ et } Y \preccurlyeq Y' \end{array}$$

Cette relation est un ordre bien fondé dont Ω est l'élément minimal. On a, par exemple, la relation $X = (\lambda x.\lambda y.\Omega)\Omega \preccurlyeq (\lambda x.\lambda y.x)z = M$. On note que le préfixe X se réduit en $X' = \lambda y.\Omega$ alors qu'on a $M \to \lambda y.z = M'$. On a $X' \preccurlyeq M'$ ce qui signifie que la relation entre X et M est conservée après les réductions de ces termes. Cette propriété de monotonie est énoncée ci-dessous et illustrée sur la figure 1.2.

Résultat 1.6 (Monotonie) *Si* $X \preccurlyeq Y$ *et* $X \twoheadrightarrow X'$ *alors il existe un préfixe* Y' *tel que* $Y \twoheadrightarrow Y'$ *et* $X' \preccurlyeq Y'$*.*

Si X préfixe un terme Y et si X se réduit vers X' alors Y peut se réduire vers un terme préfixé par X'.

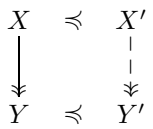


FIG. 1.4 – *Monotonie*

On distingue un sous-ensemble de $\mathbf{\Lambda}$ noté $\mathbf{V}$. Les termes de cet ensemble sont appelés *valeurs*. Les valeurs sont les abstractions.

$$V \in \mathbf{V} ::= \lambda x.X$$

L'ensemble $\mathbf{V}$ est particulièrement remarquable du fait de la propriété suivante : $\mathbf{V}$ est un sous-ensemble strict de $\mathbf{\Lambda}$ qui est stable par réduction et par substitution. Par contraste, l'ensemble des applications n'est pas stable par réduction et l'ensemble des variables n'est pas stable par substitution. En se concentrant plus particulièrement sur cet ensemble particulier de termes, on énonce la propriété de stabilité du λ-calcul.

Résultat 1.7 (Stabilité) *Si $M \twoheadrightarrow V$, il existe un préfixe X de M tel que, pour tout Y, si $Y \preccurlyeq M$ et $Y \twoheadrightarrow V'$, on a $X \preccurlyeq Y$.*

Cette propriété peut être intuitivement comprise de la façon suivante. Si un terme M peut se réduire vers une valeur V, seul un préfixe X de M sert effectivement dans cette réduction : en d'autres mots, tout préfixe de M minoré (au sens de $\preccurlyeq$) par X peut se réduire vers une valeur. Ce préfixe particulier est appelé **préfixe de stabilité** et est noté $\mathcal{P}_S(M)$. Pour illustrer cette propriété, on considère la réduction suivante : $M = (\lambda x.\lambda y.x)(\lambda z.z)u \to (\lambda x.\lambda y.x)u \to \lambda y.u = V$. Dans ce cas, on a $\mathcal{P}_S(M) = (\lambda x.\lambda y.\Omega)\Omega$. Ce préfixe se réduit de la façon suivante : $\mathcal{P}_S(M) \to \lambda y.\Omega = V'$. Les préfixes stricts de $\mathcal{P}_S(M)$ ne se réduisent pas vers une valeur. Par exemple, on a $Y = (\lambda x.\Omega)\Omega \to \Omega$. Du fait de la propriété de monotonie, on a $V' \preccurlyeq V$. Le résultat de stabilité met en lumière le fait que les sous-termes de M qui sont conservés dans $\mathcal{P}_S(M)$ jouent un rôle dans l'obtention de la valeur V. Par contraste, les autres sous-termes sont effacés car ils n'interviennent pas dans l'obtention de V. La définition suivante formalise cette distinction.

Définition 1.1 (Sous-terme critique) *Si $M \twoheadrightarrow V$, alors le sous-terme $(C[\,],N)$ de M est critique si et seulement s'il existe un terme N' tel que le terme $C[N']$ ne se réduit pas vers une valeur.*

Un sous-terme critique N intervient de manière cruciale dans l'obtention d'une valeur : en effet, son remplacement par un autre sous-terme N' peut éventuellement conduire à une réduction n'aboutissant plus à une valeur. On illustre cette définition en reprenant l'exemple précédent. Le sous-terme $N = (\lambda z.z)u$ de M n'est pas critique : en effet, quel que soit le terme N', on a $\mathcal{P}_S(M) \preccurlyeq (\lambda x.\lambda y.x)N' = M'$. Du fait de la propriété de monotonie, on obtient $M' \to M''$ où $V' \preccurlyeq M''$ ce qui implique que M'' est une valeur. Les deux résultats précédents se rapprochent du lemme de généricité introduit par Barendregt dans [7]. Cette propriété s'appuie sur la notion de terme résoluble. Un terme M est **résoluble** si et seulement s'il existe des termes $N_1, \ldots, N_n$ tels que $MN_1 \ldots N_n \twoheadrightarrow I$. Cette définition permet d'énoncé le lemme de généricité.

Résultat 1.8 (Généricité) *Si M est non-résoluble, si N est une forme normale et si on a $C[M] \twoheadrightarrow N$, alors pour tout P, on a $C[P] \twoheadrightarrow N$.*

Si le terme $C[M]$ où M est non-résoluble, aboutit à une forme normale, alors le terme $C[P]$ obtenu en remplaçant M par n'importe quel autre terme P se réduit également vers la même forme normale. Intuitivement, la réduction $C[M] \twoheadrightarrow N$ signifie que le terme non-résoluble M ne participe pas à l'obtention de N. Le remplacement de M par P n'a donc pas d'effet : le terme $C[P]$ se réduit aussi vers N. Plus généralement, ce résultat rejoint l'intuition mentionnée plus tôt

au sujet du résultat de stabilité : une réduction ne fait intervenir qu'un préfixe du terme initial. D'une part, le résultat de stabilité énonce le fait que seul un préfixe du terme initial contribue à l'obtention d'une valeur. D'autre part, le lemme de généricité signifie que l'obtention d'une forme normale N à partir d'un terme $C[M]$ ne fait intervenir qu'un préfixe X de ce terme, dans lequel le sous-terme M est effacé.

La présentation du λ-calcul faite dans cette section ne se veut pas exhaustive. En particulier, nous nous limitons volontairement aux propriétés que nous examinerons dans les chapitres suivants au moment de l'étude des variantes du λ-calcul. Certains aspects du λ-calcul qui sont ici omis ou traités de façon partielle sont exposés dans [7]. La première variante du λ-calcul que nous est étudions est le λ-calcul étiqueté.

1.2 Le λ-calcul étiqueté

Dans cette section, nous rappelons la syntaxe et les propriétés essentielles du λ-calcul étiqueté introduit dans [29]. Dans ce calcul, des étiquettes sont ajoutées aux termes du λ-calcul et la réduction devient étiquetée. Les théorèmes de confluence, des développements finis et de standardisation du λ-calcul sont conservés. A l'aide des étiquettes, on peut nommer les radicaux. Le fait que les résidus d'un radical ont le même nom que ce radical constitue la propriété fondamentale du λ-calcul étiqueté. Plus généralement, le nom d'un radical relie de façon univoque ce dernier à la famille de radicaux à laquelle il appartient. En effet, les étiquettes enregistrent intuitivement les réductions passées qui ont abouti au terme présent : deux radicaux de même nom ont donc été créés de la même façon. En d'autres mots, les étiquettes contiennent les relations de dépendance des termes présents vis-à-vis des termes passés. Les étiquettes permettent donc d'enrichir la propriété de stabilité : les étiquettes d'une valeur obtenue à l'issue d'une réduction définissent le préfixe minimal nécessaire et suffisant pour obtenir une valeur.

Soit $\mathbf{A} = \{a,b,c,\ldots\}$ un ensemble dénombrable de lettres. La syntaxe des étiquettes, des termes, des préfixes, des contextes et des valeurs du λ-calcul étiqueté est définie de la façon suivante.

Etiquettes	$\alpha,\beta \in \mathbf{E}$	$::= a \mid \alpha\beta \mid \lceil\alpha\rceil \mid \lfloor\alpha\rfloor$
Termes	$M, N \in \mathbf{\Lambda}_e$	$::= x^\alpha \mid (\lambda x.M)^\alpha \mid (MN)^\alpha$
Préfixes	X, Y	$::= (\lambda x.X)^\alpha \mid (XY)^\alpha \mid x^\alpha \mid \Omega$
Contextes	$C \in \mathbf{\Lambda}_e[\,]$	$::= [\,] \mid (\lambda x.C)^\alpha \mid (CX)^\alpha \mid (XC)^\alpha$
Valeurs	$V \in \mathbf{V}_e$	$::= (\lambda x.X)^\alpha$

Une étiquette du λ-calcul étiqueté est soit une lettre a, soit une étiquette surlignée $\lceil\alpha\rceil$ ou soulignée $\lfloor\alpha\rfloor$, soit la concaténation $\alpha\beta$ de deux étiquettes α et β. Les sous-termes du λ-calcul étiqueté sont munis d'une étiquette. La syntaxe des préfixes, des contextes et des valeurs est adaptée en conséquence. On choisit toutefois de ne pas étiqueter le vide $[\,]$ des contextes car le sous-terme qui pourrait remplacer le vide contient déjà une étiquette. De même, la limite Ω d'un préfixe n'a pas d'étiquette. L'étiquette de tête d'un terme, d'un préfixe ou d'un contexte est obtenue par l'opérateur τ.

$$\begin{array}{rrr} \tau(x^\alpha) = \alpha & \tau((\lambda x.X)^\alpha) = \alpha & \tau((XY)^\alpha) = \alpha \\ \tau((\lambda x.C)^\alpha) = \alpha & \tau((CX)^\alpha) = \alpha & \tau((XC)^\alpha) = \alpha \end{array}$$

Cette opération n'est pas définie sur un contexte vide $[\,]$ ou sur la limite Ω d'un préfixe. Les définitions des variables liées et libres sont adaptées directement du λ-calcul, de même que les

relations d'égalité modulo α-conversion $\equiv_e$ et la relation de préfixe $\preccurlyeq_e$. En l'absence d'ambiguïté, ces relations seront notées $\equiv$ et $\preccurlyeq$.

$$\begin{array}{rcl@{\qquad}rcl}
\mathtt{BV}(x^\alpha) &=& \emptyset & \mathtt{FV}(x^\alpha) &=& \{x\} \\
\mathtt{BV}((XY)^\alpha) &=& \mathtt{BV}(X) \cup \mathtt{BV}(Y) & \mathtt{FV}((XY)^\alpha) &=& \mathtt{FV}(X) \cup \mathtt{FV}(Y) \\
\mathtt{BV}((\lambda x.X)^\alpha) &=& \{x\} \cup \mathtt{BV}(X) & \mathtt{FV}((\lambda x.X)^\alpha) &=& \mathtt{FV}(X) - \{x\}
\end{array}$$

$$\begin{array}{rcll}
x^\alpha &\preccurlyeq& x^\alpha & \\
\Omega &\preccurlyeq& X & \\
(\lambda x.X)^\alpha &\preccurlyeq& (\lambda x.X')^\alpha & \text{si } X \preccurlyeq X' \\
(XY)^\alpha &\preccurlyeq& (X'Y')^\alpha & \text{si } X \preccurlyeq X' \text{ et } Y \preccurlyeq Y'
\end{array}$$

La définition de la substitution $\{x\backslash X\}_e$ nécessite une adaptation pour décider du sort de l'étiquette de la variable x^α en cas de substitution. Pour cela, on définit l'opération de concaténation notée par un point "$\cdot$".

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\alpha \cdot x^\beta &=& x^{\alpha\beta} & \alpha \cdot (XY)^\beta &=& (XY)^{\alpha\beta} & \alpha \cdot (\lambda x.X)^\beta &=& (\lambda x.X)^{\alpha\beta} \\
\alpha \cdot (\lambda x.C[\,])^\beta &=& (\lambda x.C[\,])^{\alpha\beta} & \alpha \cdot (C[\,]\,Y)^\beta &=& (C[\,]\,Y)^{\alpha\beta} & \alpha \cdot (XC[\,])^\beta &=& (XC[\,])^{\alpha\beta}
\end{array}$$

L'opération de concaténation $\alpha \cdot X$ effectue simplement une concaténation de α avec l'étiquette de tête de X. Comme pour τ, cette opération n'est pas définie pour le vide $[\,]$ et la limite Ω. Cette définition permet maintenant de définir la substitution.

$$\begin{array}{rcll}
x^\alpha\{x\backslash X\}_e &=& \alpha \cdot X & \\
y^\alpha\{x\backslash X\}_e &=& y^\alpha & \text{si } x \neq y \\
(YY')^\alpha\{x\backslash X\}_e &=& (Y\{x\backslash X\}_e Y'\{x\backslash X\}_e)^\alpha & \\
(\lambda x.Y)^\alpha\{x\backslash X\}_e &=& (\lambda x.Y)^\alpha & \\
(\lambda y.Y)^\alpha\{x\backslash X\}_e &=& (\lambda z.(Y\{y \leftarrow z\}\{x\backslash X\}_e))^\alpha & \text{où } z = \mathsf{Conv}_\alpha(x,y,Y,X)
\end{array}$$

$$\begin{array}{rcll}
[\,]\{x\backslash X\}_e &=& [\,] & \\
(CY)^\alpha\{x\backslash X\}_e &=& C\{x\backslash X\}_e X\{x\backslash Y\}_e\alpha & \\
(YC)^\alpha\{x\backslash X\}_e &=& (Y\{x\backslash X\}_e C\{x\backslash X\})^\alpha & \\
(\lambda x.C)^\alpha\{x\backslash X\}_e &=& (\lambda x.C)^\alpha & \\
(\lambda y.C)^\alpha\{x\backslash X\}_e &=& (\lambda z.C\{y \leftarrow z\}\{x\backslash X\}_e)^\alpha & \text{où } z = \mathsf{Conv}_\alpha(x,y,Y,X)
\end{array}$$

La substitution de x par X dans le terme Y (notée $Y\{x\backslash X\}_e$) remplace les occurrences libres x^α de la variable x dans Y par le terme $\alpha \cdot X$. Comme pour le λ-calcul, cette définition fait en sorte de ne pas capturer des variables libres de X en renommant éventuellement certaines variables liées de Y. Le renommage d'une variable liée par une abstraction fait appel à l'opération $\{y \leftarrow z\}$. Cette opération remplace les occurrences y^α du terme par z^α. Cette α-conversion ne modifie pas les étiquettes contrairement à une substitution $X\{y\backslash z^\gamma\}$. En l'absence d'ambiguïté, on utilisera la notation $Y\{x\backslash X\}$ en lieu et place de $Y\{x\backslash X\}_e$.

Les radicaux du λ-calcul étiqueté sont de la forme $R = ((\lambda x.M)^\alpha N)^\beta$. Dans ce cas, le **nom** du radical R est l'étiquette portée par l'abstraction, ce que l'on écrit $\mathsf{nom}(R) = \alpha$. Les règles de réduction du λ-calcul étiqueté sont définies ci-dessous.

$$(\beta_e) \qquad ((\lambda x.M)^\alpha N)^\beta \to_e \beta \cdot \lceil\alpha\rceil \cdot M\{x\backslash\lfloor\alpha\rfloor \cdot N\}_e$$

$$(\nu_e)\ \frac{X \to_e X'}{(XY)^\alpha \to_e (X'Y)^\alpha} \qquad (\mu_e)\ \frac{Y \to_e Y'}{(XY)^\alpha \to_e (XY')^\alpha} \qquad (\xi_e)\ \frac{X \to_e X'}{(\lambda x.X)^\alpha \to_e (\lambda x.X')^\alpha}$$

Les règles de contexte (ν_e), (μ_e) et (ξ_e) sont de simples adaptations du λ-calcul. La règle de réduction de base est la β_e-réduction. Comme l'illustre la figure 1.5, cette règle encadre le corps de l'abstraction contractée entre le nom du radical surligné $\lceil\alpha\rceil$ (en haut du terme) et le nom du radical souligné $\lfloor\alpha\rfloor$ (au niveau des variables substituées). Les étiquettes de l'application et des variables sont conservées dans le contractum, par concaténation, à la même place que dans le radical. De ce fait, si un radical est créé par le haut par cette contraction, son nom contient, en surligné, le nom du radical qui l'a créé. Symétriquement, si un radical est créé par le bas, son nom contient, en souligné, le nom du radical qui l'a créé. On formalise cette remarque à l'aide des relations $\prec$ et $\preceq$ définies ci-dessous.

$$\begin{array}{ll}
\alpha \preceq \beta & \text{si } \alpha \prec \beta \text{ ou } \alpha = \beta \\
\alpha \prec \lceil\alpha\rceil & \\
\alpha \prec \lfloor\alpha\rfloor & \\
\alpha \prec \beta\gamma & \text{si } \alpha \preceq \beta \text{ ou } \alpha \preceq \gamma \\
\alpha \prec \beta & \text{si } \alpha \prec \gamma \text{ et } \gamma \prec \beta
\end{array}$$

La relation $\prec$ (respectivement $\preceq$) est un ordre strict (resp. un ordre) sur les étiquettes. Comme annoncé, si la contraction d'un radical de nom α crée un radical de nom β, on a $\alpha \prec \beta$. Ces remarques justifient l'intuition donnée en introduction de cette section : les étiquettes du λ-calcul enregistrent l'histoire des réductions passées qui ont donné naissance aux radicaux présents. La notion de résidu d'un radical est adaptée de façon naturelle de la notion du λ-calcul sans étiquette, comme illustré sur la figure 1.6. On suppose $X \xrightarrow{r}_e Y$ où $r = (C[\,],R)$. Soit $s = (C'[\,],S)$ un radical de X. On note $(C[\,],R')$ le contractum de r.

Si $r = s$, alors s n'a pas de résidu dans Y.

Si r **et** s **sont deux sous-expressions disjointes**, alors en supposant, par exemple, que r est à gauche de s, on a $X = C_0[(X_1X_2)^\alpha]$ avec $C[\,] = C_0[(C_1[\,]\,X_2)^\alpha]$ et $C'[\,] = C_0[(X_1C_2[\,])^\alpha]$. Le résidu de s dans Y est $(C_0[(C_1[R']\,C_2[\,])^\alpha],S)$.

Si s **contient** r, alors on a $S = C_1[R]$ où $C_1[\,] \neq [\,]$. Par conséquent, $s' = (C'[\,],C_1[R'])$ est un radical de Y et s' est le radical résidu de s dans Y.

Si r **contient** s, on pose $R = ((\lambda x.Z)^\alpha Z')^\beta$ et $R' = \beta \cdot \lceil\alpha\rceil \cdot Z\{x\backslash\lfloor\alpha\rfloor \cdot Z'\}$.

1. Si $C'[\,] = C[((\lambda x.C_1[\,])^\alpha Z')^\beta]$, deux cas sont à envisager.
   (a) Si $C_1[\,] \neq [\,]$, on pose $C''[\,] = C[\beta \cdot \lceil\alpha\rceil \cdot C_1\{x\backslash\lfloor\alpha\rfloor \cdot Z'\}[\,]]$. Le résidu de s dans Y est le radical $(C''[\,],S\{x\backslash\lfloor\alpha\rfloor \cdot Z'\})$.
   (b) Si $C_1[\,] = [\,]$, on pose $C''[\,] = C[\,]$ et $(C[\,],\beta \cdot \lceil\alpha\rceil \cdot S\{x\backslash\lfloor\alpha\rfloor \cdot Z'\})$ est le résidu de s.
2. Si $C'[\,] = C[((\lambda x.Z)^\alpha C_1[\,])^\beta]$. Si la variable x n'a pas d'occurrence libre dans Z, s n'a pas de résidu dans Y. On suppose désormais que la variable x a une occurrence (C_2,x^γ) libre dans Z (on a $Z = C_2[x^\gamma]$). On peut supposer, en renommant éventuellement des variables liées, qu'on a $\mathtt{FV}(Z') \cap \mathtt{BV}(Z) = \emptyset$. De là, on a $R' = \beta \cdot \lceil\alpha\rceil \cdot C_1\{x\backslash\lfloor\alpha\rfloor \cdot Z'\}[\lfloor\alpha\rfloor \cdot Z']$.
   (a) Si $C_1[\,] = [\,]$ et $C_2[\,] = [\,]$, alors le radical $(C[\,],\beta \cdot \lceil\alpha\rceil \cdot \gamma \cdot \lfloor\alpha\rfloor \cdot S)$ est le résidu de s dans Y.
   (b) Si $C_1[\,] = [\,]$ et $C_2[\,] \neq [\,]$, on pose $C''[\,] = C[\beta \cdot \lceil\alpha\rceil \cdot C_2\{x\backslash\lfloor\alpha\rfloor \cdot Z'\}[\,]]$. Le radical $(C'',\gamma \cdot \lfloor\alpha\rfloor \cdot S)$ est un résidu de s dans Y.
   (c) Si $C_1[\,] \neq [\,]$ et $C_2[\,] = [\,]$, on pose $C''[\,] = C[\beta \cdot \lceil\alpha\rceil \cdot \gamma \cdot \lfloor\alpha\rfloor \cdot C_1[\,]]$. Le radical $(C''[\,],S)$ est le résidu de s dans Y.
   (d) Sinon, on pose $C''[\,] = C[\beta \cdot \lceil\alpha\rceil \cdot C_2\{x\backslash\lfloor\alpha\rfloor \cdot Z'\}[\gamma \cdot \lfloor\alpha\rfloor \cdot C_1[\,]]]$. Le radical $(C''[\,],S)$ est un résidu de s dans Y.

Une étude attentive de cette définition aboutit à la remarque fondamentale suivante.

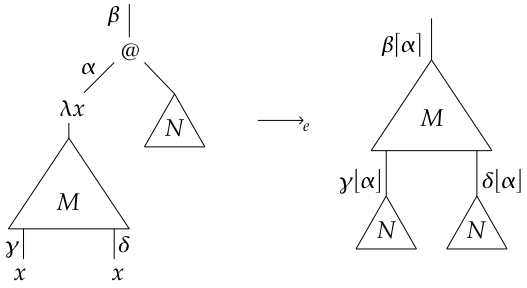
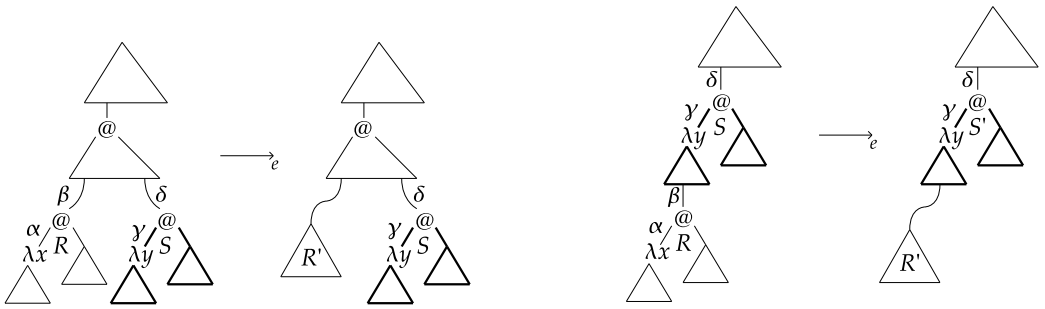
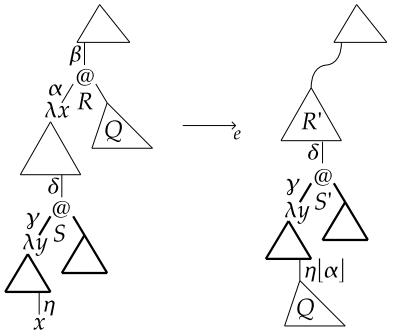


FIG. 1.5 – *Réduction dans le λ-calcul étiqueté*

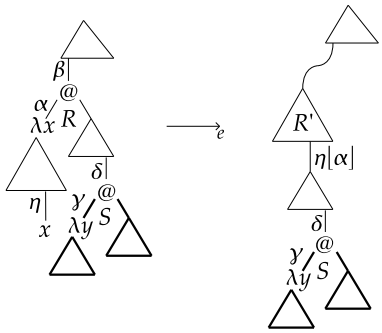


R et S sont disjoints

S contient R



S est dans l'abstraction de R



S est dans l'argument de R

FIG. 1.6 – *Résidus d'un radical S après la contraction du radical S*

Résultat 1.9 *On suppose $M \to_e M'$. Si R est un radical de M et si S est un résidu de R dans M', alors on a $\mathsf{nom}(R) = \mathsf{nom}(S)$.*

Au cours d'une réduction, tous les résidus d'un même radical ont le même nom. La réciproque est fausse. Cependant, si les étiquettes du terme initial sont des lettres distinctes, l'ensemble des radicaux portant le même nom constitue une famille de radicaux qui ont tous été créés de la même manière. Ces propriétés n'étant pas utilisées dans cette dissertation, nous ne développons pas davantage cette notion introduite et étudiée dans [29]. Comme annoncé précédemment, les propriétés fondamentales du λ-calcul sont conservées par le λ-calcul étiqueté.

Résultat 1.10 (Confluence) *Si $M \twoheadrightarrow_e M_1$ et $M \twoheadrightarrow_e M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_e N$ et $M_2 \twoheadrightarrow_e N$.*

Le λ-calcul étiqueté est confluent, comme le montre la figure 1.7. Ce résultat se prouve de la même manière que la confluence du λ-calcul par valeur étiqueté (p. 53).

Résultat 1.11 (Développements finis) *Soit $\mathcal{F}$ est un ensemble de radicaux de M.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Les développements de $\mathcal{F}$ finissent tous sur un même terme N.*
3. *L'ensemble des résidus d'un radical R de M dans N est indépendant du développement considéré.*

Les réductions relatives à un ensemble de radicaux sont finies. Ceci implique qu'une réduction infinie s'explique toujours par une création perpétuelle de radicaux. La preuve de ce résultat peut être adaptée, de façon élémentaire, de la preuve intuitive du théorème des développements finis (p. 60) dans le cadre du λ-calcul par valeur étiqueté.

Résultat 1.12 (Standardisation) *Si $M \twoheadrightarrow_e N$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_e N$ telle que $\mathcal{R}$ est standard.*

Tout terme atteignable par une réduction du λ-calcul étiqueté peut être atteint par une réduction standard. On peut montrer ce résultat en procédant de la même façon que pour le λ-calcul par valeur (p. 34).

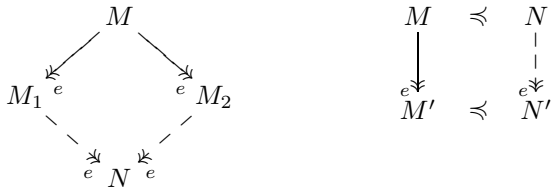
Comme dans le λ-calcul, la réduction du λ-calcul étiqueté est monotone, ce qui est exprimé de façon formelle par l'énoncé suivant.

Résultat 1.13 (Monotonie) *Si $X \preccurlyeq Y$ et $X \to_e X'$, alors il existe un préfixe Y' tel que $Y \twoheadrightarrow_e Y'$ et $X' \preccurlyeq Y'$.*

Si le terme étiqueté X préfixe le terme Y et si X se réduit vers X', alors le terme Y peut se réduire vers un terme Y' qui est préfixé par X'. Ce résultat est illustré sur la figure 1.7. Dans le cadre du λ-calcul par valeur (p. 40), on exposera une preuve de monotonie qui pourrait s'adapter au λ-calcul étiqueté. On concrétise cette propriété de monotonie en reprenant l'exemple employé dans la section précédente. Le terme $M = ((\lambda x.(\lambda y.x^d)^c)^b z^f)^a$ est préfixé par $X = ((\lambda x.(\lambda y.\Omega)^c)^b \Omega)^a$ qui se réduit vers $X' = (\lambda y.\Omega)^{a\lceil b\rceil c}$. On a bien la réduction $M \to_e (\lambda y.z^{d\lfloor b\rfloor f})^{a\lceil b\rceil c} = M'$ avec $X' \preccurlyeq M'$. De même que la propriété de monotonie, la propriété de stabilité est conservée dans le λ-calcul étiqueté. Elle est illustrée sur la figure 1.7.

Résultat 1.14 (Stabilité) *Si $M \twoheadrightarrow_e V$, il existe préfixe X de M tel que, pour tout Y, si $Y \preccurlyeq M$ et $Y \twoheadrightarrow_e V'$, on a $X \preccurlyeq Y$.*

Si un terme M se réduit vers une valeur, il existe un préfixe minimum de M qui se réduit vers une valeur. Dans le cadre du λ-calcul par valeur, la propriété de stabilité sera prouvée (p. 43). Cette preuve est aisément adaptable au λ-calcul étiqueté. Intuitivement, la propriété de stabilité signifie que seuls les sous-termes conservés dans le préfixe minimum $X = \mathcal{P}_S(M)$ jouent un rôle dans l'obtention d'une valeur ; les autres sous-termes ne contribuent pas. On illustre cette intuition



Confluence Monotonie

FIG. 1.7 – *Confluence et monotonie dans le λ-calcul étiqueté*

en reprenant l'exemple de la section précédente. On considère la réduction étiquetée suivante.

$$\begin{aligned} M = ((\lambda x.(\lambda y.x^d)^c)^b((\lambda z.z^h)^g u^i)^f)^a &\to_e ((\lambda x.(\lambda y.x^d)^c)^b u^{f\lceil g\rceil h\lfloor g\rfloor i})^a \\ &\to_e (\lambda y.u^{d\lfloor b\rfloor f\lceil g\rceil h\lfloor g\rfloor i})^{a\lceil b\rceil c} = V \end{aligned}$$

Le plus petit préfixe de M qui se réduit vers une valeur est $X = ((\lambda x.(\lambda y.\Omega)^c)^b\Omega)^a$. On observe que les sous-termes de M conservés dans X sont, dans cet exemple, exactement les sous-termes dont l'étiquette de tête est une lettre de l'étiquette de tête de V. Cette remarque est généralisée et rapprochée de la notion de sous-terme critique dans la suite de ce paragraphe. Cette notion, définie précédemment dans le cadre du λ-calcul, s'adapte de façon directe au λ-calcul étiqueté.

Définition 1.2 (Sous-terme critique) *Si $M \twoheadrightarrow_e V$, alors le sous-terme $(C[\,],N)$ de M est critique si et seulement s'il existe un terme N' tel que le terme $C[N']$ ne se réduit pas vers une valeur.*

La dénomination de sous-terme critique se justifie par le fait que le remplacement d'un tel sous-terme peut, éventuellement, conduire à une réduction qui ne mène plus à une valeur. Pour formaliser et généraliser la remarque issue de l'exemple précédent, selon laquelle le préfixe minimum de M menant à une valeur est décrit par les lettres présentes dans l'étiquette $\tau(V)$, on introduit les opérations $|\alpha|$ et $[\![Y]\!]_A$. L'ensemble $|\alpha|$ est constitué des lettres contenues dans l'étiquette α. L'opération de A-préfixe $[\![Y]\!]_A$ retourne le plus grand préfixe de Y dont toutes les étiquettes sont formées à partir de lettres de $A \subseteq \mathbf{A}$.

$$\begin{aligned} |a| &= \{a\} \\ |\lceil\alpha\rceil| &= |\alpha| \\ |\lfloor\alpha\rfloor| &= |\alpha| \\ |\alpha\beta| &= |\alpha| \cup |\beta| \end{aligned}$$

$$\begin{aligned} [\![\Omega]\!]_A &= \Omega \\ [\![X]\!]_A &= \Omega && \text{si } X \neq \Omega \text{ et } |\tau(X)| \not\subseteq A \\ [\![x^\alpha]\!]_A &= x^\alpha && \text{si } |\alpha| \subseteq A \\ [\![(\lambda x.X)^\alpha]\!]_A &= (\lambda x.[\![X]\!]_A)^\alpha && \text{si } |\alpha| \subseteq A \\ [\![(XY)^\alpha]\!]_A &= ([\![X]\!]_A\,[\![Y]\!]_A)^\alpha && \text{si } |\alpha| \subseteq A \end{aligned}$$

Après la réduction d'un terme, les étiquettes du terme obtenu contiennent les interactions entre les sous-termes qui ont contribué à obtenir ce terme. Dans le dernier exemple, l'obtention de la valeur a bien nécessité la contraction du radical de nom b dont l'étiquette de l'application est a et l'étiquette de tête du corps est c. Dans le cas d'une réduction $M \twoheadrightarrow_e V$ qui aboutit à une valeur, l'étiquette de tête $\tau(V)$ de cette valeur contient les étiquettes des sous-termes de M qui sont intervenus pour obtenir V. Cette intuition est formalisée par le résultat suivant.

Résultat 1.15 *Si M vérifie* `INIT` *et $M \twoheadrightarrow_e V$, alors on a $\mathcal{P}_S(M) = [\![M]\!]_{|\tau(V)|}$.*

Si un terme M dont les étiquettes sont des lettres distinctes se réduit vers une valeur V, le préfixe de stabilité de M est le $|\tau(V)|$-préfixe de M. Ce préfixe est le préfixe minimal de M qui se réduit vers une valeur. Ce résultat montre que les étiquettes du λ-calcul permettent d'exprimer la stabilité. On note cependant que seules les lettres de l'étiquette de tête sont exploitées. Les soulignements et surlignements qui hiérarchisent cette étiquette ne sont pas utilisés à ce stade. Ces informations

qui décrivent l'ordre de création des radicaux seront utilisées dans le chapitre 6 pour exprimer la propriété d'indépendance (p. 148). La connexion entre les étiquettes et la propriété de stabilité établie par le résultat 1.15 peut être vue sous un angle plus local en considérant les sous-termes critiques.

Résultat 1.16 *Si* $\texttt{INIT}(M)$ *et* $M \twoheadrightarrow_e V$*, le sous-terme* $(C[\,],N)$ *est critique si et seulement si la lettre* $\tau(N)$ *appartient à l'ensemble* $|\tau(V)|$*.*

Si les étiquettes de M sont des lettres distinctes et si M se réduit vers V, les sous-termes critiques de M sont ceux dont l'étiquette est une lettre de $\tau(V)$. En d'autres mots, l'étiquette de tête de la valeur permet de déterminer l'ensemble des sous-termes critiques. Et, en utilisant le résultat 1.15, on obtient que cet ensemble de sous-termes correspond aux sous-termes de M qui sont conservés dans le préfixe minimal $[\![M]\!]_{|\tau(V)|}$ de M qui se réduit vers une valeur.

Les étiquettes du λ-calcul permettent d'identifier les parties d'un terme qui contribuent à l'obtention d'une valeur. Cette propriété permet de faire le lien avec la stabilité du λ-calcul. Dans les chapitres suivants, nous montrerons que l'information portée par les étiquettes ne se réduit pas à l'identification des sous-termes qui interviennent au cours d'une réduction. Les étiquettes apportent également une description précise des interactions entre ces sous-termes : nous dirons que les étiquettes contiennent l'histoire d'une réduction.

1.3 Irréversibilité des contextes

Nous prouvons dans cette section les théorèmes d'irréversibilité des contextes et des chemins. Ces résultats montrent qu'au contraire du λ-calcul, aucune *coïncidence syntaxique* ne peut intervenir dans le λ-calcul étiqueté. Nous utiliserons, en particulier, ces propriétés lorsque nous étendrons le λ-calcul avec les références.

Un avantage appréciable du λ-calcul étiqueté est qu'il permet d'éviter les *coïncidences syntaxiques*. Ainsi, dans le λ-calcul, le terme $M = I(Ix)$ a deux radicaux : $R_1 = M$ et $R_2 = Ix$. En contractant l'un ou l'autre radical, on obtient dans les deux cas le terme $N = Ix$. Pourtant, ces deux réductions sont bien différentes. L'obtention du même terme n'est qu'accidentelle. Un symptôme de cette coïncidence syntaxique est le fait que la notion de résidu dans N n'est pas identique pour les deux façons d'obtenir N : si on a contracté R_1 (respectivement R_2), le radical N est le résidu de R_2 (resp. R_1). Le λ-calcul étiqueté permet de s'affranchir de ces coïncidences. En posant $M' = ((\lambda y.y^c)^b((\lambda y.y^g)^f x^h)^d)^a$, on obtient, en contractant le radical externe, $N_1 = ((\lambda y.y^g)^f x^h)^{a\lceil b\rceil c\lfloor b\rfloor d}$ et, en contractant le radical interne, $N_2 = ((\lambda y.y^c)^b x^{d\lceil f\rceil g\lfloor f\rfloor h})^a$. Cette réduction est le premier exemple illustré sur la figure 1.8.

Les coïncidences syntaxiques se manifestent d'une autre façon : le calcul est réversible. En effet, comme le montre l'exemple du terme $\Delta\Delta \to \Delta\Delta$, il est possible qu'une réduction aboutisse plusieurs fois sur le même terme. Cette réversibilité du calcul se manifeste en fait par une propriété plus générale. Dans le λ-calcul, $M \twoheadrightarrow M'$ implique bien $C[M] \twoheadrightarrow C[M']$. En revanche, la réduction $C[M] \twoheadrightarrow C[M']$ n'implique pas $M \twoheadrightarrow M'$. Ainsi, entre $C[M]$ et $C[M']$, le contexte $C[\,]$ peut disparaître car le contexte est impliqué dans une réduction. Puis ce contexte peut être recréé à l'identique par une autre réduction. Dans le λ-calcul, il n'est pas possible de distinguer le contexte original du contexte recréé. Un exemple de ce fait est donné par le contexte $C[\,] = [\,]\,x$ et le terme $M = (\lambda y.xx)x = C[\lambda y.xx]$. On a $M \to xx = C[x]$ mais on n'a pas $\lambda y.xx \to x$. Le contexte initial et le contexte final sont syntaxiquement égaux, mais cette égalité est *accidentelle*. Le λ-calcul étiqueté permet, là aussi, de s'affranchir de ces accidents. Dans l'exemple précédent, en posant $M' = ((\lambda y.(x^d x^f)^c)^b x^g)^a$ et $C'[\,] = ([\,]\,x^f)^a$, on obtient $M' \to_e (x^d x^f)^{a\lceil b\rceil c} = C'_e[x^d]$ avec

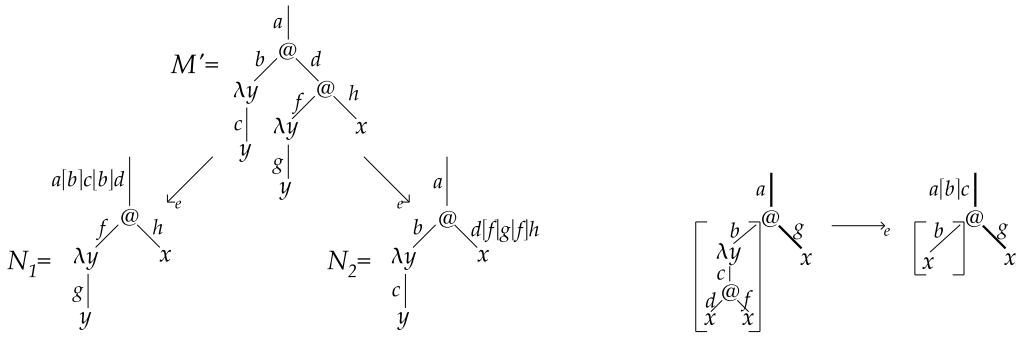


FIG. 1.8 – *Les étiquettes du λ-calcul éliminent les coïncidences syntaxiques.*

$C''[\,] = ([\,]\, x^f)^{a\lceil b\rceil c}$. Le contexte final ne coïncide plus avec le contexte initial, comme le montre le deuxième exemple de la figure 1.8.

Dans cette section, on prouve la propriété suivante d'irréversibilité, dans le λ-calcul étiqueté.

Théorème 1.1 (Irréversibilité des contextes) *Soit $C[\,]$ un contexte. On a $M \twoheadrightarrow_e M'$ si et seulement si $C[M] \twoheadrightarrow_e C[M']$.*

Bien sûr, le sens direct est vrai, aussi bien dans le λ-calcul que dans le λ-calcul étiqueté. La réciproque n'est vraie que dans le λ-calcul étiqueté. Cette propriété affirme que si un terme peut se réduire vers un autre terme et que ces deux termes admettent le même contexte $C[\,]$, alors la réduction du sous-terme placé sous le contexte $C[\,]$ est séparable du contexte. Le théorème d'irréversibilité des contextes est un corollaire immédiat de la propriété plus générale suivante.

Lemme 1.1 *Soit $C[\,]$ un contexte. Si on a $\mathcal{R} : C[M] \twoheadrightarrow_e C[M']$, alors cette réduction s'écrit $\mathcal{R} : C[M] \to_e C[M_1] \to_e \ldots \to_e C[M_n] \to_e C[M']$ avec $M \to_e M_1 \to_e \ldots \to_e M_n \to_e M'$.*

Ce lemme est une version renforcée du théorème d'irréversibilité des contextes. Si un terme $C[M]$ se réduit vers un autre terme $C[M']$, non seulement la réduction de M à M' peut être séparée du contexte $C[\,]$, mais en plus, la réduction de $C[M]$ à $C[M']$ laisse le contexte inchangé et ne contracte que des radicaux placés sous ce contexte. On peut interprété cette propriété d'une autre façon : si le contexte $C[\,]$ disparaît à une étape de la réduction (du fait d'une contraction à l'intérieur de $C[\,]$ ou à la limite du vide de $C[\,]$), ce contexte ne peut plus réapparaître. Cette interprétation justifie la dénomination d'*irréversibilité des contextes*.

Pour montrer le lemme 1.1, on prouve un résultat plus général qui porte sur des contextes ayant un nombre fini de vides. On introduit ci-dessous la syntaxe des contextes à plusieurs vides où I est un sous-ensemble fini d'entiers.

Si $I = \emptyset$	$C\overrightarrow{[]}_i^I$	$:= M$
Si $I = \{i_0\}$	$C\overrightarrow{[]}_i^I$	$:= []_{i_0} \mid (C\overrightarrow{[]}_i^I\ M)^\alpha \mid (M\ C\overrightarrow{[]}_i^I)^\alpha \mid (\lambda x.C\overrightarrow{[]}_i^I)^\alpha$
Sinon	$C\overrightarrow{[]}_i^I$	$:= (\lambda x.C\overrightarrow{[]}_i^I)^\alpha \mid (C\overrightarrow{[]}_i^{I_1}\ C\overrightarrow{[]}_i^{I_2})^\alpha$ où I_1 et I_2 forment une partition de I

Si I est vide, alors $C\overrightarrow{[]}_i^I$ ne contient aucun vide. $C\overrightarrow{[]}_i^I$ est donc un terme M. Si I est un singleton $\{i_0\}$, $C\overrightarrow{[]}_i^I$ contient un unique vide. Il s'agit d'un contexte classique dans lequel le vide a été indicé par i_0. Si I est un singleton, on utilisera la notation plus légère suivante : $C\overrightarrow{[]}_i^I = C[]_{i_0}$. Si I contient au moins deux éléments, $C\overrightarrow{[]}_i^I$ peut être une abstraction $(\lambda x.C'\overrightarrow{[]}_i^I)^\alpha$. Donc le corps est aussi un contexte qui contient les vides indicés par I. Ou bien, $C\overrightarrow{[]}_i^I$ est une application $(C_1\overrightarrow{[]}_i^{I_1}\ C_2\overrightarrow{[]}_i^{I_2})^\alpha$

dont les vides se répartissent entre membre gauche et membre droit suivant la partition de I_1 et I_2. Le membre gauche (respectivement droit) est un contexte contenant les vides indicés par I_1 (resp. I_2). Si $I = \{i_1, i_2, \ldots, i_n\}$, le contexte $C\overrightarrow{[]}_i^I$ pourra être noté $C[]_{i_1}[]_{i_2}\ldots[]_{i_n}$ ou $C\overrightarrow{[]}_i^J\overrightarrow{[]}_i^K$ si J et K forment une partition de I.

Le théorème d'irréversibilité des contextes repose essentiellement sur les propriétés fondamentales suivantes portant sur les étiquettes du λ-calcul étiqueté.

Lemme 1.2 *1. Si $R \xrightarrow{R}_e R'$, alors $\tau(R) \prec \tau(R')$.*
2. Si $M \to_e M'$, alors $\tau(M) \preceq \tau(M')$.

Preuve : Pour le premier point, on a nécessairement $R = ((\lambda x.M)^\alpha N)^\beta$. De là, on obtient $R' = \beta \cdot \lceil\alpha\rceil \cdot M\{x \backslash \lfloor\alpha\rfloor \cdot N\}$ et donc $\tau(R) = \beta \prec \beta \cdot \lceil\alpha\rceil \cdot \tau(M) = \tau(R')$. Pour le deuxième point, deux cas sont à considérer. Si M est le radical contracté, le point précédent permet de conclure. Sinon, on a $\tau(M) = \tau(M')$. □

Ce lemme élémentaire indique que l'étiquette de tête d'un terme réduit contient l'étiquette de tête du terme initial. Et cette relation est stricte si le terme considéré est le radical contracté. Cette propriété implique notamment que si une réduction survient à l'intérieur ou à la limite d'un contexte, le contexte initial contient l'étiquette de tête du radical qui est strictement différente de l'étiquette de tête du contractum. De ce fait, le contexte est modifié par la réduction : il a disparu. Cette remarque intuitive est exploitée dans la preuve du lemme suivant qui constitue le résultat de base de cette partie. Ce lemme est une version du théorème d'irréversibilité des contextes généralisée aux contextes à plusieurs vides.

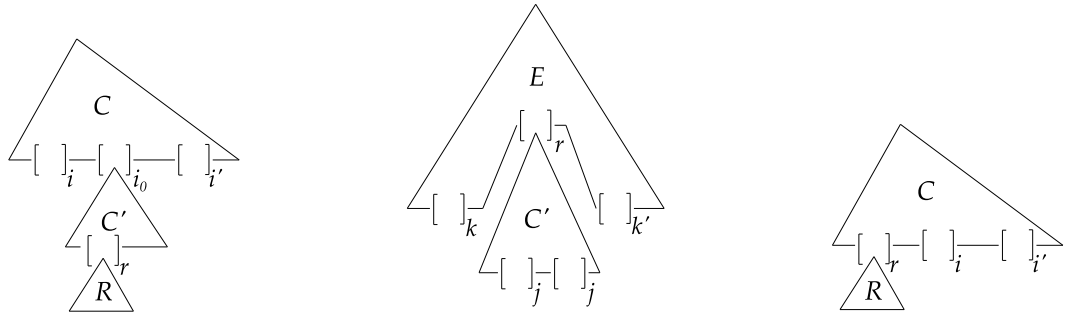
Lemme 1.3 (Irréversibilité) *Si $C\overrightarrow{[M_i]}_i^I \to_e^\star C\overrightarrow{[M'_i]}_i^I$, alors cette réduction peut se décomposer en $C\overrightarrow{[M_i]}_i^I \to_e C\overrightarrow{[M_i^1]}_i^I \to_e \cdots \to_e C\overrightarrow{[M_i^{n-1}]}_i^I \to_e C\overrightarrow{[M'_i]}_i^I$ où, pour tout $i \in I$, on a la réduction $M_i \twoheadrightarrow_e M_i^1 \twoheadrightarrow_e \cdots \twoheadrightarrow_e M_i^{n-1} \twoheadrightarrow_e M'_i$.*

Preuve : On procède par récurrence sur la longueur n de la réduction.

Base ($n = 0$) Immédiat

Récurrence ($n > 0$) On considère le premier pas de réduction : $M = C\overrightarrow{[M_i]}_i^I \to_e N \to_e^{n-1} M'$. Soit $D[]_r$ (avec $r \notin I$) le contexte du radical R qui est réduit au cours de cette réduction. On a : $C\overrightarrow{[M_i]}_i^k = D[R]_r$ et $N = D[R']_r$ où $R \to_e R'$. On examine la position de $[]_r$ par rapport aux vides de $C\overrightarrow{[]}_i^I$. Les trois cas sont illustrés sur la figure 1.9.

1. Si $[]_r$ est inclus dans $[]_{i_0}$ pour $i_0 \in I$.
   On pose $I_0 = I - \{i_0\}$. Il existe un contexte $C'[]_r$ tel que $D[]_r = C\overrightarrow{[M_i]}_i^{I_0}[C'[]_r]_{i_0}$. On peut donc écrire $N = C\overrightarrow{[M_i]}_i^{I_0}[C'[R']_r]_{i_0}$, avec $M_{i_0} = C'[R]_r \to_e C'[R']_r = M_{i_0}^1$. Comme $C\overrightarrow{[]}_i^I$ est un contexte de $N = C\overrightarrow{[M_i]}_i^{I_0}[C'[R']_r]_{i_0}$, on peut conclure par récurrence.
2. Si les $[]_j$ sont strictement inclus dans $[]_r$ pour $j \in J \subseteq I$.
   On pose $K = I - J$. Il existe un contexte $C'\overrightarrow{[]}_i^J$, non réduit à $[]_{j_0}$, qui vérifie les relations $D[C'\overrightarrow{[]}_i^J]_r = C\overrightarrow{[]}_i^J\overrightarrow{[M_i]}_i^K$ et $C'\overrightarrow{[M_i]}_i^J = R$. On considère le sous-contexte intersection de $D[]_r$ et $C\overrightarrow{[M_j]}_j^J\overrightarrow{[]}_i^K$ que l'on peut écrire $E[]_r\overrightarrow{[]}_i^K$. On a, par définition, les égalités suivantes :
   $$C\overrightarrow{[M_i]}_i^J\overrightarrow{[]}_i^K = E[R]_r\overrightarrow{[]}_i^K \qquad D[]_r = E[]_r\overrightarrow{[M_i]}_i^K \qquad E[C'\overrightarrow{[]}_i^J]_r\overrightarrow{[]}_i^K = C\overrightarrow{[]}_i^I$$
   On en déduit $N = D[R']_r = E[R']_r\overrightarrow{[M_i]}_i^K$ et $M' = C\overrightarrow{[M'_i]}_i^I = E[C'\overrightarrow{[M'_i]}_i^J]_r\overrightarrow{[M'_i]}_i^K$. On utilise l'hypothèse de récurrence sur le contexte $E[]_r\overrightarrow{[]}_i^K$ et la réduction de N à



Cas 1 Cas 2 Cas 3

FIG. 1.9 – *Les trois cas de figures du lemme 1.3*

M'. On obtient $R \to_e R' \to_e^\star C'\overrightarrow{[M'_i]_i}^J$. Comme $C'\overrightarrow{[]_i}^J$ n'est pas réduit à $[]_{j_0}$ et que $C'\overrightarrow{[M_i]_i}^J = R$, on a $\tau(C'\overrightarrow{[]_i}^J) = \tau(R)$. En utilisant le lemme 1.2, on obtient d'une part $\tau(R) \prec \tau(R')$, et d'autre part $\tau(R') \preceq \tau(C'\overrightarrow{[]_i}^J) = \tau(R)$, ce qui aboutit à une contradiction. Ce cas n'est donc pas possible.

3. Si les vides $[]_r$ et $[]_i$ pour $i \in I$ sont disjoints.
   On considère le contexte $E[]_r\overrightarrow{[]_i}^I$ qui est l'intersection de $D[]_r$ et de $C\overrightarrow{[]_i}^I$. Ce contexte vérifie :
   (a) $D[]_r = E[]_r\overrightarrow{[M_i]_i}^I$
   (b) $C\overrightarrow{[]_i}^I = E[R]_r\overrightarrow{[]_i}^I$
   (c) $N = E[R']_r\overrightarrow{[M_i]_i}^I$
   (d) $M' = C\overrightarrow{[M'_i]_i}^I = E[R]_r\overrightarrow{[M'_i]_i}^I$
   On applique l'hypothèse de récurrence à $N = E[R']_r\overrightarrow{[M_i]_i}^I \to_e^{n-1} E[R]_r\overrightarrow{[M'_i]_i}^I = M'$. On obtient : $R' \to_e^\star R$ ce qui contredit le lemme 1.2. Ce cas n'est donc pas possible. □

Si un terme $C\overrightarrow{[M_i]_i}^I$ se réduit vers un autre terme $C\overrightarrow{[M'_i]_i}^I$, alors les réductions de M_i à M'_i (pour $i \in I$) peuvent être séparées du contexte $C\overrightarrow{[]_i}^I$ et la réduction de $C\overrightarrow{[M_i]_i}^I$ à $C\overrightarrow{[M'_i]_i}^I$ ne contracte aucun radical interne au contexte $C\overrightarrow{[]_i}^I$. Contrairement au λ-calcul, ce résultat implique qu'un contexte ne peut disparaître puis réapparaître à l'identique. On déduit de ce résultat le lemme 1.1 ce qui prouve le théorème d'irréversibilité des contextes.

On s'intéresse dans la suite de cette partie à un cas particulier particulièrement utile du lemme 1.3. En effet un chemin issu de la racine du terme peut être vu comme un contexte. Dans le terme $M = (\lambda x.(((\lambda x.x^e)^d y^f)^c y^g)^b)^a$, le chemin menant au radical $R = (((\lambda x.x^e)^d y^f)^c y^g)^b$ peut être vu comme le contexte $C\overrightarrow{[]_i}^{\{1,2\}} = (\lambda x.([]_1[]_2)^b)^a$. On définit formellement la syntaxe des nœuds et des chemins de la façon suivante.

Nœud	$\theta \in \mathbf{N} ::= \lambda \mid @_i$	$i \in \{1,2\}$
Chemin-contexte	$\kappa \in \mathbf{K} ::= \alpha_1\theta_1\alpha_2\theta_2 \ldots \alpha_n\theta_n$	$n \in \mathbb{N}$
Chemin	$\varphi \in \mathbf{\Phi} ::= \alpha\theta_1\alpha_1\theta_2\alpha_2 \ldots \theta_n\alpha_n$	$n \in \mathbb{N}$

Intuitivement, un chemin relie la racine de l'arbre de syntaxe d'un terme à un nœud de cet arbre. Syntaxiquement, un chemin est une suite alternée d'étiquettes et de nœuds qui commence par une étiquette et qui peut être éventuellement vide. Cette suite correspond à la succession d'étiquettes et de nœuds rencontrés entre la racine et le nœud final. Dans un chemin, un nœud peut être un λ, pour les chemins traversant une abstraction, un $@_1$ si le chemin traverse une application

vers le membre gauche de l'application, ou un $@_2$ si le chemin traverse une application vers le membre droit. Deux types de chemin sont utilisés. Un chemin-contexte peut être soit un chemin vide (que l'on note $\bot$), soit une suite alternée qui finit sur un nœud. Un chemin est une suite alternée qui finit sur une étiquette. On utilise librement la concaténation pour simplifier les notations : si κ, κ', sont les chemins-contextes $\alpha_1\theta_1 \dots \alpha_i\theta_i$, et $\beta_1\theta'_1 \dots \beta_j\theta'_j$, alors $\kappa\kappa'$ est le chemin-contexte $\alpha_1\theta_1 \dots \alpha_i\theta_i\beta_1\theta'_1 \dots \beta_j\theta'_j$. Si φ est le chemin $\gamma_1\theta''_1 \dots \gamma_k\theta''_k$, alors $\kappa\varphi$ est le chemin $\alpha_1\theta_1 \dots \alpha_i\theta_i\gamma_1\theta''_1 \dots \gamma_k\theta''_k$. De là, on définit naturellement une notion de préfixe entre chemins et chemins-contextes.

$$\begin{array}{l}
\kappa \preceq \kappa' \iff \exists \kappa'' \in \mathbf{K} \,.\, \kappa\kappa'' = \kappa' \\
\kappa \prec \varphi \iff \exists \varphi' \in \mathbf{\Phi} \,.\, \kappa\varphi' = \varphi \\
\varphi \prec \kappa \iff \exists (\kappa',\theta) \in \mathbf{K} \times \mathbf{N} \,.\, \varphi\theta\kappa' = \kappa \\
\varphi \preceq \varphi' \iff \varphi = \varphi' \text{ ou } \exists (\varphi'',\theta) \in \mathbf{\Phi} \times \mathbf{N} \,.\, \varphi\theta\varphi'' = \varphi'
\end{array}$$

La relation $\preceq$ est la relation de préfixe sur les suites finies alternées d'étiquettes et de nœuds. Cette relation est un ordre bien fondé sur $\mathbf{K} \cup \mathbf{\Phi}$ dont le plus petit élément est $\bot$.

La notion de chemin-contexte est étroitement liée à la notion de contexte. Un chemin-contexte correspond au chemin d'un contexte $C[\,]$ menant de la racine au vide du contexte. Cette association est formalisée par la fonction σ suivante.

$$\begin{array}{rlrl}
\sigma([\,]) &= \bot & \sigma((C[\,]N)^\alpha) &= \alpha@_1\sigma(C[\,]) \\
\sigma((\lambda x.C[\,])^\alpha) &= \alpha\lambda\sigma(C[\,]) & \sigma((MC[\,])^\alpha) &= \alpha@_2\sigma(C[\,])
\end{array}$$

Le chemin-contexte associé au contexte $[\,]$ est le chemin vide $\bot$. Le chemin-contexte associé à $C[\,]$ est la séquence des étiquettes et des nœuds rencontrés en parcourant l'arbre syntaxique correspondant au contexte depuis la racine de $C[\,]$ vers son vide $[\,]$. Par contraste, un chemin relie la racine à l'étiquette d'un nœud sans contenir le nœud ; il finit donc par une étiquette. Si on reprend l'exemple employé plus haut, $\kappa = a\lambda b@_1$ est le chemin-contexte associé au contexte $C_1[\,] = (\lambda x.([\,]\, y^g)^c)^b)^a$, c'est-à-dire $\sigma(C_1[\,]) = \kappa$. Et le chemin φ menant au radical R est $\varphi = \kappa c = a\lambda b@_1 c$. De façon générale, si $(C[\,],R)$ est un radical, on appellera *chemin menant au radical* le chemin $\sigma(C[\,])\tau(R)$. On peut généraliser la fonction σ pour pouvoir obtenir le chemin-contexte menant à un vide d'un contexte à plusieurs trous. Dans ce cas, on mentionne l'indice du vide dont on veut le chemin-contexte. Si $i_0 \in I$, $\sigma(i_0,C\overrightarrow{[]}^I_i)$ est le chemin-contexte menant au vide indicé par i_0. Cette fonction σ se définit formellement de la façon suivante.

$$\begin{array}{rlll}
\sigma(i_0,[]_{i_0}) &= \bot & & \\
\sigma(i_0,(\lambda x.C\overrightarrow{[]}^I_i)^\alpha) &= \alpha\lambda\sigma(C\overrightarrow{[]}^I_i) & & \\
\sigma(i_0,(C_1\overrightarrow{[]}^{I_1}_i\ C_2\overrightarrow{[]}^{I_2}_i)^\alpha) &= \alpha@_1\sigma(i_0,C_1\overrightarrow{[]}^{I_1}_i) & & \text{si } i_0 \in I_1 \\
\sigma(i_0,(C_1\overrightarrow{[]}^{I_1}_i\ C_2\overrightarrow{[]}^{I_2}_i)^\alpha) &= \alpha@_2\sigma(i_0,C_2\overrightarrow{[]}^{I_2}_i) & & \text{si } i_0 \in I_2
\end{array}$$

Comme $i_0 \in I$ et comme I_1 et I_2 forment une partition de I, la définition ci-dessus est bien correcte. Le chemin $\sigma(i_0,C\overrightarrow{[]}^I_i)$ est la suite d'étiquettes et de nœuds rencontrés en parcourant l'arbre de syntaxe correspondant à $\overrightarrow{[]}^I_i$ de la racine vers le vide $[]_{i_0}$.

Si on peut associer un chemin-contexte à un contexte, l'inverse est aussi vrai. Si M est un terme et κ est un chemin contexte de M, alors on peut associer à κ le contexte de M qui est réduit au chemin-contexte κ. Si $M = (\lambda x.(((\lambda x.x^e)^d y^f)^c y^g)^b)^a$, le contexte associé au chemin-contexte $\kappa = a\lambda b@_1 c@_1$ est $C\overrightarrow{[]}^{\{0,1,2\}}_i = (\lambda x.(([]_0[]_1)^c[]_2)^b)^a$ où $\sigma(0,C\overrightarrow{[]}^{\{0,1,2\}}_i) = \kappa$. Plus généralement, on associe un contexte et un indice à un chemin-contexte κ de M en définissant la fonction $\zeta : \mathbf{\Lambda}_e \times \mathbf{K} \to \mathbf{\Lambda}_e[]_i$.

– La fonction ζ est définie à l'aide de $\zeta' : \mathbf{\Lambda}_e \times \mathbf{K} \times 2^{\mathbb{N}} \to \mathbf{\Lambda}_e[]_i$. On a $\zeta(M,\kappa) = \zeta'(M,\kappa,\emptyset)$.

- Si $\kappa = \bot$, alors $\zeta'(M,\kappa,I) = [\,]_0$.
- Si $M = (\lambda x.M_1)^\alpha$ et $\kappa = \alpha\lambda\kappa'$, alors $\zeta'(M,\kappa,I) = (\lambda x.\zeta'(M_1,\kappa',I))^\alpha$.
- Si $M = (M_1M_2)^\alpha$ et $\kappa = \alpha@_1\kappa'$, et alors $\zeta'(M,\kappa,I) = (\zeta'(M_1,\kappa',I \cup \{i_0\})\ [\,]_{i_0})^\alpha$ avec $i_0 \notin I \cup \{0\}$.
- Si $M = (M_1M_2)^\alpha$ et $\kappa = \alpha@_2\kappa'$, alors $\zeta'(M,\kappa,I) = ([\,]_{i_0}\,\zeta'(M_2,\kappa',I\cup\{i_0\}))^\alpha$ avec $i_0 \notin I\cup\{0\}$.

On construit un contexte de M qui ne contient que les nœuds mentionnés dans le chemin-contexte κ. Les sous-termes que κ ne traverse pas sont remplacés par des vides. Par construction, on a $\sigma(0,\zeta(M,\kappa)) = \kappa$. Cette définition nous permet d'obtenir le théorème d'irréversibilité des chemins.

Théorème 1.2 (Irréversibilité des chemins) *Si $M_0 \to_e M_1 \to_e \ldots \to_e M_n$ et si φ_i, pour $i \in \{1 \ldots n\}$, est le chemin menant au radical contracté entre M_{i-1} et M_i, alors si $i \leq j \leq n$, φ_i n'est pas un chemin de M_j.*

Preuve : Supposons par l'absurde qu'il existe un indice $1 \leq i < j \leq n$ tels que φ_i est un chemin de M_j. Il existe un chemin-contexte κ tel que $\varphi_i = \kappa\alpha$. Comme κ est un chemin-contexte de M_{i-1} et de M_j, alors il existe un contexte $C\overrightarrow{[]}^I_i$ tel que $C\overrightarrow{[]}^I_i = \zeta(M_j,\kappa) = \zeta(M_{i-1},\kappa)$. De là, il existe deux suites de termes $\{N_i\}_{i\in I}$ et $\{N'_i\}_{i\in I}$ telles que N_0 est le radical contracté entre M_{i-1} et M_i et $M_{i-1} = C\overrightarrow{[N_i]}^I_i$ et $M_j = C\overrightarrow{[N'_i]}^I_i$. En utilisant le lemme 1.3, on obtient en particulier $N_0 \twoheadrightarrow_e N'_0$ avec $\varphi_i = \kappa\tau(N_0)$ et $\varphi_i = \kappa\tau(N'_0)$. En utilisant le lemme 1.2, on obtient $\tau(N_0) \prec \tau(N'_0)$ ce qui aboutit à une contradiction. □

Si on contracte un radical dont le chemin associé est φ, ce chemin (ou un chemin préfixé par ce chemin) ne peut plus réapparaître dans les termes dans la suite de la réduction, d'où l'appellation *irréversibilité des chemins*. Cette irréversibilité suggère une analogie temporelle entre chemins et dates. En effet, de même que la même date ne peut survenir deux fois, le même chemin menant au radical contracté ne peut se répéter. Cette analogie sera utilisée plus tard pour comparer les réductions de deux termes qui acceptent le même contexte. Les chemins menant aux radicaux communs aux deux réductions permettront de *synchroniser* ces réductions, ce qui facilitera la comparaison. La propriété d'irréversibilité des chemins sera aussi utilisée au moment d'ajouter les références au langage. Les chemins menant aux radicaux permettront de donner un nom aux locations créées : le théorème 1.2 assure que le même nom ne peut être donné deux fois.

Chapitre 2

λ-calcul par valeur et étiquettes

Les langages fonctionnels tels que Lisp, Objective Caml, SML/NJ ou Haskell [20, 28, 40, 34] sont fondés sur le λ-calcul. Si plusieurs radicaux peuvent coexister dans un terme du λ-calcul, au moment d'évaluer ce terme, par exemple dans un interpréteur, on peut se demander quel radical réduire en premier. En présence d'effets de bord, l'ordre d'évaluation est critique puisque le résultat obtenu peut dépendre de cet ordre. Plus simplement, dans le λ-calcul pur, si l'abstraction d'un radical $R = (\lambda x.M)N$ ne contient pas d'occurrences de x, réduire le terme N peut sembler inutile, puisque ce terme disparaîtra de toute façon quand R sera contracté. La réduction de N pourrait même boucler alors que R est normalisable. En revanche, si le corps M de l'abstraction contient plusieurs occurrences de x, contracter R dupliquerait les réductions à faire dans N. Faire ces réductions avant la contraction de R revient à les factoriser. Dans la mesure où on s'attend à ce que l'argument d'une fonction soit utilisé, la stratégie d'évaluation adoptée par Objective Caml et SML/NJ est en appel par valeur. Les arguments des applications sont réduits jusqu'à obtenir des valeurs. Puis l'application est réduite. Au moment de l'étude de la propriété de non-interférence, exposée dans le chapitre 5, nous avons souhaité étudier cette propriété au sein d'un λ-calcul étendu par des traits impératifs permettant de manipuler la mémoire. La présence d'effets de bord nous a poussé à définir un ordre d'évaluation. Nous avons choisi, comme pour Objective Caml et SML/NJ, une évaluation en appel par valeur, c'est-à-dire une évaluation dans laquelle les arguments des β-radicaux sont des valeurs. Ce choix a motivé l'étude dans le λ-calcul de ces réductions particulières où les radicaux contractés sont de la forme $(\lambda x.M)V$.

Dans la section 2.1, nous introduisons les règles de la réduction par valeur $\rightarrow_v$. Ces règles s'appliquent aux termes du λ-calcul. Par abus, nous appellerons *λ-calcul par valeur* le langage constitué des termes du λ-calcul et muni de la réduction par valeur. Nous montrons que la réduction par valeur vérifie les mêmes propriétés essentielles que la réduction $\rightarrow$ classique. On obtient donc une propriété de confluence. On montre également que le λ-calcul par valeur vérifie le théorème des développements finis. En utilisant une définition de la réduction standard spécifique au λ-calcul par valeur (différente de celle du λ-calcul), on prouve que le théorème de standardisation est vérifié. Les propriétés de monotonie et de stabilité du λ-calcul sont également conservées.

On constate dans la section 2.2 que les étiquettes du λ-calcul ne vérifient plus, en utilisant la réduction par valeur, la propriété de stabilité énoncée dans le résultat 1.15. On introduit de nouvelles étiquettes spécifiques au λ-calcul par valeur. Ce calcul étiqueté est confluent. En utilisant une démonstration intuitive, fondée sur une notion d'*imbrication des radicaux* qui tient compte des imbrications futures, on montre le théorème des développements finis. Le λ-calcul par valeur étiqueté conserve la propriété de standardisation. Et les nouvelles étiquettes permettent d'obtenir la propriété de stabilité correspondant au résultat 1.15.

2.1 Le λ-calcul par valeur

Par abus, on appelle λ-calcul par valeur, le calcul dont les termes sont les termes du λ-calcul et dont les règles de réduction sont définies ci-dessous.

$$(\beta_v) \qquad (\lambda x.X)V \to_v X\{x\backslash V\}$$

$$(\nu_v)\ \frac{X \to_v X'}{XY \to_v X'Y} \qquad (\mu_v)\ \frac{Y \to_v Y'}{XY \to_v XY'} \qquad (\xi_v)\ \frac{X \to_v X'}{\lambda x.X \to_v \lambda x.X'}$$

La β_v-réduction est une β-réduction où l'argument du β-radical est une valeur. Dans le cadre présent où les valeurs sont les abstractions, il peut sembler inutile d'utiliser cette notion. Cependant, cette dernière présente un avantage en terme de modularité si de nouveaux termes tels que les entiers sont ajoutés à la syntaxe du langage. Ces entiers, considérés comme des valeurs, ne nécessiteraient pas de changement des règles de réduction. Les règles de contexte sont simplement adaptées. Par conséquent, si les réductions sont vues comme des relations, on a $\to_v \subseteq \to$. Autrement énoncé, si $X \to_v Y$, alors on a $X \to Y$. La notion de radical est modifiée en conséquence : les β_v-radicaux sont les sous-termes de la forme $(\lambda x.X)V$. De même que dans le λ-calcul, on a $X \to_v X'$ si et seulement s'il existe un contexte C tel que $X = C[(\lambda x.Y)V]$ et $X' = C[Y\{x\backslash V\}]$. Pour définir la notion de β_v-résidu, on s'appuie sur la définition de β-résidu. On commence par observer que la notion de β_v-radical vérifie la propriété fondamentale suivante.

Lemme 2.1 *Si R est un β_v-radical et N est un terme, alors $R\{x\backslash N\}$ est un β_v-radical.*

Preuve : Ce résultat tient au fait que l'ensemble des valeurs est stable par substitution. □

Ce lemme énonce le fait que l'ensemble des β_v-radicaux est stable par substitution. A fortiori, cet ensemble est stable par substitution par une valeur. Cela permet d'obtenir le résultat suivant.

Lemme 2.2 *Si $X \to X'$, les β-résidus dans X' des β_v-radicaux de X sont des β_v-radicaux.*

Preuve : Elémentaire, par inspection des cas. □

Les β-résidus d'un β_v-radical sont des β_v-radicaux. Cette propriété est vraie pour une réduction $\to$ et, à plus forte raison, pour une réduction $\to_v$. On peut donc définir la notion de β_v**-résidu** de la façon suivante, en utilisant la notion de β-résidu : si $X \to_v X'$, un β_v-radical r' de X' est β_v-résidu d'un β_v-radical r de X si et seulement si r' est β-résidu de r. On illustre cette définition avec la réduction $R = (\lambda x.I\lambda z.x)(yy) \to I\lambda z.yy$. Après la contraction de R, le β-résidu du β_v-radical $R' = I\lambda z.x$ est le β_v-radical $I\lambda z.yy$: ce dernier est le β_v-résidu de R'. On examine un deuxième exemple qui met en lumière une différence fondamentale entre le λ-calcul et le λ-calcul par valeur. Le terme $R = I_1(I_2\ I_3)$ (où $I_1 = I_2 = I_3 = \lambda x.x$) contient deux β-radicaux : R et I_2I_3. Seul ce dernier est un β_v-radical. Le terme R se réduit par la réduction $R \to_v I_1I_3 = R'$. Alors que R' est le β-résidu du β-radical R, le β_v-radical R' n'est pas un β_v-résidu d'un β_v-radical de R : le β_v-radical R' est donc créé par la contraction de R. La réduction en appel par valeur entraîne donc des changements dans la relation de création entre les radicaux par rapport au λ-calcul : un nouveau cas de création apparaît. En plus de la création "par le haut" et "par le bas", un β_v-radical peut être créé "par la droite" lorsque l'argument d'une application devient une valeur. Ceci constitue la particularité fondamentale du λ-calcul par valeur. Nous verrons que cette dernière a un impact dans cette section sur la définition de réduction standard et, dans la section suivante, sur la propriété de stabilité. Dans la suite de cette section on montre que le λ-calcul par valeur est localement confluent et confluent. On montre aussi que le théorème des développements finis est conservé et que, pour une définition spécifique de réduction standard, le théorème de standardisation est vérifié. Les propriétés de monotonie et de stabilité sont également conservées.

2.1.1 Confluence

Les propriétés de confluence locale et de confluence du λ-calcul par valeur s'obtiennent en adaptant les preuves correspondantes dans le λ-calcul. On commence par examiner une propriété classique de permutation des substitutions, qui ne fait pas intervenir la réduction par valeur.

Lemme 2.3 *Si $x \neq y$ et si $x \notin \mathtt{FV}(Y)$, alors on a $Z\{x\backslash X\}\{y\backslash Y\} = Z\{y\backslash Y\}\{x\backslash X\{y\backslash Y\}\}$.*

Preuve : On pose $Z_1 = Z\{x\backslash X\}\{y\backslash Y\}$ et $Z_2 = Z\{y\backslash Y\}\{x\backslash X\{y\backslash Y\}\}$. On procède par récurrence sur la taille de Z.

1. Si $Z = x$, on a $Z_1 = X\{y\backslash Y\}$ et, comme $x \neq y$, on a $Z_2 = X\{y\backslash Y\}$.
2. Si $Z = y$, on a $Z_1 = Y$ et $Z_2 = Y\{x\backslash X\{y\backslash Y\}\}$. Comme $x \notin \mathtt{FV}(Y)$, alors on a $Z_2 = Y$.
3. Si $Z = \Omega$ ou si $Z = z$ avec $z \neq x$ et $z \neq y$, alors on a $Z_1 = Z = Z_2$.
4. Si $Z = \lambda z.Z'$ ou $Z = Z'Z''$, on conclut par hypothèse de récurrence. □

Moyennant une condition de bord portant sur les variables substituées, on peut inverser l'ordre de deux substitutions. Cette propriété classique du λ-calcul est exploitée par le lemme suivant.

Lemme 2.4 *1. Si $X \to_v X'$, alors $X\{y\backslash Y\} \to_v X'\{y\backslash Y\}$.*
2. Si $Y \to_v Y'$, alors $X\{y\backslash Y\} \twoheadrightarrow_v X\{y\backslash Y'\}$.
3. Si $X \to_v X'$ et $Y \to_v Y'$ alors $X\{y\backslash Y\} \twoheadrightarrow_v X'\{y\backslash Y'\}$.

Preuve : On prouve le premier point par par récurrence sur la taille de X. Les cas $X = x$ et $X = \Omega$ sont impossibles du fait de l'hypothèse $X \to_v X'$.

1. Si $X = (\lambda x.X_1)V \to_v X_1\{x\backslash V\} = X'$, alors, en renommant éventuellement x, on peut supposer $x \neq y$ et que x n'est pas libre dans Y. Par stabilité des valeurs par substitution, le terme $V\{y\backslash Y\}$ est une valeur. De ce fait, le terme $X\{y\backslash Y\} = (\lambda x.X_1\{y\backslash Y\})V\{y\backslash Y\}$ est bien un β_v-radical et on a la réduction $X\{y\backslash Y\} \to_v X_1\{y\backslash Y\}\{x\backslash V\{y\backslash Y\}\}$. On obtient donc, en utilisant le lemme 2.3, la réduction $X\{y\backslash Y\} \to_v X'\{y\backslash Y\}$.
2. Si $X = \lambda x.X_1 \to_v \lambda x.X_1' = X'$ ou si $X = X_1X_2 \to_v X_1'X_2$, ou si $X = X_1X_2 \to_v X_1X_2'$, alors on conclut par hypothèse de récurrence.

De même, on montre le deuxième point par récurrence sur la taille de X.

1. Si $X = x$ où $x \neq y$ ou si $X = \Omega$, alors on a $X\{y\backslash Y\} = X = X\{y\backslash Y'\}$.
2. Si $X = y$, alors on a $X\{y\backslash Y\} = Y \to_v Y' = X\{y\backslash Y'\}$.
3. Si $X = \lambda x.X_1$ ou si $X = X_1X_2$, alors on conclut par hypothèse de récurrence.

Le troisième point est un corollaire direct des deux premiers points. □

Le premier point énonce la compatibilité de la réduction $\to_v$ avec la substitution à gauche. Le deuxième point énonce la compatibilité de la réduction $\twoheadrightarrow_v$ avec la substitution à droite. Le troisième point est un corollaire des deux premiers points : la réduction $\twoheadrightarrow_v$ est compatible avec la substitution vue comme une fonction de deux termes vers un terme. De ce fait, les réductions par valeur vérifient exactement les mêmes propriétés de compatibilité avec la substitution que $\to$ et $\twoheadrightarrow$. Cela permet d'obtenir, comme dans le λ-calcul, le théorème de confluence locale.

Théorème 2.1 (Confluence locale) *Si $X \to_v X'$ et $X \to_v X''$, alors il existe un terme Y tel que $X' \twoheadrightarrow_v Y$ et $X'' \twoheadrightarrow_v Y$.*

Preuve : On procède par récurrence sur la taille de X. Les cas $X = x$ et $X = \Omega$ sont impossibles puisque $X \to_v X'$.

1. Si $X = (\lambda x.X_1)V \to_v X_1\{x\backslash V\} = X'$. Soit r le radical contracté entre X et X''. Trois cas sont à envisager.
   (a) Si r est le radical X, alors le résultat est immédiat.
   (b) Si r est dans X_1 alors $X'' = (\lambda x.X_1'')V$ avec $X_1 \to_v X_1''$. De là, on a $X'' \to_v X_1''\{x\backslash V\}$. Le lemme 2.4 permet de conclure.

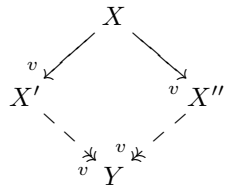


FIG. 2.1 – *Le λ-calcul par valeur est localement confluent*

(c) Si r est dans V, alors $X'' = (\lambda x.X_1)V'$ où $V \to_v V'$. De là, on a $X'' \twoheadrightarrow_v X_1\{x\backslash V'\}$. Le lemme 2.4 permet de conclure.

2. Le cas symétrique du cas précédent se traite de façon symétrique.
3. Si $X = \lambda x.X_1 \to_v \lambda x.X_1' = X'$ et si $X_1 \to_v \lambda x.X_1'' = X_1''$, alors l'hypothèse de récurrence permet de conclure.
4. Si $X = X_1X_2$ et si X n'est pas le radical réduit entre X et X' d'une part, et X et X'' d'autre part, alors, comme précédemment, l'utilisation de l'hypothèse de récurrence sur X_1 et X_2 donne le résultat. □

Comme l'illustre la figure 2.1, le λ-calcul par valeur est localement confluent. On prouve maintenant la propriété de confluence en utilisant la méthode de Tait et Martin-Löf. Pour cela, on définit ci-dessous la **relation des réductions parallèles** $\rightrightarrows_v$.

$$\begin{array}{rcll} x & \rightrightarrows_v & x & \\ \Omega & \rightrightarrows_v & \Omega & \\ XY & \rightrightarrows_v & X'Y' & \text{si } X \rightrightarrows_v X' \text{ et } Y \rightrightarrows_v Y' \\ (\lambda x.X)V & \rightrightarrows_v & X'\{x\backslash V'\} & \text{si } X \rightrightarrows_v X' \text{ et } V \rightrightarrows_v V' \\ \lambda x.X & \rightrightarrows_v & \lambda x.X' & \text{si } X \rightrightarrows_v X' \end{array}$$

Intuitivement, si $X \rightrightarrows_v Y$, alors Y peut être obtenu en contractant des radicaux ou des résidus de radicaux présents dans X. Ces radicaux sont indépendants dans le sens où tous ces radicaux sont des résidus de radicaux de X : aucun de ces radicaux n'est donc créé par la contraction d'un radical de X. Cette notion intuitive justifie l'appellation de *réductions parallèles*. En revanche, si dans le réduction $\mathcal{R} : X \xrightarrow{R_1}_v Y \xrightarrow{R_2}_v Z$, le radical R_2 est créé par R_1, alors, en général, on n'a pas $X \rightrightarrows_v Z$. La réduction $\mathcal{R}$ est une *séquence* de pas de réduction : la première réduction précède nécessairement la deuxième. Ces intuitions sur la relation $\rightrightarrows_v$ sont formalisées par les propriétés suivantes.

Lemme 2.5 *1. Si $X \to_v X'$, alors $X \rightrightarrows_v X'$.*
2. Si $X \rightrightarrows_v X'$, alors $X \twoheadrightarrow_v X'$.

Ces propriétés dont les preuves sont élémentaires font le lien entre $\to_v$ et $\rightrightarrows_v$. La deuxième propriété montre qu'il suffit de montrer la confluence locale forte de $\rightrightarrows_v$ pour montrer la confluence de $\to_v$. Comme précédemment, la preuve de la confluence locale forte de $\rightrightarrows_v$ s'appuie fondamentalement sur la propriété suivante de la substitution.

Lemme 2.6 *Si $X \rightrightarrows_v X'$ et $V \rightrightarrows_v V'$, alors $X\{x\backslash V\} \rightrightarrows_v X'\{x\backslash V'\}$.*

Preuve : On procède par récurrence sur la taille de X.

1. Si $X = y$ et $y \neq x$, alors on a $X' = y$. De là, on a $X\{x\backslash V\} = y \rightrightarrows_v y = X'\{x\backslash V'\}$.
2. Si $X = x$, on a $X' = x$. De là $X\{y\backslash V\} = V \rightrightarrows_v V' = X'\{x\backslash V'\}$.
3. Si $X = \Omega$, on a $X' = \Omega$. Le résultat est alors immédiat.
4. Si $X = X_1X_2$, deux cas sont possibles :

Confluence locale forte de $\rightrightarrows_v$ Confluence de $\to_v$

FIG. 2.2 – *Propriétés de confluence de $\rightrightarrows_v$ et $\to_v$*

(a) Si $X' = X'_1X'_2$ où $X_1 \rightrightarrows_v X'_1$ et $X_2 \rightrightarrows_v X'_2$, on utilise l'hypothèse de récurrence sur X_1 et X_2. On obtient $X_1\{x\backslash V\} \rightrightarrows_v X'_1\{x\backslash V'\}$ et $X_2\{x\backslash V\} \rightrightarrows_v X'_2\{x\backslash V'\}$. On a donc bien $X\{x\backslash V\} \rightrightarrows_v X'\{x\backslash V'\}$.

(b) Si $X = (\lambda y.Y)W$ et $X' = Y'\{y\backslash W'\}$ avec $Y \rightrightarrows_v Y'$ et $W \rightrightarrows_v W'$. Par hypothèse de récurrence, on a $Y\{x\backslash V\} \rightrightarrows_v Y'\{x\backslash V'\}$ et $W\{x\backslash V\} \rightrightarrows_v W'\{x\backslash V'\}$. De là, on obtient $X\{x\backslash V\} \rightrightarrows_v Y'\{x\backslash V'\}\{y\backslash W'\{x\backslash V'\}\}$. Comme $X'\{x\backslash V'\} = Y'\{y\backslash W'\}\{x\backslash V'\}$, on peut conclure par le lemme 2.4.

5. Si $X = \lambda y.Y$, alors on a $X' = \lambda y.Y'$ où $Y \rightrightarrows_v Y'$. On conclut comme précédemment par hypothèse d'induction. □

La relation $\rightrightarrows_v$ est compatible avec la substitution. Dans ce cas, c'est la relation élémentaire qui vérifie cette propriété de compatibilité et non la fermeture réfléxive transitive $\twoheadrightarrow_v$ de $\to_v$. De ce fait, on obtient une propriété de confluence locale forte pour $\rightrightarrows_v$.

Lemme 2.7 (Confluence locale forte) *Si $X \rightrightarrows_v X'$ et $X \rightrightarrows_v X''$, il existe un terme Y tel que $X' \rightrightarrows_v Y$ et $X'' \rightrightarrows_v Y$.*

Preuve : On procède par récurrence sur la taille de X.

1. Si $X = x$ ou $X = \Omega$, on a nécessairement $X' = X''$.
2. Si $X = X_1X_2$, plusieurs cas sont à considérer.
   (a) Si $X_1 = \lambda x.Z_1$ et $X_2 = V$ et si $X' = Z'_1\{x\backslash V'\}$ et $X'' = Z''_1\{x\backslash V''\}$ où $Z_1 \rightrightarrows_v Z'_1$, $V \rightrightarrows_v V'$, $Z_1 \rightrightarrows_v Z''_1$ et $V \rightrightarrows_v V''$, alors par hypothèse de récurrence, il existe deux termes Y_1 et W tels que $Z'_1 \rightrightarrows_v Y_1$, $Z''_1 \rightrightarrows_v Y_1$, $V' \rightrightarrows_v W$ et $V'' \rightrightarrows_v W$. De là, en utilisant le lemme 2.6, on obtient $X' \rightrightarrows_v Y_1\{x\backslash W\}$ et $X'' \rightrightarrows_v Y_1\{x\backslash W\}$.
   (b) Si $X_1 = \lambda x.Z_1$ et $X_2 = V$ et si $X' = Z'_1\{x\backslash V'\}$ et $X'' = (\lambda x.Z''_1)V''$ où $Z_1 \rightrightarrows_v Z''_1$, $V \rightrightarrows_v V'$, $Z_1 \rightrightarrows_v Z'_1$ et $V \rightrightarrows_v V''$, alors par hypothèse de récurrence, il existe deux termes Y_1 et W tels que $Z'_1 \rightrightarrows_v Y_1$, $Z''_1 \rightrightarrows_v Y_1$, $V' \rightrightarrows_v W$ et $V'' \rightrightarrows_v W$. En utilisant le lemme 2.6, on obtient d'une part $X' \rightrightarrows_v Y_1\{x\backslash W\}$. D'autre part, par définition de $\rightrightarrows_v$, on a $X'' \rightrightarrows_v Y_1\{x\backslash W\}$.
   (c) Le cas symétrique se traite de façon symétrique.
   (d) Si $X' = X'_1X'_2$ et $X'' = X''_1X''_2$ où $X_1 \rightrightarrows_v X'_1$, $X_2 \rightrightarrows_v X'_2$, $X_1 \rightrightarrows_v X''_1$ et $X_2 \rightrightarrows_v X''_2$, alors on conclut par hypothèse de récurrence.
3. Si $X = \lambda x.X_1$, on a nécessairement $X' = \lambda x.X'_1$ où $X_1 \rightrightarrows_v X'_1$ et $X'' = \lambda x.X''_1$ où $X_1 \rightrightarrows_v X''_1$. On conclut par hypothèse de récurrence. □

La relation des réductions parallèles est fortement localement confluente. En exploitant les propriétés élémentaires énoncées par le lemme 2.5, on en déduit que la réduction par valeur $\to_v$ est confluente. Ces propriétés de $\rightrightarrows_v$ et $\to_v$ sont illustrées sur la figure 2.2.

Théorème 2.2 (Confluence) *Si $M \to_v^\star M_1$ et $M \to_v^\star M_2$, alors il existe un terme N vérifiant $M_1 \to_v^\star N$ et $M_2 \to_v^\star N$.*

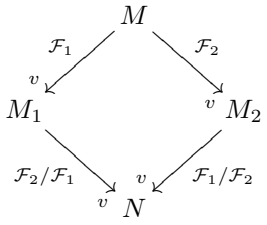


FIG. 2.3 – *Le λ-calcul par valeur vérifie le lemme des déplacements parallèles.*

Preuve : Cette propriété est une conséquence immédiate de la confluence locale forte de $\rightrightarrows_v$ et du lemme 2.5. □

Le technique de Tait et Martin-Löf nous permet donc d'obtenir la confluence du λ-calcul par valeur. Pour prouver cette propriété, nous aurions pu mettre à profit les travaux sur les systèmes de réécriture de terme de Klop [24]. En effet, le λ-calcul par valeur peut être vu comme un système de réécriture dont les règles de réécriture sont $(\lambda x.M)\lambda y.N \to_v M\{x\backslash\lambda y.N\}$ pour $x, y \in \mathbf{X}$. Ce système est linéaire à gauche et n'a pas de paire critique. La théorie des systèmes de réécriture garantit donc la propriété de confluence. Nous avons préféré utiliser ici une méthode de preuve concrète et indépendante de résultats non prouvés dans le cadre de ce mémoire.

2.1.2 Développements finis

La théorème de développements finis pour le λ-calcul par valeur s'obtient de façon quasiment directe à partir du théorème correspondant dans le λ-calcul. En effet, par définition de β_v-résidu, les notions de résidu des β_v-radicaux coïncident dans le λ-calcul et dans le λ-calcul par valeur. De ce fait, le théorème des développements finis du λ-calcul est applicable et prouve le théorème des développements finis pour le λ-calcul par valeur.

Théorème 2.3 (Développements finis) *Soit $\mathcal{F}$ un ensemble de β_v-radicaux de M.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Tous les développements de $\mathcal{F}$ finissent sur un même terme N.*
3. *L'ensemble des résidus d'un β_v-radical R de M dans N est indépendant du développement considéré.*

Preuve : Ce résultat est un corollaire du théorème des développements finis du λ-calcul. □

Si $\mathcal{F}$ est un ensemble de β_v-radicaux d'un terme, les développements de $\mathcal{F}$ sont de longueur finie et terminent sur un même terme. De plus l'ensemble des résidus d'un radical du terme initial est le même quel que soit le développement considéré. Ce résultat implique le lemme des déplacements parallèles.

Lemme 2.8 (Lemme des déplacements parallèles) *On suppose que $\mathcal{F}_1$ et $\mathcal{F}_2$ sont deux ensembles de β_v-radicaux du terme M. Si $M \xrightarrow{\mathcal{F}_1}_v M_1$ et $M \xrightarrow{\mathcal{F}_2}_v M_2$ et si $\mathcal{F}'_1$ (respectivement $\mathcal{F}'_2$) est l'ensemble des résidus de $\mathcal{F}_1$ (resp. $\mathcal{F}_2$) dans M_2 (resp. M_1), alors il existe un terme N tel que $M_1 \xrightarrow{\mathcal{F}'_2}_v N$ et $M_2 \xrightarrow{\mathcal{F}'_1}_v N$.*

Ce résultat, illustré sur la figure 2.3, a pour corollaire immédiat le résultat suivant.

Corollaire 2.1 *On considère deux réductions $\mathcal{R}_1 : M \twoheadrightarrow_v M_1$ et $\mathcal{R}_2 : M \twoheadrightarrow_v M_2$ issues d'un terme M. Soit M' le terme qui vérifie $(\mathcal{R}_2/\mathcal{R}_1) : M_1 \twoheadrightarrow_v M'$ et $(\mathcal{R}_1/\mathcal{R}_2) : M_2 \twoheadrightarrow_v M'$. Si S est un radical de M, les résidus de S par la réduction $\mathcal{R}_1;(\mathcal{R}_2/\mathcal{R}_1)$ et par la réduction $\mathcal{R}_2;(\mathcal{R}_1/\mathcal{R}_2)$ vérifient $S/(\mathcal{R}_1;(\mathcal{R}_2/\mathcal{R}_1)) = S/(\mathcal{R}_2;(\mathcal{R}_1/\mathcal{R}_2))$.*

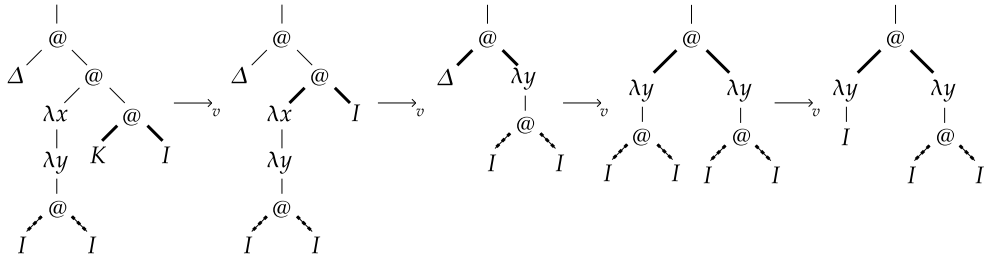


FIG. 2.4 – *Réduction par valeur du terme* $M = \Delta((\lambda x.\lambda y.II)(KI))$

Si $\mathcal{R}_1$ et $\mathcal{R}_2$ sont deux réductions du λ-calcul par valeur issues d'un terme M, les ensembles des résidus d'un radical de M coïncident pour les réductions $\mathcal{R}_1;(\mathcal{R}_2/\mathcal{R}_1)$ et $\mathcal{R}_2;(\mathcal{R}_1/\mathcal{R}_2)$. Ce résultat énonce la cohérence de la définition de réduction-résidu vis-à-vis de la notion de radical résidu.

2.1.3 Standardisation

Comme le montre le contre-exemple suivant, le théorème de standardisation n'est pas conservé tel quel dans le λ-calcul par valeur. On considère la réduction du terme $M = \Delta((\lambda x.\lambda y.II)(KI))$ où $I = \lambda x.x$, $\Delta = \lambda x.xx$ et $K = \lambda x.\lambda y.y$. Le terme M contient deux β_v-radicaux $R_1 = II$ et $R_2 = KI$. Le radical R_1 est à gauche de R_2.

$$\mathcal{R}_0 : M \xrightarrow{R_2}_v \Delta((\lambda x.\lambda y.II)I) \xrightarrow{R_3}_v \Delta\lambda y.II \xrightarrow{R_4}_v (\lambda y.II)\lambda y.II \xrightarrow{R'_1}_v (\lambda y.I)\lambda y.II = M'$$

Cette réduction est illustrée sur la figure 2.4. La première réduction consiste à contracter $R_2 = KI$. Ce radical crée le radical $R_3 = (\lambda x.\lambda y.II)I$ par la droite. Ce dernier est contracté à son tour et crée le radical $R_4 = \Delta\lambda y.II$ par la droite. Après la contraction de R_4, le radical $R_5 = (\lambda y.II)\lambda y.II$ est créé. Dans ce terme, on contracte le radical $R'_1 = ((\lambda y.[\,])\lambda y.II,II)$ qui s'avère être un résidu de R_1. Supposons qu'il existe une réduction standard $\mathcal{R} : M \twoheadrightarrow_v M'$. Comme R_2 et un résidu de R_1 sont contractés, et comme R_1 est le radical le plus à gauche dans M, alors R_1 est nécessairement le premier radical contracté de $\mathcal{R}$. Cette réduction se décompose donc de la façon suivante.

$$\mathcal{R} : M \xrightarrow{R_1}_v \Delta((\lambda x.\lambda y.I)(KI)) \twoheadrightarrow_v M'$$

Le terme $M_1 = \Delta((\lambda x.\lambda y.I)(KI))$ ne contient qu'un seul radical. Ce radical $R'_2 = KI$ est le résidu de R_2. En contractant R'_2, on obtient le terme M_2 qui contient un unique radical $R'_3 = (\lambda x.\lambda y.I)I$. En contractant R'_3, un nouveau radical $R'_4 = (\lambda y.z)\lambda y.z$ est créé. Sa contraction crée à nouveau un radical $R'_5 = (\lambda y.I)\lambda y.I$ dont la contraction aboutit à une forme normale. On obtient finalement la réduction suivante.

$$\mathcal{R} : M \xrightarrow{R_1}_v \Delta((\lambda x.\lambda y.I)(KI)) \xrightarrow{R'_2}_v \Delta((\lambda x.\lambda y.I)I) \xrightarrow{R'_3}_v \Delta\lambda y.I \xrightarrow{R'_4}_v (\lambda y.I)\lambda y.I \xrightarrow{R'_5}_v I$$

Cette réduction est illustrée sur la figure 2.5. A partir de la deuxième étape, les termes ne contiennent qu'un radical, cette réduction est la seule réduction issue de M qui commence par contracter R_1. Pourtant, cette réduction n'aboutit pas à M' ce qui constitue un contre-exemple de la propriété de standardisation. En examinant la réduction $\mathcal{R}_0$, on remarque qu'un résidu de R_1 (en pointillé sur la figure 2.4) est dupliqué alors que R_1 est le radical le plus à gauche de M. En fait, la contraction de R_2 dans M crée le radical R_3 qui est à gauche du résidu de R_1. On en déduit que dans le λ-calcul par valeur, les radicaux les plus à gauche ne vérifient plus les propriétés mentionnées dans le résultat 1.4. Ce constat n'est pas surprenant dans la mesure où la notion de radical est plus restrictive dans le λ-calcul par valeur. Dans le λ-calcul, le terme M contient

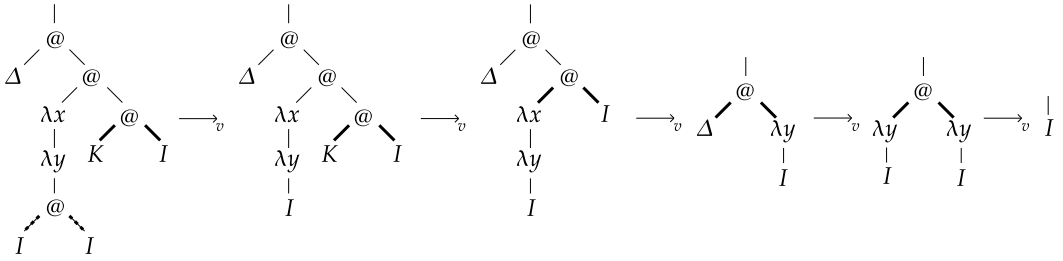
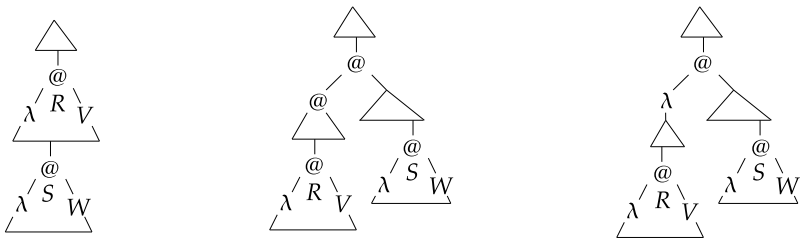


FIG. 2.5 – *Réduction par valeur du terme* $M = \Delta((\lambda x.\lambda y.II)(KI))$



1. $R <_{st} S$ 2a. $R <_{st} S$ 2b. $S <_{st} R$

FIG. 2.6 – *Ordre* $<_{st}$ *entre deux radicaux d'un terme*

deux radicaux supplémentaires : le terme M lui-même et le sous-terme $(\lambda x.\lambda y.II)(KI)$. Dans la réduction standard correspondant à $\mathcal{R}_0$ dans le λ-calcul, le premier radical contracté est M. Cette réduction n'est donc pas une réduction par valeur.

De façon générale, le théorème de standardisation vise à montrer que tous les termes N issus d'un terme initial M par réduction sont atteignables à partir de M par une réduction appartenant à une classe particulière de réductions. Dans le λ-calcul, ces réductions dites standard contractent les radicaux de gauche à droite. Plus précisément, si R est le radical contracté dans M, les résidus d'un radical S de M qui vérifie $S <_g R$, c'est-à-dire S est strictement à gauche de R, ne sont pas contractés dans la suite de la réduction. Dans le λ-calcul par valeur, du fait qu'un radical peut être créé *par la droite*, la définition d'une réduction standard est plus subtile. On introduit un **ordre strict total** $<_{st}$ entre radicaux d'un même terme. Cette relation est définie en fonction des positions respectives des radicaux dans le terme.

1. Si R contient strictement S alors $R <_{st} S$.
2. Si R et S sont disjoints, il existe des contextes $C[\,]$, $C_1[\,]$ et $C_2[\,]$ tels que $M = C[C_1[R]C_2[S]]$.
   (a) Si $C_1[R]$ est une application, alors $R <_{st} S$.
   (b) Si $C_1[R]$ est une abstraction, alors $S <_{st} R$.

L'ordre $<_{st}$ est illustré sur la figure 2.6. On note $\leq_{st}$ l'ordre associé. Cette relation vise à retrouver les propriétés du lemme 1.4 qui ne sont plus vérifiées par l'ordre $<_g$ dans le λ-calcul par valeur. En particulier, si R est plus petit, au sens de $<_{st}$, que le radical contracté, on souhaite que R ait un unique résidu que ce résidu soit plus petit que les radicaux créés par cette contraction. Comme pour $<_g$, le premier point indique que si le radical R contient le radical S, alors $R <_{st} S$. De même, si une application contient S dans son membre droit, R dans son membre gauche et si ce membre gauche est une application, alors on a $R <_{st} S$. Ce cas 2a est aussi vérifié par l'ordre $<_g$. La différence entre $<_g$ et $<_{st}$ réside dans le cas 2b où un radical R est situé dans une abstraction qui est le membre gauche d'une application dont le membre droit contient S. Dans ce cas, on a $S <_{st} R$. La raison de cette divergence s'explique intuitivement de la façon suivante : une réduction du membre

droit $C_2[S] \xrightarrow{S}_v C_2[S']$ peut créer *par la droite* le radical $T = (\lambda x.C_1[R])C_2[S']$. La contraction de S peut donc créer un radical T qui contient le résidu de R et qui vérifie donc $T <_{st} R$. Comme on vise les propriétés du lemme 1.4, ceci implique nécessairement $S <_{st} R$. Plus formellement, le lemme suivant montre que les propriétés du résultat 1.4 sont vérifiées par $<_{st}$.

Lemme 2.9 *On suppose $\mathcal{R} : M \xrightarrow{S}_v N$. Soit R un radical de M qui vérifie $R <_{st} S$.*

1. *R a un unique résidu R' dans N.*
2. *Si T' est un radical créé de N alors $R' <_{st} T'$.*
3. *Si T vérifie $R \leq_{st} T$ et si T' est un résidu de T, alors on a $R' \leq_{st} T'$.*

Preuve : On raisonne par cas sur $R <_{st} S$. On note S' le contractum de S.

1. Si R contient strictement S, alors il existe des contextes C et C' tels que $M = C[R]$ et $R = C'[S]$. Deux cas sont possibles pour C'. Si $C' = (\lambda x.C''[\,])V$, alors $N = C[R']$ où $R' = (\lambda x.C''[S'])V$ est l'unique résidu de R dans N. Si T' est un radical créé par la contraction de S, alors T est contenu dans $C''[S']$, donc dans R'. Par conséquent, on a $R' <_{st} T'$. Si T est un radical de M qui vérifie $R \leq_{st} T$, trois cas sont à envisager.
   (a) Si R contient T, chaque résidu T' de T est contenu dans R'. On a donc $R' \leq_{st} T'$.
   (b) Si $M = C_0[(\lambda x.C_1[T])C_2[R]]$, alors $N = C_0[(\lambda x.C_1[T])C_2[R']]$ où T est l'unique résidu de T dans N. Ceci implique $R' <_{st} T$.
   (c) Si $M = C_0[C_1[R]\ C_2[T]]$ où $C_1[R]$ est une application, alors $N = C_0[C_1[R']\ C_2[T]]$. T est l'unique résidu de T dans N. Comme $C_1[R']$ est une application, on a $R' <_{st} T$.

   Si $C' = (\lambda x.M_1)\lambda x.C''[\,]$, on procède de façon similaire au cas précédent.
2. Si $M = C[(\lambda x.C'[S])C''[R]]$, alors $N = C[(\lambda x.C'[S'])C''[R]]$. R est l'unique résidu de R dans N. Un radical T' créé par la contraction de S est contenu dans $C'[S']$ et vérifie donc $R <_{st} T'$. Soit T un radical de M qui vérifie $R \leq_{st} T$. Trois cas sont à envisager.
   (a) Si R contient T, alors T a un unique résidu T dans N qui est aussi contenu dans R. On a donc $R \leq_{st} T$.
   (b) Si $M = C_0[(\lambda x.C_1[T])C_2[R]]$, trois cas sont à envisager.
      i. Si $M = C_0[(\lambda x.C_1[T])C_2'[(\lambda x.C'[S])C''[R]]]$, alors $N = C_0[(\lambda x.C_1[T])C_2''[R]]$ où $C_2'' = C_2'[(\lambda x.C'[S'])C''[\,]]$. Dans ce cas, T a un unique résidu T dans N qui vérifie $R <_{st} T$.
      ii. Si $M = C[(\lambda x.C'[S])C_2'[(\lambda x.C_1[T])C_2[R]]]$, alors on a $N = C_0'[(\lambda x.C_1[T])C_2[R]]$ où $C_0' = C[(\lambda x.C'[S'])C_2'[\,]]$. T a un unique résidu T dans N qui vérifie $R <_{st} T$.
      iii. Si $C_0 = C$, chaque résidu T' de T par la contraction de S est contenu dans $C'[S']$, ce qui implique $R <_{st} T'$.
   (c) Si $M = C_0[C_1[R]\ C_2[T]]$ où $C_1[R]$ est une application, alors trois cas sont à envisager.
      i. Si $M = C_0[C_1'[(\lambda x.C'[S])C''[R]]\ C_2[T]]$, alors $N = C_0[C_1''[R]\ C_2[T]]$ où on pose $C_1'' = C_1'[(\lambda x.C'[S'])C''[\,]]$. Dans ce cas, T a un unique résidu T dans N. Comme $C_1''[R]$ est une application, on a $R <_{st} T$.
      ii. Si $M = C[(\lambda x.C'[S])C_1'[C_1[R]\ C_2[T]]]$, alors $N = C_0'[C_1[R]\ C_2[T]]$ où on pose $C_0' = C[(\lambda x.C'[S'])C_1'[\,]]$. Dans ce cas, T a un unique résidu T dans N. Comme $C_1[R]$ est une application, on a $R <_{st} T$.
      iii. Le cas $C = C_0$ est exclu car la partie gauche de C est une abstraction alors que celle de C_0 est une application.
3. Si $M = C[C'[R]\ C''[S]]$ où $C'[R]$ est une application, alors $N = C[C'[R]\ C''[S']]$. R a un unique résidu R dans N. Chaque radical T' créé par la contraction de S est contenu dans $C''[S']$ et vérifie donc $R <_{st} T'$. Soit T un radical de M qui vérifie $R \leq_{st} T$. Trois cas sont à envisager.

(a) Si R contient T, alors T a un unique résidu dans R qui est aussi contenu dans R. On obtient donc $R \leq_{st} T$.

(b) Si $M = C_0[(\lambda x.C_1[T])C_2[R]]$, trois cas sont à envisager.

i. Si $M = C_0[(\lambda x.C_1[T])\ C'_2[C'[R]\ C''[S]]]$, alors on a $N = C_0[(\lambda x.C_1[T])\ C''_2[R]]$ où $C''_2 = C'_2[C'[\,]\ C''[S']]$. Dans ce cas, T a un unique résidu T dans N et $R <_{st} T$.

ii. Si $M = C[C'_1[(\lambda x.C_1[T])\ C_2[R]]\ C''[S]]$, alors on a $N = C'_0[(\lambda x.C_1[T])\ C_2[R]]$ où $C'_0 = C[C'_1[\,]\ C''[S']]$. Dans ce cas, T a un unique résidu T dans N et $R <_{st} T$.

iii. Le cas $C = C_0$ est exclu.

(c) Si $M = C_0[C_1[R]\ C_2[T]]$ où $C_1[R]$ est une application, alors trois cas sont à envisager.

i. Si $M = C_0[C'_1[C'[R]\ C''[S]]\ C_2[T]]$, alors on a $N = C_0[C''_1[R]\ C_2[T]]$ où on pose $C''_1 = C'_1[C'[\,]\ C''[S']]$. Dans ce cas, T a un unique résidu T dans N. Et comme $C''_1[R]$ est une application, T vérifie $R <_{st} T$.

ii. Si $M = C[C'_1[C_1[R]C_2[T]]C''[S]]$, alors $N = C'_0[C_1[R]C_2[T]]$ où $C'_0 = C[C'_1[\,]C''[S']]$. T a un unique résidu T dans N. Et comme $C_1[R]$ est une application, T vérifie $R <_{st} T$.

iii. Si $C_0 = C$, alors chaque résidu T' de T est contenu dans $C''[S']$ et vérifie donc $R <_{st} T'$. □

Si R est un radical de M plus petit que S au sens de $<_{st}$, et si S est contracté, alors R a un unique résidu R' qui est plus petit que tous les radicaux créés par la contraction de S. De plus, si un radical T de M est plus grand que R, les résidus de T sont aussi plus grands que R'. Ce résultat peut s'étendre aux familles de radicaux. Si $\mathcal{F}$ est un ensemble de radicaux et R un radical de M, on note $R <_{st} \mathcal{F}$ si et seulement si pour tout radical S de $\mathcal{F}$, on a $R <_{st} S$.

Corollaire 2.2 *On suppose $\mathcal{R} : M \xrightarrow{\mathcal{F}}_v N$. Soit R un radical de M qui vérifie $R <_{st} \mathcal{F}$.*

1. *$R/\mathcal{R}$ est un singleton $\{R'\}$.*
2. *Si T' est un radical de M' qui n'est pas résidu d'un radical de M, alors on a $R' <_{st} T'$.*
3. *Si T vérifie $R \leq_{st} T$ et si T' est un résidu de T dans N, alors on a $R' \leq_{st} T'$.*

Si R est un radical de M plus petit que l'ensemble de radicaux $\mathcal{F}$ et si $M \xrightarrow{\mathcal{F}}_v N$, alors R a un unique résidu R' dans N. Ce résidu est plus petit que tous les radicaux créés par le développement de $\mathcal{F}$. Et si un radical T de M est plus grand que R, les résidus de T par un développement de $\mathcal{F}$ sont aussi plus grands que R'.

La définition de réduction standard est adaptée à l'aide de la relation d'ordre $\leq_{st}$. La réduction $M_0 \xrightarrow{R_1} M_1 \xrightarrow{R_2} M_2 \xrightarrow{R_3} \cdots \xrightarrow{R_n} M_n$ est dite **standard pour** $\leq_{st}$ si et seulement si pour tout i,j tels que $1 \leq i < j \leq n$ le radical R_j n'est pas un résidu d'un radical R'_j de M_{i-1} tel que $R'_j \leq_{st} R_i$. Dans la suite de cette section, on appellera réduction standard une réduction standard pour $\leq_{st}$.

Théorème 2.4 (Standardisation) *Si $M \twoheadrightarrow_v N$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_v N$ telle que $\mathcal{R}$ est standard pour $\leq_{st}$.*

Preuve : On adapte au cas présent la preuve donnée par Klop dans [24]. La réduction de M à N s'écrit : $M \xrightarrow{R_1}_v M_1 \xrightarrow{R_2}_v M_2 \xrightarrow{R_3}_v \ldots \xrightarrow{R_{n-1}}_v M_{n-1} \xrightarrow{R_n}_v N$. En posant, pour $1 \leq i \leq n$, $\mathcal{F}_i^0 = \{R_i\}$, on obtient :

$$M \xrightarrow{\mathcal{F}_1^0}_v M_1 \xrightarrow{\mathcal{F}_2^0}_v M_2 \xrightarrow{\mathcal{F}_3^0}_v \ldots \xrightarrow{\mathcal{F}_{n-1}^0}_v M_{n-1} \xrightarrow{\mathcal{F}_n^0}_v N$$

Soit $\mathcal{A}_1$ l'ensemble des radicaux de M qui ont un résidu dans un des ensembles de la suite $\{\mathcal{F}_i^0\}_{1\leq i\leq n}$. Soit S^1 le radical minimum pour $\leq_{st}$ de $\mathcal{A}_1$ et $\mathcal{F}_{k_1}^0$ la première famille à laquelle un résidu de S^1 appartient. En utilisant le lemme des déplacements parallèles, on construit le

diagramme suivant.

$$\begin{array}{ccc} M & \xrightarrow{\mathcal{F}_1^0}_v & M_1 \xrightarrow{\mathcal{F}_2^0}_v M_2 \\ \downarrow{\scriptstyle S^1} & & \downarrow{\scriptstyle S^1/\mathcal{F}_1^0} \\ M_0^1 & \xrightarrow[v]{\mathcal{F}_1^0/S^1} & M_1^1 \end{array}$$

On a $S^1 \leq_{st} \mathcal{F}_1^0$. Deux cas sont à envisager pour $S^1/\mathcal{F}_1^0$.

1. Si $S^1 \in \mathcal{F}_1^0$ (i.e. $k_1 = 1$), on a $S^1/\mathcal{F}_1^0 = \emptyset$ et $M_1 = M_1^1$.
2. Si $S^1 \notin \mathcal{F}_1^0$ (i.e. $k_1 > 1$). Comme $S^1 <_{st} \mathcal{F}_1^0$, en appliquant le corollaire 2.2, on obtient que $S^1/\mathcal{F}_1^0 = \mathcal{S}_1^1 = \{S_1^1\}$. Soit R' un radical de $\mathcal{F}_2^0$. Deux cas sont possibles.
   (a) R' est un résidu de R qui est un radical de M. Par définition de S^1, R vérifie $S^1 \leq_{st} R$. Par le corollaire 2.2, on obtient $S_1^1 \leq_{st} R'$.
   (b) R' n'est pas un résidu d'un radical de M : ce radical est donc créé par le développement de $\mathcal{F}_1^0$. Par le corollaire 2.2, on obtient $S_1^1 <_{st} R'$.

   Par conséquent, on a $S_1^1 \leq_{st} \mathcal{F}_2^0$. Le raisonnement en deux cas appliqué ici à M, S^1 et $\mathcal{F}_1^0$ peut donc s'appliquer à M_1, $\mathcal{S}_1^1$ et $\mathcal{F}_2^0$, et ainsi de suite, jusqu'au moment où on traitera le cas M_{k_1-1}, $S_{k_1-1}^1$ et $\mathcal{F}_{k_1}^0$.

Par conséquent, en posant, pour $i \in \{2 \dots n\}$, $\mathcal{S}_i^1 = S^1/(\mathcal{F}_1^0; \mathcal{F}_2^0; \dots; \mathcal{F}_i^0)$ et $\mathcal{F}_1^1 = \mathcal{F}_1^0/\{S^1\}$ et $\mathcal{F}_i^1 = \mathcal{F}_i^0/\mathcal{S}_{i-1}^1$, on obtient le diagramme suivant.

$$\begin{array}{ccccccccccc} M & \xrightarrow{\mathcal{F}_1^0}_v & M_1 & \xrightarrow{\mathcal{F}_2^0}_v & M_2 & \xrightarrow{\mathcal{F}_3^0}_v & M_3 & \xrightarrow{\mathcal{F}_4^0}_v & \cdots & \xrightarrow{\mathcal{F}_{n-1}^0}_v & M_{n-1} \xrightarrow{\mathcal{F}_n^0}_v M_n = N \\ \downarrow{\scriptstyle S_1} & & \downarrow{\scriptstyle \mathcal{S}_1^1} & & \downarrow{\scriptstyle \mathcal{S}_2^1} & & \downarrow{\scriptstyle \mathcal{S}_3^1} & & & & \downarrow{\scriptstyle \mathcal{S}_{n-1}^1} \quad \downarrow{\scriptstyle \emptyset} \\ M_0^1 & \xrightarrow[v]{\mathcal{F}_1^1} & M_1^1 & \xrightarrow[v]{\mathcal{F}_2^1} & M_2^1 & \xrightarrow[v]{\mathcal{F}_3^1} & M_3^1 & \xrightarrow[v]{\mathcal{F}_4^1} & \cdots & \xrightarrow[v]{\mathcal{F}_{n-1}^1} & M_{n-1}^1 \xrightarrow[v]{\mathcal{F}_n^1} M_n^1 = N \end{array}$$

D'après le raisonnement précédent sur les sous-diagrammes, on sait que si k vérifie $k_1 \leq k \leq n-1$, alors on a $\mathcal{S}_k^1 = \emptyset$, $\mathcal{F}_{k+1}^0 = \mathcal{F}_{k+1}^1$ et $M_{k+1}^1 = M_{k+1}$. Et si k vérifie $1 \leq k \leq k_1 - 1$, alors $\mathcal{S}_k^1$ est un singleton.

Tant que la suite $\{\mathcal{F}_i^l\}_{1 \leq i \leq n}$ contient un ensemble non vide, on peut poursuivre la construction du diagramme en ajoutant une ligne $l+1$. On considère l'extension illustrée sur la figure 2.7, dans laquelle les entiers l et k vérifient $2 \leq l \leq p$ et $1 \leq k \leq n$. On pose $\mathcal{S}_k^l = S^l/\mathcal{F}_1^{l-1}; \mathcal{F}_2^{l-1}; \dots; \mathcal{F}_k^{l-1}$ et $\mathcal{F}_k^l = \mathcal{F}_k^{l-1}/\mathcal{S}_{k-1}^l$. On sait, par construction, que si l vérifie $1 \leq l \leq p$, il existe un rang k_l tel que pour tout k, on a

1. si $1 \leq k \leq k_l - 1$, alors $\mathcal{S}_k^l$ est un singleton,
2. si $k_l \leq k \leq n-1$, alors $\mathcal{S}_k^l = \emptyset$.

On souhaite tout d'abord montrer que la réduction de M à M_0^p est standard. On procède par l'absurde. Pour cela, on suppose, pour $i < j$, que le radical S^j de M_0^{j-1} est résidu d'un radical R^j de M_0^{i-1} tel que $R^j <_{st} S^i$. Par définition de S^j et k_j, l'ensemble $\mathcal{F}_{k_j}^{j-1}$ contient (au moins) un résidu T^j de S^j et les ensembles $\mathcal{F}_k^{j-1}$ pour $k < k_j$ ne contiennent pas de résidu de S^j. On s'intéresse au sous-diagramme correspondant à cette situation.

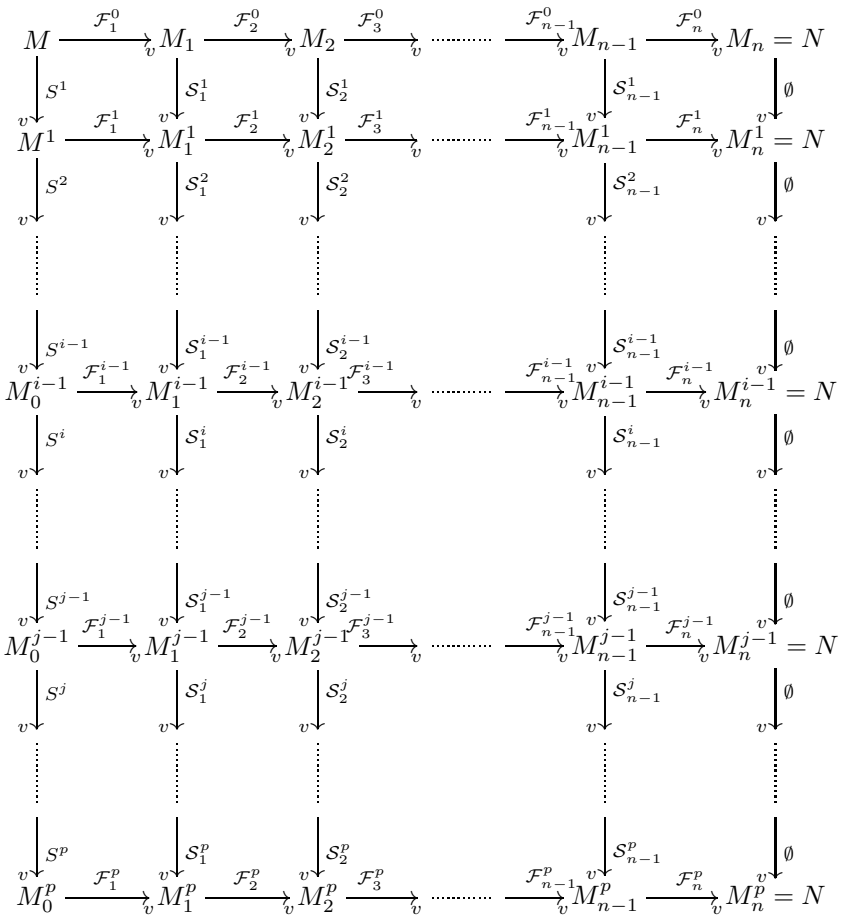
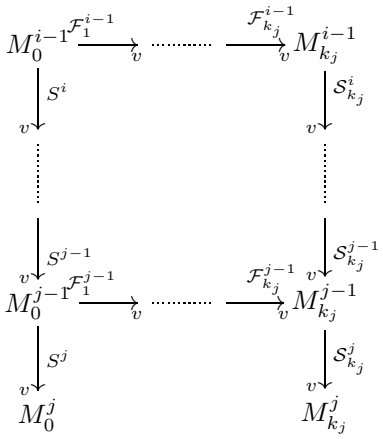


FIG. 2.7 – *Diagramme de construction d'une réduction standard*

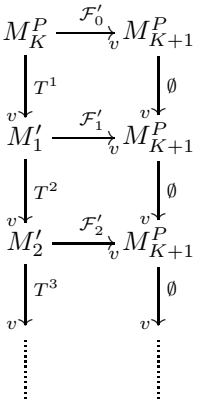


On sait que l'ensemble $\mathcal{S}^j_{k_j}$ est un singleton contenant l'unique résidu de S^j par la réduction de M_0^{j-1} à $M_{k_j}^{j-1}$. On a donc $\mathcal{S}^j_{k_j} = \{T^j\}$. Comme T^j est résidu de S^j par la réduction de M_0^{j-1} à $M_{k_j}^{j-1}$ et que S^j est résidu de R^j par la réduction de M_0^{i-1} à M_0^{j-1}, alors, T^j est résidu de R^j par la réduction de M_0^{i-1} à M_0^{j-1} puis $M_{k_j}^{j-1}$. Par le corollaire 2.1, on obtient que T^j est aussi résidu de R^j par la réduction de M_0^{i-1} à $M_{k_j}^{i-1}$ puis $M_{k_j}^{j-1}$. Comme $T^j \in \mathcal{F}_{k_j}^{j-1}$, alors T^j est résidu d'un radical de $\mathcal{F}_{k_j}^{i-1}$. Par conséquent, R^j a un résidu dans $\mathcal{F}_{k_j}^{i-1}$ et vérifie $R^j <_{st} S^i$. On obtient là une contradiction, ce qui prouve que la réduction de M à M_0^p est standard.

On souhaite maintenant montrer que la réduction construite ici entre M et M_0^p finit par aboutir à N. On raisonne par l'absurde et on suppose qu'à chaque étape $p > 0$, il existe un indice k tel que $1 \leq k \leq n$ et $\mathcal{F}_k^p$ est non vide, ce qui implique qu'on peut construire une étape supplémentaire $p+1$ et ainsi de suite. Les réductions de M_0^p à M_0^{p+1} contractent toujours, par construction, un radical alors que les réductions de M_n^p à M_n^{p+1}, n'en contractent jamais. On en déduit qu'il existe deux entiers P et K tels que la réduction $M_K^P \xrightarrow{\mathcal{S}_K^{P+1}}_v M_K^{P+1} \xrightarrow{\mathcal{S}_K^{P+2}}_v M_K^{P+2} \xrightarrow{\mathcal{S}_K^{P+3}}_v \ldots$ contracte une infinité de radicaux alors que la réduction $M_{K+1}^P \xrightarrow{\mathcal{S}_{K+1}^{P+1}}_v M_{K+1}^{P+1} \xrightarrow{\mathcal{S}_{K+1}^{P+2}}_v M_{K+1}^{P+2} \xrightarrow{\mathcal{S}_{K+1}^{P+3}}_v \ldots$ n'en contracte aucun. Plus précisément, pour tout $p \geq P$, on a :

1. $\mathcal{S}^p_{K+1} = \emptyset$,
2. il existe un indice $p' \geq p$ tel que $\mathcal{S}^{p'}_K$ est un singleton.

On obtient donc le diagramme (infini) suivant dans lequel les radicaux T^i sont les uniques éléments des singletons mentionnés ci-dessus. On pose $\mathcal{F}'_0 = \mathcal{F}^{P-1}_{K+1}$ et $\mathcal{F}'_i = \mathcal{F}'_{i-1}/T^i$ pour $i \geq 1$.



Comme, par construction, $T^i/\mathcal{F}'_{i-1}$ est vide, cela implique, par le corollaire 2.2, que T^i appartient à l'ensemble $\mathcal{F}'_{i-1}$ (sinon il y aurait un unique résidu). Par conséquent, les radicaux T^i sont des radicaux ou des résidus de radicaux de $\mathcal{F}'_0$. La réduction de M_K^P à M'_j est donc relative à $\mathcal{F}'_0$ pour tout j ce qui apporte une contradiction au théorème des développements finis. Il existe donc un

rang p tel que les ensembles $\mathcal{F}_k^p$ pour $k \in \{1 \dots n\}$ sont tous vides ce qui implique $M^P = N$. □

La preuve utilisée ici est axiomatique. Elle utilise le théorème des développements finis 2.3, le lemme des déplacements parallèles 2.8, son corollaire 2.1 de cohérence de la notion de résidu et le corollaire 2.2 des propriétés de l'ordre $\leq_{st}$ sur les radicaux. Comme ces théorèmes, corollaires et lemmes sont vrais dans le λ-calcul et le λ-calcul étiqueté, cette preuve pourrait être utilisée pour prouver la standardisation du λ-calcul ou du λ-calcul étiqueté. Elle sera utilisée pour prouver la standardisation du λ-calcul par valeur étiqueté.

Considérons à présent, à titre d'illustration, le contre-exemple que nous avions utilisé pour montrer que l'ordre des radicaux utilisé dans le λ-calcul n'était pas adapté au λ-calcul par valeur. Comme nous l'avons vu, $M = \Delta((\lambda x.\lambda y.II)(KI))$ contient deux radicaux disjoints $R_1 = II$ et $R_2 = KI$. Si R_1 est bien plus petit que R_2 au sens de $<_g$, on a en revanche $R_2 <_{st} R_1$. La réduction standard associée à la réduction $\mathcal{R}_0$ commence donc par contracter R_2. Par la suite, la réduction $\mathcal{R}_0$ contracte toujours un radical qui contient les autres radicaux du terme. On en déduit donc que la réduction $\mathcal{R}_0$ est, en réalité, une réduction standard.

Une autre approche, plus abstraite, de la standardisation a été introduite par Gonthier, Lévy et Melliès [19]. Ces derniers définissent la standardisation simplement en fonction de la notion de résidu et de la relation d'inclusion entre radicaux. Cette présentation abstraite a l'avantage de s'appliquer aussi bien aux systèmes de réécriture de termes (TRS) qu'au λ-calcul. Cette notion de réduction standard, que nous nommerons **réduction abstraitement standard**, est définie de la façon suivante : la réduction $M_1 \xrightarrow{R_1}_v M_2 \xrightarrow{R_2}_v \dots \xrightarrow{R_{n-2}}_v M_{n-1} \xrightarrow{R_{n-1}}_v M_n$ est abstraitement standard si pour tout couple i,j d'entiers qui vérifient $1 \leq i < j \leq n$, il n'existe pas de réduction $M_i = N_i \xrightarrow{S_{i+1}}_v N_{i+1} \xrightarrow{S_{i+2}}_v N_{i+2} \xrightarrow{S_{i+3}}_v \dots \xrightarrow{S_{j-1}}_v N_{j-1} \xrightarrow{T_{j-1}}_v M_j$ telle que :

1. Pour k tel que $i \leq k \leq j-2$, et $R_i = T_i$:
   (a) T_k et S_{k+1} sont des radicaux disjoints de N_k.
   (b) T_{k+1} est un résidu de T_k dans N_{k+1}.
   (c) $N_k \xrightarrow{T_k}_v M_{k+1}$
   (d) R_k est un résidu de S_k dans M_{k+1}.
2. S_j contient T_{j-1}.

$$\begin{array}{ccccccccccc}
M_{i-1} & \xrightarrow{R_{i-1}}_v & M_i & \xrightarrow{S_{i+1}}_v & N_{i+1} & \xrightarrow{S_{i+2}}_v & N_{i+2} & \xrightarrow{S_{i+3}}_v & \cdots\cdots & \xrightarrow{S_{j-1}}_v N_{j-1} & \xrightarrow{S_j}_v \\
 & & \downarrow{\scriptstyle R_i} & & \downarrow{\scriptstyle T_{i+1}} & & \downarrow{\scriptstyle T_{i+2}} & & & \downarrow{\scriptstyle T_{j-1}} & \\
 & & M_{i+1} & \xrightarrow{R_{i+1}}_v & M_{i+2} & \xrightarrow{R_{i+2}}_v & M_{i+3} & \xrightarrow{R_{i+3}}_v & \cdots\cdots & \xrightarrow{R_{j-1}}_v M_j & \xrightarrow{R_j}_v M_{j+1}
\end{array}$$

Intuitivement, la réduction ci-dessus consiste à permuter les contractions des radicaux R_{i+1}, R_{i+2}, …, R_{j-1} avec la contraction de R_i. Si, en effectuant cette opération, le radical S_j (dont un résidu est R_j) contient T_{j-1} (qui est un résidu de R_i), alors cette réduction n'est pas abstraitement standard. En d'autres mots, cette réduction permet de faire passer un radical dont R_j est un résidu *au-dessus* d'un résidu de R_i. Le fait que S_j contienne T_{j-1} signifie intuitivement que S_j doit être contracté avant T_{j-1} pour obtenir une réduction standard. On montre que la définition de réduction standard pour $\leq_{st}$ est bien conforme à cette définition plus abstraite.

Lemme 2.10 *Une réduction standard pour $\leq_{st}$ est abstraitement standard.*

Preuve : On procède par l'absurde, en supposant qu'il existe une réduction entre M_i et M_j qui vérifie les propriétés décrites dans la définition de réduction abstraitement standard. On définit la propriété $\mathcal{P}(k)$ de la façon suivante :

1. Si le radical S de N_k est résidu d'un radical S' créé dans N_l pour $l \in \{i+1, \dots, k\}$ par la contraction de S_l, alors $T_k <_{st} S$.
2. $T_k <_{st} S_{k+1}$

Le premier point signifie intuitivement que les résidus des radicaux créés entre M_i et N_k sont plus grands, au sens de $<_{st}$ que le radical T_k. Le deuxième point signifie que T_k (un résidu de R_i) est plus petit au sens de $<_{st}$ que le radical S_{k+1} (dont R_{k+1} est un résidu). On montre que cette propriété est vérifiée pour $k \in \{i, \ldots, j-1\}$.

On commence par montrer $\mathcal{P}(i)$. Le radical R_{i+1} est résidu du radical S_{i+1}. Comme la réduction $M_i \xrightarrow{R_i}_v M_{i+1} \xrightarrow{R_{i+1}}_v M_{i+2}$ est standard pour $\leq_{st}$, alors on a $R_i \leq_{st} S_{i+1}$. Comme R_i et S_{i+1} sont disjoints, on obtient bien $R_i <_{st} S_{i+1}$.

Soit $k \in \{i, \ldots, j-2\}$. On suppose que la propriété $\mathcal{P}(k)$ est vraie. On veut montrer $\mathcal{P}(k+1)$. La situation est illustrée par le diagramme suivant.

$$\begin{array}{ccccccccccc}
M_{i-1} & \xrightarrow{R_{i-1}}_v & M_i & \xrightarrow{S_{i+1}}_v & \cdots\cdots & \xrightarrow{S_k}_v & N_k & \xrightarrow{S_{k+1}}_v & N_{k+1} & \xrightarrow{S_{k+2}}_v & \\
 & & \downarrow{\scriptstyle R_i} & & & & \downarrow{\scriptstyle T_k} & & \downarrow{\scriptstyle T_{k+1}} & & \\
 & & M_{i+1} & \xrightarrow{R_{i+1}}_v & \cdots\cdots & \xrightarrow{S_k}_v & M_{k+1} & \xrightarrow{R_{k+1}}_v & M_{k+2} & \xrightarrow{R_{k+2}}_v &
\end{array}$$

On montre tout d'abord le premier point. Soit S un radical de N_{k+1} qui est résidu d'un radical S' créé dans N_l (pour $l \in \{i+1, \ldots, k+1\}$) par la contraction de S_l. Deux cas sont à envisager.

1. Si $l \in \{i+1, \ldots, k\}$, alors S est résidu d'un radical S'' de N_k. Par $\mathcal{P}(k)$, on obtient $T_k <_{st} S''$. Le lemme 2.9 permet d'en déduire $T_{k+1} <_{st} S$.
2. Si $l = k+1$, alors S est créé par la contraction de S_k. Comme $T_k <_{st} S_k$, on en déduit par le lemme 2.9 $T_{k+1} <_{st} S$.

On examine le deuxième point de $\mathcal{P}(k+1)$. Deux cas de figure sont possibles pour S_{k+2}.

1. Si S_{k+2} est créé ou est le résidu d'un radical créé entre M_i et N_{k+1}, le point montré précédemment implique $T_{k+1} <_{st} S_{k+2}$.
2. Si S_{k+2} est résidu d'un radical S'_{k+2} présent dans M_i. Par conséquent, d'après le lemme 2.1, R_{k+2} est résidu de S'_{k+2}. La définition de réduction standard pour $\leq_{st}$ implique $R_i <_{st} S'_{k+2}$. De là, le lemme 2.9 donne $T_{k+1} <_{st} S_{k+2}$.

La propriété $\mathcal{P}(k)$ est donc vraie pour $k \in \{i, \ldots, j-1\}$. En particulier, on a $T_{j-1} <_{st} S_j$, ce qui contredit le fait que S_j contient T_{j-1}. □

2.1.4 Stabilité

On examine dans cette partie les propriétés de monotonie et de stabilité du λ-calcul par valeur. La preuve de la propriété de monotonie est adaptée de la preuve pour le λ-calcul. Pour prouver la propriété de stabilité, on utilise de façon cruciale la propriété de standardisation. Intuitivement, si $M \twoheadrightarrow_v N$, la réduction standard issue de M et aboutissant à N peut être vue comme une réduction minimale, qui ne contracte que les radicaux nécessaires pour atteindre N. La définition de la réduction standard permet de définir la réduction de tête qui est la réduction minimale permettant d'atteindre une valeur (si une valeur est atteignable). Cette réduction de tête ne fait intervenir qu'un préfixe de M. Ce préfixe est constitué des sous-termes qui contribuent à l'obtention d'une valeur.

Lemme 2.11 *Si $X \preccurlyeq Y$ et $V \preccurlyeq W$, alors $X\{x\backslash V\} \preccurlyeq Y\{x\backslash W\}$.*

Preuve : On procède par induction sur la structure de X.

1. Si $X = x$, on a bien $X\{x\backslash V\} = V \preccurlyeq W = Y\{x\backslash W\}$.
2. Si $X = y$ ou $X = \Omega$, le résultat est élémentaire.
3. Si $X = \lambda y.X_1$, on a nécessairement $Y = \lambda y.Y_1$ où $X_1 \preccurlyeq Y_1$. L'hypothèse d'induction donne $X_1\{x\backslash V\} \preccurlyeq Y_1\{x\backslash W\}$. De là, on a $X\{x\backslash V\} = \lambda y.X_1\{x\backslash V\} \preccurlyeq \lambda y.Y_1\{x\backslash W\} = Y\{x\backslash W\}$.

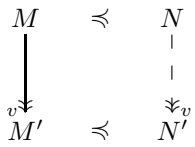


FIG. 2.8 – *Monotonie dans le λ-calcul par valeur*

4. Si $X = X_1X_2$, on conclut, comme précédemment, par hypothèse d'induction. □

La relation de préfixe est compatible avec la substitution. Cette propriété est cruciale pour montrer la réduction par valeur est monotone. Cette propriété de monotonie est illustrée sur la figure 2.8. La preuve de ce lemme est adaptée de la preuve dans le cas du λ-calcul.

Lemme 2.12 (Monotonie) *Si $M \preccurlyeq N$ et $M \twoheadrightarrow_v M'$, alors il existe un terme N' tel que $N \twoheadrightarrow_v N'$ et $M' \preccurlyeq N'$.*

Preuve : On considère dans un premier temps une réduction $M \to_v M'$ élémentaire. On procède induction sur la réduction $M \to_v M'$.

1. Si $M = (\lambda x.M_1)V_1 \to_v M_1\{x\backslash V_1\} = M'$, on a nécessairement $N = (\lambda x.N_1)W_1$ avec $M_1 \preccurlyeq N_1$ et $V_1 \preccurlyeq W_1$. De là, $N \to_v N_1\{x\backslash W_1\} = N'$. On conclut par le lemme 2.11.
2. Si $M = M_1M_2 \to_v M_1'M_2 = M'$ avec $M_1 \to_v M_1'$, on a nécessairement $N = N_1N_2$ avec $M_1 \preccurlyeq N_1$ et $M_2 \preccurlyeq N_2$. Par hypothèse d'induction, on obtient un terme N_1' tel que $N_1 \to_v N_1'$ avec $M_1' \preccurlyeq N_1'$. En posant $N' = N_1'N_2$, on obtient bien le résultat voulu : $N \to_v N'$ et $M' \preccurlyeq N'$.
3. Les autres cas se traitent comme précédemment en utilisant l'hypothèse d'induction.

Ce résultat se généralise immédiatement à une réduction en n étapes ($n \geq 0$). □

Si le terme M minore le terme N et si M se réduit vers M', alors le terme N se réduit vers un terme N' qui est minoré par M'. Pour montrer le théorème de stabilité, on utilise l'opération d'**intersection** entre deux préfixes X et X' d'un même terme. L'intersection de X et X' est le plus grand préfixe commun à X et X'. Plus formellement, cette opération est définie récursivement de la façon suivante.

$$\begin{aligned} X \cap \Omega &= \Omega & \qquad \Omega \cap X &= \Omega \\ x \cap x &= x & \qquad \lambda x.X \cap \lambda x.Y &= \lambda x.(X \cap Y) \\ XX' \cap YY' &= (X \cap Y)(X' \cap Y') \end{aligned}$$

L'intersection de deux abstractions est l'abstraction dont le corps est l'intersection des corps des abstractions. L'intersection d'une application est l'application dont le membre gauche (respectivement droit) est l'intersection des membres gauches (resp. droits) des applications. L'intersection est une opération interne sur les préfixes d'un terme M. Cette opération est réflexive, symétrique et admet M comme élément neutre et Ω comme élément absorbant. L'intersection vérifie la propriété suivante vis-à-vis des valeurs.

Lemme 2.13 *Si V et V' sont des valeurs qui vérifient $V \preccurlyeq W$ et $V' \preccurlyeq W$, le préfixe $V \cap V'$ est une valeur qui vérifie $V \cap V' \preccurlyeq W$.*

Preuve : Ce résultat est élémentaire. □

L'intersection de deux valeurs qui sont des préfixes d'une même valeur est une valeur. L'intersection vérifie également une propriété moins élémentaire de commutation avec la substitution.

Lemme 2.14 *Si X et Y sont des préfixes d'un même terme, et si V et W sont des valeurs qui sont des préfixes d'une même valeur, alors on a $X\{x\backslash V\} \cap Y\{x\backslash W\} = (X \cap Y)\{x\backslash V \cap W\}$.*

Preuve : On pose $Z = (X \cap Y)\{x\backslash V \cap W\}$. On procède par induction sur la structure de X.

1. Si $X = \Omega$, on a bien $(\Omega \cap Y)\{x\backslash V \cap W\} = \Omega = \Omega\{x\backslash V\} \cap Y\{x\backslash W\}$.
2. Le cas $X = y$ (où $y \neq x$) est élémentaire.
3. Si $X = x$, Y peut être de deux formes.
   (a) Si $Y = \Omega$, on est ramené au premier cas.
   (b) Si $Y = x$, on a $(X \cap Y)\{x\backslash V \cap W\} = V \cap W = X\{x\backslash V\} \cap Y\{x\backslash W\}$.
4. Si $X = X_1X_2$, Y peut être de deux formes.
   (a) Si $Y = \Omega$, on est ramené au premier cas.
   (b) Si $Y = Y_1Y_2$, on a $Z = (X_1 \cap Y_1)\{x\backslash V \cap W\}(X_2 \cap Y_2)\{x\backslash V \cap W\}$. Par hypothèse d'induction, on obtient l'égalité suivante.
   $$Z = (X_1\{x\backslash V\} \cap Y_1\{x\backslash W\})(X_2\{x\backslash V\} \cap Y_2\{x\backslash W\}) = X\{x\backslash V\} \cap Y\{x\backslash W\}$$
5. Si $X = \lambda y.X_1$, on obtient le résultat en utilisant, comme précédemment, l'hypothèse d'induction. □

La substitution dans l'intersection entre X et Y de la variable x par l'intersection entre V et W est l'intersection entre la substitution dans X de x par V et la substitution dans Y de x par W. Ce résultat est utilisé dans le lemme technique suivant qui examine les propriétés de la réduction standard.

Lemme 2.15

1. *Si la réduction $\mathcal{R} : (\lambda x.M)N \to_v (\lambda x.M')N \twoheadrightarrow_v P$ est standard, alors il existe un terme M_0 tel que $P = (\lambda x.M_0)N$.*
2. *Si la réduction $\mathcal{R} : (\lambda x.M)V \to_v (\lambda x.M)V' \twoheadrightarrow_v P$ est standard, alors il existe un terme M_0 et une valeur V_0 tels que $P = (\lambda x.M_0)V_0$.*
3. *Si N est une application et si la réduction $\mathcal{R} : NM \to_v NM' \twoheadrightarrow_v P$ est standard, alors il existe un terme M_0 tel que $P = NM_0$.*

Intuitivement, cette série de propriétés montre qu'au cours d'une réduction standard, si, à une étape, le radical minimal (pour $\leq_{st}$) du terme obtenu n'est pas réduit, un résidu de ce radical perdure jusqu'à la fin de la réduction standard.

Preuve : On montre successivement ces propriétés. Soit S le radical contracté au cours de la première réduction de $\mathcal{R}$. Dans tous les cas, on procède par récurrence sur la longueur n de $\mathcal{R}$. Le cas de base $n = 1$ est élémentaire. On suppose maintenant $n > 1$.

1. On pose $\mathcal{R}_1 : (\lambda x.M)N \to_v (\lambda x.M')N$. Le radical S est contenu dans M. Soit R' le radical contracté dans $(\lambda x.M')N$. On considère les trois cas possibles.
   (a) Si R' est créé par la contraction de S, alors R' est dans M'. On a donc la réduction $(\lambda x.M')N \xrightarrow{R'}_v (\lambda x.M'')N$. On conclut par hypothèse de récurrence.
   (b) Le cas où R' est un résidu d'un radical R de N est exclu par le fait que $\mathcal{R}$ est standard.
   (c) Si R' est un résidu d'un radical R de M, alors R' est dans M'. On conclut comme précédemment par hypothèse de récurrence.
2. On pose $\mathcal{R}_1 : (\lambda x.M)V \to_v (\lambda x.M)V'$. Le radical S est contenu dans V. Soit R' le radical contracté dans $(\lambda x.M)V'$. On considère les quatre cas possibles.
   (a) Le cas $R' = (\lambda x.M)V'$ est exclu par le fait que la réduction est standard.
   (b) Si R' est créé par la contraction de S, alors R' est dans V'. On a donc la réduction $(\lambda x.M)V' \xrightarrow{R'}_v (\lambda x.M)V''$. On conclut par hypothèse de récurrence.
   (c) Si R' est un résidu d'un radical R de M, on a $(\lambda x.M)V' \xrightarrow{R'}_v (\lambda x.M')V'$. On conclut par la propriété précédente.
   (d) Si R' est un résidu d'un radical R de V, alors R' est dans V'. On conclut comme précédemment par hypothèse de récurrence.

3. On pose $\mathcal{R}_1 : NM \to_v NM'$. Le radical S est contenu dans M. Soit R' le radical contracté dans NM'. On considère les trois cas possibles.
   (a) Si R' est créé par la contraction de S, alors R' est dans M'. On a donc la réduction $NM' \xrightarrow{R'}_v NM''$. On conclut par hypothèse de récurrence.
   (b) Le cas où R' est résidu d'un radical R de N est exclu par le fait que $\mathcal{R}$ est standard.
   (c) Si R' est résidu d'un radical R de M, alors on a $NM' \xrightarrow{R'}_v NM''$. On conclut par hypothèse de récurrence. □

Ce résultat nous amène à introduire la notion de **réduction en tête** qui est la réduction standard qui contracte à chaque étape le radical minimal pour l'ordre total $\leq_{st}$. L'intérêt de cette définition réside dans le résultat suivant.

Lemme 2.16 *Si la réduction $\mathcal{R} : M \twoheadrightarrow_v V$ est standard, alors il existe une réduction de tête $\mathcal{R}_t : M \twoheadrightarrow_v V'$ et une réduction standard $\mathcal{R}_s : V' \twoheadrightarrow_v V$ telles que $\mathcal{R} = \mathcal{R}_t; \mathcal{R}_s$.*

Preuve : Pour prouver ce résultat, il suffit de montrer que la réduction standard menant de M à la première valeur atteinte au cours de la réduction $\mathcal{R}$ est une réduction de tête. C'est pourquoi, on suppose désormais que V est la première valeur atteinte par $\mathcal{R}$ (les termes intermédiaires entre M et V ne sont pas des valeurs). On veut montrer que $\mathcal{R}$ est une réduction de tête. On procède par récurrence sur la longueur de $\mathcal{R}$.

Base Si $M = V$, on obtient directement le résultat voulu $\mathcal{R}_t = \mathcal{R} = \emptyset$.

Récurrence Si $\mathcal{R} : M \xrightarrow{S}_v M' \twoheadrightarrow_v V$ est une réduction de longueur $n+1$. On considère les différents cas possibles pour M.

1. Si $M = (\lambda x.M_1)W$. Comme $\mathcal{R}$ est standard et comme $\mathcal{R}$ aboutit à une valeur, le lemme 2.15 implique que le radical contracté est nécessairement M. La réduction $\mathcal{R}$ s'écrit donc $M \xrightarrow{M}_v M' = M_1\{x\backslash W\} \twoheadrightarrow_v V$. On décompose $\mathcal{R}$ en $\mathcal{R}_0; \mathcal{R}'$ en posant $\mathcal{R}_0 : M \xrightarrow{M}_v M_1\{x\backslash W\}$ et $\mathcal{R}' : M_1\{x\backslash W\} \twoheadrightarrow_v V$. La réduction standard $\mathcal{R}'$ est, par hypothèse de récurrence, une réduction de tête. La réduction $\mathcal{R}$ est donc également une réduction de tête.
2. Si $M = (\lambda x.M_1)M_2$ où M_2 n'est pas une valeur. Comme $\mathcal{R}$ est standard et aboutit à une valeur, en utilisant le lemme 2.15, on obtient que $\mathcal{R}$ s'écrit nécessairement $(\lambda x.M_1)M_2 \to_v (\lambda x.M_1)M_2^1 \to_v \ldots \to_v (\lambda x.M_1)M_2^k \to_v (\lambda x.M_1)W \twoheadrightarrow_v V$ où M_2^i pour $i \in \{1 \ldots k\}$ n'est pas une valeur. Par hypothèse de récurrence, la réduction standard de M_2 à W est une réduction de tête. La réduction $\mathcal{R}_0 : (\lambda x.M_1)M_2 \twoheadrightarrow_v (\lambda x.M_1)W$ est en conséquence une réduction de tête. En utilisant le cas précédent, on obtient que $\mathcal{R}' : (\lambda x.M_1)W \twoheadrightarrow_v V$ est une réduction de tête. La réduction $\mathcal{R} = \mathcal{R}_0; \mathcal{R}'$ est donc bien une réduction de tête.
3. Si $M = M_1M_2$ où M_1 est une application. Comme $\mathcal{R}$ est standard et aboutit à une valeur, en utilisant le lemme 2.15, on obtient que la réduction $\mathcal{R}$ s'écrit nécessairement $M_1M_2 \to_v M_1^1M_2 \to_v M_1^2M_2 \to_v \ldots \to_v M_1^kM_2 \to_v (\lambda x.M_3)M_2 \twoheadrightarrow_v V$ où, pour i tel que $1 \leq i \leq k$, le terme M_1^i est une application. Par hypothèse de récurrence, la réduction standard $M_1 \twoheadrightarrow_v \lambda x.M_3$ est une réduction de tête. On conclut en utilisant les points précédents. □

Une réduction standard peut se décomposer en la composition d'une réduction en tête et d'une réduction standard. Ceci signifie intuitivement qu'une réduction standard est une réduction de tête jusqu'au moment où le radical minimal pour $\leq_{st}$ n'est pas réduit. La réduction qui commence à ce moment est standard. Ce résultat de décomposition, associé aux propriétés de la réduction standard mentionnées dans le lemme 2.15, permet prouver que si M se réduit vers une valeur, alors M se réduit par une réduction en tête vers une valeur. Ce résultat est exploité dans le preuve du théorème de stabilité.

Théorème 2.5 (Stabilité) *Si $M \twoheadrightarrow_v V$, il existe un préfixe X de M tel que, pour tout Y, si $Y \preccurlyeq M$ et $Y \twoheadrightarrow_v V'$, on a $X \preccurlyeq Y$.*

Preuve : On considère la réduction $M \twoheadrightarrow_v V$. Soient X_1 et X_2 deux préfixes de M qui se réduisent vers une valeur. En utilisant le théorème de standardisation et le lemme 2.16, on sait qu'il existe des réductions $\mathcal{R}_0 : M \twoheadrightarrow_v V_0$, $\mathcal{R}_1 : X_1 \twoheadrightarrow_v V_1$ et $\mathcal{R}_2 : X_2 \twoheadrightarrow_v V_2$ qui sont des réductions en tête aboutissant à des valeurs et dont les termes intermédiaires ne sont pas des valeurs. On montre par récurrence sur la longueur n de la réduction $\mathcal{R}_0$ que V_1 et V_2 sont des préfixes de V_0 et que le préfixe $X = X_1 \cap X_2$ de M se réduit également vers la valeur $V_1 \cap V_2$.

Base Si $M = V_0$, le terme X_i (pour $i \in \{1,2\}$) est un préfixe d'une valeur. Par conséquent, deux cas sont possibles : le terme X_i est une valeur ou $X_i = \Omega$. Comme X_i se réduit vers une valeur, X_i est nécessairement une valeur et on a $X_i = V_i$. Par conséquent, en utilisant le lemme 2.13, $X_1 \cap X_2$ est également une valeur.

Récurrence On suppose que la réduction $\mathcal{R}_0 : M \to_v M' \twoheadrightarrow_v V_0$ est de longueur $n+1$. M n'est donc pas une valeur. On en déduit que X_i (pour $i \in \{1,2\}$) n'est pas une valeur. On écrit la réduction $\mathcal{R}_i : X_i \to_v X_i' \twoheadrightarrow_v V_i$. On procède par cas sur la structure de M.

1. Si $M = (\lambda x.N)W$, alors $M' = N\{x\backslash W\}$. Comme X_i (pour $i \in \{1,2\}$) est un préfixe de M qui se réduit vers une valeur, on a nécessairement $X_i = (\lambda x.Y_i)W_i$ avec $Y_i \preccurlyeq N$ et $W_i \preccurlyeq W$ et où W_i est une valeur. Comme $X_i \twoheadrightarrow_v V_i$ est une réduction de tête, on a $X_i' = Y_i\{x\backslash W_i\}$. Le lemme 2.11 donne $X_i' \preccurlyeq M'$. On a $X_1 \cap X_2 = (\lambda x.Y_1 \cap Y_2)(W_1 \cap W_2)$ où, d'après le lemme 2.15, $W_1 \cap W_2$ est une valeur préfixe de W. On obtient la réduction $X_1 \cap X_2 \to_v (Y_1 \cap Y_2)\{x\backslash W_1 \cap W_2\}$. Le lemme 2.14 donne $(Y_1 \cap Y_2)\{x\backslash W_1 \cap W_2\} = X_1' \cap X_2'$. Par hypothèse d'induction, on obtient que $X_1' \cap X_2'$ se réduit vers une valeur. On en déduit que $X_1 \cap X_2$ se réduit vers une valeur.

2. Si $M = (\lambda x.N)P$ où P n'est pas une valeur, la réduction $\mathcal{R}_0$ s'écrit nécessairement $M = (\lambda x.N)P \twoheadrightarrow_v (\lambda x.N)W_0 \twoheadrightarrow_v V_0$ où $\mathcal{R}_0' : P \twoheadrightarrow_v W_0$ est une réduction en tête. Comme X_i (pour $i \in \{1,2\}$) est un préfixe de M qui se réduit vers une valeur, X_i est nécessairement de la forme $X_i = (\lambda x.Y_i)Z_i$ où $Y_i \preccurlyeq N$ et $Z_i \preccurlyeq P$. Comme $\mathcal{R}_i$ est une réduction en tête qui aboutit à une valeur, cette réduction s'écrit nécessairement $\mathcal{R}_i : (\lambda x.Y_i)Z_i \twoheadrightarrow_v (\lambda x.Y_i)W_i \twoheadrightarrow_v V_i$ où W_i est une valeur et $\mathcal{R}_i' : Z_i \twoheadrightarrow_v W_i$ est une réduction en tête dont les termes intermédiaires ne sont pas des valeurs. La réduction $\mathcal{R}_0'$ est de longueur inférieure à n. En utilisant l'hypothèse de récurrence, on obtient, pour $i \in \{1,2\}$, la relation $W_i \preccurlyeq W$. Et on obtient la réduction en tête $Z_1 \cap Z_2 \twoheadrightarrow_v W_1 \cap W_2$. Par conséquent, on a $X_1 \cap X_2 = (\lambda x.Y_1 \cap Y_2)(Z_1 \cap Z_2) \twoheadrightarrow_v (\lambda x.Y_1 \cap Y_2)(W_1 \cap W_2)$. En utilisant le lemme 2.14, on obtient $(\lambda x.Y_1 \cap Y_2)(W_1 \cap W_2) = ((\lambda x.Y_1)W_1) \cap ((\lambda x.Y_2)W_2)$. Pour conclure, on applique l'hypothèse de récurrence sur $\mathcal{R}_0'' : (\lambda x.N)W_0 \twoheadrightarrow_v V_0$ et $\mathcal{R}_i'' : (\lambda x.Y_i)W_i \twoheadrightarrow_v V_i$ (pour $i \in \{1,2\}$).

3. Le cas $M = NP$ où N est une application se traite de façon similaire aux cas précédents.

On a montré l'existence d'un plus petit préfixe X de M parmi les préfixes de M qui se réduisent vers une valeur. Le lemme de monotonie 2.12 implique que tout préfixe plus grand que X se réduit également vers une valeur. □

Si un terme M se réduit vers une valeur V, alors il existe un préfixe minimal X de M qui se réduit vers une valeur. Par monotonie, tous les préfixes de M minoré par X se réduisent également vers une valeur. Le λ-calcul par valeur vérifie donc, comme le λ-calcul, la théorème de stabilité.

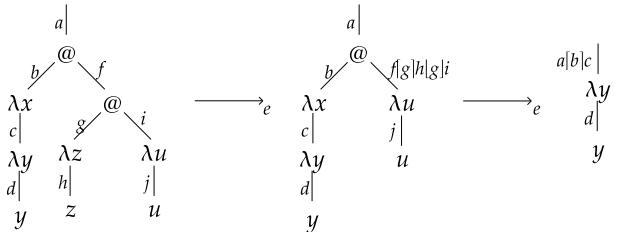


FIG. 2.9 – *Réduction par valeur de* $M = ((\lambda x.(\lambda y.y^d)^c)^b\ ((\lambda z.z^h)^g\ (\lambda u.u^j)^i)^f)^a$

2.2 Le λ-calcul par valeur étiqueté

Les étiquettes et la réduction étiquetée du λ-calcul permettent de conserver toutes les propriétés du λ-calcul : le calcul étiqueté est confluent ; les théorèmes des développements finis et de standardisation sont vérifiés. De plus, le résultat 1.15 montre que les étiquettes expriment la propriété de stabilité : si un terme M (qui vérifie $\texttt{INIT}(M)$) se réduit vers une valeur V, l'étiquette de tête $\tau(V)$ permet de caractériser le préfixe minimum de M qui se réduit vers une valeur. Ce préfixe est obtenu à partir de M en effaçant les sous-termes étiquetés par une lettre n'appartenant pas à l'étiquette $\tau(V)$.

La réduction par valeur $\rightarrow_v$ vérifie les mêmes propriétés fondamentales que la réduction classique $\rightarrow$ du λ-calcul. Comme nous l'avons vu dans la section précédente, le λ-calcul par valeur est confluent et vérifie les théorèmes des développements finis, de standardisation et de stabilité. On peut se demander si ces propriétés sont conservées dans le λ-calcul étiqueté par valeur (où les radicaux contractés sont de la forme $((\lambda x.M)^\alpha V)^\beta$). Il est clair que les propriétés de confluence ou des développements finis se déduisent, comme dans la section précédente, de manière assez directe des propriétés correspondantes du λ-calcul étiqueté. Comme pour le λ-calcul par valeur, la propriété de standardisation nécessite une adaptation de la définition d'une réduction standard. Cependant, la nature de ce problème est la même dans le λ-calcul par valeur, avec ou sans étiquettes. En revanche, les étiquettes du λ-calcul étiqueté n'expriment plus la stabilité, comme le montre la réduction suivante.

$$\begin{array}{rcl} M = ((\lambda x.(\lambda y.y^d)^c)^b\ ((\lambda z.z^h)^g\ (\lambda u.u^j)^i)^f)^a & \rightarrow_e & ((\lambda x.(\lambda y.y^d)^c)^b\ (\lambda u.u^j)^{f\lceil g\rceil h\lfloor g\rfloor i})^a \\ & \rightarrow_e & (\lambda y.y^d)^{a\lceil b\rceil c} \end{array}$$

Au cours de cette réduction, qui est illustrée sur la figure 2.9, les radicaux $R_1 = ((\lambda z.z^h)^g(\lambda u.u^j)^i)^f$ et $R_2 = ((\lambda x.(\lambda y.y^d)^c)^b\ (\lambda u.u^j)^{f\lceil g\rceil h\lfloor g\rfloor i})^a$ sont successivement contractés. Ces radicaux sont bien des radicaux par valeur. En réalité, R_1 est le seul radical par valeur de M alors que R_2 est le seul radical par valeur dans le terme obtenu. Cette réduction est donc la seule réduction par valeur qui aboutit à une valeur. Le préfixe associé à l'étiquette de tête de la valeur obtenue est $X = ((\lambda x.(\lambda y.\Omega)^c)^b\,\Omega)^a$. Conformément au résultat 1.15, ce préfixe se réduit bien, dans le λ-calcul étiqueté, vers une valeur.

$$X = ((\lambda x.(\lambda y.\Omega)^c)^b\,\Omega)^a \rightarrow_e (\lambda y.\Omega)^{a\lceil b\rceil c}$$

En revanche, comme Ω n'est pas une valeur, le terme X ne contient pas de radical par valeur ; X est une forme normale si on considère la réduction par valeur. On en déduit que les étiquettes du λ-calcul n'expriment pas la stabilité pour la réduction par valeur. Ceci s'explique par le fait que la notion de radical par valeur est plus forte : un radical par valeur est un radical (au sens général) dont le membre droit est une valeur. Pour obtenir une valeur à partir de M, il est nécessaire de réduire l'application en tête de M. Comme le membre droit de cette application n'est pas une valeur, il est

nécessaire de calculer dans ce membre droit afin d'obtenir une valeur et donc un radical en tête de M. Plus synthétiquement, on observe que le calcul dans le membre droit contribue crucialement à l'obtention de la valeur finale : ce membre droit est donc présent dans le radical minimum de M qui aboutit à une valeur. Plus précisément, ce préfixe est $Y = ((\lambda x.(\lambda y.\Omega)^c)^b\ ((\lambda z.z^g)^g\ (\lambda u.\Omega)^i)^f)^a$. Ce terme se réduit de la façon suivante.

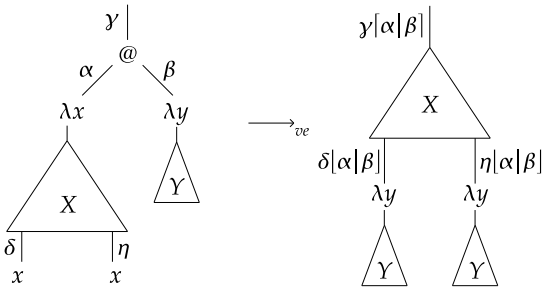
$$Y \to_e ((\lambda x.(\lambda y.\Omega)^c)^b\ (\lambda u.\Omega)^{f\lceil g\rceil h\lfloor g\rfloor i}) \to_e (\lambda y.\Omega)^{a\lceil b\rceil c}$$

Cette réduction confirme bien que les étiquettes n'expriment pas la propriété de stabilité pour la réduction par valeur.

Dans cette section, nous présentons le λ-calcul par valeur étiqueté. A l'image du λ-calcul étiqueté, ce langage conserve les propriétés fondamentales du λ-calcul par valeur. Dans la partie 2.2.1, nous montrons que ce langage est localement confluent et confluent. Dans la partie 2.2.2, nous prouvons le théorème des développements finis. La preuve utilisée est intuitive et fait appel à une relation d'imbrication étendue entre radicaux d'un ensemble de radicaux. Le théorème de standardisation est obtenu dans la partie 2.2.3 en utilisant la preuve employée dans la section précédente. Enfin, nous montrons dans la partie 2.2.4 que les étiquettes expriment la stabilité.

Dans la section précédente, nous avons vu que la notion de création de radical était modifiée dans le λ-calcul par valeur par rapport au λ-calcul. Dans le λ-calcul, la contraction d'un radical crée un radical si et seulement si, du fait de cette contraction, le membre gauche d'une application devient une abstraction. Dans le λ-calcul par valeur, deux conditions sont nécessaires et suffisantes pour qu'une application soit un radical : une application est un radical si et seulement si (1) le membre gauche est une abstraction et (2) si le membre droit est une valeur. Par conséquent, la contraction d'un radical peut contribuer à la création d'un radical en remplissant l'une de ces deux conditions. On suppose $M \xrightarrow{R}_v M'$. On examine d'abord le cas correspondant à la condition (1) ci-dessus. On considère un sous-terme NV de M. Le membre droit de cette application est une valeur. On suppose que son membre gauche n'est pas une abstraction. Si, du fait de la contraction de R, le membre gauche N devient une abstraction $\lambda x.P$, alors R crée un nouveau radical $(\lambda x.P)V'$ dans M' (la valeur V a pu subir une substitution). Ce cas de création correspond au cas habituel de création des radicaux dans le λ-calcul. On examine maintenant le cas correspondant à la condition (2). On considère un sous-terme $(\lambda x.N)P$ de M. On suppose que P n'est pas une valeur. Dans le λ-calcul, ce sous-terme est un radical alors que ce n'est pas le cas pour la réduction par valeur. On suppose que, du fait de la contraction de R, le sous-terme P devient une valeur V. Dans ce cas, le radical R crée un nouveau radical $(\lambda x.N')V$ dans M'. Ce cas de création n'existe pas dans le λ-calcul. Dans le λ-calcul par valeur, la création d'un radical ne dépend donc pas seulement du calcul dans le membre gauche de l'application comme c'est le cas dans le λ-calcul. Il dépend aussi du calcul dans le membre droit, puisque l'obtention d'une valeur est nécessaire pour former un radical. Dans le chapitre précédent, nous avons noté que les étiquettes du λ-calcul expriment les dépendances vis-à-vis des sous-termes du terme initial. En particulier, le nom d'un radical créé contient strictement les noms des radicaux qui ont contribué à sa création. Pour définir les étiquettes du λ-calcul par valeur, on rend compte du fait que la création d'un radical $((\lambda x.M)^\alpha V)^\beta$ dépend à la fois de l'histoire de l'abstraction $(\lambda x.M)^\alpha$ et de l'histoire de la valeur V. Les étiquettes du λ-calcul par valeur expriment cette double dépendance.

$$\begin{array}{rcll}
\alpha, \beta, \gamma & ::= & a & \text{Lettre} \\
& | & \alpha\beta & \text{Concaténation} \\
& | & \lceil \alpha' \rceil & \text{Surlignement} \\
& | & \lfloor \alpha' \rfloor & \text{Soulignement} \\
\alpha', \beta' & ::= & \alpha | \beta & \text{Nom}
\end{array}$$



$$((\lambda x.X)^{\alpha}V)^{\gamma} \to_{ve} \gamma \cdot \lceil \alpha|\beta \rceil \cdot X\{x \backslash \lfloor \alpha|V \rfloor\} \text{ où } V = (\lambda y.Y)^{\beta} \text{ et } \beta = \tau(V)$$

FIG. 2.10 – *Réduction dans le λ-calcul par valeur étiqueté*

Comme dans le λ-calcul étiqueté, une étiquette peut être une lettre a ou la concaténation $\alpha\beta$ de deux étiquettes α et β. Dans le λ-calcul étiqueté classique, une étiquette peut être aussi une étiquette surlignée $\lceil \alpha \rceil$ ou soulignée $\lfloor \alpha \rfloor$. Ces étiquettes sont créées par la contraction d'un radical de nom α. Intuitivement, l'étiquette α représente l'histoire du calcul ayant contribué à la création du radical, c'est-à-dire à l'obtention d'une abstraction à gauche d'une application. Par contraste, dans le λ-calcul par valeur, la création d'un radical dépend à la fois du calcul de l'abstraction (à gauche) et du calcul de la valeur (à droite) du radical. Cette double dépendance nous amène à modifier la notion de *nom* de radical. Dans le cas présent, un nom $\alpha|\beta$ fait à la fois intervenir l'histoire α du calcul de l'abstraction et l'histoire β du calcul de la valeur à droite du radical par valeur. Comme nous le verrons plus tard, la contraction d'un radical de nom α' crée un surlignement $\lceil \alpha' \rceil$ et des soulignements $\lfloor \alpha' \rfloor$ de ce nom. Ces surlignements ou soulignements peuvent donc aussi s'écrire sous la forme $\lceil \alpha|\beta \rceil$ ou $\lfloor \alpha|\beta \rfloor$. A l'exception des étiquettes, la syntaxe du λ-calcul étiqueté est reprise. En particulier, les syntaxes des termes, des préfixes, des contextes et des valeurs du λ-calcul par valeur étiqueté sont définies de la façon suivante.

$$\begin{array}{lrl}
\text{Termes} & M, N \in \mathbf{\Lambda}_{ve} & ::= x^{\alpha} \mid (\lambda x.M)^{\alpha} \mid (MN)^{\alpha} \\
\text{Préfixes} & X, Y & ::= (\lambda x.X)^{\alpha} \mid (XY)^{\alpha} \mid x^{\alpha} \mid \Omega \\
\text{Contextes} & C & ::= [\,] \mid (\lambda x.C)^{\alpha} \mid (CX)^{\alpha} \mid (XC)^{\alpha} \\
\text{Valeurs} & V & ::= (\lambda x.X)^{\alpha}
\end{array}$$

Les définitions de l'opérateur point "$\cdot$", de la substitution et de l'étiquette de tête s'adaptent de façon immédiate à la syntaxe du λ-calcul par valeur étiqueté. La réduction étiquetée de ce langage est définie de la façon suivante.

$$(\beta_{ve}) \qquad ((\lambda x.X)^{\alpha}V)^{\gamma} \to_{ve} \gamma \cdot \lceil \alpha|\beta \rceil \cdot X\{x \backslash \lfloor \alpha|V \rfloor\} \text{ où } \beta = \tau(V)$$

$$\mathsf{nom}(((\lambda x.X)^{\alpha}V)^{\gamma}) = \alpha|\tau(V)$$

$$\lfloor \alpha|(\lambda x.X)^{\beta} \rfloor = (\lambda x.X)^{\lfloor \alpha|\beta \rfloor}$$

Cette réduction est similaire à la réduction étiquetée du λ-calcul. La différence réside dans le nom du radical qui est surligné ou souligné. Dans le λ-calcul, ce nom est l'étiquette de l'abstraction. Dans le λ-calcul par valeur, ce nom $\alpha|\beta$ est constitué de l'étiquette α de l'abstraction et de l'étiquette de tête β de la valeur. Ce nom reflète la dépendance double évoquée auparavant. Concrètement, comme le montre la figure 2.10 dans le cas où $V = (\lambda y.Y)^{\beta}$, la (β_{ve})-réduction encadre le corps M de l'abstraction par le nom du radical. Ce nom est surligné en haut de M et souligné à l'emplacement des occurrences des variables x substituées. La notation $\lfloor \alpha|V \rfloor$ est un raccourci syntaxique commode ; le terme $\lfloor \alpha|V \rfloor$ est la valeur V dans laquelle l'étiquette de tête β

est remplacée par $\lfloor\alpha|\beta\rfloor$. Dans le cadre actuel, où les seules valeurs sont les abstractions, il peut sembler inutile d'introduire cette notation. Cependant, cette notation a l'avantage de la modularité si on ajoute d'autres types de termes au λ-calcul par valeur, par exemple les entiers n^β. Ces entiers seraient considérés comme des valeurs et on aurait $\lfloor\alpha|n^\beta\rfloor = n^{\lfloor\alpha|\beta\rfloor}$. Cette modularité permettrait de garder la règle de réduction inchangée. Les règles de contexte du λ-calcul par valeur étiqueté s'adaptent simplement, de la façon suivante.

$$(\nu_{ve})\ \frac{X \to_{ve} X'}{(XY)^\alpha \to_{ve} (X'Y)^\alpha} \qquad (\mu_{ve})\ \frac{Y \to_{ve} Y'}{(XY)^\alpha \to_{ve} (XY')^\alpha} \qquad (\xi_{ve})\ \frac{X \to_{ve} X'}{(\lambda x.X)^\alpha \to_{ve} (\lambda x.X')^\alpha}$$

De même, la notion de résidu dans le λ-calcul par valeur étiqueté est une simple synthèse de la notion de résidu dans le λ-calcul par valeur et dans le λ-calcul étiqueté. On suppose $X \xrightarrow{r}_{ve} Y$ où $r = (C,R)$. Soit $s = (C',S)$ un radical de X. On note (C,R') le contractum de r.

Si $r = s$. Alors s n'a pas de résidu dans Y.

Si r et s sont disjoints. Si, par exemple, r est à gauche de s, on a $X = C_0[(X_1X_2)^\alpha]$ avec $C = C_0[(C_1X_2)^\alpha]$ et $C' = C_0[(X_1C_2)^\alpha]$. Le résidu de s dans Y est $(C_0[(C_1[R']C_2)^\alpha],S)$.

Si s contient r. On a $S = C_1[R]$ où $C_1 \neq [\,]$. Par conséquent, $s' = (C',C_1[R'])$ est un radical de Y et s' est le radical résidu de s dans Y.

Si r contient s. On pose $R = ((\lambda x.Z)^\alpha V)^\gamma$ où $\beta = \tau(V)$ et donc $R' = \gamma \cdot \lceil\alpha|\beta\rceil \cdot Z\{x\backslash\lfloor\alpha|V\rfloor\}$.

1. Si $C' = C[((\lambda x.C_1)^\alpha V)^\gamma]$, on pose $C'' = C[\gamma \cdot \lceil\alpha|\beta\rceil \cdot C_1\{x\backslash\lfloor\alpha|V\rfloor\}]$.
   (a) Si $C_1 \neq [\,]$, $(C'',S\{x\backslash\lfloor\alpha|V\rfloor\})$ est le résidu de s dans Y.
   (b) Si $C_1 = [\,]$, on a $C'' = C$ et $(C,\gamma \cdot \lceil\alpha|\beta\rceil \cdot S\{x\backslash\lfloor\alpha|V\rfloor\})$ est le résidu de s dans Y.
2. Si $C' = C[((\lambda x.Z)^\alpha C_1[\,])^\gamma]$. On a nécessairement $C_1[\,] = (\lambda y.C_1'[\,])^\beta$. Si la variable x n'a pas d'occurrence libre dans Z, s n'a pas de résidu dans Y. Si la variable x a une occurrence libre dans Z caractérisée par le contexte $C_2[\,]$ avec $Z = C_2[x^\delta]$.
   (a) Si $C_1 \neq [\,]$ et $C_2 = [\,]$, on pose $C'' = C[\gamma \cdot \lceil\alpha|\beta\rceil \cdot \delta \cdot (\lambda y.C_1'[\,])^{\lfloor\alpha|\beta\rfloor}]$ et (C'',S) est le résidu de s dans Y.
   (b) Sinon, on pose $C'' = C[\gamma \cdot \lceil\alpha|\beta\rceil \cdot C_2\{x\backslash\lfloor\alpha|V\rfloor\}[\delta \cdot (\lambda y.C_1'[\,])^{\lfloor\alpha|\beta\rfloor}]]$ et (C'',S) est un résidu de s dans Y.

On observe que le terme résidu S' de S est inchangé ($S' = S$) ou bien a subi une substitution $S' = S\{x\backslash\lfloor\alpha|V\rfloor\}$ voire une concaténation en tête : $S' = \gamma\cdot\lceil\alpha|\beta\rceil\cdot S\{x\backslash\lfloor\alpha|V\rfloor\}$. C'est ici que le fait d'avoir exclu les variables des valeurs se justifie : pour se rapprocher du λ-calcul étiqueté, on souhaite conserver la propriété fondamentale suivante : si R' est un résidu de R, alors ces radicaux ont le même nom. Si nous avions décidé de considérer les variables comme des valeurs, cette propriété aurait été perdue comme le montre le terme suivant : $M = ((\lambda x.((\lambda y.y^f)^d x^g)^c)^b(\lambda z.z^i)^h)^a$. Si on inclut les variables dans l'ensemble des valeurs, ce terme contient deux radicaux $R_1 = M$ et $R_2 = ((\lambda y.y^f)^d x^g)^c$. En contractant R_1, on obtient le terme $M' = ((\lambda y.y^f)^d(\lambda z.z^i)^{g\lfloor b|h\rfloor})^{a\lceil b|h\rceil c}$. Ce terme contient un résidu $R_2' = M'$ de R_2. On note que ces radicaux ont des noms différents : on a $\mathsf{nom}(R_2) = d|g$ et $\mathsf{nom}(R_2') = d|g\lfloor b|h\rfloor$. En effet, la contraction de R_1 a influé sur le membre droit du radical R_1 du fait d'une substitution. Cette substitution a modifié l'histoire de cette valeur et donc le nom du radical résidu. Ceci va à l'encontre de l'intuition de valeur. Comme nous l'avons mentionné dans le premier chapitre, l'ensemble des valeurs est stable par réduction et substitution. L'exemple mentionné ci-dessus nous amène à introduire une contrainte plus forte : l'étiquette de tête d'une valeur, qui représente intuitivement son histoire, est stable par réduction et substitution. L'ensemble des abstractions respecte cette contrainte. En incluant les variables, cette contrainte n'aurait pas été vérifiée.

Lemme 2.17 *On suppose $V = (\lambda x.X)^\alpha$. Si $V \to_{ve} V'$ et si $V'' = V\{x\backslash Y\}$, alors V' et V'' sont des valeurs qui vérifient $\tau(V) = \tau(V') = \tau(V'') = \alpha$.*

Preuve : Par inspection des différents cas. □

Cette propriété permet d'obtenir la propriété suivante qui est une propriété centrale du λ-calcul par valeur étiqueté.

Lemme 2.18 (Conservation du nom des radicaux) *Si $X \rightarrow_{ve} X'$ et si R' est un résidu dans X' d'un radical R de X, les noms de ces radicaux vérifient* $\mathsf{nom}(R) = \mathsf{nom}(R')$.

Preuve : Cette propriété s'obtient en utilisant le lemme 2.17 et en inspectant les différents cas de figure mentionnés dans la définition de radical résidu dans le λ-calcul étiqueté. □

Après une réduction, le nom d'un radical résidu est identique au nom du radical dont il est le résidu. Ce lemme est l'analogue du résultat 1.9 dans le cadre du λ-calcul étiqueté.

Remarque A ce stade de la réflexion, il est légitime de se demander s'il est toujours nécessaire de souligner ou surligner le nom de la valeur d'un radical au moment de sa contraction. Pour illustrer cette question, on considère la réduction du terme $M = ((\lambda x.x^c)^b(\lambda y.y^f)^d)^a$. Ce terme appartient à la fois au λ-calcul par valeur étiqueté et au λ-calcul étiqueté. Dans ces langages, ce terme se réduit de la façon suivante.

$$M \rightarrow_{ve} (\lambda y.y^f)^{a\lceil b|d\rceil c\lfloor b|d\rfloor} \qquad\qquad M \rightarrow_{e} (\lambda y.y^f)^{a\lceil b\rceil c\lfloor b\rfloor d}$$

On observe que les étiquettes de tête des résultats contiennent bien les mêmes lettres. Dans cet exemple où la valeur $(\lambda y.y^f)^d$ à droite du radical M contracté intervient directement dans le résultat, il peut sembler inutile de répéter l'étiquette de cette valeur dans le surlignement. En poursuivant ce raisonnement, on pourrait disposer de deux réductions différentes pour les radicaux $((\lambda x.X)^\alpha V)^\gamma$ selon que la variable liée x est présente ou non dans le corps X de l'abstraction. En réalité, ce raisonnement n'aboutit pas puisque même si une variable x est présente dans X, cette variable (où la valeur qui le remplace) pourrait disparaître du fait de la contraction d'autres radicaux. La question de l'intervention de la variable x dans le résultat final est indécidable. Ceci justifie l'intégration systématique de l'étiquette de la valeur dans le nom du radical surligné ou souligné.

2.2.1 Confluence

On examine dans cette partie les propriétés de confluence locale et de confluence du λ-calcul par valeur étiqueté. Comme le λ-calcul par valeur vérifie ces propriétés, une réduction étiquetée adaptée à ce langage devrait conserver ces propriétés fondamentales. On prouve dans un premier temps la confluence locale avant d'examiner la confluence. Ces preuves se basent en partie sur les propriétés syntaxiques suivantes, vérifiées par l'opération "$\cdot$", et bien connues dans le cadre du λ-calcul étiqueté.

Lemme 2.19 *1.* $\alpha \cdot (\beta \cdot X) = \alpha\beta \cdot X$
2. $(\alpha \cdot X)\{y\backslash Y\} = \alpha \cdot (X\{y\backslash Y\})$
3. $\lfloor\alpha|V\rfloor\{x\backslash X\} = \lfloor\alpha|V\{x\backslash X\}\rfloor$

Preuve : Le premier point se prouve par cas sur X. On pose $X_1 = \alpha \cdot (\beta \cdot X)$ et $X_2 = (\alpha\beta) \cdot X$.

1. Si $X = x^\gamma$, on a $X_1 = \alpha \cdot x^{\beta\gamma} = x^{\alpha\beta\gamma}$ et $X_2 = x^{\alpha\beta\gamma}$.
2. Le cas $X = \Omega$ est élémentaire. Les autres cas sont similaires aux cas précédents.

Le deuxième point se prouve par induction sur X. On pose $X_1 = (\alpha \cdot X)\{y\backslash Y\}$ et $X_2 = \alpha \cdot (X\{y\backslash Y\})$.

1. Si $X = x^\beta$, on a $X_1 = x^{\alpha\beta} = X_2$.
2. Si $X = y^\beta$, alors on a $X_1 = y^{\alpha\beta}\{y\backslash Y\} = \alpha\beta \cdot Y = \alpha \cdot \beta \cdot Y = X_2$.
3. Le cas $X = \Omega$ est élémentaire. Les cas $X = (\lambda z.Z)^\beta$ et $X = (Z_1 Z_2)^\beta$ se traitent directement par hypothèse d'induction.

Pour le troisième point, on observe que la valeur V est une abstraction $(\lambda y.Y)^\beta$. Après un renommage éventuel, on peut supposer $x \neq y$. De là, on obtient $\lfloor\alpha|(\lambda y.Y)^\beta\rfloor\{x\backslash X\} = (\lambda y.Y\{x\backslash X\})^{\lfloor\alpha|\beta\rfloor}$ et $\lfloor\alpha|(\lambda y.Y)^\beta\{x\backslash X\}\rfloor = (\lambda y.Y\{x\backslash X\})^{\lfloor\alpha|\beta\rfloor}$. □

Le premier point de ce lemme énonce le fait que concaténer à X successivement deux étiquettes α et β avec l'opération "$\cdot$" revient à concaténer l'étiquette concaténée $\alpha\beta$. Le deuxième point montre que les opérations de substitution et de concaténation "$\cdot$" peuvent être permutées. Le troisième point est une variante du deuxième point : les opérations de substitution et de soulignement par l'étiquette α peuvent être permutées. On note que ce point aurait été faux si les variables avaient été des valeurs, comme le montre l'exemple suivant : les termes $\lfloor\alpha|x^\beta\rfloor\{x\backslash y^\gamma\} = y^{\lfloor\alpha|\beta\rfloor\gamma}$ et $\lfloor\alpha|x^\beta\{x\backslash y^\gamma\}\rfloor = y^{\lfloor\alpha|\beta\gamma\rfloor}$ sont bien différents. Ces propriétés élémentaires permettent d'obtenir un résultat d'interversion des substitutions analogue au lemme 2.4.

Lemme 2.20 *Si $x \neq y$ et si $x \notin \mathtt{FV}(Y)$, alors $Z\{x\backslash X\}\{y\backslash Y\} = Z\{y\backslash Y\}\{x\backslash X\{y\backslash Y\}\}$*

Preuve : On pose $Z_1 = Z\{x\backslash X\}\{y\backslash Y\}$ et $Z_2 = Z\{y\backslash Y\}\{x\backslash X\{y\backslash Y\}\}$. On procède par récurrence sur la taille de Z.

1. Si $Z = x^\alpha$, on a $Z_1 = (\alpha \cdot X)\{y\backslash Y\}$. Comme $x \neq y$, on obtient $Z_2 = \alpha \cdot (X\{y\backslash Y\})$. Le lemme 2.19 permet de conclure.
2. Si $Z = y^\alpha$, on a $Z_1 = \alpha \cdot Y$ et $Z_2 = \alpha \cdot Y\{x\backslash X\{y\backslash Y\}\}$. Comme $x \notin \mathtt{FV}(Y)$, on a $Z_2 = \alpha \cdot Y$.
3. Les cas $Z = z^\alpha$ où $z \notin \{x,y\}$ et $Z = \Omega$ sont élémentaires. Les autres cas se traitent par hypothèse de récurrence □

Ce résultat d'interversion des substitutions est un résultat fondamental : intuitivement, il signifie que, sous certaines conditions, changer l'ordre de deux substitutions ne modifie pas le résultat obtenu. Comme nous le verrons par la suite, ceci implique que si deux réductions élémentaires sont possibles, il est possible de faire confluer les résultats de ces réductions vers un terme commun. La formalisation de cette intuition se fait à l'aide des trois résultats intermédiaires suivants.

Lemme 2.21 *1. Si $X \to_{ve} X'$ alors $\alpha \cdot X \to_{ve} \alpha \cdot X'$.*
2. Si $V \to_{ve} V'$ alors $\lfloor\alpha|V\rfloor \to_{ve} \lfloor\alpha|V'\rfloor$.

Preuve : Pour montrer le premier point, on procède par cas sur la réduction $X \to_{ve} X'$.

1. Si $X = ((\lambda y.Y)^\beta V)^\delta$ où $\tau(V) = \gamma$ et $X \to_{ve} \delta \cdot \lceil\beta|\gamma\rceil \cdot Y\{y\backslash\lfloor\beta|V\rfloor\} = X'$, on a, en utilisant le lemme 2.19, $\alpha \cdot X = ((\lambda y.Y)^\beta V)^{\alpha\delta} \to_{ve} (\alpha\delta) \cdot \lceil\beta|\gamma\rceil \cdot Y\{y\backslash\lfloor\beta|V\rfloor\} = \alpha \cdot X'$
2. Si $X = (X_1X_2)^\beta$ et $X_2 \to_{ve} X_2'$ et $X' = (X_1X_2')^\beta$, alors on obtient directement la réduction $\alpha \cdot X = (X_1X_2)^{\alpha\beta} \to_{ve} (X_1X_2')^{\alpha\beta} = X'$.
3. Les cas $X = (X_1X_2)^\beta \to_{ve} (X_1'X_2)^\beta$ et $X = (\lambda x.X_1)^\beta \to_{ve} (\lambda x.X_1')^\beta$ où $X_1 \to_{ve} X_1'$ sont similaires au cas précédent.

Le deuxième point est prouvé de manière similaire. □

Comme dans le cas du λ-calcul étiqueté [29], le premier point de ce lemme énonce le fait que la concaténation par l'opération "$\cdot$" est compatible avec la réduction étiquetée du λ-calcul par valeur. Le deuxième point énonce la compatibilité de l'opération de soulignement avec la réduction étiquetée du λ-calcul par valeur. Ce résultat s'interprète intuitivement de la façon suivante : si $X \to_{ve} X'$, l'étiquette de tête β du terme X se retrouve en tête de l'étiquette de tête de X'. Modifier cette étiquette par concaténation avec α ne fait que concaténer l'étiquette α avec l'étiquette de tête du terme X' obtenu.

Lemme 2.22 *Si $X \to_{ve} X'$ alors $X\{y\backslash Y\} \to_{ve} X'\{y\backslash Y\}$.*

Preuve : On procède par récurrence sur la taille de X.

1. Les cas $X = x^\alpha$ et $X = \Omega$ sont impossibles du fait de l'hypothèse $X \to_{ve} X'$.

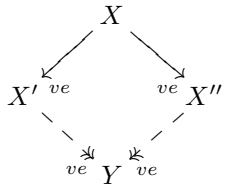


FIG. 2.11 – *Confluence locale du λ-calcul par valeur étiqueté*

2. Si $X = (((\lambda x.X_1)^\alpha)V)^\gamma \to_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_1\{x\backslash\lfloor\alpha|V\rfloor\} = X'$ (avec $\tau(V) = \beta$), alors en renommant éventuellement x, on peut supposer $x \neq y$ et $x \notin \mathtt{FV}(Y)$. On a la réduction $X\{y\backslash Y\} \to_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_1\{y\backslash Y\}\{x\backslash\lfloor\alpha|V\{y\backslash Y\}\rfloor\} = X''$. Avec le lemme 2.20, on obtient $X'\{y\backslash Y\} = \gamma\cdot\lceil\alpha|\beta\rceil\cdot X_1\{x\backslash\lfloor\alpha|V\rfloor\}\{y\backslash Y\} = \gamma\cdot\lceil\alpha|\beta\rceil\cdot X_1\{y\backslash Y\}\{x\backslash\lfloor\alpha|V\rfloor\{y\backslash Y\}\}$. En utilisant le lemme 2.19, on obtient finalement $X'\{y\backslash Y\} = X''$.
3. On utilise l'hypothèse de récurrence pour traiter les autres cas. □

Le deuxième résultat intermédiaire est un résultat de compatibilité. Si on voit la substitution $X\{y\backslash Y\}$ comme une fonction à deux arguments (un argument gauche X et un argument droit Y), alors la substitution est compatible à gauche avec la réduction étiquetée $\to_{ve}$.

Lemme 2.23 *Si $Y \to_{ve} Y'$ alors $X\{y\backslash Y\} \twoheadrightarrow_{ve} X\{y\backslash Y'\}$.*

Preuve : On procède par récurrence sur la taille de X.

1. Si $X = y^\alpha$, on a, en utilisant le lemme 2.21, $X\{y\backslash Y\} = \alpha \cdot Y \to_{ve} \alpha \cdot Y' = X\{y\backslash Y'\}$.
2. Si $X = x^\alpha$ où $x \neq y$ ou si $X = \Omega$, le résultat est élémentaire. Les cas $X = (\lambda x.X_1)^\alpha$ et $X = (X_1X_2)^\alpha$ se traitent par hypothèse de récurrence. □

Le troisième lemme intermédiaire montre la compatibilité à droite de la substitution avec la réduction étiquetée $\twoheadrightarrow_{ve}$. Ces deux derniers résultats de compatibilité se combinent de façon directe pour donner le corollaire suivant.

Corollaire 2.3 *Si $X \twoheadrightarrow_{ve} X'$ et $Y \twoheadrightarrow_{ve} Y'$ alors $X\{y\backslash Y\} \twoheadrightarrow_{ve} X'\{y\backslash Y'\}$.*

Ce résultat exprime la compatibilité de la substitution avec la relation $\twoheadrightarrow_{ve}$. Ces résultats intermédiaires se combinent pour prouver le théorème de confluence locale du λ-calcul par valeur étiqueté.

Théorème 2.6 (Confluence locale) *Si $X \to_{ve} X'$ et $X \to_{ve} X''$ alors il existe un terme Y tel que $X' \twoheadrightarrow_{ve} Y$ et $X'' \twoheadrightarrow_{ve} Y$.*

Preuve : On reprend le schéma de la preuve du théorème de confluence locale dans le λ-calcul par valeur. Le cas crucial est : $X = ((\lambda x.X_1)^\alpha V)^\gamma \to_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_1\{x\backslash\lfloor\alpha|V\rfloor\} = X'$ avec $\beta = \tau(V)$. Soit R le radical contracté entre X et X''. Trois cas sont à envisager.

1. Si R est le radical X, alors le résultat est trivial.
2. Si R est dans X_1 alors $X'' = ((\lambda x.X_1'')^\alpha V)^\gamma$ avec $X_1 \to_{ve} X_1''$. De là, on obtient la réduction $X'' \to_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_1''\{x\backslash\lfloor\alpha|V\rfloor\}$. Le lemme 2.22 permet de conclure.
3. Si R est dans V, alors $X'' = ((\lambda x.X_1)^\alpha V')^\gamma$ où $V \to_{ve} V'$. De là, on obtient la réduction $X'' \twoheadrightarrow_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_1\{x\backslash\lfloor\alpha|V'\rfloor\}$. Le lemme 2.23 permet de conclure.

Les autres cas se traitent de la même façon que pour le théorème 2.1. □

La propriété de confluence locale est illustrée sur la figure 2.11. Comme dans le cas du λ-calcul, du λ-calcul étiqueté ou du λ-calcul par valeur, la propriété de confluence locale est faible. Ce résultat n'est pas suffisant pour montrer la confluence du λ-calcul par valeur étiqueté. Comme dans la partie précédente, on emploie une technique de preuve spécifique, qui s'inspire largement de la méthode de Tait et Martin-Löf. Cette technique consiste à trouver une relation qui est contenue

dans la relation $\twoheadrightarrow_{ve}$ et qui vérifie une propriété de confluence locale forte. Cette relation, appelée relation des réductions parallèles et notée $\rightrightarrows_{ve}$, est définie de la façon suivante.

$$
\begin{array}{rll}
x^\alpha & \rightrightarrows_{ve} x^\alpha & \\
\Omega & \rightrightarrows_{ve} \Omega & \\
(XY)^\alpha & \rightrightarrows_{ve} (X'Y')^\alpha & \text{si } X \rightrightarrows_{ve} X' \text{ et } Y \rightrightarrows_{ve} Y' \\
((\lambda x.X)^\alpha V)^\gamma & \rightrightarrows_{ve} \gamma \cdot \lceil \alpha | \beta \rceil \cdot X'\{x \backslash \lfloor \alpha | V' \rfloor\} & \text{si } \tau(V) = \beta \text{ et } X \rightrightarrows_{ve} X' \text{ et } V \rightrightarrows_{ve} V' \\
(\lambda x.X)^\alpha & \rightrightarrows_{ve} (\lambda x.X')^\alpha & \text{si } X \rightrightarrows_{ve} X'
\end{array}
$$

La définition de $\rightrightarrows_{ve}$ s'inspire largement de la relation $\rightrightarrows_v$ introduite dans la partie précédente. Ainsi, si $X \rightrightarrows_{ve} Y$, alors Y peut être intuitivement obtenu en contractant des radicaux ou des résidus de radicaux présents dans X. Ces radicaux sont indépendants dans le sens où tous ces radicaux sont des résidus de radicaux de X ; aucun n'est donc créé par la contraction d'un radical de X. Ceci justifie l'appellation de *réductions parallèles*. Par contraste, une réduction $\mathcal{R} : X \xrightarrow{R_1}_{ve} Y \xrightarrow{R_2}_{ve} Z$ où le radical R_2 est créé par R_1 est considérée comme une séquence de pas de réduction : la première réduction précède nécessairement la deuxième car le radical contracté dans cette dernière est créé par R_1. Ces intuitions sur la relation $\rightrightarrows_{ve}$ sont formalisées par les propriétés suivantes.

Lemme 2.24 *1. Si $X \rightarrow_{ve} X'$, alors $X \rightrightarrows_{ve} X'$.*
2. Si $X \rightrightarrows_{ve} X'$, alors $X \twoheadrightarrow_{ve} X'$.

Ce lemme permet de faire le lien entre la réduction étiquetée et la relation des réductions parallèles. En particulier, le premier point énonce le fait que la relation $\rightrightarrows_{ve}$ contient la réduction étiquetée élémentaire. Comme attendu, le deuxième point montre que la relation des réductions parallèles est contenue dans la fermeture réflexive et transitive de la réduction étiquetée. On montre dans la suite de cette partie, que la relation des réductions parallèles est localement fortement confluente. On emploie pour cela, la même technique de preuve que pour la confluence locale de $\rightarrow_{ve}$. On obtient donc des lemmes intermédiaires qui portent sur la compatibilité de la concaténation et de la substitution avec la relation des réductions parallèles.

Lemme 2.25 *1. Si $X \rightrightarrows_{ve} X'$, alors $\alpha \cdot X \rightrightarrows_{ve} \alpha \cdot X'$.*
2. Si $V \rightrightarrows_{ve} V'$, alors $\lfloor \alpha | V \rfloor \rightrightarrows_{ve} \lfloor \alpha | V' \rfloor$.

Preuve : On procède par cas sur la réduction $X \rightrightarrows_{ve} X'$.

1. Si $X = ((\lambda x.X_1)^\delta V)^\gamma \rightrightarrows_{ve} \gamma \cdot \lceil \delta | \beta \rceil \cdot X_1'\{x \backslash \lfloor \delta | V' \rfloor\} = X'$ où $X_1 \rightrightarrows_{ve} X_1'$, $V \rightrightarrows_{ve} V'$ et $\beta = \tau(V)$. On obtient $\alpha \cdot X = ((\lambda x.X_1)^\delta V)^{\alpha\gamma} \rightrightarrows_{ve} \alpha\gamma \cdot \lceil \delta | \beta \rceil \cdot X_1'\{x \backslash \lfloor \delta | V' \rfloor\} = \alpha \cdot X'$.
2. Les autres cas sont similaires. □

Ce lemme, qui est analogue au lemme 2.21, énonce la compatibilité des opérations de concaténation et de surlignement avec la relation des réductions parallèles. Le résultat suivant est crucial pour l'obtention de la confluence locale forte de $\rightrightarrows_{ve}$.

Lemme 2.26 *Si $X \rightrightarrows_{ve} X'$ et $V \rightrightarrows_{ve} V'$, alors $X\{x \backslash V\} \rightrightarrows_{ve} X'\{x \backslash V'\}$.*

Preuve : On procède par récurrence sur la taille de X.

1. Si $X = y^\alpha$ avec $y \neq x$ ou si $X = \Omega$, alors le résultat est immédiat.
2. Si $X = x^\alpha$, on a $X' = x^\alpha$. De là $X\{y \backslash V\} = \alpha \cdot V$. En utilisant le lemme 2.25, on obtient $\alpha \cdot V \rightrightarrows_{ve} \alpha \cdot V' = X'\{x \backslash V'\}$, ce qui permet de conclure.
3. Si $X = (X_1 X_2)^\alpha$, deux cas sont possibles :
   (a) Si $X' = (X_1' X_2')^\gamma$ où $X_1 \rightrightarrows_{ve} X_1'$ et $X_2 \rightrightarrows_{ve} X_2'$, on utilise l'hypothèse de récurrence sur X_1 et X_2. On obtient $X_1\{x \backslash V\} \rightrightarrows_{ve} X_1'\{x \backslash V'\}$ et $X_2\{x \backslash V\} \rightrightarrows_{ve} X_2'\{x \backslash V'\}$. On a donc bien $X\{x \backslash V\} \rightrightarrows_{ve} X'\{x \backslash V'\}$.

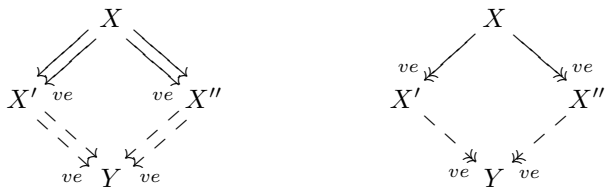


FIG. 2.12 – *Confluence locale forte de $\rightrightarrows_{ve}$ et confluence de $\to_{ve}$*

(b) Si $X = ((\lambda y.Y)^\alpha W)^\gamma$ et $X' = \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{y\backslash\lfloor\alpha|W'\rfloor\}$ avec $\beta = \tau(W)$, $Y \rightrightarrows_{ve} Y'$ et $W \rightrightarrows_{ve} W'$. En renommant éventuellement y, on peut supposer $x \neq y$ et $y \notin \mathtt{FV}(V)$. En utilisant l'hypothèse de récurrence, on obtient les relations $Y\{x\backslash V\} \rightrightarrows_{ve} Y'\{x\backslash V'\}$ et $W\{x\backslash V\} \rightrightarrows_{ve} W'\{x\backslash V'\}$. De là, comme $X\{x\backslash V\} = ((\lambda y.Y\{x\backslash V\})^\alpha W\{x\backslash V\})^\gamma$, on obtient la relation $X\{x\backslash V\} \rightrightarrows_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{x\backslash V'\}\{y\backslash\lfloor\alpha|W'\{x\backslash V'\}\rfloor\} = Z$. Avec le lemme 2.21, on obtient $Z = \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{x\backslash V'\}\{y\backslash\lfloor\alpha|W'\rfloor\{x\backslash V'\}\}$. Le lemme 2.20 permet de conclure : on a $Z = \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{y\backslash\lfloor\alpha|W'\rfloor\}\{x\backslash V'\} = X'\{x\backslash V'\}$.

4. Si $X = (\lambda y.Y)^\alpha$, alors on conclut comme précédemment par hypothèse de récurrence. □

La substitution, vue comme une fonction de deux termes, est compatible avec la relation des réductions parallèles. Ce résultat correspond au corollaire 2.3 à la différence près qu'on obtient ici le résultat pour la relation élémentaire $\rightrightarrows_{ve}$ et non pour la fermeture réflexive et transitive $\twoheadrightarrow_{ve}$. De ce fait, ce résultat entraîne, comme attendu, la confluence locale forte de $\rightrightarrows_{ve}$.

Lemme 2.27 (Confluence locale forte) *Si $X \rightrightarrows_{ve} X'$ et $X \rightrightarrows_{ve} X''$, il existe un terme Y tel que $X' \rightrightarrows_{ve} Y$ et $X'' \rightrightarrows_{ve} Y$.*

Preuve : On procède par récurrence sur la taille de X.

1. Les cas $X = x^\alpha$ et $X = \Omega$ sont élémentaires.
2. Si $X = (X_1X_2)^\gamma$, quatre cas sont possibles.
   (a) Si $X' = (X'_1X'_2)^\gamma$ et $X'' = (X''_1X''_2)^\gamma$ avec $X_1 \rightrightarrows_{ve} X'_1$, $X_2 \rightrightarrows_{ve} X'_2$, $X_1 \rightrightarrows_{ve} X''_1$ et $X_2 \rightrightarrows_{ve} X''_2$. Par hypothèse de récurrence, il existe deux termes Y_1 et Y_2 tels que $X'_1 \rightrightarrows_{ve} Y_1$, $X''_1 \rightrightarrows_{ve} Y_1$, $X'_2 \rightrightarrows_{ve} Y_2$ et $X''_2 \rightrightarrows_{ve} Y_2$. On obtient $X' \rightrightarrows_{ve} (Y_1Y_2)^\gamma$ et $X'' \rightrightarrows_{ve} (Y_1Y_2)^\gamma$.
   (b) Si $X = ((\lambda x.X_3)^\alpha V)^\gamma$ et $X' = \gamma \cdot \lceil\alpha|\beta\rceil \cdot X'_3\{x\backslash\lfloor\alpha|V'\rfloor\}$ et $X'' = ((\lambda x.X''_3)^\alpha V'')^\gamma$ où $\beta = \tau(V) = \tau(V') = \tau(V'')$, $X_3 \rightrightarrows_{ve} X'_3$, $X_3 \rightrightarrows_{ve} X''_3$, $V \rightrightarrows_{ve} V'$ et $V \rightrightarrows_{ve} V''$. Par hypothèse de récurrence, il existe des termes Y_3 et W tels que $X'_3 \rightrightarrows_{ve} Y_3$, $X''_3 \rightrightarrows_{ve} Y_3$, $V' \rightrightarrows_{ve} W$ et $V'' \rightrightarrows_{ve} W$. De là, le lemme 2.25 donne $\lfloor\alpha|V'\rfloor \rightrightarrows_{ve} \lfloor\alpha|W\rfloor$. En utilisant les lemmes 2.26 et 2.25, on obtient $X' \rightrightarrows_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y_3\{x\backslash\lfloor\alpha|W\rfloor\}$. On obtient par ailleurs $X'' \rightrightarrows_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y_3\{x\backslash\lfloor\alpha|W\rfloor\}$.
   (c) Le cas où $X = ((\lambda x.X_3)^\alpha V)^\gamma$ et $X' = ((\lambda x.X'_3)^\alpha V')^\gamma$ et $X'' = \gamma \cdot \lceil\alpha|\beta\rceil \cdot X''_3\{x\backslash\lfloor\alpha|V''\rfloor\}$ est symétrique du cas précédent.
   (d) Si $X = ((\lambda x.X_3)^\alpha V)^\gamma$, $X' = \gamma \cdot \lceil\alpha|\beta\rceil \cdot X'_3\{x\backslash\lfloor\alpha|V'\rfloor\}$ et $X'' = \gamma \cdot \lceil\alpha|\beta\rceil \cdot X''_3\{x\backslash\lfloor\alpha|V''\rfloor\}$. où $\beta = \tau(V) = \tau(V')$ $X_3 \rightrightarrows_{ve} X'_3$, $X_3 \rightrightarrows_{ve} X''_3$, $V \rightrightarrows_{ve} V'$ et $V \rightrightarrows_{ve} V''$. Par hypothèse de récurrence, il existe des termes Y_3 et W tels que $X'_3 \rightrightarrows_{ve} Y_3$, $X''_3 \rightrightarrows_{ve} Y_3$, $V' \rightrightarrows_{ve} W$ et $V'' \rightrightarrows_{ve} W$. Le lemme 2.25 donne $\lfloor\alpha|V'\rfloor \rightrightarrows_{ve} \lfloor\alpha|W\rfloor$ et $\lfloor\alpha|V''\rfloor \rightrightarrows_{ve} \lfloor\alpha|W\rfloor$. De là, en utilisant les lemmes 2.25 et 2.26, on obtient $X' \rightrightarrows_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y_3\{x\backslash\lfloor\alpha|W\rfloor\}$. et $X'' \rightrightarrows_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y_3\{x\backslash\lfloor\alpha|W\rfloor\}$.
3. Si $X = (\lambda x.X_1)^\alpha$, on a nécessairement $X' = (\lambda x.X'_1)^\alpha$ où $X_1 \rightrightarrows_{ve} X'_1$ et $X'' = (\lambda x.X''_1)^\alpha$ où $X_1 \rightrightarrows_{ve} X''_1$. On conclut par hypothèse de récurrence comme précédemment. □

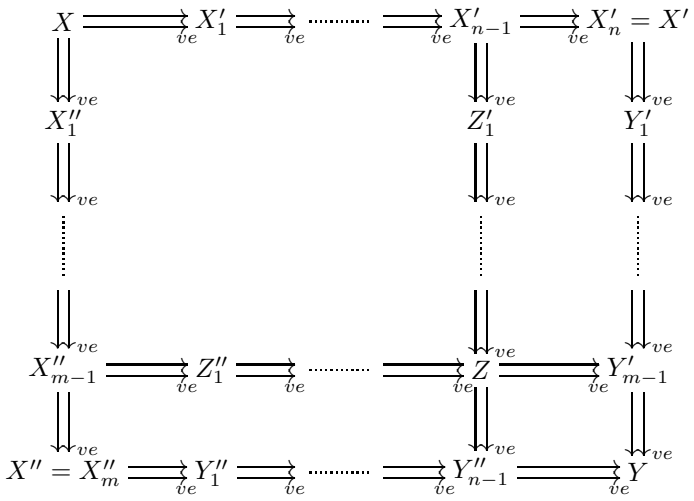


FIG. 2.13 – *Preuve de la confluence* $\rightarrow_{ve}$

Selon ce résultat, qui est illustré sur la figure 2.12, si un terme X se réduit en une étape de réductions parallèles vers deux termes X' et X'', alors il existe un terme Y atteignable depuis X' et X'' en une étape de réductions parallèles. Ce résultat aboutit au théorème de confluence de la réduction étiquetée du λ-calcul par valeur.

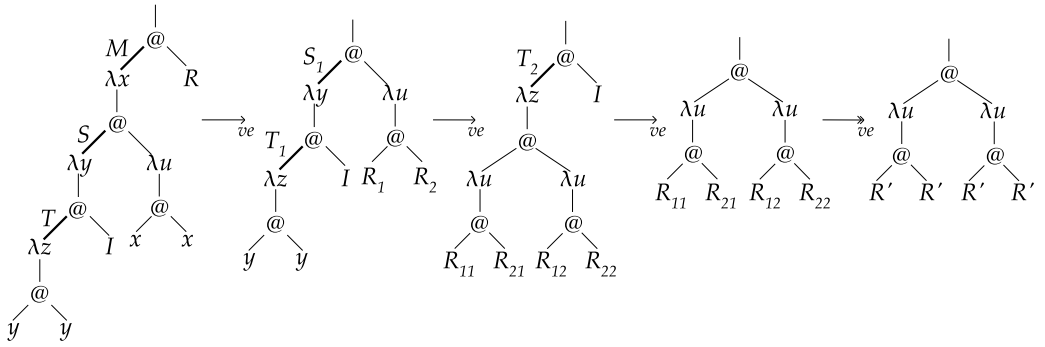
Théorème 2.7 (Confluence) *Si $X \twoheadrightarrow_{ve} X'$ et $X \twoheadrightarrow_{ve} X''$, alors il existe un terme Y qui vérifie $X' \twoheadrightarrow_{ve} Y$ et $X'' \twoheadrightarrow_{ve} Y$.*

Preuve : On considère la propriété suivante. Si on a les réductions $X \rightrightarrows_{ve} X'_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} X'_n = X'$ et $X \rightrightarrows_{ve} X''_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} X''_m = X''$, alors il existe un terme Y tel que $X' \rightrightarrows_{ve} Y'_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} Y'_m$ et $X'' \rightrightarrows_{ve} Y''_1 \ldots \rightrightarrows_{ve} Y''_n$ avec $Y = Y'_m = Y'_n$. En utilisant le lemme 2.24, cette propriété implique de façon élémentaire la confluence de $\rightarrow_{ve}$. On montre cette propriété par récurrence sur la borne supérieure p de $\{n,m\}$. Le cas $p = 0$ est élémentaire. Le cas $p = 1$ est prouvé par le lemme 2.27. On suppose désormais $p > 1$. Si $n = 0$ ou $m = 0$, le résultat est élémentaire. On suppose désormais $n,m > 0$. Cette preuve est illustrée sur la figure 2.13. On applique l'hypothèse de récurrence sur $X \rightrightarrows_{ve} X'_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} X'_{n-1}$ et $X \rightrightarrows_{ve} X''_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} X''_{m-1}$. On obtient un terme Z tel que $X'_{n-1} \rightrightarrows_{ve} Z'_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} Z'_{m-1} = Z$ et $X''_{m-1} \rightrightarrows_{ve} Z''_1 \ldots \rightrightarrows_{ve} Z''_{n-1} = Z$. On applique à nouveau l'hypothèse de récurrence, d'une part sur $X'_{n-1} \rightrightarrows_{ve} X'$ et $X'_{n-1} \rightrightarrows_{ve} Z'_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} Z$, et d'autre part sur $X''_{m-1} \rightrightarrows_{ve} X''$ et $X''_{m-1} \rightrightarrows_{ve} Z''_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} Z$. On obtient deux termes Y'_{m-1} et Y''_{n-1} tels que $Z \rightrightarrows_{ve} Y'_{m-1}$, $X' \rightrightarrows_{ve} Y'_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} Y'_{m-1}$, $Z \rightrightarrows_{ve} Y''_{n-1}$ et $X'' \rightrightarrows_{ve} Y''_1 \rightrightarrows_{ve} \ldots \rightrightarrows_{ve} Y''_{n-1}$. On conclut par le lemme 2.27. □

Le λ-calcul par valeur étiqueté est confluent : si un terme X se réduit vers X' et X'' il existe un terme Y qui est atteignable à la fois depuis X' et X''. Cette propriété est illustrée sur la figure 2.12. La preuve de confluence s'appuie essentiellement sur la propriété de confluence locale forte de la relation des réductions parallèles $\rightrightarrows_{ve}$.

2.2.2 Développements finis

Dans le cas du λ-calcul par valeur, le théorème des développements finis s'obtient simplement à partir du théorème des développements finis du λ-calcul. Dans le cas du λ-calcul par valeur étiqueté, on ne peut pas déduire cette propriété à partir du théorème des développements finis du λ-calcul étiqueté pour la simple raison que la syntaxe des étiquettes du λ-calcul par valeur étiqueté est légèrement différente de celle du λ-calcul étiqueté. En réalité, cette différence de syntaxe ne



$$R = II \qquad S = (\lambda y.(\lambda z.yy)I) \qquad T = (\lambda z.yy)I$$
$$\mathcal{F} = \{M, R, S, T\}$$

FIG. 2.14 – *Développement de $\mathcal{F}$ dans $M = (\lambda x.(\lambda y.(\lambda z.yy)I)\lambda u.xx)R$*

change pas réellement la nature du problème. Mais c'est un bon prétexte pour se pencher sur la démonstration de cette propriété fondamentale du λ-calcul. La preuve mentionnée dans cette partie s'applique au λ-calcul par valeur ou non et étiqueté ou non.

Si M est un terme qui contient un ensemble de radicaux $\mathcal{F} = \{R_1, \ldots, R_n\}$, la propriété essentielle du théorème des développements finis réside dans le fait que toutes les réductions relatives à $\mathcal{F}$ sont finies. En d'autres termes, il n'existe aucune réduction relative à $\mathcal{F}$ qui soit infinie. On illustre cette propriété avec le terme $M = (\lambda x.(\lambda y.(\lambda z.yy)I)\lambda u.xx)R$ (où $I = \lambda v.v$ et $R = II$) dans lequel les étiquettes sont omises pour des raisons de lisibilité. On considère l'ensemble de radical de M suivant : $\mathcal{F} = \{M, R, S, T\}$ où $S = (\lambda y.(\lambda z.yy)I)$ et $T = (\lambda z.yy)I$. Un développement de $\mathcal{F}$ est illustré sur la figure 2.14. La première réduction consiste à contracter M. Cette réduction duplique le radical R. L'ensemble résidu $\mathcal{F}_1$ de $\mathcal{F}$ vérifie $\mathcal{F}_1 = \{S_1, T_1, R_1, R_2\}$ et contient quatre éléments, c'est-à-dire autant d'éléments que $\mathcal{F}$. On note que si R était disjoint de S dans M, les résidus de R sont contenus dans S_1 qui est le résidu de S. La deuxième réduction consiste à contracter S_1. Cette réduction duplique à nouveau les résidus de R. L'ensemble résidu de $\mathcal{F}$ est $\mathcal{F}_2 = \{T_2, ,R_{11}, R_{12}, R_{21}, R_{22}\}$ et contient cinq éléments. On note que T_2, le résidu de T, contient les résidus de R, alors que T était disjoint de R dans M. La troisième réduction contracte T_2. Puis les réductions suivantes contractent les résidus de R. Cette réduction montre que le cardinal des ensembles résidus de $\mathcal{F}$ ne diminue pas nécessairement après une réduction du fait des duplications qui peuvent se produire. On note en outre que les relations d'imbrication entre les radicaux ne restent pas figées au cours du calcul. Cette relation, notée $<$, est définie de la façon suivante : on a $X < Y$ si et seulement si X contient strictement Y. La relation $<$ est bien entendu un ordre strict partiel. L'imbrication au sens large est notée $\leq$. Dans l'exemple mentionné ici, des résidus de des radicaux initialement disjoints peuvent être imbriqués après quelques réductions. Ainsi, si on considère le multi-ensemble des profondeurs d'imbrication des radicaux de $\mathcal{F}$ et de ses résidus, on note que ce multi-ensemble ne décroît pas du fait des nouvelles imbrications qui apparaissent au cours du développement. Cette profondeur est définie de la façon suivante.

$$\mathcal{P}_{\mathcal{F}}(U) = \mathtt{max}(\{0\} \cup \{1 + \mathcal{P}_{\mathcal{F}}(U')/U' \in \mathcal{F} \text{ et } U' < U\})$$

Comme on a $M < S < T$, on a $\mathcal{P}_{\mathcal{F}}(T) = 2$ et $\mathcal{P}_{\mathcal{F}}(S) = 1$. Plus globalement, le multi-ensemble correspondant à $\mathcal{F} = \{M, R, S, T\}$ est $\mathcal{M} = \{\!\{0, 1, 1, 2\}\!\}$. Après la contraction de M, le multi-ensemble correspondant à $\mathcal{F}_1 = \{S_1, T_1, R_1, R_2\}$ est $\mathcal{M}_1 = \{\!\{0, 1, 1, 1\}\!\}$. Ce dernier multi-ensemble est donc bien strictement inférieur à $\mathcal{M}$ au sens de l'ordre $\leq_m$ sur les multi-ensembles d'entiers. En revanche, après la contraction de S_1, le multi-ensemble correspondant à

$\mathcal{F}_2 = \{T_2, , R_{11}, R_{12}, R_{21}, R_{22}\}$ est $\mathcal{M}_2 = \{\!\{0, 1, 1, 1, 1\}\!\}$. On a donc $\mathcal{M}_2 \not<_m \mathcal{M}_1$. La relation d'imbrication n'est donc pas directement exploitable pour montrer que les développements sont finis.

Pour prouver le théorème des développements finis, la technique classique, exposée dans [7], consiste à attribuer des entiers à tous les sous-termes de M. Ces entiers associés aux sous-termes permettent de définir une norme sur les termes. En attribuant les entiers aux sous-termes de M de façon astucieuse, on peut montrer qu'au cours d'une réduction relative à $\mathcal{F}$, la norme des termes obtenus diminue strictement. Cette technique de preuve, certes efficace, a l'inconvénient d'être peu intuitive : elle est peu révélatrice sur les raisons fondamentales qui font que le théorème des développements finis est vrai.

On adopte ici une approche plus concrète, qui met en lumière les mécanismes profonds des développements finis. Cette preuve s'inspire des travaux préliminaires (non publiés) engagés par Curien, Lévy et Melliès et tire profit des constatations relevées sur l'exemple mentionné plus haut. Nous avons vu, en particulier, que la relation $<$ n'était pas adaptée car deux radicaux initialement disjoints pouvaient devenir imbriqués au cours du développement. Plus précisément, on observe sur la figure 2.14 que la contraction du radical M substitue le radical R aux occurrences de x. De ce fait, les résidus de R sont injectés dans le résidu S_1 du radical S. Plus fondamentalement, comme l'abstraction M contient S, que S contient la variable liée par M et que la partie droite de M contient le radical R, le résidu de S (après la contraction de M) contient les résidus de R : ceci nous amène à dire que R est imbriqué dans S dans le futur, c'est-à-dire après la contraction de M. Cette situation sera formellement décrite plus loin par les relations $\nearrow$ et $\nwarrow$. Dans le cas de S et R, cette imbrication future est directe. En revanche, l'imbrication de R dans T est indirecte : T contient la variable liée par S et S contient dans sa partie droite la variable liée par M. De ce fait, après la contraction de M, le résidu S_1 de S contient les résidus R_1 et R_2 de R dans sa partie droite. Comme précédemment, la contraction de S_1 conduit à l'injection des résidus de R dans le résidu T_2 de T. Cette situation sera formellement décrite plus loin par les relations $\nearrow$, $\overset{\nearrow}{\scriptstyle a}$ et $\nwarrow$. Dans l'approche utilisée ici, ces *imbrications futures* sont prises en compte. Nous introduisons une imbrication $\Subset_{\mathcal{F}}$ qui met en relation (dans le terme initial) tous les radicaux qui sont en relation par $<$ dans le terme initial ou dont des résidus seront en relation par $<$ au cours d'un développement. Cette propriété est formalisée dans le lemme 2.33. Pour définir cette imbrication étendue $\Subset_{\mathcal{F}}$, nous exploitons la relation d'imbrication $<$. Cette dernière porte sur les termes, ce qui est peu commode lorsqu'on veut l'utiliser avant et après une réduction. Si Y et Z sont des sous-termes de X et $X \to_{ve} X'$, la relation $Y < Z$ n'est pas très exploitable dans X' car la notion de résidu de sous-terme est délicate à définir. En revanche, la notion de résidu de radical est précisément et simplement définie. On s'intéressera donc par la suite plus particulièrement à l'imbrication des radicaux. Dans cette optique, on introduit trois notations qui raffinent la relation $<$ pour les radicaux et qui sont illustrées sur la figure 2.15. Comme l'a montré l'exemple de la figure 2.14, les variables liées jouent un rôle particulier dans ces définitions. Pour les manipuler aisément et précisément, on introduit la notation suivante : on a $\texttt{subFV}(x,X,Y)$ si et seulement si $X = C[Y]$ et $Y = C'[x]$ et $(C[C'[\,]],x)$ est une occurrence libre de x dans X. Intuitivement $\texttt{subFV}(x,X,Y)$ signifie que Y est un sous-terme de X qui contient une occurrence de x libre dans X. Cette notation nous permet de définir les relations $\nearrow$, $\overset{\nearrow}{\scriptstyle a}$ et $\nwarrow$.

$S \nearrow R$ si et seulement si $R = ((\lambda x.X)^\alpha V)^\beta$ et $X \leq S$ et $\texttt{subFV}(x,X,S)$

$S \overset{\nearrow}{\scriptstyle a} R$ si et seulement si $R = ((\lambda x.X)^\alpha V)^\beta$ et $S = ((\lambda y.Y)^\gamma W)^\delta$ et $X \leq S$ et $\texttt{subFV}(x,X,W)$

$R \nwarrow S$ si et seulement si $R = ((\lambda x.X)^\alpha V)^\beta$ et $V \leq S$

La relation $S \nearrow R$ signifie que S contient la variable liée par R. Ceci s'exprime formellement par le fait que l'abstraction du radical R contient le radical S et que la variable liée par R a une occurrence

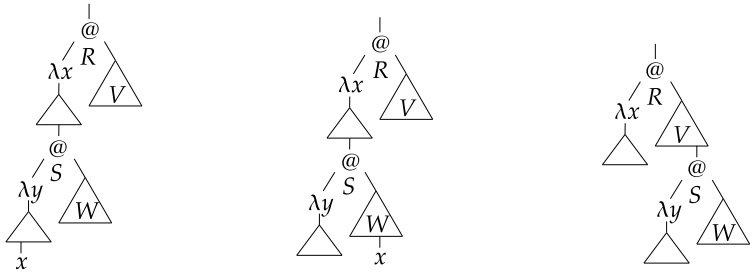


FIG. 2.15 – *Relations d'imbrication sur les radicaux*

dans S qui est libre dans X. Cette relation est raffinée par la relation $S \underset{a}{\nearrow} R$ qui impose de plus que la partie argument de S contient une variable liée par R. Symétriquement, la relation $R \nwarrow S$ signifie que la partie argument du radical R contient le radical S. Comme l'exemple mentionné plus haut l'a illustré, la notion d'imbrication future dépend essentiellement de la position des variables liées par les radicaux : les relations $\nearrow$ et $\underset{a}{\nearrow}$ témoignent de l'attention portée aux variables liées des radicaux. On veut en effet capturer le fait qu'un radical T présent dans un argument d'un radical R peut passer, quand R est contracté, sous un autre radical S si $S \nearrow R$. Si $S \underset{a}{\nearrow} R$, on obtient de plus que T passe dans la partie argument de S après la contraction de R. Dans l'exemple de la figure 2.14, on a en particulier les relations $S \nearrow M$ et $S \underset{a}{\nearrow} M$. On a aussi $T \nearrow S$ et $M \nwarrow R$. Les relations $\nearrow$, $\underset{a}{\nearrow}$ et $\nwarrow$ vérifient les premières propriétés élémentaires suivantes.

Lemme 2.28 (Propriétés élémentaires)

1. *Si* $R \underset{a}{\nearrow} S$, *alors* $R \nearrow S$.
2. *Si* $S \nearrow R$ *et* $R < T \leq S$, *alors* $T \nearrow R$.
3. *Si* $R \nwarrow S \nearrow T$, *alors* $R \nwarrow T$ *ou* $R \underset{a}{\nearrow} T$.
4. *Si* $R \nwarrow S \underset{a}{\nearrow} T$, *alors* $R \nwarrow T$ *ou* $R \underset{a}{\nearrow} T$.

Preuve : On examine ces propriétés successivement.

1. La première propriété est élémentaire.
2. Comme T contient S, alors T contient une variable liée par R. Deux cas sont possibles : (1) T contient l'abstraction de R donc R (puisque T est un radical) ce qui est exclu par l'hypothèse $R < T$; (2) l'abstraction de R contient T ce qui donne bien $T \nearrow R$.
3. Comme les radicaux R et T contiennent un radical S, alors trois cas sont possibles :
   (a) Le cas $R = T$ est exclu par le fait que l'abstraction de T contient S alors que la partie argument de R contient S.
   (b) Si R contient T, T est dans la partie argument de R i.e. $R \nearrow T$.
   (c) Si T contient R, alors comme S est inclus dans l'argument de R, R est dans l'abstraction de T et une variable liée par T est dans la partie argument de R i.e. $R \underset{a}{\nearrow} T$.
4. La quatrième propriété est un corollaire du premier et du troisième point. □

Le premier point énonce le fait que la relation $\underset{a}{\nearrow}$ est contenue dans la relation $\nearrow$. Le deuxième point indique que si T est contenu dans R et si T contient le radical S qui contient la variable liée par R alors T contient la variable liée par R et donc $T \nearrow R$. Les troisième et quatrième points montrent que si S est dans la partie droite de R et si S contient la variable liée par T, alors deux cas sont possibles : soit T est dans la partie droite de R (i.e. $R \nwarrow T$) soit la partie droite de R contient la variable liée par T (i.e. $R \underset{a}{\nearrow} T$). En analysant l'exemple de la figure 2.14, nous avions noté que l'imbrication future de R dans T résultait de la conjonction des éléments suivants.

1. M contient R dans sa partie droite, c'est-à-dire $M \nwarrow R$.
2. La variable liée par M a une occurrence dans la partie droite de S, c'est-à-dire $S \underset{a}{\nearrow} M$.
3. La variable liée par S a une occurrence dans T, c'est-à-dire $T \nearrow S$.

Plus synthétiquement, la conjonction $T \nearrow S \underset{a}{\nearrow} M \nwarrow R$ entraîne l'imbrication future de R dans T. De même, la propriété $S \nearrow M \nwarrow R$ entraîne l'imbrication future de R dans S. Pour formaliser cette notion d'*imbrication future*, on introduit les notations suivantes.

$$R \ll_{T_1 \ldots T_n} S \text{ si et seulement si } R \nearrow T_1 \underset{a}{\nearrow} T_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} T_n \nwarrow S$$

$$R \ll_{\mathcal{F}} S \text{ si et seulement si } R \ll_{T_1 \ldots T_n} S \text{ où } \{T_i\}_{1 \leq i \leq n} \text{ est une suite de radicaux de } \mathcal{F}$$

$$R \Subset_{\mathcal{F}} S \text{ si et seulement si } R < S \text{ ou } R \ll_{\mathcal{F}} S.$$

La relation $\ll_{\mathcal{F}}$ capture la notion d'*imbrication future* mentionnée plus haut via la notation plus précise $\ll_{T_1 \ldots T_n}$. La relation $\Subset_{\mathcal{F}}$ étend $\ll_{\mathcal{F}}$ avec la relation d'imbrication $<$ habituelle. On montre que cette nouvelle relation est, comme $<$, un ordre strict.

Lemme 2.29 *La relation $\ll_{\mathcal{F}}$ est un ordre strict.*

Preuve : Si on a $R \ll_{\mathcal{F}} S$, alors il existe une suite $\{T_i\}_{1 \leq i \leq n}$ de radicaux de $\mathcal{F}$ telle qu'on a la relation $R \nearrow T_1 \underset{a}{\nearrow} T_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} T_n \nwarrow S$. Par conséquent T_n contient R dans son abstraction et S dans sa partie argument. On en déduit que $\ll_{\mathcal{F}}$ est non-réflexive. On examine maintenant la transitivité de cette relation. On suppose $R \ll_{R_1 \ldots R_n} S \ll_{S_1 \ldots S_m} T$ où $R_1 \ldots R_n$ et $S_1 \ldots S_m$ sont des suites de radicaux de $\mathcal{F}$. On a donc $R \nearrow R_1 \underset{a}{\nearrow} R_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} R_n \nwarrow S \nearrow S_1 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} S_m \nwarrow T$. L'utilisation itérative des propriétés 3 et 4 du lemme 2.28 peut aboutir à deux issues :

1. Il existe un entier m_0 tel que $1 \leq m_0 \leq m$ et $R \nearrow R_1 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} R_n \underset{a}{\nearrow} S_{m_0} \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} S_m \nwarrow T$.
2. On obtient $R \nearrow R_1 \underset{a}{\nearrow} R_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} R_n \nwarrow S_m \nwarrow T$ ce qui implique immédiatement $R \nearrow R_1 \underset{a}{\nearrow} R_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} R_n \nwarrow T$.

Dans les deux cas, on obtient $R \ll_{\mathcal{F}} T$. □

On souhaite montrer que la relation d'imbrication étendue $\Subset_{\mathcal{F}}$ est également un ordre strict. Comme les relations $<$ et $\ll_{\mathcal{F}}$ sont des ordres stricts, il est clair que la relation $\Subset_{\mathcal{F}}$ est non-réflexive. Il reste à montrer que cette dernière est transitive. Pour cela, on s'appuie sur les propriétés suivantes, vérifiées par $<$ et $\ll_{\mathcal{F}}$.

Lemme 2.30 *1. Si $R < S \ll_{\mathcal{F}} T$, alors $R < T$ ou $R \ll_{\mathcal{F}} T$.*
2. Si $R \ll_{\mathcal{F}} S < T$, alors $R \ll_{\mathcal{F}} T$.

Preuve : On examine successivement les deux propriétés.

1. On suppose $S \ll_{U_1 \ldots U_n} T$ où $U_1 \ldots U_n$ est une suite de radicaux de $\mathcal{F}$. Comme R et chacun des U_i contient S, on a pour tout $i \in \{1 \ldots n\}$ soit $U_i < R$, soit $R < U_i$ soit $U_i = R$. Au total, trois cas sont à envisager :
   (a) Si $R < U_n$, comme $U_n < T$, on a bien sûr $R < T$.
   (b) Si $U_1 < R$, comme $R < S$ et $S \nearrow U_1$, on a, d'après le lemme 2.28, $R \nearrow U_1$. On a donc $R \nearrow U_1 \underset{a}{\nearrow} U_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} U_n \nwarrow T$, c'est-à-dire $R \ll_{U_1 \ldots U_n} T$.
   (c) Sinon, il existe un indice $i > 1$ tel que $U_i < R \leq U_{i-1}$. Comme $U_{i-1} \underset{a}{\nearrow} U_i$, on a $U_{i-1} \nearrow U_i$. De là, le lemme 2.28 donne $R \nearrow U_i$. On obtient donc $R \ll_{U_i \ldots U_n} T$.
2. On suppose $R \ll_{U_1 \ldots U_n} S$ où $U_1 \ldots U_n$ est une suite de radicaux de $\mathcal{F}$. On a $U_n \nwarrow S$ et $S < T$, d'où $U_n \nwarrow T$. Ceci implique $R \ll_{U_1 \ldots U_n} T$. □

La transitivité de la relation d'imbrication étendue $\Subset_{\mathcal{F}}$ découle directement de ce résultat. En conjonction avec la propriété de non-réflexivité de $\Subset_{\mathcal{F}}$, on obtient directement la propriété suivante.

Lemme 2.31 *La relation $\Subset_{\mathcal{F}}$ est un ordre strict.*

Preuve : Comme $<$ et $\ll_{\mathcal{F}}$ sont des ordres stricts, la relation $\Subset_{\mathcal{F}}$ est non-réflexive. Le lemme 2.30 permet de prouver la transitivité de $\Subset_{\mathcal{F}}$. □

L'examen de l'exemple de la figure 2.14 a montré que la relation d'imbrication $<$ n'était pas adaptée à la preuve des développements finis du fait de l'apparition de nouvelles imbrications au cours du développement. Ainsi, après la contraction de M, le radical R_1 est imbriqué dans S_1 alors que R (dont R_1 est un résidu) n'est pas imbriqué dans S (dont S_1 est le résidu). L'introduction des relations d'imbrication future $\ll_{\mathcal{F}}$ et d'imbrication étendue $\Subset_{\mathcal{F}}$ avait pour but de corriger cette difficulté. Ainsi, toujours dans cet exemple, on a bien $S \Subset_{\mathcal{F}} R$ car $S \ll_{\mathcal{F}} R$. Plus généralement, dans la suite de cette partie, on prouve que si deux résidus R' et S' de radicaux R et S de $\mathcal{F}$ vérifient $R \Subset_{\mathcal{F}'} S$ (où $\mathcal{F}'$ est l'ensemble résidu de $\mathcal{F}$) alors $R \Subset_{\mathcal{F}} S$. Pour prouver cette propriété qui est formellement énoncée dans le lemme 2.33, on utilise les propriétés du lemme technique suivant.

Lemme 2.32 *Si $X \xrightarrow{T}_{ve} X'$, et si R' et S' sont des résidus des radicaux R et S de X, les propriétés suivantes sont vérifiées.*

1. *Si $R' < S'$, alors $R < S$ ou $R \ll_T S$.*
2. *Si $R' \nearrow S'$, alors $R \nearrow S$ ou $R \nearrow T \underset{a}{\nearrow} S$.*
3. *Si $R' \underset{a}{\nearrow} S'$, alors $R \underset{a}{\nearrow} S$ ou $R \underset{a}{\nearrow} T \underset{a}{\nearrow} S$.*
4. *Si $R' \nwarrow S'$, alors $R \nwarrow S$ ou $R \underset{a}{\nearrow} T \nwarrow S$.*
5. *On suppose que $U'_1, U'_2, \ldots, U'_n$ sont des résidus des radicaux $U_1, U_2, \ldots, U_n$ de X. Si on a la relation $R' \ll_{U'_1 \ldots U'_n} S'$, alors on a $R \ll_{T_1 \ldots T_m} S$ où $T_1, T_2, \ldots, T_m$ est une suite extraite de la suite $T, U_1, T, U_2, T, \ldots, T, U_n, T$ qui contient tous les radicaux de $U_1, \ldots, U_n$.*

Preuve : On examine successivement les cinq propriétés.

1. (a) Si $T < R$ et si R ne contient pas S, comme $R' < S'$, R contient nécessairement la variable liée par T i.e. $R \nearrow T$. Et S est nécessairement dans la partie droite de T, i.e. $T \nwarrow S$. On a donc $R \ll_T S$.
   (b) Si $R < T$. Comme R' contient S', on a nécessairement $R < S$.
   (c) Si $T \not< R$ et $R \not< T$, alors le sous-terme R n'est pas modifié par la contraction de T : on a $R' = R$. Par conséquent, $R' < S'$ implique $R < S$.
2. On suppose $R \not\nearrow S$: la variable liée par S n'apparaît pas libre dans R. Dans ce cas, comme $R' \nearrow S'$, R contient nécessairement la variable liée par T i.e. $R \nearrow T$. Et la variable liée par S apparaît libre dans la partie droite de T i.e. $T \underset{a}{\nearrow} S$. On a donc bien $R \nearrow T \underset{a}{\nearrow} S$.
3. On procède de la même façon que dans le cas précédent. On suppose $R \not\underset{a}{\nearrow} S$: donc la variable liée par S n'apparaît pas libre dans la partie droite de R. Dans ce cas, comme $R' \underset{a}{\nearrow} S'$, la partie droite de R contient nécessairement la variable liée par T i.e. $R \underset{a}{\nearrow} T$. Et la variable liée par S apparaît libre dans la partie droite de T, i.e. $T \underset{a}{\nearrow} S$. On a donc bien $R \underset{a}{\nearrow} T \underset{a}{\nearrow} S$.
4. On suppose $R \not\nwarrow S$: la partie droite de R ne contient pas S. Comme $R' \nwarrow S'$, la partie droite de R contient nécessairement la variable liée par T i.e. $R \underset{a}{\nearrow} T$. Et la partie droite de T contient S i.e. $T \nwarrow S$. On a donc bien $R \underset{a}{\nearrow} T \nwarrow S$.
5. On a $R' \nearrow U'_1 \underset{a}{\nearrow} U'_2 \underset{a}{\nearrow} \ldots \underset{a}{\nearrow} U'_n \nwarrow S'$. Par application des points précédents, on a :

$$\begin{array}{ll} & (R \nearrow U_1 \vee R \nearrow T \underset{a}{\nearrow} U_1) \\ \wedge & (U_1 \underset{a}{\nearrow} U_2 \vee U_1 \underset{a}{\nearrow} T \underset{a}{\nearrow} U_2) \\ \wedge & (U_2 \underset{a}{\nearrow} U_3 \vee U_2 \underset{a}{\nearrow} T \underset{a}{\nearrow} U_3) \\ \ldots & \\ \wedge & (U_{n-1} \underset{a}{\nearrow} U_n \vee U_{n-1} \underset{a}{\nearrow} T \underset{a}{\nearrow} U_n) \\ \wedge & (U_n \nwarrow S \vee U_n \underset{a}{\nearrow} T \nwarrow S) \end{array}$$

   On obtient bien le résultat voulu. □

Le premier point justifie l'appellation de la relation $\ll_T$: si des résidus de R et S sont imbriqués alors soit ils sont eux-mêmes imbriqués, soit ils sont en relation par la relation d'imbrication future $\ll_T$. Les quatre points suivants énoncent le fait que si des résidus de R et S sont en relation par $\nearrow$, $\overset{a}{\nearrow}$, $\nwarrow$ ou $\ll_{T_1 \ldots T_m}$, alors R et S sont eux-mêmes en relation ou bien sont en relation indirecte via le radical T contracté. Ce lemme technique permet de prouver le lemme suivant, qui fait de $\Subset_{\mathcal{F}}$ une relation adaptée pour montrer le théorème des développements finis.

Lemme 2.33 *Soit $\mathcal{F}$ un ensemble de radicaux de X. On suppose $T \in \mathcal{F}$ et $X \xrightarrow{T}_{ve} X'$. Soit $\mathcal{F}'$ l'ensemble résidu de $\mathcal{F}$ par cette réduction. Soient R' et S' deux radicaux de X' qui sont des résidus des radicaux R et S de X. Si $R' \Subset_{\mathcal{F}'} S'$, alors on a $R \Subset_{\mathcal{F}} S$.*

Preuve : Ce résultat est une application directe des points 1 et 5 du lemme précédent. □

Comme attendu, si des résidus de R et S sont en relation par $\Subset_{\mathcal{F}'}$ où $\mathcal{F}'$ est l'ensemble résidu de $\mathcal{F}$ par la réduction qui contracte un élément de $\mathcal{F}$, alors les résidus R et S sont eux-mêmes en relation par $\Subset_{\mathcal{F}}$. On montre ainsi que la difficulté relevée pour $<$ dans l'exemple de la figure 2.14 est levée : aucune imbrication étendue n'est créée au cours d'un développement. Cela constitue la propriété cruciale de la preuve. Cette propriété est exploitée en redéfinissant la profondeur $\mathcal{P}_{\mathcal{F}}(R)$ d'un radical de X.

$$\mathcal{P}_{\mathcal{F}}(R) = \mathtt{max}(\{0\} \cup \{1 + \mathcal{P}_{\mathcal{F}}(S) \mid S \in \mathcal{F} \text{ et } S \Subset_{\mathcal{F}} R\})$$

On montre que la profondeur d'un résidu R' d'un radical R de $\mathcal{F}$ est strictement inférieure à celle de R si on contracte un radical de $\mathcal{F}$ qui contient R.

Lemme 2.34 *Soit $\mathcal{F}$ un ensemble de radicaux de X et T un élément de $\mathcal{F}$. On suppose $X \xrightarrow{T}_{ve} X'$. Soit $\mathcal{F}'$ l'ensemble des résidus de $\mathcal{F}$ par cette réduction. Soit R' un résidu d'un radical R de $\mathcal{F}$. Si $T < R$ alors on a $\mathcal{P}_{\mathcal{F}'}(R') < \mathcal{P}_{\mathcal{F}}(R)$.*

Preuve : On montre cette propriété par récurrence sur $\mathcal{P}_{\mathcal{F}'}(R') = n$.

1. Si $n = 0$, alors comme $T < R$, on a $\mathcal{P}_{\mathcal{F}}(R) \geq 1 > \mathcal{P}_{\mathcal{F}'}(R')$.
2. Si $n > 1$, alors il existe une suite de $\mathcal{F}'$ telle que $R'_1 \Subset_{\mathcal{F}'} R'_2 \Subset_{\mathcal{F}'} \ldots \Subset_{\mathcal{F}'} R'_n \Subset_{\mathcal{F}'} R'$. Pour $1 \leq i \leq n$, on note R_i le radical de $\mathcal{F}$ dont R'_i est un résidu. D'après le lemme 2.33, on a $R_1 \Subset_{\mathcal{F}} R_2 \Subset_{\mathcal{F}} \ldots \Subset_{\mathcal{F}} R_n \Subset_{\mathcal{F}} R$ (où pour $1 \leq i \leq n$, on a $R_i \neq T$). On montre maintenant, qu'il existe une suite de radicaux de $\mathcal{F}$ telle que $S_1 \Subset_{\mathcal{F}} S_2 \Subset_{\mathcal{F}} \ldots \Subset_{\mathcal{F}} S_{n+1} \Subset_{\mathcal{F}} R$. On procède par cas sur la relation $R_n \Subset_{\mathcal{F}} R$.
   (a) Si $R_n < R$, comme on a $T < R$ et $T \neq R_n$, seules deux positions relatives de R_n et T sont possibles.
      i. Si $R_n < T$, la chaîne d'imbrication $R_1 \Subset_{\mathcal{F}} R_2 \Subset_{\mathcal{F}} \ldots \Subset_{\mathcal{F}} R_n \Subset_{\mathcal{F}} T \Subset_{\mathcal{F}} R$ permet de conclure : on a $\mathcal{P}_{\mathcal{F}}(R) \geq n + 1 > \mathcal{P}_{\mathcal{F}'}(R')$.
      ii. Si $T < R_n$, on a $\mathcal{P}_{\mathcal{F}'}(R'_n) = n-1$. Par hypothèse de récurrence, on obtient une suite $S_1, S_2, \ldots, S_n$ de radicaux de $\mathcal{F}$ qui vérifie $S_1 \Subset_{\mathcal{F}} S_2 \Subset_{\mathcal{F}} \ldots \Subset_{\mathcal{F}} S_n \Subset_{\mathcal{F}} R_n \Subset_{\mathcal{F}} R$, ce qui permet de conclure.
   (b) Si $R_n \ll_{\mathcal{F}} R$, il existe une suite $T_1, \ldots, T_m$ d'éléments de $\mathcal{F}$ qui vérifie $R_n \ll_{T_1 \ldots T_m} R$. Comme T_m et T contiennent R, deux positions relatives sont possibles.
      i. Si $T \leq T_m$, comme $T_m < R_n$, on a $T < R_n$. On retrouve le cas 2(a)ii. L'utilisation de l'hypothèse de récurrence sur R_n permet de conclure.
      ii. Si $T_m < T$, on a $R_n \ll_{T_1 \ldots T_m} T$. La chaîne $R_1 \Subset_{\mathcal{F}} R_2 \Subset_{\mathcal{F}} \ldots \Subset_{\mathcal{F}} R_n \Subset_{\mathcal{F}} T \Subset_{\mathcal{F}} R$ permet de conclure. □

Si T et R sont des radicaux de $\mathcal{F}$, alors, après la contraction de T, la profondeur $\mathcal{P}_{\mathcal{F}'}(R')$ d'un résidu R' de R (où $\mathcal{F}'$ est l'ensemble résidu de $\mathcal{F}$) est strictement plus petite que la profondeur de R. On illustre ce résultat à l'aide de l'exemple de la figure 2.14. Dans cet exemple, on a $M < R$

et $M < S < T \ll_{SM} R$. La profondeur de R est donc $\mathcal{P}_{\mathcal{F}}(R) = 3$. Après la contraction de M, on a $S_1 < T_1 \ll_{S_1} R_1$ et donc $\mathcal{P}_{\mathcal{F}_1}(R_1) = 2$. Cette profondeur a donc strictement diminué après la contraction de M. Cette propriété est exploitée pour montrer que les développements sont finis.

Théorème 2.8 (Développements finis) *Soit $\mathcal{F}$ un ensemble de radicaux de X. Toute réduction relative à $\mathcal{F}$ termine.*

Pour prouver ce résultat, on considère le multi-ensemble des profondeurs des radicaux de $\mathcal{F}$ au cours d'un développement. Dans le cadre de l'exemple de la figure 2.14, les radicaux de $\mathcal{F}$ vérifient $M < S < T \ll_{SM} R$, $M < S$ et $M < S < T$. Le multi-ensemble des profondeurs de $\mathcal{F} = \{M, R, S, T\}$ est donc $\mathcal{M} = \{\!\{0, 3, 1, 2\}\!\}$. Après la contraction de M, les radicaux de $\mathcal{F}_1 = \{S_1, T_1, R_1, R_2\}$ vérifient $S_1 < T_1$ et $S_1 < T_1 \ll_{S_1} R_i$ ($i \in \{1, 2\}$). De là, le multi-ensemble correspondant est $\mathcal{M}_1 = \{\!\{0, 1, 2, 2\}\!\}$. On a bien $\mathcal{M}_1 <_m \mathcal{M}$. Après la contraction de S_1, les radicaux de $\mathcal{F}_2 = \{T_2, ,R_{11}, R_{12}, R_{21}, R_{22}\}$ vérifient $T_2 < R_{ij}$ (i,$j \in \{1, 2\}$). Le multi-ensemble correspondant est $\mathcal{M}_2 = \{\!\{0, 1, 1, 1, 1\}\!\}$. On a bien $\mathcal{M}_2 <_m \mathcal{M}_1$.

Preuve : On considère la réduction $\mathcal{R} : X \xrightarrow{T}_{ve} X'$ où $T \in \mathcal{F}$. Soit $\mathcal{F}'$ l'ensemble des résidus de $\mathcal{F}$ par cette réduction. Pour chaque radical R de $\mathcal{F}$ dont l'ensemble des résidus (éventuellement vide) est $\{R'_1, \ldots, R'_n\}$, on compare les multi-ensembles $\mathcal{M} = \{\!\{\mathcal{P}_{\mathcal{F}}(R)\}\!\}$ et $\mathcal{M}' = \{\!\{\mathcal{P}_{\mathcal{F}}(R'_i) \mid i = 1 \ldots n\}\!\}$. On note $\leq_m$ l'ordre bien fondé sur les multi-ensembles d'entiers. On procède par cas sur les positions relatives de R et T.

1. Si $R < T$ ou si R et T sont disjoints, le radical R a un unique résidu R'_1 dans X'. D'après le lemme 2.33, on a $\mathcal{P}_{\mathcal{F}'}(R'_1) \leq \mathcal{P}_{\mathcal{F}}(R)$ ce qui donne bien $\mathcal{M}' \leq_m \mathcal{M}$.
2. Si $T < R$, deux situations sont possibles :
   (a) Si R n'a pas de résidu, on a $\mathcal{M}' = \emptyset <_m \mathcal{M}$.
   (b) Si R a n résidus $R'_1 \ldots R'_n$, alors on obtient, en utilisant le lemme 2.34, que ces derniers vérifient $\mathcal{P}_{\mathcal{F}'}(R'_i) < \mathcal{P}_{\mathcal{F}}(R)$ pour $1 \leq i \leq n$. On obtient donc $\mathcal{M}' <_m \mathcal{M}$.
3. Si $T = R$, le radical T n'a pas de résidu donc $\mathcal{M}' = \emptyset <_m \mathcal{M}$.

De cette façon, on obtient :

$$\{\!\{\mathcal{P}_{\mathcal{F}}(R) \mid R \in \mathcal{F}\}\!\} = \bigcup_{R \in \mathcal{F}} \{\!\{\mathcal{P}_{\mathcal{F}}(R)\}\!\} <_m \bigcup_{R \in \mathcal{F}} \{\!\{\mathcal{P}_{\mathcal{F}}(R') \mid R' \in R/\mathcal{R}\}\!\} = \{\!\{\mathcal{P}_{\mathcal{F}}(R') \mid R' \in \mathcal{F}'\}\!\}$$

Comme l'ordre $\leq_m$ est bien fondé, on obtient que les réductions relatives à $\mathcal{F}$ terminent. □

L'énoncé précédent est l'énoncé intitulé (FD) dans [7]. Pour obtenir l'énoncé (FD!) et retrouver un résultat correspondant au théorème 2.3, on prouve que les réductions relatives à un ensemble de radicaux vérifient une propriété de confluence locale. Pour prouver cette propriété, on reprend la technique employée pour montrer la confluence locale. On commence par montrer des propriétés de compatibilité des opérations de concaténation, de substitution et de soulignement avec la relation de réduction.

Lemme 2.35

1. *Si $X \xrightarrow{(C,R)}_{ve} X'$, alors $\alpha \cdot X \xrightarrow{(\alpha \cdot C, R)}_{ve} \alpha \cdot X'$.*
2. *Si $X \xrightarrow{(C,R)}_{ve} X'$, alors $X\{y \backslash Y\} \xrightarrow{(C^\star, R^\star)}_{ve} X'\{y \backslash Y\}$ où $C^\star = C\{y \backslash Y\}$ et $R^\star = R\{y \backslash Y\}$.*
3. *Si $Y \xrightarrow{(C,R)}_{ve} Y'$, alors $X\{y \backslash Y\} \xrightarrow{\mathcal{F}}_{ve} X\{y \backslash Y'\}$ avec $\mathcal{F} = \{(C_i[\gamma_1 \cdot C], R) \mid 1 \leq i \leq n\}$ où les occurrences libres de y dans X sont $\{(C_i[\,], y_i^\gamma) \mid 1 \leq i \leq n\}$.*
4. *Si $V \xrightarrow{(C,R)}_{ve} V'$, alors $\lfloor \alpha | V \rfloor \xrightarrow{(\lfloor \alpha | C \rfloor, R)}_{ve} \lfloor \alpha | V' \rfloor$.*

Preuve : On obtient ce résultat en adaptant les preuves employées 2.21, 2.22 et 2.23. □

Ces résultats, qui correspondent aux lemmes 2.21, 2.23 et 2.22, permettent d'obtenir la propriété suivante de permutation de l'ordre des radicaux contractés.

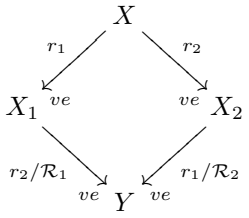


FIG. 2.16 – *Permutation locale*

Lemme 2.36 (Permutation locale) *Si* $\mathcal{R}_1 : X \xrightarrow{r_1}_{ve} X_1$ *et* $\mathcal{R}_2 : X \xrightarrow{r_2}_{ve} X_2$ *sont des réductions issues de* X*, il existe un terme* Y *tel que* $X_1 \xrightarrow{r_2/\mathcal{R}_1}_{ve} Y$ *et* $X_2 \xrightarrow{r_1/\mathcal{R}_2}_{ve} Y$*.*

Preuve : On pose $r_1 = (C_1,R_1)$ et $r_2 = (C_2,R_2)$. On procède par récurrence sur la taille de X.

1. Les cas $X = x^\alpha$ et $X = \Omega$ sont impossibles puisque $X \rightarrow_{ve} X_1$.
2. Si $X = (YZ)^\gamma$, on peut supposer, sans perte de généralité, que R_1 est à gauche de R_2. Trois cas sont à envisager.
   (a) Si $r_1 = ([\,],X)$, alors on a $X = ((\lambda x.Y')^\alpha V)^\gamma \rightarrow_v \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{x\backslash\lfloor\alpha|V\rfloor\} = X_1$.
      i. Si r_2 est de la forme $(((\lambda x.C_2'[\,])^\alpha V)^\gamma,R_2)$, on a $X \xrightarrow{r_2}_{ve} ((\lambda x.Y_2')^\alpha V)^\gamma = X_2$ avec $Y' \xrightarrow{(C_2',R_2)}_{ve} Y_2'$. On en déduit que le radical r_2 a un unique résidu r_2' dans X_1 qui vérifie $r_2' = (\gamma \cdot \lceil\alpha|\beta\rceil \cdot C_2'\{x\backslash\lfloor\alpha|V\rfloor\},R_2\{x\backslash\lfloor\alpha|V\rfloor\})$. On obtient, en utilisant les points (1) et (2) du lemme 2.35, $X_1 \xrightarrow{r_2'}_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y_2'\{x\backslash\lfloor\alpha|V\rfloor\}$. Par ailleurs, le radical r_1 a un unique résidu $r_1' = ([\,],((\lambda x.Y_2')^\alpha V)^\gamma)$ dans X_2. On obtient donc $X_2 \xrightarrow{r_1'}_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y_2'\{x\backslash\lfloor\alpha|V\rfloor\}$.
      ii. Si r_2 est de la forme $(((\lambda x.Y')^\alpha C_2'[\,])^\gamma,R_2)$, on a $X \xrightarrow{r_2}_{ve} (((\lambda x.Y')^\alpha V_2)^\gamma$ avec $V \xrightarrow{(C_2',R_2)}_{ve} V_2$. On en déduit que le radical r_2 a autant de résidus dans X_1 que la variable liée par X a d'occurrences libres dans Y'. Si ces n occurrences ($n \geq 0$) de x sont caractérisées, pour $i \in \{1 \ldots n\}$, par C^i, γ_i et $X = ((\lambda x.C^i[x^{\gamma_i}])^\alpha V)$, alors les radicaux $r_2^i = (\gamma \cdot \lceil\alpha|\beta\rceil \cdot C^i[\gamma_i \cdot C_2'[\,]],R_2)$ sont les résidus de r_2 dans X_1. On obtient, en utilisant les points (3) et (4) du lemme 2.35, $X_1 \xrightarrow{r_2/\mathcal{R}_1}_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{x\backslash\lfloor\alpha|V_2\rfloor\}$. Par ailleurs, r_1 a un unique résidu r_1' dans X_2 : $r_1' = ([\,],((\lambda x.Y')^\alpha V_2)^\gamma)$. On obtient donc $X_2 \xrightarrow{r_1'}_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot Y'\{x\backslash\lfloor\alpha|V_2\rfloor\}$.
   (b) Si Y contient r_1 et r_2, on a $r_1 = ((C_1'[\,]\,Z)^\gamma,R_1)$ et $r_2 = ((C_2'[\,]\,Z)^\gamma,R_2)$ et $X_1 = (Y_1Z)^\gamma$ et $X_2 = (Y_2Z)^\gamma$. On a aussi $\mathcal{S}_1 : Y \xrightarrow{s_1}_{ve} Y_1$ et $\mathcal{S}_2 : Y \xrightarrow{s_2}_{ve} Y_2$ où $s_1 = (C_1',R_1)$ et $s_2 = (C_2',R_2)$. On applique l'hypothèse de récurrence à Y. On obtient $Y_1 \xrightarrow{s_2/\mathcal{S}_1}_{ve} Y_3$ et $Y_2 \xrightarrow{s_1/\mathcal{S}_2}_{ve} Y_3$. Il existe une relation bijective entre les ensembles $r_1/\mathcal{R}_2$ et $s_1/\mathcal{S}_2$. Si $s = (D,S) \in s_1/\mathcal{S}_2$, alors $r = ((D[\,]\,Z)^\gamma,S) \in r_1/\mathcal{R}_2$. Réciproquement, si $r \in r_1/\mathcal{R}_2$, alors r est nécessairement de la forme $r = ((C[\,]Z)^\gamma,R)$ et $s = (C,R) \in s_1/\mathcal{S}_2$. On obtient donc $X_2 = (Y_2Z)^\gamma \xrightarrow{r_1/\mathcal{R}_2}_{ve} (Y_3Z)^\gamma$. Ce raisonnement s'applique aussi aux ensembles $r_2/\mathcal{R}_1$ et $s_2/\mathcal{S}_1$. On obtient $X_1 = (Y_1Z)^\gamma \xrightarrow{r_2/\mathcal{R}_1}_{ve} (Y_3Z)^\gamma$.
   (c) Si Y contient r_1 et Z contient r_2. On a $r_1 = ((C_1\ Z)^\gamma,R_1)$ et $r_2 = ((YC_2)^\gamma,R_2)$ et $X_1 = (Y_1Z)^\gamma$ et $X_2 = (YZ_2)^\gamma$. Le radical r_1 a un unique résidu r_1' dans X_2 qui vérifie $r_1' = ((C_1\ Z_2)^\gamma,R_1)$. On obtient $X_2 = (YZ_2)^\gamma \xrightarrow{r_1'}_{ve} (Y_1Z_2)^\gamma$. De même, r_2 a un unique résidu r_2' dans X_1 qui vérifie $r_2' = ((Y_1C_2)^\gamma,R_2)$. On obtient $X_1 = (Y_1Z)^\gamma \xrightarrow{r_2'}_{ve} (Y_1Z_2)^\gamma$.
   (d) Si Z contient r_1 et r_2, on procède de la même façon que dans le cas 2b.
3. Le cas $X = (\lambda x.X')^\alpha$ se traite de la même façon que le cas 2b. □

Ce résultat précise le résultat de confluence locale obtenu dans la partie précédente. Si X peut se réduire de façon élémentaire vers X_1 et X_2 en contractant, respectivement, les radicaux r_1 et r_2, alors en développant respectivement les résidus de r_2 et r_1, les réductions issues de X_1 et X_2 convergent vers un même terme Y. Cette propriété est illustrée sur la figure 2.16. De plus, dans ce cadre, la notion de résidu ne dépend pas de la réduction choisie, comme le montre le résultat suivant.

Lemme 2.37 *On suppose $\mathcal{R}_1 : X \xrightarrow{R_1}_{ve} X_1$ et $\mathcal{R}_2 : X \xrightarrow{R_2}_{ve} X_2$. Si S est un radical de X, on a $S/(\mathcal{R}_1; R_2/\mathcal{R}_1) = S/(\mathcal{R}_2; R_1/\mathcal{R}_2)$.*

Preuve : Le résultat s'obtient de façon élémentaire en inspectant toutes les positions possibles de S par rapport à R_1 et R_2. □

En combinant le théorème 2.8 et les lemmes 2.36 et 2.37, on obtient l'énoncé (FD!) du théorème des développements finis.

Théorème 2.9 (Développements finis) *Soit $\mathcal{F}$ un ensemble de radicaux de X.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Tous les développements de $\mathcal{F}$ finissent sur un même terme Y.*
3. *L'ensemble des résidus d'un radical R de X dans Y est indépendant du développement considéré.*

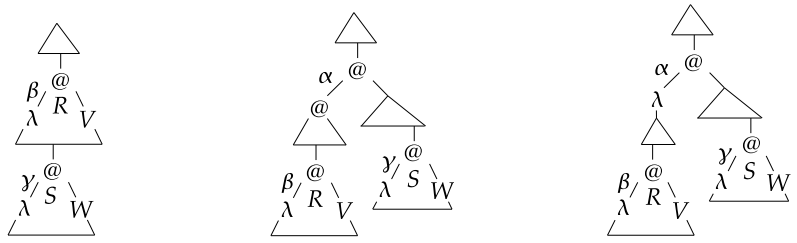
Preuve : Le premier point est obtenu par le théorème 2.8. Pour obtenir le deuxième point, un utilise une méthode itnroduite par Newman en 1942 [33]. On montre que s'il existe deux développements $\mathcal{R}_1$, $\mathcal{R}_2$ de $\mathcal{F}$ qui se terminent sur des termes distincts Y_1 et Y_2, alors il existe un terme X^1 et $\mathcal{F}'$ une famille de radicaux de X^1 tels que $X \xrightarrow{R}_{ve} X^1$ et $R \in \mathcal{F}$ et il existe deux développements de $\mathcal{F}'$ qui aboutissent à des termes distincts. Pour cela, on pose $\mathcal{R}_1 : X \xrightarrow{R_1}_{ve} X_1 \twoheadrightarrow_{ve} Y_1$ et $\mathcal{R}_2 : X \xrightarrow{R_2}_{ve} X_2 \twoheadrightarrow_{ve} Y_2$.

1. Si $R_1 = R_2$, alors $X_1 = X_2 = X^1$. Il existe donc deux développements de $\mathcal{F}' = \mathcal{F}/R_1$ qui aboutissent à des valeurs distinctes Y_1 et Y_2.
2. Si $R_1 \neq R_2$, on utilise le lemme 2.36. On obtient un terme X_3 tel que $X_1 \xrightarrow{R_2/R_1}_{ve} X_3$ et $X_2 \xrightarrow{R_1/R_2}_{ve} X_3$. Avec le lemme 2.37, on obtient $\mathcal{G} = \mathcal{F}/(R_1; (R_2/R_1)) = \mathcal{F}/(R_2; (R_1/R_2))$. On considère un développement $\mathcal{R}_3$ de $\mathcal{G}$. En utilisant le théorème des développements finis (FD), on sait que $\mathcal{G}$ se termine sur un terme Y_3.
   (a) Si $Y_3 \neq Y_1$, alors on pose $X^1 = X_1$ et $\mathcal{F}' = \mathcal{F}/R_1$. Il existe deux développements de $\mathcal{F}'$ qui se terminent sur Y_1 et Y_3.
   (b) Si $Y_3 = Y_1$, alors on pose $X^1 = X_2$ et $\mathcal{F}' = \mathcal{F}/R_2$. Il existe deux développements de $\mathcal{F}'$ qui se terminent sur Y_1 et Y_2.

On peut donc construire un développement infini $X \to_v X^1 \to_v X^2 \to_v \ldots$ de $\mathcal{F}$, ce qui contredit le théorème 2.8. Les développements se terminent donc tous sur un même terme Y. La combinaison du lemme 2.37 et du théorème des développements finis (FD) prouve le troisième point. □

2.2.3 Standardisation

Comme pour le λ-calcul par valeur, en utilisant la définition de réduction standard du λ-calcul classique, le théorème de standardisation n'est pas vérifié dans le λ-calcul par valeur étiqueté. Pour s'en convaincre, il suffit d'étiqueter les exemples des figures 2.4 et 2.5. L'utilisation de l'ordre strict $<_g$ des radicaux du λ-calcul ne permet pas d'obtenir une classe de réduction standard dans le λ-calcul par valeur étiqueté. Pour définir la classe des réductions standard du λ-calcul par valeur étiqueté, on adapte l'ordre strict des radicaux $<_{st}$ du λ-calcul par valeur à la syntaxe du λ-calcul par valeur étiqueté. Cet ordre strict est noté $<^e_{st}$.



1. $R <^e_{st} S$ 2a. $R <^e_{st} S$ 2b. $S <^e_{st} R$

FIG. 2.17 – *Ordre strict $<^e_{st}$ entre deux radicaux d'un terme*

1. Si R contient strictement S alors $R <^e_{st} S$.
2. Si R et S sont disjoints, il existe des contextes C, C_1 et C_2 tels que $M = C[(C_1[R]\, C_2[S])^\alpha]$.
   (a) Si $C_1[R]$ est une application, alors $R <^e_{st} S$.
   (b) Si $C_1[R]$ est une abstraction, alors $S <^e_{st} R$.

Cette définition est illustrée sur la figure 2.17. L'ordre associé est $\leq^e_{st}$. La définition de réduction standard est adaptée : la réduction $M_0 \xrightarrow{R_1} M_1 \xrightarrow{R_2}_{ve} M_2 \xrightarrow{R_3}_{ve} \cdots \xrightarrow{R_n}_{ve} M_n$ est standard pour $\leq^e_{st}$ si et seulement si pour tout i, j tels que $1 \leq i < j \leq n$ le radical R_j n'est pas un résidu d'un radical R'_j de X_{i-1} tel que $R'_j \leq^e_{st} R_i$. Cet ordre vérifie les mêmes propriétés que $\leq_g$ dans le λ-calcul classique et $\leq_{st}$ dans le λ-calcul par valeur.

Lemme 2.38 *On suppose $M \xrightarrow{S}_{ve} N$. Soit R un radical de M qui vérifie $R <^e_{st} S$.*

1. *R a un unique résidu R' dans N.*
2. *Si T' est un radical créé de N alors $R' <^e_{st} T'$.*
3. *Si T vérifie $R \leq^e_{st} T$, alors chaque résidu T' de T dans N vérifie $R' \leq^e_{st} T'$.*

Preuve : On adapte la preuve du lemme 2.9 à la syntaxe du λ-calcul par valeur étiqueté. □

Si $M \xrightarrow{S}_{ve} N$ et si R est un radical plus petit que S, alors R a un unique résidu dans N. De plus, les radicaux créés sont strictement plus grands que le résidu de R. Enfin, les résidus des radicaux plus grands que R sont plus grands que le résidu de R. Ce résultat s'étend, de façon élémentaire, aux familles de radicaux, comme le montre le corollaire suivant.

Corollaire 2.4 *On suppose $\mathcal{R} : M \xrightarrow{\mathcal{F}}_{ve} N$. Soit R un radical de M qui vérifie $R <^e_{st} \mathcal{F}$.*

1. *$R/\mathcal{R}$ est un singleton $\{R'\}$*
2. *Si T' est un radical de M' qui n'est pas résidu d'un radical de M, alors $R' <^e_{st} T'$.*
3. *Si T vérifie $R \leq^e_{st} T$ et si T' est un résidu de T dans N, alors on a $R' \leq^e_{st} T'$.*

Si $M \xrightarrow{\mathcal{F}}_{ve} N$ et si R est un radical plus petit que tous les éléments de $\mathcal{F}$, alors R a un unique résidu R' dans N. De plus, les résidus des radicaux créés au cours du développement de $\mathcal{F}$ sont strictement plus grands que le résidu de R. Enfin, les résidus des radicaux plus grands que R sont plus grands que le résidu de R. Ce résultat, qui correspond exactement au corollaire 2.2 introduit dans le cadre du λ-calcul par valeur, constitue, avec le théorème des développements finis, la propriété fondamentale qui permet d'obtenir le théorème de standardisation.

Théorème 2.10 (Standardisation) *Si $X \twoheadrightarrow_{ve} Y$, il existe une réduction $\mathcal{R} : X \twoheadrightarrow_{ve} Y$ telle que $\mathcal{R}$ est standard pour $\leq^e_{st}$.*

Preuve : On adapte la preuve du théorème 2.4 en utilisant le corollaire 2.4 et le théorème 2.9 en lieu et place du corollaire 2.2 et du théorème 2.3. □

Dans le λ-calcul par valeur étiqueté, le théorème de standardisation est vérifié : si X se réduit en Y alors Y est atteignable par une réduction standard pour $\leq^e_{st}$ issue de X.

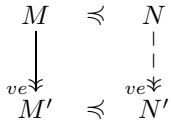


FIG. 2.18 – *Monotonie*

2.2.4 Stabilité

Dans le λ-calcul étiqueté, d'après le résultat 1.15, si $X \twoheadrightarrow_e V$ et si toutes les étiquettes de X sont des lettres distinctes, alors l'étiquette de tête de V caractérise le plus petit préfixe de X qui se réduit vers une valeur : ce préfixe est obtenu à partir de X en effaçant les sous-termes de X qui sont étiquetés par une lettre n'intervenant pas dans $\tau(V)$. Comme nous l'avons observé au début de cette section, cette propriété n'est plus vraie si on ne considère que des réductions par valeur, où la partie droite des radicaux contractés sont des valeurs. Ceci nous a conduit à introduire de nouvelles étiquettes pour retrouver cette propriété en tenant compte des spécificités de la réduction par valeur. Alors que dans le λ-calcul étiqueté, le nom d'un radical $R = ((\lambda x.X)^\alpha V)^\beta$ est l'étiquette α de l'abstraction, dans le λ-calcul par valeur étiqueté, le nom de R est constitué de la juxtaposition $\alpha|\tau(V)$ de l'étiquette de l'abstraction et de la valeur. Cette juxtaposition rend compte du fait que la création du radical R dépend à la fois du calcul de l'abstraction $(\lambda x.X)^\alpha$ et du calcul de la valeur V.

Dans cette partie, on montre non seulement que le λ-calcul par valeur étiqueté vérifie les propriétés de monotonie et de stabilité, mais aussi que les étiquettes expriment la propriété de stabilité de la même manière que pour le λ-calcul étiqueté. Pour exprimer les propriétés de monotonie et de stabilité, nous utilisons une relation de préfixe $\preccurlyeq$.

$$\begin{array}{rcll} x^\alpha & \preccurlyeq & x^\alpha & \\ \Omega & \preccurlyeq & X & \\ (\lambda x.X)^\alpha & \preccurlyeq & (\lambda x.X')^\alpha & \text{si } X \preccurlyeq X' \\ (XY)^\alpha & \preccurlyeq & (X'Y')^\alpha & \text{si } X \preccurlyeq X' \text{ et } Y \preccurlyeq Y' \end{array}$$

Cette relation est une simple adaptation de la relation employée dans le λ-calcul étiqueté à la syntaxe du λ-calcul par valeur étiqueté. Avec cette relation, on retrouve de façon assez directe la propriété de monotonie du lemme 2.12.

Lemme 2.39 (Monotonie) *Si $X \preccurlyeq Y$ et $X \twoheadrightarrow_{ve} X'$ alors il existe un terme Y' tel que $Y \twoheadrightarrow_{ve} Y'$ et $X' \preccurlyeq Y'$.*

Preuve : On adapte la preuve du lemme 2.12. □

La réduction par valeur étiquetée est monotone : si X se réduit vers X' et minore un terme Y, alors Y se réduit vers un terme Y' qui est minoré par Y. Cette propriété est illustrée sur la figure 2.18. De même, le théorème de stabilité s'obtient de la même façon que dans le λ-calcul par valeur. On utilise de façon cruciale le théorème de standardisation. En particulier, si $X \twoheadrightarrow_{ve} V$, on isole les réductions qui contribuent à l'obtention d'une valeur grâce au résultat suivant, qui correspond au lemme 2.16

Lemme 2.40 *Si $\mathcal{R} : M \twoheadrightarrow_{ve} V$ est standard, alors $\mathcal{R}$ peut se décomposer en $\mathcal{R} = \mathcal{R}_t;\mathcal{R}_s$ où $\mathcal{R}_t : M \twoheadrightarrow_{ve} V'$ est une réduction de tête et $\mathcal{R}_s : V' \twoheadrightarrow_{ve} V$ est standard.*

Preuve : On adapte la preuve du lemme 2.16 à la syntaxe du λ-calcul par valeur étiqueté. □

Une réduction standard $\mathcal{R}$ qui mène à une valeur peut se décomposer en la composition d'une réduction en tête $\mathcal{R}_t$ qui mène à une valeur avec une réduction standard $\mathcal{R}'$. Seule la réduction $\mathcal{R}_t$ contribue à l'obtention d'une valeur. Les radicaux contractés au cours de $\mathcal{R}'$ sont internes à

la valeur obtenue à l'issue de $\mathcal{R}_t$. Cette propriété alliée au théorème de standardisation permet d'obtenir le théorème de stabilité.

Théorème 2.11 (Stabilité) *Si $M \twoheadrightarrow_{ve} V$, il existe un préfixe X de M tel que, pour tout Y, si $Y \preccurlyeq M$ et $Y \twoheadrightarrow_{ve} V'$, on a $X \preccurlyeq Y$.*

Preuve : La preuve du théorème 2.5 est adaptée. □

Si un terme M se réduit vers une valeur, il existe un plus petit préfixe X de M qui se réduit vers une valeur. On observe que l'étiquette de tête de la valeur obtenue en partant de X est la même que l'étiquette de tête de V. Les résultats obtenus jusqu'à maintenant sont de simples adaptations des résultats obtenus dans le cadre du λ-calcul par valeur. La suite de cette partie est consacrée à prouver un résultat correspondant au théorème 1.15 : on montre que les étiquettes du λ-calcul par valeur étiqueté expriment la stabilité. Pour cela, on rappelle, en les adaptant, les définitions de l'opérateur $|\cdot|$ et du A-préfixe $[\![\cdot]\!]_A$.

$$\begin{array}{rcll} [\![X]\!]_A &=& \Omega & \text{si } |\tau(M)| \not\subseteq A \\ [\![x^\alpha]\!]_A &=& x^\alpha & \text{si } |\alpha| \subseteq A \\ [\![(\lambda x.X)^\alpha]\!]_A &=& (\lambda x.[\![X]\!]_A)^\alpha & \text{si } |\alpha| \subseteq A \\ [\![(XY)^\alpha]\!]_A &=& ([\![X]\!]_A\,[\![Y]\!]_A)^\alpha & \text{si } |\alpha| \subseteq A \\ [\![\Omega]\!]_A &=& \Omega & \end{array} \qquad \begin{array}{rcl} |a| &=& \{a\} \\ |\lceil\alpha|\beta\rceil| &=& |\alpha| \cup |\beta| \\ |\lfloor\alpha|\beta\rfloor| &=& |\alpha| \cup |\beta| \\ |\alpha\beta| &=& |\alpha| \cup |\beta| \end{array}$$

Si α est une étiquette, $|\alpha|$ en l'ensemble des lettres qui interviennent dans α. Si A est un ensemble de lettres et si X est un terme, $[\![X]\!]_A$ est le plus grand préfixe de X dont les étiquettes de tous les sous-termes sont constituées de lettres de A. Ces opérations vérifient les propriétés suivantes.

Lemme 2.41 *1. Si $|\alpha| \subseteq A$, alors $\alpha \cdot [\![X]\!]_A = [\![\alpha \cdot X]\!]_A$.*
2. Si $|\alpha| \subseteq A$, alors $\lfloor\alpha|[\![X]\!]_A\rfloor = [\![\lfloor\alpha|W\rfloor]\!]_A$.
3. $[\![X\{x\backslash V\}]\!]_A = [\![X]\!]_A\{x\backslash[\![V]\!]_A\}$

Preuve : On montre successivement ces trois propriétés.

1. On procède par cas, en posant $Y_1 = \alpha \cdot [\![X]\!]_A$ et $Y_2 = [\![\alpha \cdot X]\!]_A$.
   (a) Si $X = x^\beta$ et si $|\beta| \subseteq A$, on a $Y_1 = x^{\alpha\beta} = Y_2$. Si $|\beta| \not\subseteq A$, on a $Y_1 = \Omega = Y_2$.
   (b) Les autres cas se traitent de façon similaire.
2. On procède par cas comme pour le cas précédent.
3. On pose $Z_1 = [\![X]\!]_A\{x\backslash[\![V]\!]_A\}$ et $Z_2 = [\![X\{x\backslash V\}]\!]_A$. On procède par induction sur X.
   (a) Si $[\![X]\!]_A = \Omega$, c'est-à-dire $|\tau(X)| \not\subseteq A$ ou $X = \Omega$, alors $Z_1 = Z_2 = \Omega$.
   (b) Le cas $X = y^\alpha$ où $x \neq y$ est élémentaire.
   (c) Si $X = x^\alpha$ avec $|\alpha| \subseteq A$, alors le premier point donne $Z_1 = \alpha \cdot [\![V]\!]_A = [\![\alpha \cdot V]\!]_A = Z_2$.
   (d) Les cas $X = (\lambda y.Y)^\alpha$ et $X = (X_1X_2)^\alpha$ se traitent par hypothèse d'induction. □

Le premier point indique que, si $|\alpha| \subseteq A$, l'opération de concaténation avec α commute avec l'opération de A-préfixe. De façon très liée, l'opération de soulignement par α commute avec l'opération de A-préfixe. Enfin, le troisième point donne une propriété de commutation entre la substitution et l'opération de A-préfixe. Ces propriétés syntaxiques sont exploitées dans la preuve du lemme intermédiaire suivant.

Lemme 2.42 *Si $X \twoheadrightarrow_{ve} V$, alors il existe une valeur V' telle que $[\![X]\!]_{|\tau(V)|} \twoheadrightarrow_{ve} V'$.*

Preuve : D'après le lemme 2.40, il existe une réduction en tête $\mathcal{R} : X \twoheadrightarrow_{ve} W$. On peut supposer que W est la première valeur atteinte au cours de la réduction $\mathcal{R}$. Soit A un ensemble d'étiquettes tel que $|\tau(W)| \subseteq A$. On montre $[\![X]\!]_A \twoheadrightarrow_{ve} [\![W]\!]_A$. On procède par récurrence sur la longueur n de cette réduction.

1. Si $n = 0$, alors le résultat est trivial.

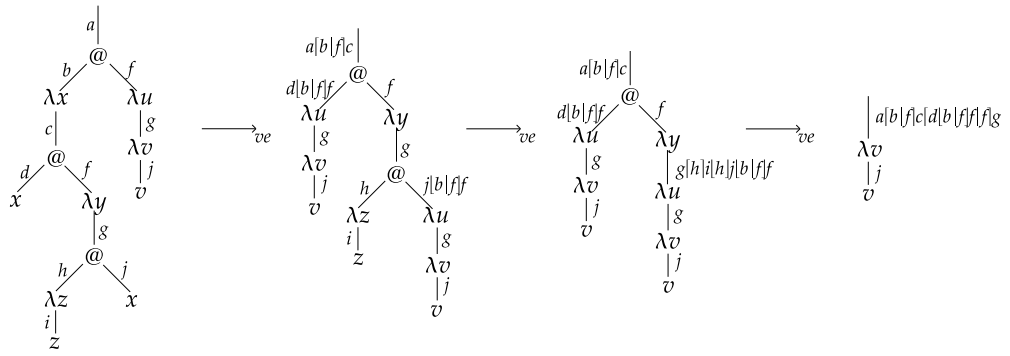


FIG. 2.19 – *Réduction de* $M = ((\lambda x.(x^d(\lambda y.((\lambda z.z^i)^h x^j)^g)^f)^c)^b(\lambda u.(\lambda v.v^j)^g)^f)^a$

2. Si $n > 0$, on procède par cas sur la forme de X.
   (a) Les cas $X = x^\alpha$ ou $X = \Omega$ sont impossibles car ils contredisent l'hypothèse $X \twoheadrightarrow_{ve} W$.
   (b) Le cas $X = (\lambda x.X_1)^\alpha$ est impossible puisque $\mathcal{R}$ est de longueur strictement positive et W est la première valeur atteinte au cours de cette réduction.
   (c) Si $X = (X_1X_2)^\gamma$, alors comme $\mathcal{R}$ est une réduction en tête qui aboutit à une valeur, la réduction $\mathcal{R}$ se décompose de la façon suivante.

$$\begin{aligned}\mathcal{R} : X &\twoheadrightarrow_{ve} ((\lambda x.X_3)^\alpha X_2)^\gamma && \text{avec } \beta = \tau(V)\\ &\twoheadrightarrow_{ve} ((\lambda x.X_3)^\alpha W_1)^\gamma\\ &\rightarrow_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_3\{x\backslash\lfloor\alpha|W_1\rfloor\}\\ &\twoheadrightarrow_{ve} W\end{aligned}$$

   La réduction $\mathcal{R}_1 : X_1 \twoheadrightarrow_{ve} (\lambda x.X_3)^\alpha$ est une réduction en tête dont la valeur finale est la première valeur atteinte. De même pour les réductions $\mathcal{R}_2 : X_2 \twoheadrightarrow_{ve} W_1$ et $\mathcal{R}_3 : \gamma \cdot \lceil\alpha|\beta\rceil \cdot X_3\{x\backslash\lfloor\alpha|W_1\rfloor\} \twoheadrightarrow_{ve} W$. Ces réductions en tête sont de longueur strictement inférieures à n. Comme α et β sont des sous-étiquettes de $\tau(W)$, on a $|\alpha| \subseteq A$ et $|\beta| \subseteq A$. De là, on applique l'hypothèse de récurrence aux réductions $\mathcal{R}_1$ et $\mathcal{R}_2$. On obtient $[\![X_1]\!]_A \twoheadrightarrow_{ve} (\lambda x.[\![Y_3]\!]_A)^\alpha$ et $[\![X_2]\!]_A \twoheadrightarrow_{ve} [\![W_1]\!]_A$. Comme γ est une sous-étiquette de $\tau(W)$, on a $|\gamma| \subseteq A$. On obtient donc la réduction suivante.

$$\begin{aligned}[\![X]\!]_A = ([\![X_1]\!]_A[\![X_2]\!]_A)^\gamma &\twoheadrightarrow_{ve} ((\lambda x.[\![X_3]\!]_A)^\alpha[\![W_1]\!]_A)^\gamma\\ &\rightarrow_{ve} \gamma \cdot \lceil\alpha|\beta\rceil \cdot [\![X_3]\!]_A\{x\backslash\lfloor\alpha|[\![W_1]\!]_A\rfloor\}\end{aligned}$$

   Le lemme 2.41 donne $\gamma \cdot \lceil\alpha|\beta\rceil \cdot [\![X_3]\!]_A\{x\backslash\lfloor\alpha|[\![W_1]\!]_A\rfloor\} = [\![\gamma \cdot \lceil\alpha|\beta\rceil \cdot X_3\{x\backslash\lfloor\alpha|W_1\rfloor\}]\!]_A$. On conclut par hypothèse de récurrence sur la réduction $\mathcal{R}_3$. □

Si un terme X se réduit vers une valeur V, le $|\tau(V)|$-préfixe de X se réduit également vers une valeur. Cette propriété signifie intuitivement que l'étiquette de tête de la valeur obtenue V définit un sur-ensemble des sous-termes de X qui contribuent, au cours de la réduction de X, à l'obtention d'une valeur. Pour illustrer cette propriété, nous examinons la réduction par valeur étiquetée du terme $M = ((\lambda x.(x^d(\lambda y.((\lambda z.z^i)^h x^j)^g)^f)^c)^b(\lambda u.(\lambda v.v^j)^g)^f)^a$. Cette réduction est illustrée sur la figure 2.19. Le préfixe obtenu à partir de l'étiquette de tête $a\lceil b|f\rceil c\lceil d\lfloor b|f\rfloor f|f\rceil g$ de la valeur finale est $X = ((\lambda x.(x^d(\lambda y.(\Omega\Omega)^g)^f)^c)^b(\lambda u.(\lambda v.\Omega)^g)^f)^a$. Le préfixe de stabilité de M est $\mathcal{P}_S(M) = ((\lambda x.(x^d(\lambda y.\Omega)^f)^c)^b(\lambda u.(\lambda v.\Omega)^g)^f)^a$. On a bien $\mathcal{P}_S(M) \preccurlyeq X$. En ajoutant la contrainte `INIT` sur le terme initial, on obtient le résultat suivant qui montre que les étiquettes permettent de déterminer exactement le préfixe de stabilité.

Théorème 2.12 *Si X vérifie* `INIT` *et si $X \twoheadrightarrow_{ve} V$, alors on a $\mathcal{P}_S(X) = [\![X]\!]_{|\tau(V)|}$.*

Preuve : On pose $A = |\tau(V)|$. Par le lemme 2.42, on obtient qu'il existe une valeur W telle que $[\![X]\!]_A \twoheadrightarrow_{ve} W$. Si Y est un préfixe de X qui vérifie $Y \twoheadrightarrow_{ve} V'$, alors les propriétés de monotonie et

de confluence du langage impliquent que l'étiquette de tête de V' est la même que celle de W et V. On en déduit que Y contient au moins les étiquettes de A, ce qui signifie, en utilisant l'hypothèse `INIT`, que l'on a la relation $[\![X]\!]_A \preccurlyeq Y$. □

Si les étiquettes de X sont des lettres distinctes et si X se réduit vers une valeur V, le préfixe de stabilité de X est le préfixe associé aux lettres de $\tau(V)$. En d'autres mots, ce préfixe est le préfixe minimum de X qui se réduit vers une valeur. Ce résultat signifie que les étiquettes du λ-calcul par valeur expriment la propriété de stabilité.

Dans ce chapitre, nous avons examiné le λ-calcul par valeur qui est une variante du λ-calcul dans laquelle les radicaux réduits ont pour membres droits des valeurs. Ces réductions particulières ont un intérêt pratique puisque dans des langages comme Objective Caml ou SML/NJ, les arguments appliqués à une fonction sont réduits vers des valeurs avant que l'application soit elle-même réduite. Nous avons montré que le λ-calcul par valeur est localement confluent et confluent. Le théorème des développements finis est directement hérité du λ-calcul. Le λ-calcul par valeur vérifie également le théorème de standardisation mais la définition de réduction standard dans le λ-calcul par valeur n'est pas la même que dans le λ-calcul. Cette différence s'explique par le fait qu'il existe trois façons de créer un radical dans le λ-calcul par valeur. En plus des créations *par le haut* et *par le bas* déjà présentes dans le λ-calcul, il existe une création *par la droite* lorsque le membre droit d'une application devient une valeur. Ce nouveau cas de création a aussi des conséquences pour la propriété de stabilité du λ-calcul par valeur. Alors que dans le cas du λ-calcul, les étiquettes permettaient de déterminer le préfixe de stabilité, cette propriété n'est plus vraie dans le λ-calcul par valeur. Pour retrouver cette relation entre étiquettes et stabilité, nous avons introduit de nouvelles étiquettes. Le langage étiqueté obtenu reste localement confluent et confluent. L'examen des propriétés fondamentales du λ-calcul par valeur étiqueté nous fournit l'occasion d'exposer une démonstration intuitive et élégante du théorème des développements finis. Cette preuve se fonde sur une relation d'imbrication étendue entre radicaux qui tient compte à la fois des imbrications présentes et des imbrications futures qui pourraient être créées au cours d'un développement. Enfin, le λ-calcul par valeur étiqueté vérifie le théorème de standardisation pour une notion de réduction standard directement adaptée de celle introduite pour le λ-calcul par valeur. Comme souhaité, nous avons montré que les étiquettes du λ-calcul par valeur expriment bien la notion de stabilité. Cette propriété sera exploitée dans le chapitre 5 au moment de l'examen de la propriété de non-interférence dans le cadre du λ-calcul par valeur : nous montrerons que les notions d'interférence et de sous-terme critique ne coïncident pas.

Chapitre 3

λ-calcul faible et étiquettes

Jusqu'à présent, nous avons considéré des calculs dits *forts*, c'est-à-dire qui autorisent la règle de contexte ξ :

$$(\xi)\ \frac{M \to N}{\lambda x.M \to \lambda x.N}$$

Par contraste, un calcul *faible* n'autorise pas cette règle ; les contractions internes aux abstractions ne sont pas permises. En cela, le calcul faible est plus proche des implémentations des langages fonctionnels [28, 40]. Pourtant, si le calcul fort a fait l'objet de nombreux travaux, le calcul faible n'a pas autant été étudié. La notion de partage dans le λ-calcul fort a notamment été étudiée en profondeur dans les multiples travaux d'Abadi, Asperti, Coppola, Gonthier, Guerrini, Lamping, Lawall, Lévy, Mairson, Martini [5, 6, 18, 25, 27, 29]. Pour exprimer le partage dans le calcul fort, les termes sont représentés par des graphes dont les sous-contextes peuvent être partagés. Par exemple, dans le terme $(\lambda x.xa(xb))(\lambda y.Iy)$ où $I = \lambda x.x$, il est nécessaire de partager le radical Iy indépendamment de la valeur de y. Ainsi, on doit partager le sous-contexte $I[\ ]$. Après la réduction des radicaux externes, on obtient $(\lambda y.Iy)a((\lambda y.Iy)b)$ puis $Ia((\lambda y.Iy)b)$ où le sous-contexte (partagé) $I[\ ]$ est instancié avec deux sous-termes différents : a et y. En utilisant les notations de Lamping, le partage d'un sous-contexte se fait par un nœud *fan-in* qui multiplexe des arêtes provenant des termes qui l'utilisent. Les nœuds *fan-out* permettent de remplir les vides de ces sous-contextes par une opération de démultiplexage. Les arêtes sortant du nœud fan-out pointent sur les termes correspondant à ces trous. Lamping a élaboré un calcul optimal opérant sur ces graphes. Les réductions de son calcul obéissent à la *sémantique des contextes* définie par Gonthier, Abadi et Lévy [18].

Dans le calcul faible, les contractions ne peuvent se produire sous une abstraction. Ainsi, dans le précédent exemple, le sous-terme Iy de $(\lambda x.xa(xb))(\lambda y.Iy)$ ne peut être réduit puisqu'il est contenu par l'abstraction $\lambda y.Iy$. Plus généralement, un sous-terme ne pourra être réduit que s'il ne contient plus de variable liée par un radical externe. Par conséquent, les sous-contextes ne sont plus nécessaires pour partager les sous-termes. On peut représenter les termes avec partage simplement avec des graphes acycliques orientés, au lieu de la structure cyclique utilisée par Lamping. Dans [43], Wadsworth décrit deux algorithmes pour la réduction de termes dans le λ-calcul faible avec partage. Dans le premier algorithme, les arguments des radicaux contractés sont partagés. Cependant, si une abstraction partagée est impliquée dans un radical R, cette abstraction est dupliquée avant de contracter R. Pour son deuxième algorithme, Wadsworth remarque qu'il n'est pas nécessaire de dupliquer l'abstraction en entier : les sous-termes qui ne contiennent pas la variable liée peuvent rester partagés. Des duplications inutiles sont donc évitées. La figure 3.1 illustre ces deux algorithmes pour la réduction de $M = (\lambda x.(xy)(xz))\lambda u.Iu$. Le premier radical contracté est $R_1 = M$. Dans le terme obtenu par les deux algorithmes, l'argument $\lambda u.(\lambda v.v)u$ du

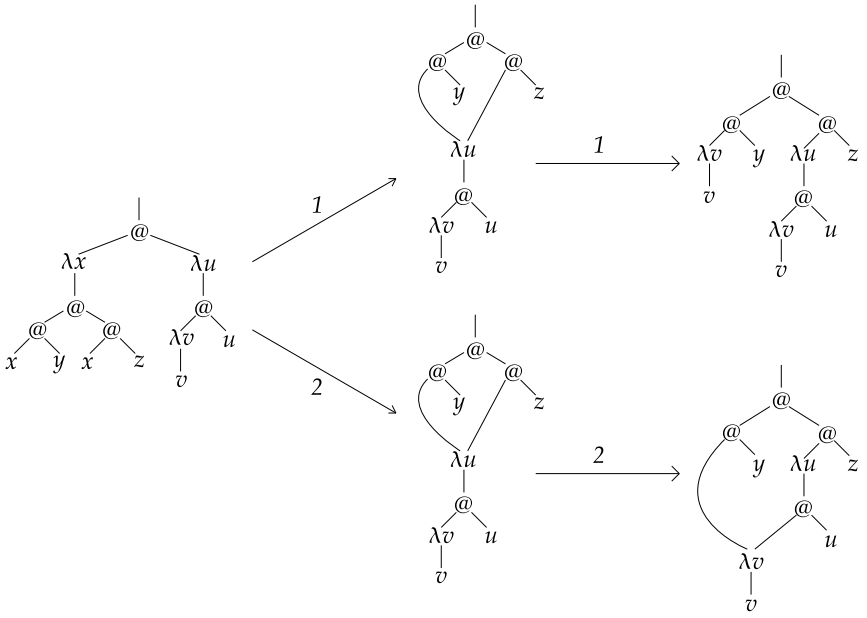


FIG. 3.1 – *Réductions de* $M = (\lambda x.(xy)(xz))\lambda u.Iu$ *par les deux algorithmes de Wadsworth*

radical contracté est partagé. Le deuxième radical contracté est $R_2 = (\lambda u.(\lambda v.v)u)y$. L'abstraction de ce radical est partagée. Avec le premier algorithme, cette dernière est entièrement départagée avant la réduction. Avec le deuxième algorithme, seul le contexte $\lambda u.([\,])u$ (le contexte minimal qui contient toutes les variables liées par l'abstraction) est départagé. Cet algorithme plus fin permet de maintenir le partage du sous-terme $\lambda v.v$ après la réduction, au contraire du premier algorithme. En s'inspirant des conclusions de Wadsworth, Shivers et Wand [38] proposent une implémentation réaliste qui représente les termes sous forme de graphes. Dans ces graphes, le nœud correspondant à une abstraction $\lambda x.M$ est non seulement relié au sous-terme correspondant à M mais aussi aux occurrences de x dans M. Les nœuds des termes sont reliés entre eux par des arêtes dans les deux sens. Cette représentation facilite la duplication des sous-termes : cette duplication s'effectue en partant des occurrences de la variable liée vers le lieur. Ceci permet d'offrir une implémentation efficace du deuxième algorithme de Wadsworth.

Dans cette partie, on présente un λ-calcul faible étiqueté qui s'inspire du deuxième algorithme de Wadsworth. Le but est d'obtenir un langage qui vérifie les mêmes propriétés fondamentales que le λ-calcul fort et dont les étiquettes expriment simplement la notion de partage tout en restant dans une syntaxe de terme habituelle : on souhaite éviter une représentation sous forme de graphes, moins commode. Dans la section 3.1, on examine les propriétés élémentaires du λ-calcul faible. On considère une variante de ce calcul qui est confluente et qui vérifie les théorèmes des développements finis et de standardisation. Dans la section 3.2, on introduit les étiquettes dans cette variante du λ-calcul faible. On montre dans un premier temps que les propriétés fondamentales du langage sont conservées. Dans un deuxième temps, on prouve, dans la section 3.3, que les étiquettes choisies expriment le partage : sous certaines conditions initiales, deux sous-termes ayant la même étiquette sont égaux.

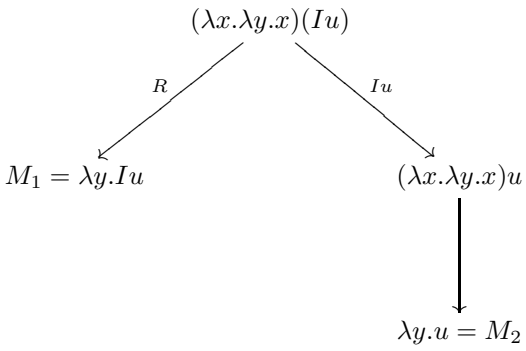


FIG. 3.2 – *Réductions de* $R = (\lambda x.\lambda y.x)(Iu)$ *dans le* λ*-calcul faible*

3.1 Le λ-calcul faible

Avant de s'intéresser à la notion de partage, nous nous penchons tout d'abord sur les propriétés élémentaires du λ-calcul faible. Si on se contente de soustraire au λ-calcul fort la règle de contexte (ξ), on obtient un calcul qui n'est pas confluent. Pour s'en convaincre, on observe les réductions du terme $R = (\lambda x.\lambda y.x)(Iu)$ qui sont illustrées sur la figure 3.2. Le terme R se réduit vers les termes M_1 et M_2. Ces derniers sont à la fois distincts et en formes normales. Cette non-confluence s'explique par le fait que le sous-terme Iu qui est initialement un radical de R, est, après contraction de R, *gelé* dans M_1 sous l'abstraction $\lambda y.Iu$. Le sous-terme Iu de M_1 n'est plus un radical. Intuitivement, ce sous-terme Iu de M_1 pourrait pourtant être considéré comme un radical puisqu'il ne dépend pas véritablement de l'abstraction qui le contient, dans la mesure où il ne contient pas la variable y liée par ce radical. En effet, le sous-terme Iu a été injecté sous l'abstraction par une substitution. Cette intuition est exploitée par Lévy et Maranget. Ils proposent dans [30] une variante du λ-calcul faible qui est inspirée par les travaux de Çağman et Hindley pour la Logique Combinatoire [11]. Dans cette variante, une nouvelle règle de contexte est ajoutée.

$$(\sigma)\ \frac{N \to N' \quad M \text{ linéaire en } x}{M\{x\backslash N\} \to M\{x\backslash N'\}}$$

Dans cette règle, la condition "M linéaire en x" signifie qu'il existe une unique occurrence de x dans M. L'usage de la substitution reflète le fait que N ne contient pas de variables liées par une abstraction de M. Ceci signifie que N ne dépend ni des abstractions de M, ni de leur argument potentiel. Ces conditions rejoignent l'intuition issue de l'exemple mentionné plus haut. Par ailleurs, la condition de linéarité évite de considérer des pas de réduction parallèles. Dans cette partie, au lieu de la règle (σ), on utilise une approche voisine. Les règles de notre variante du λ-calcul faible sont les suivantes.

$$(\beta_w)\ \ R = (\lambda x.M)N \xrightarrow{R}_w M\{x\backslash N\} \qquad (\nu_w)\ \frac{M \xrightarrow{R}_w M'}{MN \xrightarrow{R}_w M'N}$$

$$(\xi'_w)\ \frac{M \xrightarrow{R}_w M' \quad x \notin \mathtt{FV}(R)}{\lambda x.M \xrightarrow{R}_w \lambda x.M'} \qquad (\mu_w)\ \frac{N \xrightarrow{R}_w N'}{MN \xrightarrow{R}_w MN'}$$

La règle de base (β_w) est conservée du λ-calcul fort. On note simplement que la réduction $\xrightarrow{R}_w$ est annotée avec le radical contracté. En cas d'absence d'ambiguïté, on omettra cette annotation. Aux règles classiques (μ_w) et (ν_w), on ajoute une nouvelle règle (ξ'_w). Par définition de cette règle, le corps d'une abstraction peut être réduit seulement si le radical contracté ne contient pas la

variable liée par l'abstraction. Ceci montre la correction de la notation $\xrightarrow{R}_w$ puisque l'α-conversion ne change pas le radical R qui annote la réduction. Au lieu d'utiliser la substitution pour exprimer l'absence de variable liée dans un sous-terme d'une abstraction, cette règle (ξ'_w) exprime plus directement cette contrainte. De là, les radicaux de cette variante du λ-calcul faible sont donc les radicaux $(\lambda x.M)N$ du λ-calcul fort qui ne contiennent pas de variable liée. Ces radicaux vérifient la propriété suivante.

Lemme 3.1 *On suppose $M \xrightarrow{S} M'$ où S est un β_w-radical. On a $M \xrightarrow{S}_w M'$ et les β-résidus d'un β_w-radical R sont des β_w-radicaux de M'.*

Preuve : Comme R est un β_w-radical de M, il ne contient pas de variable liée par une abstraction externe. Par conséquent, ses β-résidus sont des β-radicaux qui vérifient la même propriété : ce sont donc des β_w-radicaux. □

Cette propriété permet de donner la définition suivante de β_w**-résidu** : si $M \xrightarrow{S}_w M'$, les β_w-résidus d'un β_w-radical R de M dans M' sont les β-résidus de R dans M'.

Comparativement au λ-calcul fort, on observe qu'il existe une nouvelle façon de créer des radicaux dans le λ-calcul faible en plus des créations par le haut et par le bas. Après avoir contracté $M = (\lambda x.Ix)y$, nous obtenons un radical Iy qui n'est pas un résidu d'un radical de M et qui n'est pas un radical créé au sens du λ-calcul fort. En réalité, si le sous-terme Ix est un radical pour le calcul fort, il n'est pas un radical pour le calcul faible : il est *gelé* dans M par l'occurrence de x. La contraction de l'abstraction contenant Ix crée le radical Iy en le dégelant.

L'introduction de la règle (ξ'_w) permet d'obtenir un calcul localement confluent. Comme dans le cas du calcul fort, la preuve de confluence locale s'appuie essentiellement sur les propriétés suivantes qui mettent en jeu la substitution et la réduction.

Lemme 3.2 *1. Si $M \xrightarrow{R}_w M'$, alors $M\{x\backslash N\} \xrightarrow{R\{x\backslash N\}}_w M'\{x\backslash N\}$.*
2. Si $N \xrightarrow{R}_w N'$, alors $M\{x\backslash N\} \xrightarrow{R}\!\!\!\!\rightarrow_w M\{x\backslash N'\}$.

Preuve : On prouve successivement les deux points.

1. Comme pour la preuve du lemme 2.4, on prouve ce point par récurrence sur la taille de M. Par rapport à cette preuve, il existe une nouveauté : si $M \xrightarrow{R}_w M'$ par la règle (ξ'). Dans ce cas, on a $M = \lambda y.M_1$, $M' = \lambda y.M'_1$, $M_1 \xrightarrow{R}_w M'_1$ et $y \notin \mathtt{FV}(R)$. Par hypothèse de récurrence, on obtient $M_1\{x\backslash N\} \xrightarrow{R\{x\backslash N\}}_w M'_1\{x\backslash N\}$. En renommant éventuellement y, on peut supposer $y \notin \mathtt{FV}(N)$. De là, on obtient la relation $y \notin R\{x\backslash N\}$ et, par la règle (ξ'), on a $M\{x\backslash N\} = \lambda y.M_1\{x\backslash N\} \xrightarrow{\mathtt{FV}(R\{x\backslash N\})}_w \lambda y.M'_1\{x\backslash N\} = M'\{x\backslash N\}$.
2. Le deuxième point du lemme se prouve par induction sur M en utilisant l'hypothèse selon laquelle les radicaux réduits sont bien des termes R.
   (a) Les cas $M = x$ et $M = y$ avec $y \neq x$ sont élémentaires.
   (b) Si $M = M_1M_2$, on obtient par hypothèse d'induction $M_1\{x\backslash N\} \xrightarrow{R}\!\!\!\!\rightarrow_w M_1\{x\backslash N'\}$ et $M_2\{x\backslash N\} \xrightarrow{R}\!\!\!\!\rightarrow_w M_2\{x\backslash N'\}$. On conclut avec les règles (ν_w) et (μ_w).
   (c) Si $M = \lambda y.M_1$, on obtient par hypothèse d'induction $M_1\{x\backslash N\} \xrightarrow{R}\!\!\!\!\rightarrow_w M_1\{x\backslash N'\}$. En renommant éventuellement y, on peut supposer $y \notin \mathtt{FV}(R)$. Cette relation nous permet de conclure par (ξ'_w). □

Le premier point est similaire à la propriété de compatibilité à gauche de la substitution avec la relation $\rightarrow$. Cependant, il ne s'agit pas véritablement d'une propriété de compatibilité dans la mesure où, l'appartenance (M, M') à la relation $\xrightarrow{R}_w$ implique l'appartenance de $(M\{x\backslash N\}, M'\{x\backslash N\})$ à la relation $\xrightarrow{R\{x\backslash N\}}_w$. En revanche, comme dans le calcul fort, on obtient la compatibilité à droite de $\xrightarrow{R}\!\!\!\!\rightarrow_w$ vis-à-vis de l'opération de substitution. Ce résultat permet, de la même manière que dans les sections précédentes, d'obtenir la propriété de confluence locale pour le λ-calcul faible.

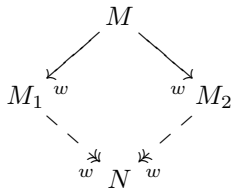


FIG. 3.3 – *Confluence locale du λ-calcul faible*

Théorème 3.1 (Confluence locale) *Si $M \to_w M_1$ et $M \to_w M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_w N$ et $M_2 \twoheadrightarrow_w N$.*
Ce résultat est illustré sur la figure 3.3.

Pour montrer la confluence du λ-calcul faible, on adapte, comme dans les sections précédentes, la méthode de Tait et Martin-Löf. On introduit la relation de $\rightrightarrows_w^{\mathcal{X}}$ paramétrée par un ensemble de variables $\mathcal{X}$.

$$\begin{array}{rll} x & \rightrightarrows_w^{\mathcal{X}} x & \\ \lambda x.M & \rightrightarrows_w^{\mathcal{X}} \lambda x.M' & \text{si } M \rightrightarrows_w^{\mathcal{X}\cup\{x\}} M' \\ MN & \rightrightarrows_w^{\mathcal{X}} M'N' & \text{si } M \rightrightarrows_w^{\mathcal{X}} M' \text{ et } N \rightrightarrows_w^{\mathcal{X}} N' \\ (\lambda x.M)N & \rightrightarrows_w^{\mathcal{X}} M'\{x\backslash N'\} & \text{si } M \rightrightarrows_w^{\mathcal{X}\cup\{x\}} M' \text{ et } N \rightrightarrows_w^{\mathcal{X}} N' \text{ et } \mathcal{X} \cap \mathtt{FV}((\lambda x.M)N) = \emptyset \end{array}$$

Comme dans les parties précédentes, $M \rightrightarrows_w M'$ signifie intuitivement que M peut se réduire vers M' en contractant simultanément plusieurs radicaux de M. On dira par la suite que M se réduit par des *réductions parallèles* vers M'. Par contraste avec les sections précédentes, la nouvelle règle de contexte (ξ'_w) nous pousse à paramétrer la relation $\rightrightarrows_w$ avec un ensemble de variables. En effet, dans le λ-calcul faible, on ne peut réduire un radical sous une abstraction que s'il ne contient pas la variable liée par cette dernière. Pour satisfaire cette contrainte, le paramètre $\mathcal{X}$ contient intuitivement les variables qui ne doivent pas être contenues par un radical. Pour cette raison, deux abstractions $\lambda x.M$ et $\lambda x.M'$ sont en relation pour $\mathcal{X}$, si leurs corps sont en relation pour l'ensemble $\mathcal{X}$ augmenté de la variable x liée dans ceux-ci. De même, un terme de la forme $(\lambda x.M)N$, c'est-à-dire un radical potentiel, est en relation avec un terme de la forme d'un contractum $M'\{x\backslash N'\}$ si toutes les variables, intuitivement liées, de $\mathcal{X}$ sont bien absentes du radical. L'interprétation de $M \rightrightarrows_w^{\mathcal{X}} M'$ est donc que M peut se réduire vers M' en contractant simultanément plusieurs radicaux de M qui ne contiennent pas de variable de $\mathcal{X}$. On note $\rightrightarrows_w$ la relation $\rightrightarrows_w^{\emptyset}$. La relation $\rightrightarrows_w^{\mathcal{X}}$ vérifie les propriétés suivantes vis-à-vis de son paramètre.

Lemme 3.3 *1. Si $\mathcal{X} \subseteq \mathcal{X}'$ et $M \rightrightarrows_w^{\mathcal{X}'} M'$, alors on a $M \rightrightarrows_w^{\mathcal{X}} M'$.*
2. Si $M \rightrightarrows_w^{\mathcal{X}} M'$ et $x \notin \mathtt{FV}(M)$, alors on a $M \rightrightarrows_w^{\mathcal{X}\cup\{x\}} M'$.

Preuve : Le premier point est une conséquence directe de la définition de $\rightrightarrows_w^{\mathcal{X}}$. Le deuxième point se montre par induction sur la taille de M. Le cas crucial est $M = (\lambda y.N)P \rightrightarrows_w^{\mathcal{X}} N'\{y\backslash P'\}$ avec $N \rightrightarrows_w^{\mathcal{X}\cup\{y\}} N'$ et $P \rightrightarrows_w^{\mathcal{X}} P'$ et $\mathcal{X} \cap \mathtt{FV}(M) = \emptyset$. En renommant éventuellement y, on peut supposer $y \neq x$. On a $x \notin \mathtt{FV}(N)$ et $x \notin \mathtt{FV}(P)$. De là, on obtient par hypothèse d'induction $N \rightrightarrows_w^{\mathcal{X}\cup\{y,x\}} N'$ et $P \rightrightarrows_w^{\mathcal{X}\cup\{x\}} P'$. Par ailleurs, on a $(\mathcal{X} \cup \{x\}) \cap \mathtt{FV}(M) = \emptyset$, ce qui permet de conclure. □

Ces propriétés sont conformes à l'intuition donnée précédemment : on a $M \rightrightarrows_w^{\mathcal{X}} M'$ si M peut se réduire vers M' en contractant simultanément des radicaux qui ne contiennent pas de variable de $\mathcal{X}$. Comme le montre le premier point du lemme précédent, en diminuant le paramètre, on relâche la contrainte. La relation $\rightrightarrows_w^{\mathcal{X}}$ est donc décroissante par rapport à son paramètre. De même, si une variable x n'intervient pas dans M, ajouter cette variable à $\mathcal{X}$ n'ajoute pas de contrainte

supplémentaire : le deuxième point indique que la relation entre deux termes M et M' ne dépend que de M, de M' et des variables de $\mathtt{FV}(M)$ et de $\mathcal{X}$.

Le lemme suivant permet de relier les relations $\rightrightarrows_w$ et $\to_w$. Il permet, en particulier, de justifier formellement l'appellation de relation de *réductions parallèles*.

Lemme 3.4 *1. Si $M \to_w M'$, alors $M \rightrightarrows_w M'$.*
2. Si $M \rightrightarrows_w^{\mathcal{X}} M'$, alors $M \twoheadrightarrow_w M'$ et si R est un radical contracté entre M et M', alors $\mathcal{X} \cap \mathtt{FV}(R) = \emptyset$.
3. Si $M \rightrightarrows_w M'$, alors $M \twoheadrightarrow_w M'$.

Preuve : Le premier point découle directement de la définition de $\rightrightarrows_w$. Pour montrer le deuxième point, on montre la propriété suivante, plus générale : si $M \rightrightarrows_w^{\mathcal{X}} M'$, alors il existe une réduction $M \xrightarrow{R_1}_w M_1 \xrightarrow{R_2}_w \ldots \xrightarrow{R_n}_w M'$ où pour tout $i \in \{1 \ldots n\}$, on a $\mathcal{X} \cap \mathtt{FV}(R_i) = \emptyset$. On montre cette propriété par induction sur M. Le cas crucial est $M = (\lambda y.N)P \rightrightarrows_w^{\mathcal{X}} N'\{x\backslash P'\}$ où $N \rightrightarrows_w^{\mathcal{X}\cup\{y\}} N'$, $P \rightrightarrows_w^{\mathcal{X}} P'$ et $\mathcal{X} \cap \mathtt{FV}(M) = \emptyset$. Par hypothèse d'induction, on obtient $N \xrightarrow{R_1}_w \ldots \xrightarrow{R_n}_w N'$ et $P \xrightarrow{S_1}_w \ldots \xrightarrow{S_p}_w P'$ avec, pour $i \in \{1 \ldots n\}$, $(\mathcal{X} \cup \{y\}) \cap \mathtt{FV}(R_i) = \emptyset$ et, pour $i \in \{1 \ldots p\}$, $\mathcal{X} \cap \mathtt{FV}(S_i) = \emptyset$. De là, par application des règles de contexte (ξ'_w), (μ_w) et (ν_w), on obtient la réduction suivante.

$$\mathcal{R} : M \xrightarrow{R_1}_w \ldots \xrightarrow{R_n}_w (\lambda y.N')P \xrightarrow{S_1}_w \ldots \xrightarrow{S_p}_w (\lambda y.N')P' = M' \xrightarrow{M'}_w N'\{y\backslash P'\}$$

Comme $\mathcal{X} \cap \mathtt{FV}(M) = \emptyset$, on a $\mathcal{X} \cap \mathtt{FV}(M') = \emptyset$. Tous les radicaux R contractés au cours de la réduction $\mathcal{R}$ vérifient $\mathcal{X} \cap \mathtt{FV}(R) = \emptyset$. Le troisième point est un corollaire immédiat du deuxième point. □

Ce lemme montre qu'un pas de réduction $\to_w$ correspond à une réduction parallèle. Et une réduction parallèle correspond à plusieurs pas de réduction. On note que le deuxième point du lemme formalise l'interprétation que nous avions donnée de $\rightrightarrows_w^{\mathcal{X}}$ auparavant.

Comme dans les sections précédentes, la propriété de confluence locale forte de $\rightrightarrows_w$ repose essentiellement sur le lemme suivant. Ce résultat est une adaptation du lemme 2.6 de compatibilité à gauche et à droite de la relation des réductions parallèles avec la substitution.

Lemme 3.5 *Si $M \rightrightarrows_w^{\mathcal{X}\cup\{x\}} M'$ et $N \rightrightarrows_w^{\mathcal{X}} N'$, alors $M\{x\backslash N\} \rightrightarrows_w^{\mathcal{X}} M'\{x\backslash N'\}$.*

Preuve : On procède par induction sur M. On pose $M_\star = M\{x\backslash N\}$ et $M'_\star = M'\{x\backslash N'\}$.

1. Si $M = x = M'$, alors on a par hypothèse $M_\star = N \rightrightarrows_w^{\mathcal{X}} N' = M'_\star$.
2. Si $M = y = M'$ avec $x \neq y$, le résultat est direct.
3. Si $M = \lambda y.P$ et $M' = \lambda y.P'$ avec $P \rightrightarrows_w^{\mathcal{X}\cup\{x,y\}} P'$, alors, par hypothèse d'induction on obtient $P\{x\backslash N\} \rightrightarrows_w^{\mathcal{X}\cup\{y\}} P'\{x\backslash N'\}$, ce qui permet de conclure.
4. Si $M = (\lambda y.P)Q$ et $M' = P'\{y\backslash Q'\}$, on a nécessairement $P \rightrightarrows_w^{\mathcal{X}\cup\{x,y\}} P'$, $Q \rightrightarrows_w^{\mathcal{X}\cup\{x\}} Q'$ et $(\mathcal{X} \cup \{x\}) \cap \mathtt{FV}(M) = \emptyset$. En particulier, on a $x \notin \mathtt{FV}(M)$, ce qui implique, $x \notin \mathtt{FV}(M')$. Par conséquent, on obtient $M_\star = M \rightrightarrows_w^{\mathcal{X}} M' = M'_\star$.
5. Le cas $M = PQ$ et $M' = P'Q'$ se traite en utilisant l'hypothèse d'induction. □

L'opération de substitution $\{x\backslash N\}$ peut potentiellement injecter dans des sous-termes de M des variables de $\mathcal{X}$ qui sont, intuitivement, liées. De ce fait, si on veut pouvoir réduire M de la même façon avant et après substitution, on a besoin d'une hypothèse renforcée pour la réduction parallèle entre M et M', dans laquelle la variable x joue le rôle d'une variable liée. C'est pourquoi, pour obtenir $M\{x\backslash N\} \rightrightarrows_w^{\mathcal{X}} M'\{x\backslash N'\}$, le lemme requiert l'hypothèse $M \rightrightarrows_w^{\mathcal{X}\cup\{x\}} M'$. De même que dans les parties précédentes, ces résultats se combinent pour obtenir la confluence locale forte de la relation $\rightrightarrows_w^{\mathcal{X}}$.

Lemme 3.6 (Confluence locale forte) *Si $M \rightrightarrows_w^{\mathcal{X}} M_1$ et $M \rightrightarrows_w^{\mathcal{X}} M_2$, il existe un terme M' tel que $M_1 \rightrightarrows_w^{\mathcal{X}} M'$ et $M_2 \rightrightarrows_w^{\mathcal{X}} M'$.*

Preuve : On procède par récurrence sur la taille de X.

Confluence locale forte de $\rightrightarrows_w^{\mathcal{X}}$ Confluence de $\rightarrow_w$

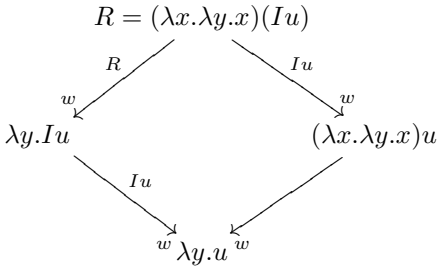
FIG. 3.4 – *Confluence du λ-calcul faible*

1. Si $M = x^\alpha$ ou $M = \Omega$, on a nécessairement $M_1 = M_2$.
2. Si $M = \lambda x.N$, on a nécessairement $M_1 = \lambda x.N_1$ où $N \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N_1$ et $M_2 = \lambda x.N_2$ où $N \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N_2$. Par hypothèse de récurrence, on obtient un terme N' tel que $N_1 \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N'$ et $N_2 \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N'$, ce qui permet de conclure.
3. Si $M = NP$ où N n'est pas une abstraction, alors on obtient le résultat comme dans le cas précédent, par une double application de l'hypothèse de récurrence.
4. Si $M = (\lambda x.N)P$, alors plusieurs cas sont à considérer.
   (a) Si $M_1 = (\lambda x.N_1)P_1$ et $M_2 = (\lambda x.N_2)P_2$ avec $N \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N_1$, $P \rightrightarrows_w^{\mathcal{X}} P_1$, $N \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N_2$ et $P \rightrightarrows_w^{\mathcal{X}} P_2$, alors, par hypothèse de récurrence, il existe des termes N' et P' tels que $N_1 \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N'$, $N_2 \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N'$, $P_1 \rightrightarrows_w^{\mathcal{X}} P'$ et $P_2 \rightrightarrows_w^{\mathcal{X}} P'$. On conclut en utilisant la définition de $\rightrightarrows_w$.
   (b) Si $M_1 = N_1\{x\backslash P_1\}$ où $N \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N_1$ et $P \rightrightarrows_w^{\mathcal{X}} P_1$ et si $M_2 = N_2\{x\backslash P_2\}$ où $N \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N_2$ et $P \rightrightarrows_w^{\mathcal{X}} P_2$, alors, par hypothèse de récurrence, il existe deux termes N' et P' tels que $N_1 \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N'$, $N_2 \rightrightarrows_w^{\mathcal{X}\cup\{x\}} N'$, $P_1 \rightrightarrows_w^{\mathcal{X}} P'$ et $P_2 \rightrightarrows_w^{\mathcal{X}} P'$. On conclut en utilisant le lemme 3.5.
   (c) Si $M_1 = N_1\{x\backslash P_1\}$ où $N \rightrightarrows_v N_1$ et $P \rightrightarrows_v P_1$ et si $M_2 = (\lambda x.N_2)P_2$ avec $P \rightrightarrows_v P_2$ et $\lambda x.N \rightrightarrows_v \lambda x.N_2$, on procède de la même façon que pour les cas précédents. □

La confluence locale forte de $\rightrightarrows_w$ implique la confluence de la réduction $\rightarrow_w$. Ces propriétés sont illustrées sur la figure 3.4.

Théorème 3.2 (Confluence) *Si $M \twoheadrightarrow_w M_1$ et $M \twoheadrightarrow_w M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_w N$ et $M_2 \twoheadrightarrow_w N$.*

L'introduction de la règle (ξ'_w) permet de retrouver la propriété de confluence de λ-calcul. En reprenant le contre-exemple mentionné au début de cette section, on obtient le diagramme suivant.



Comme Iu ne contient pas de variable liée à l'extérieur du sous-terme, la règle (ξ'_w) permet de réduire ce radical, ce qui permet de fermer ce diagramme de confluence.

Si la définition d'un λ-calcul faible confluent est un peu délicate, le théorème des développements finis s'avère plus direct dans le calcul faible que dans le calcul fort. En effet, les radicaux du λ-calcul faible vérifient une propriété qui simplifie grandement la démonstration du théorème des développements finis.

Remarque 3.1 *Si $R < S$, alors S ne contient pas la variable liée par R.*

Du fait de cette remarque, on observe que les notations $\nearrow$ et $\overset{}{\underset{a}{\nearrow}}$ qui sont utilisées dans la démonstration du théorème 2.8, deviennent inutiles : un radical ne peut contenir la variable liée par un radical externe. En d'autres termes, si deux radicaux sont disjoints, leurs résidus sont également disjoints. On illustre cette propriété par le terme suivant : $M = (\lambda x.Ix)(Jy)$ où $I = J = \lambda x.x$. Dans le λ-calcul fort, les radicaux Ix et Jy sont disjoints. Mais après contraction du radical externe, on obtient le terme $I(Jy)$ où le résidu $I(Jy)$ de Ix contient le résidu de Jy. Dans le λ-calcul faible, le sous-terme Ix de M n'est pas un radical puisqu'il contient la variable liée x. Plus généralement, la remarque 3.1 permet de simplifier la définition de la relation $\Subset_{\mathcal{F}}$ utilisée dans la preuve du théorème 2.8 : cette relation $\Subset_{\mathcal{F}}$ coïncide dans le calcul faible avec la relation d'imbrication $<$. De là, on introduit une définition singulièrement simplifiée de la profondeur $\mathcal{P}_{\mathcal{F}}(R)$ d'un radical R.

$$\mathcal{P}_{\mathcal{F}}(R) = \mathtt{max}(\{0\} \cup \{1 + \mathcal{P}_{\mathcal{F}}(S) \mid S \in \mathcal{F} \text{ et } S < R\})$$

La preuve du lemme suivant qui correspond au lemme 2.33 est directe.

Lemme 3.7 *Soit $\mathcal{F}$ un ensemble de radicaux de M et $T \in \mathcal{F}$. On suppose $M \xrightarrow{T}_w M'$. Soit $\mathcal{F}'$ l'ensemble des résidus de $\mathcal{F}$ par cette réduction. Soit R' un résidu d'un radical R tel que $R \in \mathcal{F}$. Si $T < R$ alors $\mathcal{P}_{\mathcal{F}'}(R') < \mathcal{P}_{\mathcal{F}}(R)$.*

Preuve : On montre cette propriété par récurrence sur $\mathcal{P}_{\mathcal{F}'}(R') = n$.

1. Si $n = 0$, alors comme $T < R$, on a $\mathcal{P}_{\mathcal{F}}(R) \geq 1 > \mathcal{P}_{\mathcal{F}'}(R')$.
2. Si $n > 1$, alors il existe une suite de radicaux de $\mathcal{F}'$ vérifiant : $R'_1 < R'_2 < \ldots < R'_n < R'$. Pour i tel que $1 \leq i \leq n$, on note R_i le radical de $\mathcal{F}$ dont R'_i est un résidu. Comme R'_1 contient R'_2, on a nécessairement $R_1 < R_2$. En poursuivant ce raisonnement, on obtient la relation $R_1 < R_2 < \ldots < R_n < R$ (où pour $1 \leq i \leq n$ on a $R_i \neq T$). On procède par cas sur la relation $R_n \Subset_{\mathcal{F}} R$.
   (a) Si $R_n < T < R$, la chaîne d'imbrication $R_1 < R_2 < \ldots < R_n < T < R$ permet de conclure : $\mathcal{P}_{\mathcal{F}}(R) \geq n + 1 > \mathcal{P}_{\mathcal{F}'}(R')$.
   (b) Si $T < R_n$. Comme $\mathcal{P}_{\mathcal{F}'}(R'_n) = n - 1$, on obtient par hypothèse de récurrence sur R'_n une suite $\{S_i\}_{1 \leq i \leq n}$ d'éléments de $\mathcal{F}$ qui vérifie $S_1 < S_2 < \ldots < S_n < R_n < R$, ce qui permet de conclure. □

Ce lemme montre que la profondeur des résidus des radicaux de $\mathcal{F}$ qui sont contenus dans le radical contracté diminue strictement. Ce résultat permet d'aboutir au théorème des développements finis en reprenant la preuve du théorème 2.8.

Théorème 3.3 (Développements finis) *Soit $\mathcal{F}$ un ensemble de radicaux de M.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Tous les développements de $\mathcal{F}$ finissent sur un même terme N.*
3. *L'ensemble des résidus d'un radical R de M dans N est indépendant du développement considéré.*

A partir du théorème des développements finis, le théorème de standardisation pour le λ-calcul faible s'obtient de la même façon que pour le calcul fort. Comme le montre le lemme suivant, les radicaux du calcul faible vérifient les propriétés mentionnées dans le lemme 2.9 dans le cadre du calcul fort.

Lemme 3.8 *On suppose que R et S sont deux radicaux de M qui vérifient $M \xrightarrow{S}_w N$ et $R <_g S$. Dans ces conditions, les propriétés suivantes sont vérifiées.*

1. *R a un unique résidu R' dans N.*
2. *Si T' est un radical créé de N alors $R' <_g T'$.*

3. Si T est un radical de M tel que $R \leq_g T$, alors chaque résidu T' de T dans N vérifie $R' \leq_g T'$.

Preuve : Les propriétés 1 et 3 sont des conséquences directes du lemme 2.9. On montre la propriété 2 par cas. Dans le λ-calcul faible, en comparaison du λ-calcul fort, il existe trois façons de créer un radical. Soit R' le résidu de R. Si T' est créé par le haut ou par le bas par la contraction de S, alors, comme dans le λ-calcul fort, T' est à droite de R'. Si T' est *dégelé*, on a la réduction $M = C[(\lambda x.C'[N])M_1] \xrightarrow{S}_w C[C'[N\{x\backslash M_1]] = M'$ où $N = (\lambda y.M_2)M_3$ contient la variable x et $N' = (\lambda y.M_2\{x\backslash M_1\})M_3\{x\backslash M_1\}$ est un radical de M'. En particulier S contient N et le contractum de S contient le radical créé N. Le radical R est à gauche de S : par conséquent, le radical R' est à gauche du contractum S', et donc à gauche de N'. □

Si, dans un terme M, on contracte un radical strictement à droite du radical R, alors R a un unique résidu, les résidus des radicaux à droite de R restent à droite de R et les radicaux créés sont également à droite de R. La nouveauté par rapport au calcul fort réside dans le nouveau cas de création d'un radical. Si un radical T est dégelé, il est nécessairement contenu dans le contractum du radical contracté : S est donc à droite de R. Ce lemme permet de réutiliser la preuve de standardisation donnée dans le cadre du λ-calcul par valeur.

Théorème 3.4 (Standardisation) *Si $M \twoheadrightarrow_w M'$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_w M'$ telle que $\mathcal{R}$ est standard.*

La variante du λ-calcul faible que nous avons introduite dans cette section vérifie les mêmes propriétés fondamentales que le λ-calcul. Pour cette raison, on considérera par la suite que cette variante est le λ-calcul faible.

3.2 Le λ-calcul faible étiqueté

A partir du λ-calcul faible confluent de la section précédente, nous introduisons un λ-calcul faible étiqueté. Le but est d'obtenir un système d'étiquettes qui, en plus de conserver les propriétés fondamentales du λ-calcul faible, permet d'exprimer la notion de partage. Dans cette section, nous prouvons la confluence du langage et les théorèmes des développements finis et de standardisation.

La syntaxe du λ-calcul faible étiqueté est fondée sur le λ-calcul étiqueté [29]. Cependant, la volonté d'exprimer la notion de partage de sous-termes nous pousse à utiliser un système qui exhibe plus clairement les étiquettes simples en lieu et place des concaténations d'étiquettes utilisées pour le calcul fort. On note que les étiquettes utilisées ici diffèrent de celles utilisées dans [30] pour le λ-calcul faible avec substitutions explicites du fait du rôle essentiel que les variables liées jouent ici. La syntaxe des termes et des étiquettes est définie de la façon suivante.

Termes étiquetés	M,N	$::= x^\alpha$	Variable
		$\mid (\lambda x.M)^\alpha$	Abstraction
		$\mid (MN)^\alpha$	Application
		$\mid \alpha : M$	Intercalaire
Etiquettes atomiques	α,β	$::= a$	Lettre
		$\mid \lceil \alpha' \rceil$	Surlignement
		$\mid \lfloor \alpha' \rfloor$	Soulignement
		$\mid [\alpha',\beta]$	Marquage
Etiquettes composées	α',β'	$::= \alpha_1\alpha_2\cdots\alpha_n$	$(n > 0)$

Aux variables, abstractions et applications du λ-calcul, on ajoute les intercalaires qui sont des sous-termes précédés d'une étiquette atomique. Intuitivement, ces intercalaires seront introduits au cours de la réduction. On dit que les termes x^α, $(\lambda x.M)^\alpha$, $(MN)^\alpha$ et $\alpha : M$ ont pour étiquette

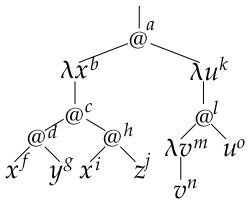


FIG. 3.5 – *Terme du λ-calcul faible étiqueté :* $((\lambda x.(((x^f\ y^g)^d\ (x^i\ z^j)^h)^c)^b\ (\lambda u.((\lambda v.v^n)^m\ u^o)^l)^k)^a$

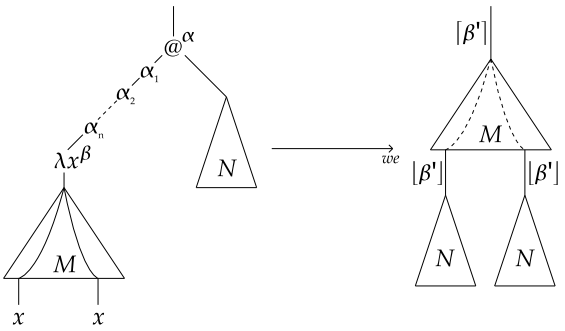


FIG. 3.6 – β_{we}*-réduction*

α. Contrairement au λ-calcul fort, les étiquettes ne se concatènent pas ; on garde les étiquettes sous leur forme atomique. De ce fait, alors que dans le calcul fort, les étiquettes portées par un terme sont intuitivement associées aux arêtes entre les nœuds, ici, les étiquettes des termes sont associées aux nœuds. On tient compte de cette interprétation dans les illustrations de cette partie. Le terme étiqueté $((\lambda x.(((x^f\ y^g)^d\ (x^i\ z^j)^h)^c)^b\ (\lambda u.((\lambda v.v^n)^m\ u^o)^l)^k)^a$ qui correspond au terme mentionné en introduction, est représenté sur la figure 3.5.

Une étiquette atomique peut être une lettre a ou bien être formée à partir d'une étiquette composée α' par surlignement $\lceil\alpha'\rceil$, soulignement $\lfloor\alpha'\rfloor$ ou par *marquage* d'une étiquette atomique $[\alpha',\beta]$. Cette troisième façon de former une étiquette atomique à partir d'une étiquette composée est associée à la troisième façon de créer un radical. Si le surlignement (respectivement le soulignement) est associé aux radicaux créés par le haut (resp. le bas), le marquage sera utilisé pour les radicaux dégelés. Les étiquettes composées sont des séquences d'étiquettes atomiques. Elles sont utilisées pour alléger les notations dans le cas où les étiquettes atomiques s'empilent : le terme $\alpha' \circ M$ (où $\alpha' = \alpha_1\alpha_2 \ldots \alpha_n$) est un raccourci pour le terme $\alpha_1 : \alpha_2 : \cdots \alpha_n : M$. On utilisera aussi la notation $\beta' = \beta\alpha'\alpha$ pour $\beta' = \beta\alpha_1\alpha_2 \ldots \alpha_n\alpha$.

Les règles de réduction du λ-calcul faible étiqueté sont définies de la façon suivante.

$$(\beta_{we}) \qquad R = (\alpha' \circ (\lambda x.M)^\beta N)^\alpha \xrightarrow{R}_{we} \lceil\beta'\rceil : (\beta' \circledcirc{x} M)\{x \setminus \lfloor\beta'\rfloor : N\}$$
$$\text{où } \beta' = \alpha\alpha'\beta$$

$$(\nu_{we}) \ \frac{M \xrightarrow{R} M'}{(MN)^\alpha \xrightarrow{R} (M'N)^\alpha} \qquad (\lambda_{we}) \ \frac{M \xrightarrow{R} M'}{\alpha : M \xrightarrow{R} \alpha : M'}$$

$$(\mu_{we}) \ \frac{N \xrightarrow{R} N'}{(MN)^\alpha \xrightarrow{R} (MN')^\alpha} \qquad (\xi'_{we}) \ \frac{M \xrightarrow{R} M' \quad x \notin \mathtt{FV}(R)}{(\lambda x.M)^\alpha \xrightarrow{R} (\lambda x.M')^\alpha}$$

La règle de réduction de base est (β_{we}). Cette réduction est illustrée sur la figure 3.6. Le nom du radical R est β' ; ce qu'on écrira $\mathsf{nom}(R) = \beta'$. On observe sur cette figure que le corps M

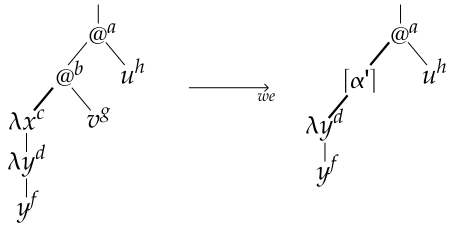


FIG. 3.7 – *Réduction* $(((\lambda x.(\lambda y.y^f)^d)^c\ v^g)^b\ u^h)^a \to_{we} (\lceil\alpha'\rceil : (\lambda y.y^f)^d\ u^h)^a$ *où* $\alpha' = bc$

de la fonction du radical est intercalé entre son nom surligné et souligné, comme pour le calcul fort. Intuitivement, la diffusion $\beta' \circledast M$ marque par β' les étiquettes situées le long des chemins (en pointillé sur la figure) de la racine de M vers les occurrences de x, comme le montre la figure 3.6. De cette façon, tous les sous-termes de M qui contenaient une occurrence libre de x ont une nouvelle étiquette dans le contractum. Les règles de contexte (ν_{we}), (μ_{we}) et (ξ'_{we}) sont des adaptations directes des règles de contexte du λ-calcul faible. On ajoute à ces règles une règle (λ_{we}) pour pouvoir calculer sous les intercalaires. Par convention, on suppose que la substitution est prioritaire sur la notation de l'intercalaire. Ainsi, $\lceil\alpha'\rceil : (M\{x\backslash N\})$ s'écrit $\lceil\alpha'\rceil : M\{x\backslash N\}$. Les opérations de substitution et de diffusion sont formellement définies ci-dessous.

$$
\begin{aligned}
x^\alpha\{x\backslash N\} &= \alpha : N\\
y^\alpha\{x\backslash N\} &= y^\alpha\\
(MP)^\alpha\{x\backslash N\} &= (M\{x\backslash N\}\, P\{x\backslash N\})^\alpha\\
(\lambda x.M)^\alpha\{x\backslash N\} &= (\lambda x.M)^\alpha\\
(\lambda y.M)^\beta\{x\backslash N\} &= (\lambda z.M\{y \leftarrow z\}\{x\backslash N\})^\beta \text{ où } z = \mathsf{Conv}_\alpha(x,y,M,N)\\
(\alpha : M)\{x\backslash N\} &= \alpha : M\{x\backslash N\}
\end{aligned}
$$

$$
\begin{aligned}
\alpha' \circledast M &= M \text{ si } x \notin \mathtt{FV}(M)\\
\alpha' \circledast x^\alpha &= x^{[\alpha',\alpha]}\\
\alpha' \circledast (MN)^\alpha &= (\alpha' \circledast M\ \ \alpha' \circledast N)^{[\alpha',\alpha]} \text{ si } x \in \mathtt{FV}((MN)^\alpha)\\
\alpha' \circledast (\lambda y.M)^\alpha &= (\lambda y.\alpha' \circledast M)^{[\alpha',\alpha]} \text{ si } x \in \mathtt{FV}((\lambda y.M)^\alpha)\\
\alpha' \circledast \alpha : M &= [\alpha',\alpha] : \alpha' \circledast M \text{ si } x \in \mathtt{FV}(\alpha : M)
\end{aligned}
$$

La substitution $\{x\backslash N\}$ remplace les occurrences x^α par des intercalaires $\alpha : N$. Comme annoncé précédemment, l'opération de diffusion $\alpha' \circledast M$ marque avec l'étiquette composée α' les étiquettes présentes sur les chemins menant de la racine de M aux occurrences de x dans M.

On retrouve dans le λ-calcul faible étiqueté certaines propriétés du calcul fort portant sur les noms des radicaux. Ainsi, les résidus d'un radical portent le même nom que le radical dont ils sont issus. Par ailleurs, le nom d'un radical contient intuitivement les noms des radicaux qui l'ont créé. Si le radical β' contient l'étiquette surlignée $\lceil\alpha'\rceil$, alors un radical nommé α' a créé β' par le haut. On illustre cette interprétation intuitive par la réduction illustrée sur la figure 3.7. Le nom du radical créé $\beta' = a\lceil\alpha'\rceil d$ contient le nom du radical qui l'a créé, c'est-à-dire α'. De même, l'étiquette soulignée $\lfloor\alpha'\rfloor$ sert à marquer les radicaux créés *par le bas* par la contraction d'un radical α'. Nous avons vu dans la section précédente, qu'il existe une troisième façon de créer des radicaux dans le λ-calcul faible. L'étiquette $[\alpha',\alpha]$ sert ainsi à marquer les radicaux *dégelés* par la contraction d'un radical α'. L'exemple de la réduction du terme $M = ((\lambda x.((\lambda y.y^f)^d x^g)^c)^b u^h)^a$, présenté sur la figure 3.8, illustre cette intuition. Le sous-terme $((\lambda y.y^f)^d x^g)^c$ de M n'est pas un radical puisqu'il contient la variable x liée par l'abstraction externe $(\lambda x.((\lambda y.y^f)^d x^g)^c)^b$. Du fait de la réduction, cette variable liée est substituée. Le sous-terme $((\lambda y.y^f)^d\ [\alpha',g] : \lfloor\alpha'\rfloor : u^h)^{[\alpha',c]}$ obtenu ne contient plus de variable liée. Il est donc *dégelé* et devient donc un radical. On observe que le nom de ce

$$M = ((\lambda x.((\lambda y.y^f)^d x^g)^c)^b u^h)^a \to_{we} \lceil\alpha'\rceil : ((\lambda y.y^f)^d\ [\alpha',g] : \lfloor\alpha'\rfloor : u^h)^{[\alpha',c]} = M'$$
où $\alpha' = ab$ et $\beta' = [\alpha',c]d$

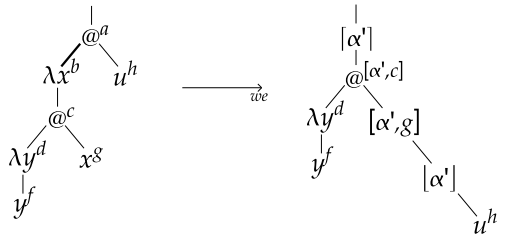


FIG. 3.8 – *Nom du radical créé*

radical est $\beta' = [\alpha',c]d$: il contient le nom du radical qui l'a créé.

On examine maintenant les propriétés de confluence. Comme pour le λ-calcul, ces propriétés s'appuient sur la propriété fondamentale de commutation des substitutions.

Lemme 3.9 *Si $x \notin \mathtt{FV}(P)$, alors on a $M\{x\backslash N\}\{y\backslash P\} = M\{y\backslash P\}\{x\backslash N\{y\backslash P\}\}$.*

Preuve : La preuve mentionné dans les parties précédentes s'étend au cas de l'intercalaire de façon élémentaire. □

L'opérateur de diffusion vérifie également une propriété de commutation avec la substitution.

Lemme 3.10 *Si $x \neq y$ et $x \notin \mathtt{FV}(N)$, alors on a $(\alpha' ⓧ M)\{y\backslash N\} = \alpha' ⓧ M\{y\backslash N\}$.*

Preuve : On procède par induction sur M. Si $x \notin \mathtt{FV}(M)$, alors on a $x \notin \mathtt{FV}(M\{y\backslash N\})$. De là, on obtient $(\alpha' ⓧ M)\{y\backslash N\} = M\{y\backslash N\} = \alpha' ⓧ M\{y\backslash N\}$. Si $x \in \mathtt{FV}(M)$, on considère les différents cas possibles.

1. Si $M = x^\alpha$, alors on a $(\alpha' ⓧ M)\{y\backslash N\} = x^{[\alpha',\alpha]} = \alpha' ⓧ M\{y\backslash N\}$.
2. Si $M = (\lambda z.M')^\alpha$, on peut supposer, après un éventuel renommage, qu'on a $z \notin \{x,y\}$ et $z \notin \mathtt{FV}(N)$. De là, on obtient les relations $(\alpha' ⓧ M)\{y\backslash N\} = (\lambda z.(\alpha' ⓧ M')\{y\backslash N\})^{[\alpha',\alpha]}$ et $\alpha' ⓧ M\{y\backslash N\} = (\lambda z.\alpha' ⓧ M'\{y\backslash N\})^{[\alpha',\alpha]}$. On peut conclure par hypothèse d'induction.
3. Les autres cas sont similaires aux cas précédents. □

Comme dans la section précédente, la propriété de confluence locale repose essentiellement sur les propriétés vérifiées par la substitution vis-à-vis de la réduction.

Lemme 3.11 *1. Si $M \xrightarrow{R}_{we} M'$, alors on a $M\{y\backslash N\} \xrightarrow{R\{y\backslash N\}}_{we} M'\{y\backslash N\}$.*
2. Si $N \xrightarrow{R}_{we} N'$, alors on a $M\{y\backslash N\} \xrightarrow{R}\!\!\!\!\to_{we} M\{y\backslash N'\}$.

Preuve : On vérifie que les étiquettes employées ici ne perturbent pas les preuves du lemme 3.2.

1. Le cas crucial est $M = (\alpha' \circ (\lambda x.P)^\beta Q)^\alpha \xrightarrow{M}_{we} \lceil\beta'\rceil : (\beta' ⓧ P)\{x\backslash\lfloor\beta'\rfloor : Q\} = M'$ avec $\beta' = \alpha\alpha'\beta$. En renommant éventuellement x, on peut supposer $x \notin \mathtt{FV}(N)$. De là, le terme $M\{y\backslash N\} = (\alpha' \circ (\lambda x.P\{y\backslash N\})^\beta Q\{y\backslash N\})^\alpha$ se réduit de la façon suivante.
$$M\{y\backslash N\} \xrightarrow{M\{y\backslash N\}}_{we} \lceil\beta'\rceil : (\beta' ⓧ P\{y\backslash N\})\{x\backslash\lfloor\beta'\rfloor : Q\{y\backslash N\}\} = M''$$
En utilisant les lemmes 3.10 et 3.9, on obtient $M'' = M'\{y\backslash N\}$.
2. Le cas crucial est $M = y^\alpha$. On a, par (λ_{we}), $M\{y\backslash N\} = \alpha : N \xrightarrow{R}_{we} \alpha : N' = M'\{y\backslash N\}$. □

Si M est en relation par $\xrightarrow{R}_{we}$ avec M', alors $M\{y\backslash N\}$ est en relation par $\xrightarrow{R\{y\backslash N\}}_{we}$ avec $M'\{y\backslash N\}$. On obtient également la compatibilité à droite de $\xrightarrow{R}\!\!\!\!\to_{we}$ avec la substitution. Un résultat supplémentaire est utilisé pour prouver la confluence locale et la confluence. Le lemme suivant énonce la compatibilité de la réduction avec la diffusion.

Lemme 3.12 *Si $M \xrightarrow{R}_{we} M'$ et $x \notin \mathtt{FV}(R)$, alors $\alpha' ⓧ M \xrightarrow{R}_{we} \alpha' ⓧ M'$.*

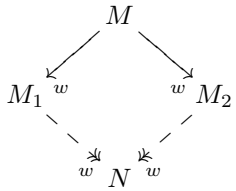


FIG. 3.9 – *Concluence locale du λ-calcul faible étiqueté*

Preuve : On prouve tout d'abord une propriété préliminaire : comme $x \notin \mathtt{FV}(R)$ et $M \xrightarrow{R}_{we} M'$, on a $x \in \mathtt{FV}(M)$ si et seulement si $x \in \mathtt{FV}(M')$. Si $x \notin \mathtt{FV}(M)$, on a bien sûr $x \notin \mathtt{FV}(M')$ puisque l'ensemble des variables libres du contractum est un sous-ensemble des variables libres du radical contracté. Réciproquement, si $x \in \mathtt{FV}(M)$, cette variable ne peut disparaître que si toutes ses occurrences appartiennent à la partie droite du radical contracté ce qui est exclu par la relation $x \notin \mathtt{FV}(R)$. On procède maintenant par cas sur la définition de la diffusion et par récurrence sur la taille de M.

1. Si $x \notin \mathtt{FV}(M)$, alors $x \notin \mathtt{FV}(M')$ et $\alpha' \circledast M = M \xrightarrow{R}_{we} M' = \alpha' \circledast M'$.
2. Si $x \in \mathtt{FV}(M)$, on procède par cas sur la réduction de M.
   (a) Le cas (β_{we}) est exclu du fait de l'hypothèse $x \notin \mathtt{FV}(R)$.
   (b) Si $M = (NP)^{\alpha} \xrightarrow{R}_{we} (N'P)^{\alpha}$ où $N \xrightarrow{R}_{we} N'$. Par induction sur N, on obtient la réduction $\alpha' \circledast N \xrightarrow{R}_{we} \alpha' \circledast N'$. Par définition de la diffusion, on a la relation $\alpha' \circledast M = (\alpha' \circledast N\ \alpha' \circledast P)^{[\alpha', \alpha]}$. La remarque préliminaire implique $x \in \mathtt{FV}(M')$. De là, on a $\alpha' \circledast M' = (\alpha' \circledast N'\ \alpha' \circledast P)^{[\alpha', \alpha]}$. Par la règle (ν_{we}), on obtient la réduction $\alpha' \circledast M \xrightarrow{R}_{we} \alpha' \circledast M'$.
   (c) Les cas (λ_{we}), (μ_{we}), (ξ'_{we}) sont similaires. □

Si la variable de diffusion n'est pas impliquée dans le radical R contracté entre M et M', la relation $\xrightarrow{R}_{we}$ est compatible avec la diffusion sur x. Ces différents résultats de compatibilité aboutissent au théorème de confluence locale, qui est illustré sur la figure 3.9.

Théorème 3.5 (Confluence locale) *Si $M \to_{we} M_1$ et $M \to_{we} M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_{we} N$ et $M_2 \twoheadrightarrow_{we} N$.*

Preuve : Le cas crucial est $M = (\alpha' \circ (\lambda x.P)^{\beta} Q)^{\alpha} \xrightarrow{M}_{we} \lceil \beta' \rceil : (\beta' \circledast P)\{x \backslash \lfloor \beta' \rfloor : Q\} = M_1$ où $\alpha' = \alpha_1 \alpha_2 \cdots \alpha_n$ et $\beta' = \alpha \alpha_1 \alpha_2 \cdots \alpha_n \beta$. Soit R le radical contracté entre M et M_2. Trois cas sont à envisager.

1. Si R est le radical M, alors le résultat est trivial.
2. Si R est dans P alors $M_2 = (\alpha' \circ (\lambda x.P')^{\beta} Q)^{\alpha}$ avec $P \xrightarrow{R}_{we} P'$ et $x \notin \mathtt{FV}(R)$. De là, on a $M_2 \xrightarrow{M_2}_{we} \lceil \beta' \rceil : (\beta' \circledast P')\{x \backslash \lfloor \beta' \rfloor : Q\}$. Par le lemme 3.12, on obtient $\beta' \circledast P \xrightarrow{R}_{we} \beta' \circledast P'$. De là, le lemme 3.11 et la règle (λ_{we}) permettent de conclure.
3. Si R est dans Q, alors $M_2 = (\alpha' \circ (\lambda x.P)^{\beta} Q')^{\alpha}$ avec où $Q \xrightarrow{R}_{we} Q'$. De là, on obtient la réduction $M_2 \xrightarrow{R}_{we} \lceil \beta' \rceil : (\beta' \circledast P)\{x \backslash \lfloor \beta' \rfloor : Q'\}$. Le lemme 3.11 et la règle (λ_{we}) permettent de conclure. □

Pour montrer la propriété de confluence, on s'appuie sur la preuve adoptée dans la partie précédente. On commence donc par étendre et adapter la définition de la relation des réductions

parallèles $\rightrightarrows^{\mathcal{X}}_{we}$.

$$
\begin{array}{rll}
x^\alpha & \rightrightarrows^{\mathcal{X}}_{we} \ x^\alpha & \\
(MN)^\alpha & \rightrightarrows^{\mathcal{X}}_{we} \ (M'N')^\alpha & \text{si } M \rightrightarrows^{\mathcal{X}}_{we} M' \text{ et } N \rightrightarrows^{\mathcal{X}}_{we} N' \\
(\alpha' \circ (\lambda x.M)^\beta N)^\alpha & \rightrightarrows^{\mathcal{X}}_{we} \ \lceil\beta'\rceil : (\beta' \circledcirc M')\{x \backslash N'\} & \text{si } \beta' = \alpha\alpha'\beta \text{ et } M \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} M' \text{ et } N \rightrightarrows^{\mathcal{X}}_{we} N' \\
 & & \text{et } \mathcal{X} \cap \mathtt{FV}((\alpha' \circ (\lambda x.M)^\beta N)^\alpha) = \emptyset \\
(\lambda x.M)^\alpha & \rightrightarrows^{\mathcal{X}}_{we} \ (\lambda x.M')^\alpha & \text{si } M \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} M' \\
\alpha : M & \rightrightarrows^{\mathcal{X}}_{we} \ \alpha : M' & \text{si } M \rightrightarrows^{\mathcal{X}}_{we} M'
\end{array}
$$

Cette définition est une adaptation de la relation des réductions parallèles au λ-calcul faible étiqueté. On l'étend naturellement au cas de l'intercalaire. Cette relation vérifie les mêmes propriétés de croissance vis-à-vis de son paramètre que la relation correspondante mentionnée dans la section précédente. De même $\rightrightarrows_{we}$ est reliée de la façon suivante à $\rightarrow_{we}$.

Lemme 3.13 *1. Si $\mathcal{X} \subseteq \mathcal{X}'$, alors $M \rightrightarrows^{\mathcal{X}'}_{we} M'$ implique $M \rightrightarrows^{\mathcal{X}}_{we} M'$.*
2. Si $M \rightrightarrows^{\mathcal{X}}_{we} M'$ et $y \notin \mathtt{FV}(M)$, alors $M \rightrightarrows^{\mathcal{X}\cup\{y\}}_{we} M'$.
3. Si $M \rightarrow_{we} M'$, alors $M \rightrightarrows_{we} M'$.
4. Si $M \rightrightarrows_{we} M'$, alors $M \twoheadrightarrow_{we} M'$.

Preuve : On adapte de façon élémentaire les preuves des lemmes 3.3 et 3.4. □

Le premier point exprime la décroissance de $\rightrightarrows^{\mathcal{X}}_{we}$ vis-à-vis de son paramètre $\mathcal{X}$. Le deuxième point montre que les variables y telles que $y \notin \mathtt{FV}(M)$ n'ont pas d'influence sur la relation $M \rightrightarrows^{\mathcal{X}}_{we} M'$. Les troisième et quatrième points établissent des liens entre $\rightrightarrows_{we}$ et $\rightarrow_{we}$. Pour aboutir à la confluence locale forte de $\rightrightarrows_{we}$, on utilise le lemme suivant qui énonce le comportement de la relation $\rightrightarrows_{we}$ vis-à-vis de la substitution.

Lemme 3.14 *Si $M \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} M'$ et $N \rightrightarrows^{\mathcal{X}}_{we} N'$, alors $M\{x\backslash N\} \rightrightarrows^{\mathcal{X}}_{we} M'\{x\backslash N'\}$.*

Preuve : On procède par induction sur M. On pose $M_\star = M\{x\backslash N\}$ et $M'_\star = M'\{x\backslash N'\}$.

1. Si $M = x^\alpha = M'$, alors on a $M_\star = \alpha : N$ et $M'_\star = \alpha : N'$. Comme par hypothèse, on a $N \rightrightarrows^{\mathcal{X}}_{we} N'$, on obtient bien $M_\star \rightrightarrows^{\mathcal{X}}_{we} M'_\star$.
2. Si $M = (\alpha' \circ (\lambda y.P)^\beta Q)^\alpha$ et $M' = \lceil\beta'\rceil : (\beta' \circledcirc P')\{y \backslash \lfloor\beta'\rfloor : Q'\}$ avec $\beta' = \alpha\alpha'\beta$, $P \rightrightarrows^{\mathcal{X}\cup\{x,y\}}_{we} P'$, $Q \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} Q'$ et $(\mathcal{X} \cup \{x\}) \cap \mathtt{FV}(M) = \emptyset$, alors on a $x \notin \mathtt{FV}(M')$ ce qui permet d'obtenir $M = M_\star \rightrightarrows^{\mathcal{X}}_{we} M' = M'_\star$.
3. Les autres cas sont élémentaires. □

Comme pour le lemme 3.5 de la section précédente, pour obtenir $M\{x\backslash N\} \rightrightarrows^{\mathcal{X}}_{we} M'\{x\backslash N'\}$, on a besoin de l'hypothèse $M \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} M'$ car la variable x joue le rôle d'une variable liée. De façon analogue au lemme 3.12 utilisé pour la preuve de la confluence locale, on introduit ici un résultat de compatibilité de la relation des réductions parallèles avec la diffusion.

Lemme 3.15 *Si $M \rightrightarrows^{\mathcal{X}}_{we} M'$ et $x \in \mathcal{X}$, alors on a $\alpha' \circledcirc M \rightrightarrows^{\mathcal{X}}_{we} \alpha' \circledcirc M'$.*

Preuve : On montre cette propriété par induction sur M.

1. Si $M = x^\alpha$, alors on a $M' = x^\alpha$ et $\alpha' \circledcirc M = x^{[\alpha',\alpha]} = \alpha' \circledcirc M'$.
2. Si $x \notin \mathtt{FV}(M)$, alors on a $x \notin \mathtt{FV}(M')$. De là, on obtient $\alpha' \circledcirc M = M \rightrightarrows^{\mathcal{X}}_{we} M' = \alpha' \circledcirc M'$.
3. Si $M = (\lambda y.N)^\alpha$ avec $x \in \mathtt{FV}(M)$, alors on a $M' = (\lambda y.N')^\alpha$ avec $N \rightrightarrows^{\mathcal{X}\cup\{y\}}_{we} N'$. Par hypothèse d'induction, on a $\alpha' \circledcirc N \rightrightarrows^{\mathcal{X}\cup\{y\}}_{we} \alpha' \circledcirc N'$. Par conséquent, on obtient la réduction $\alpha' \circledcirc M = (\lambda y.\alpha' \circledcirc N)^{[\alpha',\alpha]} \rightrightarrows^{\mathcal{X}}_{we} (\lambda y.\alpha' \circledcirc N')^{[\alpha',\alpha]} = \alpha' \circledcirc M'$.
4. Si $M = (\beta' \circ (\lambda y.N)^\beta P)^\alpha$ et $M' = \lceil\gamma'\rceil : (\gamma' \circledcirc N')\{y \backslash \lfloor\gamma'\rfloor : P'\}$ (avec $\gamma' = \alpha\beta'\beta$), on a $N \rightrightarrows^{\mathcal{X}\cup\{y\}}_{we} N'$, $P \rightrightarrows^{\mathcal{X}}_{we} P'$ et $\mathcal{X} \cap \mathtt{FV}(M) = \emptyset$. Comme $x \in \mathcal{X}$, on en déduit $x \notin \mathtt{FV}(M)$ ce qui nous ramène au deuxième cas.
5. Les autres cas sont similaires aux cas précédents. □

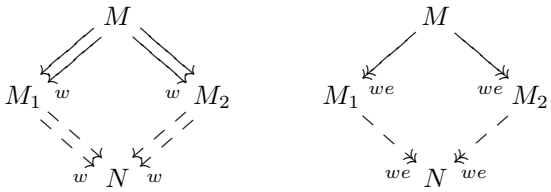


FIG. 3.10 – *Confluence locale forte de $\rightrightarrows_{we}$ et confluence de $\rightarrow_{we}$*

La combinaison des résultats précédents permet d'obtenir la confluence locale forte de $\rightrightarrows_{we}$. Ce qui implique par la suite la confluence du λ-calcul faible étiqueté présenté dans cette section.

Lemme 3.16 (Confluence locale forte) *Si $M \rightrightarrows_{we} M_1$ et $M \rightrightarrows_{we} M_2$, il existe un terme M' tel que $M_1 \rightrightarrows_{we} M'$ et $M_2 \rightrightarrows_{we} M'$.*

Preuve : Cette preuve est essentiellement similaire à la preuve du lemme 3.6. On prouve la propriété plus générale suivante. Si $M \rightrightarrows^{\mathcal{X}}_{we} M_1$ et $M \rightrightarrows^{\mathcal{X}}_{we} M_2$, alors il existe un terme M' tel que $M_1 \rightrightarrows^{\mathcal{X}}_{we} M'$ et $M_2 \rightrightarrows^{\mathcal{X}}_{we} M'$. On procède par récurrence sur la taille de M. Le cas crucial est $M = (\alpha' \circ (\lambda x.N)^{\beta}\ P)^{\alpha}$ avec $M_1 = \lceil\beta'\rceil : (\beta' \circledcirc N_1)\{x \backslash \lfloor\beta'\rfloor : P_1\}$ (où $\beta' = \alpha\alpha'\beta$) et $M_2 = (\alpha' \circ (\lambda x.N_2)^{\beta}\ P_2)^{\alpha}$. On a $N \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} N_1$, $P \rightrightarrows^{\mathcal{X}}_{we} P_1$, $N \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} N_2$ et $P \rightrightarrows^{\mathcal{X}}_{we} P_2$ et $\forall z \in \mathcal{X}\ .\ z \notin \mathtt{FV}(M)$. De là, par hypothèse de récurrence, on obtient deux termes N' et P' tels que $N_1 \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} N'$, $N_2 \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} N'$, $P_1 \rightrightarrows^{\mathcal{X}}_{we} P'$ et $P_2 \rightrightarrows^{\mathcal{X}}_{we} P'$. En utilisant la définition de $\rightrightarrows_{we}$, on obtient $M_2 \rightrightarrows^{\mathcal{X}}_{we} \lceil\beta'\rceil : (\beta' \circledcirc N')\{x \backslash \lfloor\beta'\rfloor : P'\}$ et $\lfloor\beta'\rfloor : N_1 \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} \lfloor\beta'\rfloor : N'$. Comme $x \in \mathcal{X} \cup \{x\}$, en utilisant le lemme 3.15, on obtient $\beta' \circledcirc N_1 \rightrightarrows^{\mathcal{X}\cup\{x\}}_{we} \beta' \circledcirc N_2$. De là, en utilisant le lemme 3.14, on obtient $M_1 \rightrightarrows^{\mathcal{X}}_{we} \lceil\beta'\rceil : (\beta' \circledcirc N')\{x \backslash \lfloor\beta'\rfloor : P'\}$. □

Théorème 3.6 (Confluence) *Si $M \twoheadrightarrow_{we} M_1$ et $M \twoheadrightarrow_{we} M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_{we} N$ et $M_2 \twoheadrightarrow_{we} N$.*

Ces propriétés sont illustrées sur la figure 3.10.

L'adaptation du théorème des développements finis ne présente pas de difficulté supplémentaire par rapport au théorème correspondant de la section précédente.

Théorème 3.7 (Développements finis) *Soit $\mathcal{F}$ un ensemble de radicaux de M.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Tous les développements de $\mathcal{F}$ finissent sur un même terme N.*
3. *L'ensemble des résidus d'un radical R de M dans N est indépendant du développement considéré.*

De même, on obtient le théorème de standardisation en procédant de la même façon que pour le λ-calcul faible. On obtient tout d'abord les propriétés suivantes sur les positions relatives des résidus et des radicaux créés. La propriété de standardisation en découle directement.

Lemme 3.17 *On suppose $M \xrightarrow{S}_{we} N$ et R est un radical de M situé à strictement gauche de S : $R <_g S$.*

1. *R a un unique résidu R' dans N.*
2. *Si T' est un radical créé de N alors $R' <_g T'$.*
3. *Si T est un radical de M tel que $R \leq_g T$, alors chaque résidu T' de T dans N vérifie $R' \leq_g T'$.*

Théorème 3.8 (Standardisation) *Si $M \twoheadrightarrow_{we} M'$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_{we} M'$ telle que $\mathcal{R}$ est standard.*

Nous avons montré dans cette section que le λ-calcul faible étiqueté que nous avons introduit préserve les propriétés fondamentales vérifiée par le λ-calcul faible. Ce calcul étiqueté est confluent et vérifie les propriétés des développements finis et de standardisation. Nous examinons dans la section suivante comment ces étiquettes permettent d'exprimer une notion de partage inspirée du deuxième algorithme de Wadsworth [43].

3.3 Partage de sous-termes

Dans cette section, on montre comment les étiquettes du λ-calcul faible étiqueté permettent d'exprimer la notion de partage. Pour cela, on s'inspire des travaux de Wadsworth, Shivers et Wand [43, 38]. En particulier, on observe que notre définition de la (β_{we})-réduction et plus précisément la définition de la diffusion rejoignent l'idée centrale de ces travaux : la contraction d'un radical affecte les chemins du corps de la fonction qui mènent aux variables ; les autres sous-termes peuvent en revanche être partagés. Dans la suite de cette section, on prouve que, sous certaines conditions, deux sous-termes ayant la même étiquette de tête sont égaux dans le λ-calcul faible étiqueté. Cette propriété est formellement énoncée par l'invariant suivant.

Invariant 3.1 *Le terme M vérifie l'invariant $\mathcal{P}$, ce que l'on note $\mathcal{P}(M)$, si et seulement si, pour toute paire de sous-termes (N,P) de M vérifiant $\tau(N) = \tau(P)$, on a $N = P$.*

Cet invariant est la propriété centrale de cette section. Si $\mathcal{P}(M)$ est vrai, alors tous les sous-termes ayant la même étiquette de tête sont égaux. Dans ce cas, les étiquettes expriment formellement la notion de partage. Plus concrètement, tous les termes ayant la même étiquette pourraient être partagés. Si on représente les termes par des graphes acycliques dirigés, toutes les arêtes pointant vers des termes ayant la même étiquette peuvent être dirigées vers le même nœud.

La question est maintenant de savoir si cet invariant est préservé par réduction. En fait, la réduction du λ-calcul faible étiqueté n'est pas la bonne notion pour tester la préservation du partage. En effet, si le terme que l'on considère est représenté sous une forme partagée, un radical de ce terme est, en réalité, un radical partagé. Ce radical partagé représente tous les radicaux du terme portant le même nom. On en conclut que la réduction pertinente est la réduction qui contracte l'ensemble des radicaux portant un certain nom. Cette réduction, appelée *réduction complète* et notée $\overset{\alpha'}{\Rightarrow}_{we}$, est le développement fini de l'ensemble des radicaux de M portant le nom α'. La notation $\twoheadrightarrow_{we}$ désigne une succession éventuellement vide de réductions complètes. La suite de cette section est consacrée à prouver la préservation de $\mathcal{P}$ par une réduction complète.

Dans le λ-calcul faible étiqueté, le nom d'un radical reflète son histoire. Comme on l'a vu dans la section 3.1, si un radical S est résidu d'un radical R, ces radicaux ont le même nom. Et si un radical S est créé par la contraction d'un radical R, le nom de R est contenu dans le nom de S. On formalise cette propriété centrale du λ-calcul faible étiqueté avec la relation $\prec$ définie ci-après. On dit que α' est contenu strictement dans β', ce que l'on écrit $\alpha' \prec \beta'$, dans les cas suivants.

$$\alpha' \prec \lceil \alpha' \rceil \qquad \alpha' \prec \lfloor \alpha' \rfloor \qquad \alpha' \prec [\alpha',\beta]$$

$$\alpha' \prec \beta_i \;\Rightarrow\; \alpha' \prec \beta_1 \cdots \beta_n \qquad \alpha' \prec \beta' \prec \gamma' \;\Rightarrow\; \alpha' \prec \gamma'$$

Un nom de radical α' est contenu dans une étiquette dans laquelle il apparaît souligné ou surligné. Il est aussi contenu dans un marquage dont il est le premier composant. S'il est contenu dans une étiquette atomique β_i, il est aussi contenu dans une étiquette composée contenant β_i. Cette relation est, en outre, fermée par transitivité. La relation $\prec$ est un ordre strict. Cette relation exprime l'interprétation intuitive que nous avons donnée au moment de la définition des étiquettes : on a $\alpha' \prec \beta'$ si la contraction d'un radical de nom α' participe à la création de β'. On rappelle

que la réduction $M \rightarrow_{we} N$ crée le radical S dans N si S n'est pas le résidu d'un radical R de M. Le résultat suivant formalise cette intuition.

Lemme 3.18 *Si $M \xrightarrow[we]{R} N$ et si le radical S de N est créé par cette réduction, alors on a $\mathsf{nom}(R) \prec \mathsf{nom}(S)$.*

Ce lemme se prouve par une inspection de cas élémentaire. Le nom d'un radical créé contient strictement le nom du radical qui l'a créé. L'exemple de la figure 3.8 de la page 80 illustre cette propriété fondamentale du langage.

Pour prouver la préservation de l'invariant $\mathcal{P}$, on utilise deux invariants auxiliaires. Le premier de ces invariants porte sur les étiquettes du terme.

Invariant 3.2 *Le terme M vérifie l'invariant $\mathcal{Q}$, ce que l'on note$\mathcal{Q}(M)$, si et seulement si, pour tout radical R de M et tout sous-terme N de M, on a $\mathsf{nom}(R) \not\prec \tau(N)$.*

Formellement, l'invariant $\mathcal{Q}(M)$ signifie que les noms des radicaux de M sont maximaux dans M. Plus intuitivement, cette condition s'interprète de la façon suivante : si R est un radical de M alors, aucun sous-terme de M n'a été créé par la contraction d'un radical portant le même nom que R. Cette propriété n'est pas préservée par réduction car plusieurs radicaux portant le même nom peuvent coexister dans M. En réduisant un seul de ces radicaux, on perdrait l'invariant $\mathcal{Q}$. En revanche, en réduisant tous ces radicaux, on retrouve l'invariant $\mathcal{Q}$. Plus précisément, cette propriété est préservée par une réduction complète comme le montre le lemme suivant.

Lemme 3.19 *Si $\mathcal{Q}(M)$ et $M \overset{\gamma'}{\Rightarrow}_{we} M'$, alors on a $\mathcal{Q}(M')$.*

Preuve : Soit R un radical de M' de nom α'. Soit N un sous-terme de M' dont l'étiquette de tête est β avec $\alpha' \prec \beta$.

1. Si R est un résidu d'un radical R' de M, alors $\mathsf{nom}(R) = \mathsf{nom}(R') = \alpha'$. Le cas où β est une étiquette de M est impossible du fait de $\mathcal{Q}(M)$. L'étiquette β est donc créée par la réduction de M à M'. On considère les différents cas de création.
   (a) Si $\beta = \lceil\gamma'\rceil$, alors ona $\alpha' \prec \lceil\gamma'\rceil$. Le cas $\alpha' = \gamma'$ est impossible par définition de $\overset{\gamma'}{\Rightarrow}_{we}$. On a donc $\alpha' \prec \gamma'$. Si $\gamma' = \gamma_1 \dots \gamma_n$, il existe un indice i tel que $1 \leq i \leq n$ et $\alpha' \prec \gamma_i$. Il existe donc un sous-terme P de M dont l'étiquette de tête est γ_i avec $\alpha' \prec \gamma_i$. Ceci contredit $\mathcal{Q}(M)$.
   (b) Les cas $\beta = \lfloor\gamma'\rfloor$ ou $\beta = [\gamma',\beta_1]$ sont similaires au cas précédent.
2. Si R est créé par la réduction de M à M', alors on a $\gamma' \prec \alpha' \prec \beta$. Si β est une étiquette de M, on a $\gamma' \prec \beta$, ce qui contredit $\mathcal{Q}(M)$. L'étiquette β est donc créée par la réduction de M à M'. On considère les différents cas de création.
   (a) Si $\beta = \lceil\gamma'\rceil$, on a $\gamma' \prec \beta$. Comme $\alpha' \prec \lceil\gamma'\rceil$, on a $\alpha' \preceq \gamma'$. De là, comme $\gamma' \prec \alpha'$, on a $\alpha' \prec \alpha'$, ce qui est contradictoire.
   (b) Les cas $\beta = \lfloor\gamma'\rfloor$ ou $\beta = [\gamma',\beta_1]$ sont similaires au cas précédent. □

Ce résultat de préservation de l'invariant $\mathcal{Q}$ a un corollaire important. Après une réduction complète des radicaux de nom α', aucun radical créé par la suite ne peut porter le nom α'. Les noms des radicaux contractés au cours d'une succession de réductions complètes sont donc distincts.

Le deuxième invariant utilisé pour prouver la préservation de $\mathcal{P}$ porte sur les variables. La nature de variable libre ou liée dépend du contexte de l'occurrence de la variable. Par conséquent, on ne peut pas partager une variable libre et une variable liée. Dans le cadre de notre langage étiqueté, cette contrainte se traduit par le fait qu'une occurrence de variable libre ne peut pas avoir la même étiquette qu'une occurrence de variable liée. Cette propriété est énoncée par l'invariant $\mathcal{R}$.

Invariant 3.3 *Un terme M vérifie l'invariant $\mathcal{R}$, ce que l'on note $\mathcal{R}(M)$, si et seulement si, pour toute paire (x^α,y^α) d'occurrences de variables de M ayant la même étiquette, la variable x est libre dans M si et seulement si la variable y est libre dans M.*

L'invariant $\mathcal{R}$ est préservé par une réduction étiquetée comme l'énonce le lemme suivant.

Lemme 3.20 *Si $\mathcal{R}(M)$ et $M \to_{we} M'$, alors on a $\mathcal{R}(M')$.*

Preuve : On remarque tout d'abord qu'une variable x libre (respectivement liée) dans M' ne peut venir que d'une variable x libre (resp. liée) dans M'. On considère deux sous-termes x^α et y^α de M'. Ces sous-termes viennent nécessairement de sous-termes x^α et y^α de M. Par application de $\mathcal{R}(M)$, la variable x est libre dans M si et seulement si la variable y est libre dans M. On conclut en utilisant la remarque préliminaire. □

En combinant les invariants $\mathcal{P}$, $\mathcal{Q}$ et $\mathcal{R}$, on peut aboutir au résultat central de ce chapitre, à savoir le théorème de partage. Pour cela, on utilise le lemme suivant qui énonce la préservation de $\mathcal{P}$ par la réduction complète.

Lemme 3.21 *Si $\mathcal{P}(M) \wedge \mathcal{Q}(M) \wedge \mathcal{R}(M)$ et $M \stackrel{\gamma'}{\Rightarrow}_{we} M'$, alors on a $\mathcal{P}(M')$.*

Preuve : Le nom γ' se décompose nécessairement en $\gamma' = \gamma_0\beta_1 \dots \beta_n\gamma_1$ où $\beta' = \beta_1 \dots \beta_n$. Les radicaux de M de nom γ' ont tous γ_0 pour étiquette de tête. Par $\mathcal{P}(M)$, ils sont donc tous égaux : on les note $R = (\beta' \circ (\lambda x.N)^{\gamma_1} P)^{\gamma_0}$. Et leurs contractums sont $R' = \lceil\gamma'\rceil : (\gamma' \textcircled{x} N)\{x \backslash \lfloor\gamma'\rfloor : P\}$. Comme ces radicaux de nom γ' sont tous égaux, ils sont disjoints. Par $\mathcal{R}(M)$, tous les sous-termes R de M sont des radicaux si au moins l'un d'eux est radical de M. Soient A' et B' deux sous-termes de M' qui vérifient $\tau(A') = \tau(B') = \alpha$.

1. Si α existe dans M, par $\mathcal{Q}(M)$, on a $\gamma' \not\prec \alpha$. Dans ce cas, A' et B' viennent de sous-termes A et B de M avec $\tau(A) = \tau(B) = \alpha$. On obtient donc, par $\mathcal{P}(M)$, $A = B$. Deux cas sont à envisager :
   (a) Si A' contient un contractum, alors deux cas sont possibles.
      i. Si la variable x (liée par R) est libre dans A. Le terme A est donc inclus dans l'abstraction de R. Par conséquent, A' contient $\lfloor\gamma'\rfloor : P$. Comme R ne peut être inclus dans A (les radicaux de nom γ' sont disjoints), R' est nécessairement contenu dans P. Ceci est impossible car les radicaux de nom γ' sont disjoints dans M. Ce cas est impossible.
      ii. Si A contient R. Comme tous les sous-termes R de A et B sont des radicaux, on a $A \stackrel{\gamma'}{\Rightarrow}_{we} A'$ et $A = B \stackrel{\gamma'}{\Rightarrow}_{we} B'$. En conséquence, on a bien $A' = B'$.
   (b) Si A' ne contient pas de contractum de R, on considère les cas suivants.
      i. Si A' est disjoint des contractums, alors A est disjoint des radicaux R dans M et donc $A = A'$.
      ii. Si A' est dans la partie argument d'un contractum, alors A est un sous-terme de M inclus dans la partie argument d'un radical R. On a donc $A = A'$.
      iii. Si A' est dans le corps d'un contractum, alors A est un sous-terme de M inclus dans le corps de la partie fonction d'un radical R. Le terme A ne contient pas de variable liée par R, sinon son étiquette ne serait pas la même que celle de A'. Il ne subit donc ni substitution, ni diffusion entre M et M' : on a $A = A'$.

   Dans tous ces cas, on a $A = A'$. Comme $A = B$, B ne peut pas contenir R ou une occurrence de la variable liée par R. On a donc $B' = B$ et $A' = B'$.
2. Si α est créée entre M et M', on a $\gamma' \prec \alpha$.
   (a) Si $\alpha = [\gamma',\alpha_1]$. Comme les radicaux de nom γ' sont disjoints, un sous-terme de M ne peut subir entre M et M' qu'une seule diffusion. Par conséquent, il existe dans M deux sous-termes A et B dont les étiquettes de tête sont α_1 et tels que $A' = (\gamma' \textcircled{x} A)\{x \backslash \lfloor\gamma'\rfloor : P\}$ et $B' = (\gamma' \textcircled{x} B)\{x \backslash \lfloor\gamma'\rfloor : P\}$. Par $\mathcal{P}(M)$, on obtient $A = B$, ce qui implique $A' = B'$.
   (b) Si $\alpha = \lceil\gamma'\rceil$, par $\mathcal{Q}(M)$, cette étiquette est nécessairement créée en haut des contractums des radicaux γ' qui sont tous égaux. On a donc $A' = B'$.

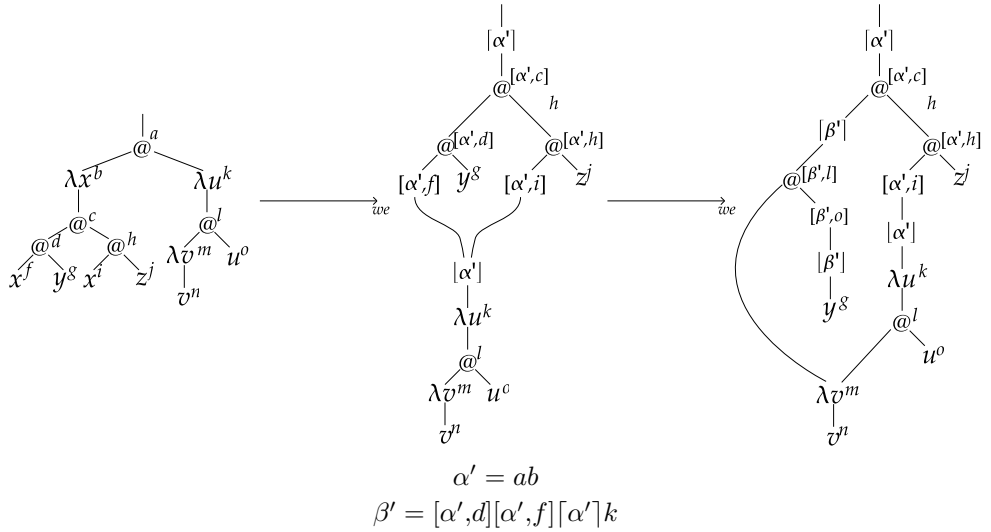


FIG. 3.11 – *Réduction étiquetée de* $M = ((\lambda x.((x^f\ y^g)^d\ (x^i\ z^j)^h)^b\ (\lambda u.((\lambda v.v^n)^m\ u^o)^l)^k)^a$

(c) Si $\alpha = \lfloor\gamma'\rfloor$, par $\mathcal{Q}(M)$, cette étiquette est créée en haut des copies de l'argument de R dans les contractums. Ces copies sont toutes égales : on a $A' = B'$. □

Si le terme M vérifie les invariants $\mathcal{P}$, Q et $\mathcal{R}$ et si M se réduit par réduction complète vers N, alors N vérifie l'invariant $\mathcal{R}$. Cette propriété de préservation permet d'obtenir le théorème de partage des sous-termes, qui correspond à une implémentation du partage par graphes acycliques dirigés de l'évaluation des termes dans le λ-calcul faible. Pour énoncer ce théorème, on utilise l'invariant `INIT` qui a été défini dans la partie 1.2.

Théorème 3.9 (Partage des sous-termes) *Si* $\texttt{INIT}(M)$ *et* $M \twoheadrightarrow_{we} N$*, alors on a* $\mathcal{P}(N)$*.*

Preuve : On remarque tout d'abord que si $\texttt{INIT}(M)$ est vrai, alors $\mathcal{P}(M) \wedge \mathcal{Q}(M) \wedge \mathcal{R}(M)$ est vrai. On peut donc conclure par le lemme précédent. □

Si les étiquettes de M sont des lettres distinctes et si M se réduit de façon complète vers N, alors les sous-termes de N qui ont la même étiquette sont égaux. Ceci signifie que les étiquettes du λ-calcul faible expriment la notion de partage. Pour illustrer ce théorème, on considère, sur la figure 3.11, la réduction du terme qui était mentionné en introduction. On observe en particulier que le partage dans le terme final est le même que sur la figure 3.1, lorsque le deuxième algorithme de Wadsworth est utilisé.

Dans ce chapitre, nous avons examiné le λ-calcul faible qui est une variante du λ-calcul dans laquelle la règle (ξ) n'est pas autorisée : on ne contracte pas les radicaux situés sous une abstraction. De façon générale, le λ-calcul faible n'est pas confluent. Dans cette partie, nous avons plus précisément considéré le λ-calcul faible proposé par Lévy et Maranget dans [30]. Ce calcul remplace la règle (ξ) par une règle (ξ') qui autorise la contraction d'un radical situé sous une abstraction si ce dernier ne contient pas d'occurrence d'une variable liée par une abstraction. Ce calcul est localement confluent et confluent. Les théorèmes des développements finis et de standardisation s'obtiennent en utilisant les mêmes techniques que celles employées pour le λ-calcul par valeur. Comme l'a montré Wadsworth dans [43], la notion de partage dans le λ-calcul faible est plus simple dans la mesure où on peut réduire faiblement des termes avec partage en représentant les termes par des graphes acycliques orientés. Plutôt que d'exposer explicitement une structure de donnée pour exprimer le partage, on préfère ici utiliser un langage étiqueté. Le λ-calcul faible étiqueté reste localement confluent et confluent, puisque les notions de confluence et de partage sont assez

largement indépendantes. De même, les théorèmes des développements finis et de standardisation s'obtiennent de la même façon que dans le chapitre précédent. Le résultat central de cette partie est le théorème de partage : si les étiquettes de M sont des lettres distinctes et si M se réduit de façon complète vers N, alors les sous-termes de N qui ont la même étiquette sont égaux. Le fait de pouvoir exprimer le partage grâce aux étiquettes dispense de choisir une structure de donnée. En restant dans le cadre d'un langage familier, on peut bénéficier en outre des résultats et des techniques développés depuis des décennies pour le λ-calcul.

Chapitre 4

Inspection de pile

Le mécanisme d'inspection de pile est utilisé à une grande échelle puisqu'il est intégré dans la JVM [17, 31] et le CLR [9, 26] qui sont les machines virtuelles de Java et $C\#$. Il permet de contrôler dynamiquement l'exécution d'un programme dont les composants peuvent avoir des origines diverses. Comme on n'accorde pas la même confiance à ses propres fonctions et aux fonctions téléchargées d'internet, ce contrôle dynamique de l'exécution d'une fonction dépend (1) son origine et (2) la chaîne d'appels conduisant à son exécution.

Plus précisément, ce système utilise un type particulier d'objet, les permissions, qui représentent les droits attribués aux fonctions. Par exemple, les objets de la classe FileIOPermission représentent les droits d'accès aux fichiers. Un de ces objets peut donner le droit d'accéder en lecture seule à un fichier spécifique. Un autre objet peut donner le droit d'accéder en lecture et écriture à l'ensemble des fichiers d'un répertoire. En fonction de l'origine du programme (internet, entreprise de logiciel, machine locale), des permissions dites *statiques* sont attribuées aux fonctions. Par exemple, une fonction issue de la bibliothèque standard de la machine virtuelle bénéficiera de permissions plus étendues qu'une fonction téléchargée d'internet. Les permissions statiques sont utilisées pour calculer les permissions *dynamiques* qui sont les permissions effectivement disponibles au moment de l'exécution de la fonction. Les permissions dynamiques sont l'intersection des permissions statiques des fonctions présentes sur la pile d'appel. Cette définition comporte une exception : une fonction peut augmenter les permissions dynamiques dans la limite de ses permissions statiques. Cette instruction, que nous noterons par la suite `grant` T `in` M, ajoute l'ensemble des permissions de T aux permissions dynamiques de M. Le contrôle d'exécution consiste à tester l'appartenance d'un ensemble de permissions à l'ensemble des permissions dynamiques. Cette instruction, qui sera notée plus tard `check` T `for` M, teste si les permissions de T appartiennent bien aux permissions dynamiques courantes. En cas de succès du test, l'exécution se poursuit normalement ; en cas d'échec, une exception de sécurité est levée. Par exemple, au moment d'accéder en lecture à un fichier, la fonction de librairie chargée d'effectuer cette lecture vérifie que la permission de la classe FileIOPermission associée à cette opération est bien présente dans les permissions dynamiques courantes.

Si l'inspection de pile peut être utile en pratique, ce mécanisme est un système *ad hoc* et les travaux de Fournet et Gordon [15] ont montré qu'il est étonnamment difficile d'exprimer quelle propriété de sécurité est garantie. Ce constat a donné lieu à de nombreux travaux visant à proposer des solutions alternatives. Il peut s'agir d'une sémantique différente comme pour Wallach, Appel et Felten [44] ou Abadi et Fournet [2]. Ou bien d'un système de type permettant de s'affranchir des tests dynamiques, comme l'ont proposé Pottier, Skalka et Smith [35]. Une autre approche a consisté à définir des analyses statiques permettant d'avoir une approximation des piles d'appel atteignables par un programme. De là, Jensen, Le Métayer et Thorn vérifient si ces piles se conforment à une

propriété de sécurité [23]. Besson, Grenier de Latour et Jensen déterminent une condition suffisante sur la pile pour assurer qu'une propriété de sécurité ne sera pas violée par la fonction analysée [8].

Dans ce chapitre, notre but est de rapprocher le mécanisme d'inspection de pile et le λ-calcul étiqueté. Plus précisément, nous nous fondons sur λ_{sec}-calcul, le langage introduit pas Fournet et Gordon dans [15] dont la syntaxe est rappelée dans la section 4.1. L'inspection de pile exploite une information locale sur l'origine des fonctions sous la forme des permissions statiques. Ces informations locales se composent pour donner une information globale sur le contexte d'appel d'une fonction : les permissions dynamiques. Dans le cas du λ-calcul étiqueté, les étiquettes, qui permettent de déterminer l'origine des termes, correspondent naturellement aux permissions statiques. En guise d'information globale, nous faisons appel aux chemins et chemins-contextes introduits dans la partie 1.3. De façon analogue à l'inspection de pile, ces chemins sont composés des étiquettes qui représentent l'information locale. Le contrôle de l'exécution consiste à conditionner la contraction d'un radical par l'information locale matérialisée par l'étiquette du radical et par l'information globale issue du chemin-contexte. Nous étudions les propriétés élémentaires du langage obtenu, le λ_t-calcul, au travers du prisme que constitue la propriété de confluence locale. Dans la section 4.3, on instancie les paramètres du λ_t-calcul pour se rapprocher davantage du mécanisme d'inspection de pile. On obtient ainsi le λ_{ts}-calcul. Ce langage se rapproche d'une variante du λ_{sec}-calcul que Fournet et Gordon avaient proposée dans [15]. On montre qu'il existe une traduction correcte de cette variante vers le λ_{ts}-calcul.

4.1 Le λ_{sec}-calcul : l'inspection de pile formalisée

Fournet et Gordon formalisent le mécanisme d'inspection de pile grâce à un langage fondé sur le λ-calcul : le λ_{sec}-calcul. Ce langage est muni des ingrédients élémentaires de l'inspections de pile. En particulier, ce langage fait intervenir les permissions et les principaux.

Permissions	$p, q \in \mathcal{P}$
Principaux	$T, S, D \subseteq \mathcal{P}$

Les permissions sont ici simplement des éléments d'un ensemble $\mathcal{P}$. Intuitivement, ces permissions sont des droits associés à certaines opérations. Ainsi, à l'action d'écrire dans un fichier, on associe la permission **FileIO**. A l'action d'afficher un message sur l'écran, on associe la permission **ScreenIO**. La notion de principal fait habituellement référence à une entité dont l'identité est utilisée. Un principal peut être un individu, une organisation ou un programme. Ici, un principal est un ensemble de permissions. Intuitivement, cet ensemble correspond aux droits accordés à un individu, une organisation, etc. Les permissions et les principaux interviennent dans le langage dont la syntaxe est définie de la façon suivante.

$M, N ::=$	x	Variable
	$\lambda x.M$	Abstraction
	MN	Application
	$T\langle M\rangle$	Cadre
	`grant` T `in` M	Terme privilégié
	`demand` T `then` M `else` N	Test
	`fail`	Echec

Quatre nouvelles constructions sont ajoutées au λ-calcul. Un cadre permet de définir les permissions statiques locales. Intuitivement, le sous-terme M du cadre $T\langle M\rangle$ est issu du principal T et est réduit avec les permissions statiques T. Le terme privilégié `grant` T `in` M permet de réduire le sous-terme M avec les permissions dynamiques supplémentaires T (dans la limite des permissions statiques de

M). Le terme $\texttt{demand}\ T\ \texttt{then}\ M\ \texttt{else}\ N$ permet de tester les permissions dynamiques. Le succès de ce test débouche sur M ; l'échec conduit à N. Le terme `fail` est utilisé pour coder un échec. Certains termes sont singularisés : les valeurs et les sorties.

$$\begin{array}{ll} \text{Valeurs} & V ::= \lambda x.M \\ \text{Sorties} & O ::= V \mid \texttt{fail} \end{array}$$

Une valeur est une abstraction. Une sortie peut être une valeur ou un échec. La réduction $\rightarrow_{\mathsf{sec}}^{S,D}$ est paramétrée par deux ensembles de permissions : l'ensemble des permissions statiques S et l'ensemble des permissions dynamiques D. Les règles de réductions sont définies sur la figure 4.1. La règle ($\text{App}_{\mathsf{sec}}$) correspond à une β-réduction en appel par valeur. Les règles ($\text{AppL}_{\mathsf{sec}}$) et ($\text{AppR}_{\mathsf{sec}}$) spécifient l'ordre d'évaluation des applications : le sous-terme gauche est réduit en premier. La règle ($\text{Demand}_{\mathsf{sec}}$) teste les permissions dynamiques pour l'ensemble de permission T. Si toutes les permissions de T appartiennent à D, alors le test est un succès. Dans ce cas, le terme $M_{\texttt{tt}}$ est évalué. Sinon, le test est un échec et le terme $M_{\texttt{ff}}$ est évalué. La règle ($\text{Ctx Frame}_{\mathsf{sec}}$) montre l'effet du cadre sur les permissions statiques et dynamiques. Intuitivement, le sous-terme M de $T\langle M\rangle$ est issu du principal T. Il se réduit donc avec les permissions statiques T. Les permissions dynamiques, qui ne peuvent dépasser les permissions statiques, sont ajustées en conséquence par une intersection. La règle ($\text{Ctx Grant}_{\mathsf{sec}}$) montre l'effet d'un terme privilégié $\texttt{grant}\ T\ \texttt{in}\ M$. Ce terme permet d'évaluer le sous-terme M avec des permissions dynamiques T supplémentaires. Ces permissions dynamiques sont, comme toujours, bornées par les permissions statiques. Les permissions dynamiques deviennent donc $S\cap(D\cup T)$. Comme le montrent les règles ($\text{Red Frame}_{\mathsf{sec}}$) et ($\text{Red Grant}_{\mathsf{sec}}$), une fois qu'on a obtenu une sortie sous un cadre ou un terme privilégié, ce dernier disparaît. Les règles ($\text{Fail Rator}_{\mathsf{sec}}$) et ($\text{Fail Rand}_{\mathsf{sec}}$) montrent que l'échec `fail` se comporte comme une exception et remonte à la tête du terme.

Le langage introduit par Fournet et Gordon s'inspire directement du mécanisme d'inspection de pile, tel qu'il est utilisé dans les machines virtuelles JVM ou CLR. Avec ce formalisme, Fournet et Gordon montrent qu'il est difficile de formuler quelle propriété de sécurité est garantie par l'inspection de pile. En particulier, la règle ($\text{Red Frame}_{\mathsf{sec}}$) entraîne la disparition des cadres autour

$$\begin{array}{ll}
(\text{App}_{\mathsf{sec}}) & (\lambda x.M)V \rightarrow_{\mathsf{sec}}^{S,D} M\{x\backslash V\} \\
(\text{AppL}_{\mathsf{sec}}) & \dfrac{M \rightarrow_{\mathsf{sec}}^{S,D} M'}{MN \rightarrow_{\mathsf{sec}}^{S,D} M'N} \\
(\text{AppR}_{\mathsf{sec}}) & \dfrac{M \rightarrow_{\mathsf{sec}}^{S,D} M'}{VM \rightarrow_{\mathsf{sec}}^{S,D} VM'} \\
(\text{Demand}_{\mathsf{sec}}) & \texttt{demand}\ R\ \texttt{then}\ M_{\texttt{tt}}\ \texttt{else}\ M_{\texttt{ff}} \rightarrow_{\mathsf{sec}}^{S,D} M_{R\subseteq D} \\
(\text{Ctx Frame}_{\mathsf{sec}}) & \dfrac{M \rightarrow_{\mathsf{sec}}^{R,R\cap D} M'}{R\langle M\rangle \rightarrow_{\mathsf{sec}}^{S,D} R\langle M'\rangle} \\
(\text{Ctx Grant}_{\mathsf{sec}}) & \dfrac{M \rightarrow_{\mathsf{sec}}^{S,S\cap(D\cup R)} M'}{\texttt{grant}\ R\ \texttt{in}\ M \rightarrow_{\mathsf{sec}}^{S,D} \texttt{grant}\ R\ \texttt{in}\ M'} \\
(\text{Red Frame}_{\mathsf{sec}}) & T\langle o\rangle \rightarrow_{\mathsf{sec}}^{S,D} o \\
(\text{Red Grant}_{\mathsf{sec}}) & \texttt{grant}\ R\ \texttt{in}\ o \rightarrow_{\mathsf{sec}}^{S,D} o \\
(\text{Fail Rator}_{\mathsf{sec}}) & \texttt{fail}\ M \rightarrow_{\mathsf{sec}}^{S,D} \texttt{fail} \\
(\text{Fail Rand}_{\mathsf{sec}}) & V\ \texttt{fail} \rightarrow_{\mathsf{sec}}^{S,D} \texttt{fail}
\end{array}$$

FIG. 4.1 – *Règles de réduction du* λ_{sec}*-calcul*

des abstractions. Le principal associé à cette abstraction est, dans ce cas, définitivement perdu. Pour illustrer cette faiblesse, on s'inspire d'un exemple mentionné dans [2]. Dans cet exemple, $S \subseteq \mathcal{P}$ est un principal d'une application locale qui bénéficie de droits étendus, en particulier **FileIO** $\in S$. Le principal U qui vérifie $U \subseteq \mathcal{P}$ correspond à un programme téléchargé d'internet qui dispose de peu de droits. En particulier, on a **FileIO** $\notin U$. Les fonctions de S (intuitivement sûres) sont nommées M alors que celles de U (potentiellement dangereuses) sont notées N. Pour alléger la notation de l'exemple, on utilise ici la notation $\rightarrow_{\mathsf{sec}} = \rightarrow_{\mathsf{sec}}^{\mathcal{P},\mathcal{P}}$.

$$\begin{aligned} M_{\mathsf{Del}} &= \lambda x.S\langle \texttt{demand}\ \{\textbf{FileIO}\}\ \texttt{then}\ (M_{\mathsf{pDel}}\ x)\ \texttt{else}\ \texttt{fail}\rangle \\ N_{\mathsf{Vir}} &= U\langle M_{\mathsf{Del}}\ V_{\mathsf{sys}}\rangle \end{aligned}$$

$$\mathcal{R}_1 : N_{\mathsf{Vir}} \rightarrow_{\mathsf{sec}} U\langle S\langle \texttt{demand}\ \{\textbf{FileIO}\}\ \texttt{then}\ (M_{\mathsf{pDel}}\ V_{\mathsf{sys}})\ \texttt{else}\ \texttt{fail}\rangle\rangle \rightarrow_{\mathsf{sec}} U\langle S\langle \texttt{fail}\rangle\rangle \twoheadrightarrow_{\mathsf{sec}} \texttt{fail}$$

L'abstraction M_{Del} est une fonction système permettant d'effacer un fichier grâce à une fonction primitive M_{pDel}. Avant de faire cette opération, un test sur la permission **FileIO** est effectué. Comme le montre la réduction $\mathcal{R}_1$ mentionnée ci-dessus, ce test offre une bonne protection face à un virus N_{Vir} venant d'internet qui tenterait d'effacer le fichier système dont le nom est V_{sys}. On considère maintenant la situation suivante.

$$\begin{aligned} M_{\mathsf{Nav}} &= \lambda x.S\langle (M_{\mathsf{Get}}\ x)\ M_{\mathsf{Del}}\rangle \\ M_{\mathsf{Get}} &= \lambda x.S\langle x\ V\rangle \\ N_{\mathsf{Plug}} &= \lambda x.U\langle \lambda y.y\ V_{\mathsf{sys}}\rangle \end{aligned}$$

Un navigateur M_{Nav} efface un fichier temporaire d'un plug-in N_{Plug} issu d'internet. Pour ce faire, le navigateur utilise le terme M_{Get} chargé d'obtenir le nom du fichier temporaire. Dans ce contexte, un plug-in provenant d'internet peut profiter des permissions statiques du navigateur pour effacer le fichier secret, comme le montre la réduction $\mathcal{R}_2$ suivante.

$$\begin{aligned} \mathcal{R}_2 : M_{\mathsf{Nav}}N_{\mathsf{Plug}} &\rightarrow_{\mathsf{sec}} S\langle (M_{\mathsf{Get}}\ N_{\mathsf{Plug}})\ M_{\mathsf{Del}}\rangle \\ &\rightarrow_{\mathsf{sec}} S\langle S\langle N_{\mathsf{Plug}}\ V\rangle\ M_{\mathsf{Del}}\rangle \\ &\rightarrow_{\mathsf{sec}} S\langle S\langle U\langle \lambda y.y\ V_{\mathsf{sys}}\rangle\ M_{\mathsf{Del}}\rangle\rangle \\ &\twoheadrightarrow_{\mathsf{sec}} S\langle (\lambda y.y\ V_{\mathsf{sys}})\ M_{\mathsf{Del}}\rangle \\ &\rightarrow_{\mathsf{sec}} S\langle M_{\mathsf{Del}}\ V_{\mathsf{sys}}\rangle \\ &\rightarrow_{\mathsf{sec}} S\langle S\langle \texttt{demand}\ \{\textbf{FileIO}\}\ \texttt{then}\ (M_{\mathsf{pDel}}\ V_{\mathsf{sys}})\ \texttt{else}\ \texttt{fail}\rangle\rangle \\ &\rightarrow_{\mathsf{sec}} S\langle S\langle M_{\mathsf{pDel}}\ V_{\mathsf{sys}}\rangle\rangle \end{aligned}$$

L'appel de M_{Get} aboutit à une abstraction dont l'origine est perdue du fait de l'application de la règle (Red Frame$_{\mathsf{sec}}$). De ce fait, le test de la permission **FileIO** est contourné et le fichier système est effacé par la primitive d'effacement.

Même si les travaux de Fournet et Gordon montrent la difficulté d'exprimer une propriété de sécurité garantie par l'inspection de pile et son caractère *ad hoc*, nous souhaitons nous inspirer de ce mécanisme pour introduire un système similaire dans le λ-calcul. La principale caractéristique de l'inspection de pile est de faire coexister une information locale (les permissions statiques) et une information globale (les permissions dynamiques). Le but est ici de mettre en application cette idée en se fondant sur le λ-calcul étiqueté dont les étiquettes expriment localement les dépendances vis-à-vis des réductions passées.

4.2 Le λ_t-calcul : un λ-calcul avec un contexte

Nous souhaitons dans cette section introduire un calcul étiqueté qui dispose d'un mécanisme de contrôle de la réduction inspiré de l'inspection de pile. Pour rester au plus près de λ_{sec}, le λ_t-calcul, qui est présenté dans cette section, est fondé sur le λ-calcul par valeur étiqueté. Sa syntaxe est

définie de la façon suivante.

Termes	$M, N \in \mathbf{\Lambda}_t ::=$	x^α	Variable
		$\mid (\lambda x.M)^\alpha_c$	Abstraction
		$\mid (MN)^\alpha$	Application
		$\mid \texttt{fail}$	Echec
Valeurs	$V \in \mathbf{V}_t ::=$	$(\lambda x.M)^\alpha_c$	
Etiquettes	$\alpha, \beta \in \mathbf{E}_t ::=$	$a \mid \alpha\beta \mid \lceil\alpha\vert\beta\rceil \mid \lfloor\alpha\vert\beta\rfloor$	

On retrouve les variables et l'application du λ-calcul étiqueté. On note que l'abstraction $(\lambda x.M)^\alpha_c$ est indicée par une condition c. Comme nous le verrons au moment de la définition de la réduction, au moment de la contraction d'un radical on teste la condition c dont les trois paramètres sont le chemin qui mène au radical, l'étiquette de l'abstraction et l'étiquette de la valeur. L'échec `fail` est utilisé dans le cas où un test échoue. Les valeurs sont les abstractions.

Intuitivement, les étiquettes du λ-calcul apportent une information locale sur l'origine du terme et sur sa dépendance vis-à-vis des réductions précédentes. On peut rapprocher cette information des permissions statiques utilisées dans le mécanisme d'inspection de pile. De même que les permissions dynamiques se construisent à partir des permissions statiques, on fonde l'information globale sur les étiquettes. Pour ce faire, nous réutilisons les notions de chemin et de chemin-contexte introduites dans la partie 1.3. Le chemin menant au radical jouera le même rôle que les permissions dynamiques.

Nœud	$\theta \in \mathbf{N}_t ::=$	$\lambda \mid @_i$	$i \in \{1,2\}$
Chemin-contexte	$\kappa \in \mathbf{K}_t ::=$	$\alpha_1\theta_1\alpha_2\theta_2 \ldots \alpha_n\theta_n$	$n \in \mathbb{N}$
Chemin	$\varphi \in \mathbf{\Phi}_t ::=$	$\alpha\theta_1\alpha_1\theta_2\alpha_2 \ldots \theta_n\alpha_n$	$n \in \mathbb{N}$
Conditions	$c \subseteq$	$\mathbf{\Phi}_t \times \mathbf{E}_t \times \mathbf{E}_t$	

Un chemin (respectivement un chemin-contexte) est une suite alternée de nœuds et d'étiquettes qui finit sur une étiquette (resp. un nœud). Intuitivement, un chemin est formé de la suite des nœuds et étiquettes rencontrés depuis la racine vers un nœud de l'arbre de syntaxe du terme. Une condition est simplement un ensemble de triplets constitués d'un chemin, et de deux étiquettes. Au moment de la contraction d'un radical, on teste simplement l'appartenance à c du triplet constitué du chemin menant à la racine et des étiquettes de tête de l'abstraction et de l'argument.

La réduction du λ_t-calcul est paramétrée par le chemin-contexte associé au contexte du radical réduit. Les règles de réductions sont formalisées de la façon suivante.

$$(\beta_t) \quad \frac{(\kappa\gamma,\alpha,\beta) \in c \quad \tau(V) = \beta}{((\lambda x.M)^\alpha_c V)^\gamma \xrightarrow{\kappa}_t \gamma \cdot \lceil\alpha\vert\beta\rceil \cdot M\{x \backslash \lfloor\alpha\vert V\rfloor\}}$$

$$(\text{Fail}) \quad \frac{(\kappa\gamma,\alpha,\beta) \notin c \quad \tau(V) = \beta}{((\lambda x.M)^\alpha_c V)^\gamma \xrightarrow{\kappa}_t \texttt{fail}}$$

$$(\nu_t) \quad \frac{M \xrightarrow{\kappa\beta@_1}_t M'}{(M\ N)^\beta \xrightarrow{\kappa}_t (M'\ N)^\beta}$$

$$(\mu_t) \quad \frac{N \xrightarrow{\kappa\beta@_2}_t N'}{(M\ N)^\beta \xrightarrow{\kappa}_t (M\ N')^\beta}$$

$$(\xi_t) \quad \frac{M \xrightarrow{\kappa\alpha\lambda}_t M'}{(\lambda x.M)^\alpha_c \xrightarrow{\kappa}_t (\lambda x.M')^\alpha_c}$$

$$(\mathtt{f}^L_t) \quad (\texttt{fail}\ M)^\beta \xrightarrow{\kappa}_t \texttt{fail}$$

$$(\mathtt{f}^R_t) \quad (M\ \texttt{fail})^\beta \xrightarrow{\kappa}_t \texttt{fail}$$

La règle de réduction centrale est adaptée de la règle (β_{ve}) du λ-calcul par valeur étiqueté. Comme annoncé précédemment, au moment de la contraction d'un radical de la forme $((\lambda x.M)^\alpha_c V)^\gamma$, on

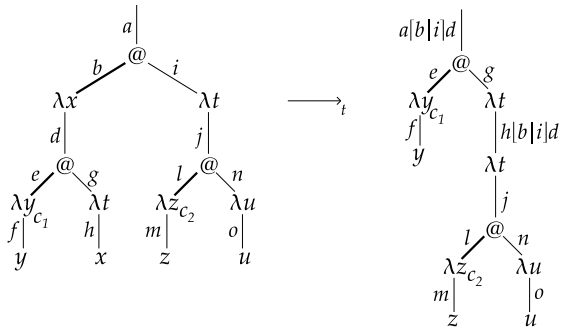


FIG. 4.2 – *Réduction de* $M = ((\lambda x.((\lambda y.y^f)^e_{c_1}(\lambda t.x^h)^g)^d)^b(\lambda t.((\lambda z.z^m)^l_{c_2}(\lambda u.u^o)^n)^j)^i)^a$

teste l'appartenance du triplet $(\kappa\gamma,\alpha,\beta)$ à la condition c associée à l'abstraction. Le chemin $\kappa\gamma$ est le chemin menant au radical contracté. Les étiquettes α et β sont les étiquettes de tête de l'abstraction et de la valeur en argument. En cas de succès du test, la contraction du radical peut avoir lieu de la même manière que dans le λ-calcul par valeur étiqueté. En cas d'échec, on obtient un terme `fail`. Les règles (ν_t), (μ_t) et (ξ_t) sont de simples adaptations des règles de contexte correspondantes du λ-calcul par valeur étiqueté. Les règles de contexte associées à `fail` sont adaptées de λ_{sec}. On remarque que dans le cadre présent, contrairement à λ_{sec}, l'ordre d'évaluation n'est pas fixé.

Les définitions de l'étiquette de tête τ, est reprise de la partie 2.2. Cette fonction n'est pas définies sur le terme `fail`. Les définitions de la fonction de concaténation " $\cdot$ " et de la substitution utilisée ici étendent les définitions correspondantes de la partie 2.2 sur le terme d'échec de la façon suivante.

$$\alpha \cdot \texttt{fail} = \texttt{fail} \qquad\qquad \texttt{fail}\{x\backslash M\} = \texttt{fail}$$

Dans la suite de cette section, on examine la propriété de confluence locale du langage. Comme le λ-calcul par valeur étiqueté est confluent, cette propriété dépend ici essentiellement des tests effectués sur c au moment de la contraction des radicaux. On détermine en particulier des conditions suffisantes sur les conditions c pour obtenir la propriété de confluence locale. Comparativement au λ-calcul par valeur étiqueté, le λ_t-calcul conserve, de façon élémentaire, les propriétés syntaxiques suivantes.

Lemme 4.1 *1.* $(\alpha \cdot M)\{x\backslash V\} = \alpha \cdot (M\{x\backslash V\})$
2. Si $x \neq y$ *et* $x \notin \texttt{FV}(P)$*, alors on a* $M\{x\backslash N\}\{y\backslash P\} = M\{y\backslash P\}\{x\backslash N\{y\backslash P\}\}$.

On conserve la commutation des opérations de concaténation d'une étiquette en tête et de substitution. Le lemme de commutation des substitutions est conservé. On conserve également la compatibilité à gauche de la substitution avec la réduction. Cette propriété est conservée car la substitution ne change pas les chemins menant aux radicaux de M.

Lemme 4.2 (Compatibilité à gauche) *Si* $M \xrightarrow{\kappa}_t M'$*, alors* $M\{x\backslash V\} \xrightarrow{\kappa}_t M'\{x\backslash V\}$.

La difficulté introduite par les tests sur le chemin menant à la racine est illustrée par l'exemple suivant $M = ((\lambda x.((\lambda y.y^f)^e_{c_1}(\lambda t.x^h)^g)^d)^b(\lambda t.((\lambda z.z^m)^l_{c_2}(\lambda u.u^o)^n)^j)^i)^a$. On omet dans ce terme les conditions qui ne nous intéressent pas. Ce terme contient les trois radicaux suivants : $R_0 = M$, $R_1 = ((\lambda y.y^f)^e_{c_1}(\lambda t.t^h)^g)^d$ et $R_2 = ((\lambda z.z^m)^l_{c_2}(\lambda u.u^o)^n)^j$. Si on veut contracter R_1 (ou respectivement R_2), il faut examiner l'appartenance à c_1 (resp. c_2) de (φ_1,e,g) où $\varphi_1 = a@_1b\lambda d$ (resp. (φ_2,l,n) où $\varphi_2 = a@_2i\lambda j$). Sur la figure 4.2, on montre la réduction du radical R_0. Pour contracter les résidus de R_1 et R_2, il faut examiner l'appartenance à c_1 (resp. c_2) de (φ'_1,e,g) où $\varphi'_1 = a\lceil b|i\rceil d$ (resp. (φ'_2,l,n) où $\varphi'_2 = a\lceil b|i\rceil d@_2g\lambda h\lfloor b|i\rfloor\lambda j$). Cet exemple illustre deux faits : (1) les étiquettes de l'abstraction et de la valeur en argument d'un radical résidu sont les mêmes que celles du radical

dont il est résidu. Cette propriété est notamment utilisée pour montrer la confluence du λ-calcul par valeur étiqueté. (2) Le chemin menant à un radical résidu peut être différent du chemin menant au radical dont il est résidu. On en déduit que, en toute généralité, le λ_t-calcul n'est pas localement confluent. En s'inspirant de cet exemple, on exprime une condition suffisante pour obtenir la confluence locale du langage. Pour cela, on utilise les notations suivantes de concaténation d'une étiquette avec un chemin à gauche $\alpha \cdot \alpha_0\theta_1\alpha_1 \dots \theta_n\alpha_n = (\alpha\alpha_0)\theta_1\alpha_1 \dots \theta_n\alpha_n$ et à droite $\alpha_0\theta_1\alpha_1 \dots \theta_n\alpha_n \cdot \alpha = \alpha_0\theta_1\alpha_1 \dots \theta_n(\alpha_n\alpha)$. La concaténation à gauche est aussi utilisée pour les chemins-contextes.

Définition 4.1 (Condition confluente) *Une condition c est confluente si et seulement si elle vérifie les deux propriétés suivantes :*

1. *$\kappa\alpha@_1\beta\lambda\varphi \in c$ si et seulement si, pour tout γ, on a $\kappa(\alpha\lceil\beta|\gamma\rceil \cdot \varphi) \in c$.*
2. *$\kappa\alpha@_2\beta\lambda\varphi \in c$ si et seulement si, pour tout (γ,φ'), on a $\kappa(\alpha\lceil\gamma|\beta\rceil \cdot \varphi' \cdot \lfloor\gamma|\beta\rfloor)\lambda\varphi \in c$.*

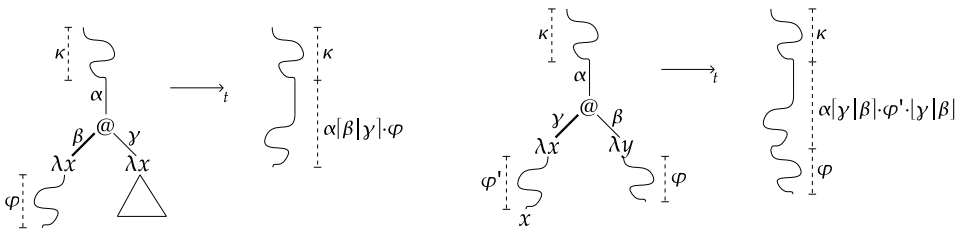
Ces conditions sont illustrées sur la figure 4.3. La première condition correspond à la transformation d'un chemin qui se termine dans l'abstraction du radical contracté. La deuxième condition correspond à la transformation d'un chemin qui se termine dans la valeur en argument d'un radical contracté. Intuitivement, ces deux conditions répondent aux difficultés soulevées dans l'exemple précédent. Ainsi, le test correspondant à la contraction d'un radical R donne le même résultat avant ou après la contraction d'un radical qui contient R que ce soit dans son abstraction ou dans sa valeur en argument. Cette intuition est justifiée par le lemme suivant.

Lemme 4.3 *Si les conditions de M sont confluentes alors les propriétés suivantes sont vérifiées.*

1. *Si $M \xrightarrow{\kappa\alpha@_1\beta\lambda}_t M'$, alors, pour tout γ, on a $\alpha \cdot \lceil\beta|\gamma\rceil \cdot M \xrightarrow{\kappa}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot M'$.*
2. *Si $V \xrightarrow{\kappa\alpha@_2}_t V'$ où $\beta = \tau(V)$, alors, pour tout (γ,κ',δ), on a $\delta \cdot \lfloor\gamma|V\rfloor \xrightarrow{\kappa(\alpha\lceil\gamma|\beta\rceil\cdot\kappa')}_t \delta \cdot \lfloor\gamma|V'\rfloor$.*

Ce lemme est une application élémentaire de la définition de condition confluente. Intuitivement, le premier point signifie que si un terme peut se réduire sous l'abstraction d'un radical R, alors ce terme peut aussi se réduire après la contraction du radical R. De façon similaire, le deuxième point signifie intuitivement que si un terme peut se réduire dans la valeur en argument d'un radical R, alors ce terme peut se réduire aussi après la contraction du radical R. Ce résultat sert crucialement dans la preuve de confluence locale qui est donnée ci-dessous. Il permet de montrer que si une réduction peut avoir lieu à l'intérieur d'un radical, alors cette réduction (éventuellement dupliquée) peut avoir lieu après la contraction du radical externe. En d'autres mots, si les conditions d'un terme sont confluentes, alors la réduction de ce terme vérifie une propriété de confluence locale.

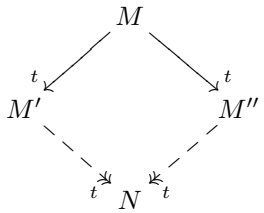
Théorème 4.1 (Confluence locale) *Si $M \xrightarrow{\kappa}_t M'$ et $M \xrightarrow{\kappa}_t M''$ et si les conditions de M sont*



Condition 1 Condition 2

FIG. 4.3 – *Condition confluente*

confluentes, alors il existe un terme N tel que $M' \xrightarrow{\kappa}\!\!\!\!\rightarrow_t N$ et $M'' \xrightarrow{\kappa}\!\!\!\!\rightarrow_t N$.



Preuve : On considère simplement les deux cas cruciaux.

1. Si $M = ((\lambda x.M_1)_c^\beta V)^\alpha$ et $M' = \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1\{x\backslash\lfloor\beta|V\rfloor\}$ où $\gamma = \tau(V)$ avec $(\kappa\alpha,\beta,\gamma) \in c$. Et si $M'' = ((\lambda x.M_1')_c^\beta V)^\alpha$ avec $M_1 \xrightarrow{\kappa\alpha@_1\beta\lambda}_t M_1'$, alors, en utilisant l'hypothèse $(\kappa\alpha,\beta,\gamma) \in c$, on a $M'' \xrightarrow{\kappa}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1'\{x\backslash\lfloor\beta|V\rfloor\}$. Comme les conditions de M_1 sont confluentes, on a, par le lemme 4.3, la réduction $\alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1 \xrightarrow{\kappa}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1'$. De là, en utilisant le lemme 4.2, on obtient $M' = \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1\{x\backslash\lfloor\beta|V\rfloor\} \xrightarrow{\kappa\alpha@_1\beta\lambda}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1'\{x\backslash\lfloor\beta|V\rfloor\}$.
2. Si $M = ((\lambda x.M_1)_c^\beta V)^\alpha$ et $M' = \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1\{x\backslash\lfloor\beta|V\rfloor\}$ où $\gamma = \tau(V)$ avec $(\kappa\alpha,\beta,\gamma) \in c$. Et si $M'' = ((\lambda x.M_1)_c^\beta V')^\alpha$ où $V \xrightarrow{\kappa\gamma@_2}_t V'$, alors, en utilisant l'hypothèse $(\kappa\alpha,\beta,\gamma) \in c$, on a $M'' \xrightarrow{\kappa}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1\{x\backslash\lfloor\beta|V'\rfloor\}$. Trois cas sont à considérer.
   (a) Si x n'a pas d'occurrence dans M_1, le résultat est immédiat.
   (b) Si $M_1 = x^\delta$, comme les conditions de M sont confluentes, alors en utilisant le lemme 4.3, on obtient $M' = \alpha \cdot \lceil\beta|\gamma\rceil \cdot \delta \cdot \lfloor\beta|V\rfloor \xrightarrow{\kappa}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot \delta \cdot \lfloor\beta|V'\rfloor$.
   (c) Sinon, on considère une occurrence de x dans M_1 caractérisée par le contexte $C[\,]$. On a $C[x^\delta] = M_1$. On pose $\kappa' = \sigma(C[\,])$. Comme les conditions de M sont confluentes, on peut utiliser le lemme 4.3. On obtient $\delta \cdot \lfloor\beta|V\rfloor \xrightarrow{\kappa(\alpha\lceil\beta|\gamma\rceil\cdot\kappa')}_t \delta \cdot \lfloor\beta|V'\rfloor$. De là, on obtient $M' = \alpha \cdot \lceil\beta|\gamma\rceil \cdot C\{x\backslash V\}[\delta \cdot \lfloor\beta|V\rfloor] \xrightarrow{\kappa}_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot C\{x\backslash V\}[\lfloor\beta|V'\rfloor]$. En procédant de même pour les autres occurrences de x, on obtient $M' \xrightarrow{\kappa}\!\!\!\!\rightarrow_t \alpha \cdot \lceil\beta|\gamma\rceil \cdot M_1\{x\backslash\lfloor\beta|V'\rfloor\}$. □

4.3 Du λ_t-calcul à l'inspection de pile

Dans cette section, on se rapproche plus encore du langage λ_{sec}-calcul introduit par Fournet et Gordon. On instancie les étiquettes du λ_t-calcul pour retrouver le mécanisme de l'inspection de pile. Les lettres des étiquettes sont définies de la façon suivante.

$$
\begin{array}{rrlll}
a,b,c \in \mathbf{A} & ::= & \mathtt{c}_T & T \subseteq \mathcal{P} & \text{lettre-cadre} \\
& | & \mathtt{g}_T & T \subseteq \mathcal{P} & \text{lettre-privilège} \\
& | & \mathtt{e} & & \text{lettre-vide}
\end{array}
$$

Une lettre peut être une lettre-cadre, une lettre-privilège ou une lettre-vide. Intuitivement, la lettre-cadre $\mathtt{c}_T$ joue le rôle du cadre $T\langle M\rangle$ du λ_{sec}-calcul en changeant les permissions statiques et dynamiques du sous-terme dont l'étiquette contient $\mathtt{c}_T$. De même, la lettre-privilège $\mathtt{g}_T$ joue le rôle d'un terme privilégié `grant` T `in` M en ajoutant les permissions de T aux permissions dynamiques. La lettre-vide n'a pas d'influence sur les permissions statiques et dynamiques. On l'utilisera pour les sous-termes qui ne modifient pas les permissions statiques et dynamiques. Pour mettre en œuvre ces interprétations des étiquettes, on utilise la fonction p qui permet d'obtenir les permissions statiques et dynamiques associées à un chemin-contexte κ ou un chemin φ issu de la racine d'un terme. Cette fonction est définie sur la figure 4.4. Pour déterminer les permissions statiques et dynamiques, la définition de la fonction p s'inspire du mécanisme d'inspection de pile. La fonction p retourne un couple (S,D) de sous-ensembles de $\mathcal{P}$ constitués, intuitivement, des permissions statiques S et dynamiques D. Pour définir ces ensembles, on procède comme pour l'inspection de

$$
\begin{array}{rcl@{\qquad}rcl}
p(\bot) & = & (\mathcal{P},\mathcal{P}) & p(\kappa\alpha) & = & p_{\mathbf{E}}(\alpha,p(\kappa)) \\
p(\varphi\theta) & = & p(\varphi) & & & \\
p_{\mathbf{E}}(\mathsf{c}_T,(S,D)) & = & (T,T\cap D) & p_{\mathbf{E}}(\mathsf{g}_T,(S,D)) & = & (S,(D\cup T)\cap S) \\
p_{\mathbf{E}}(\mathsf{e},(S,D)) & = & (S,D) & p_{\mathbf{E}}(\alpha\beta,(S,D)) & = & p_{\mathbf{E}}(\beta,p_{\mathbf{E}}(\beta,(S,D))) \\
p_{\mathbf{E}}(\lceil\alpha|\beta\rceil,(S,D)) & = & p_{\mathbf{E}}(\alpha,(S,D)) & p_{\mathbf{E}}(\lfloor\alpha|\beta\rfloor,(S,D)) & = & p_{\mathbf{E}}\beta,(S,D)
\end{array}
$$

FIG. 4.4 – *Définition de la fonction* $p_{\mathbf{E}}$

pile en examinant, en commençant par la racine, les étiquettes successives qui constituent le chemin en argument. Pour cela, on utilise la fonction $p_{\mathbf{E}}$ qui prend en argument l'étiquette α à examiner et le couple des permissions statiques et dynamiques courantes. Si α est une lettre-cadre c_T, on effectue les mêmes opérations que pour le cadre du λ_{sec}-calcul : les permissions statiques deviennent T et les permissions dynamiques sont intersectées avec T. De même, si α est une lettre-privilège, les permissions statiques sont inchangées alors que les permissions dynamiques sont augmentées de T dans la limite de ses permissions statiques. Si α est une lettre-vide, alors les permissions statiques et dynamiques sont laissées inchangées. Si α est une concaténation, on examine successivement les deux sous-étiquettes. Si α est un surlignement, on examine seulement l'étiquette de gauche qui correspond à l'abstraction réduite. En effet, on souhaite rester proche du λ_{sec}-calcul. Dans ce calcul, un éventuel cadre en argument n'intervient pas en tête du contractum. Pour les mêmes raisons, seule l'étiquette de droite d'un souligné est pris en compte.

Dans le mécanisme d'inspection de λ_{sec}-calcul, les tests concernent uniquement les permissions dynamiques. Les permissions statiques ne sont pas directement utilisées, bien que dans les implémentations de JVM et CLR, elles peuvent être testées. Par conséquent, les tests que nous utilisons dans cette section ne feront intervenir que les permissions dynamiques déterminées à partir du chemin courant avec la fonction p. Les informations locales, c'est-à-dire les étiquettes de l'abstraction ou de la valeur en argument n'interviennent pas. Les conditions utilisées dans cette section sont notées χ_T pour $T \subseteq \mathcal{P}$. Elles sont formellement définies de la façon suivante.

$$\chi_T = (\{\varphi / p(\varphi) = (S,D) \text{ et } T \subseteq D\},\mathbf{E},\mathbf{E})$$

On a $(\varphi,\alpha,\beta) \in \chi_T$ si et seulement si T est inclus dans les permissions dynamiques retournées par $p(\varphi)$, ce qu'on note $T \subseteq \varphi$. Les étiquettes α et β n'interviennent pas dans le test. Il est clair que les conditions χ_T utilisées ici ne sont pas confluentes. Cela n'est guère surprenant dans le mesure où l'ordre d'évaluation est fixé dans le λ_{sec}-calcul. Pour pouvoir effectuer une comparaison avec le λ_{sec}-calcul, on modifie les règles de contexte pour fixer l'ordre d'évaluation. On obtient le λ_{ts}-calcul dont les règles de réduction sont définies ci-dessous.

$$
\begin{array}{ll}
(\beta_{ts}) & \dfrac{T \subseteq \varphi \qquad \tau(V) = \beta}{((\lambda x.M)^{\alpha}_{\chi_T} V)^{\gamma} \xrightarrow{\kappa}_{ts} \gamma \cdot \lceil\alpha|\beta\rceil \cdot M\{x \backslash \lfloor\alpha|V\rfloor\}} \\
(\mathtt{f}_{ts}) & \dfrac{T \not\subseteq \varphi}{((\lambda x.M)^{\alpha}_{\chi_T} V)^{\gamma} \xrightarrow{\kappa}_{ts} \mathtt{fail}} \\
(\nu_{ts}) & \dfrac{M \xrightarrow{\kappa\beta@_1}_t M'}{(M\ N)^{\beta} \xrightarrow{\kappa}_{ts} (M'\ N)^{\beta}} \\
(\mu_{ts}) & \dfrac{N \xrightarrow{\kappa\beta@_2}_t N'}{(V\ N)^{\beta} \xrightarrow{\kappa}_{ts} (V\ N')^{\beta}} \\
(\mathtt{f}^L_{ts}) & (\mathtt{fail}\ M)^{\beta} \xrightarrow{\kappa}_{ts} \mathtt{fail} \\
(\mathtt{f}^R_{ts}) & (V\ \mathtt{fail})^{\beta} \xrightarrow{\kappa}_{ts} \mathtt{fail}
\end{array}
$$

La relation $\xrightarrow{\kappa}_{ts}$ est un sous-ensemble de la relation $\xrightarrow{\kappa}_{t}$. Les règles (β_t) et $(\mathtt{f}_s)$ sont réécrites en (β_{ts}) et $(\mathtt{f}_{ts})$ pour tenir compte des conditions particulières utilisées dans cette section. Les règles (ν_{ts}), (μ_{ts}), $(\mathtt{f}^L_{ts})$ et $(\mathtt{f}^R_{ts})$ permettent d'imposer l'ordre d'évaluation des applications de gauche à droite, comme dans le λ_{sec}-calcul. La règle (ξ_t) a disparu puisque, dans le λ_{sec}-calcul, on ne réduit pas sous les abstractions.

Comparativement au λ_{sec}-calcul, il subsiste encore des différences majeures puisque les règles (Red Frame$_{\mathsf{sec}}$) et (Red Grant$_{\mathsf{sec}}$) de ce dernier n'ont pas d'équivalent dans le λ_{ts}-calcul. Dans ce dernier, les lettres présentes dans l'étiquette d'un terme ne peuvent disparaître que si ce terme disparaît, par exemple si ce terme est l'argument d'une abstraction qui ne contient pas d'occurrence de sa variable liée. L'exemple mentionné dans la section 4.1 soulignait les difficultés engendrées par ces *oublis* de principaux. En réalité, le λ_{ts}-calcul correspond à une variante de l'inspection de pile qui avait été suggérée par Fournet et Gordon comme une amélioration possible de l'inspection de pile : *l'inspection de pile avec valeurs encadrées*. Cette variante n'oublie pas les principaux puisqu'elle n'utilise plus les règles (Red Frame$_{\mathsf{sec}}$) et (Red Grant$_{\mathsf{sec}}$). Dans [15], cette variante ne comporte plus de termes privilégiés, pour simplifier les preuves. Dans la syntaxe que nous mentionnons ici, nous ajoutons les termes privilégiés, ce qui constitue, en fait, une extension élémentaire. Pour simplifier la comparaison avec λ_{ts}-calcul, nous ne considérons que des tests de la forme `demand` T `then` M `else fail`, ce que nous notons `check` T `for` M. La syntaxe du λ_{sec}-calcul avec valeurs encadrées (ou λ_{secW}-calcul) est définie de la façon suivante.

$$
\begin{array}{rcll}
M,N \in \mathbf{\Lambda}^W_{\mathsf{sec}} & ::= & x & \text{Variable} \\
& | & \lambda x.M & \text{Abstraction} \\
& | & MN & \text{Application} \\
& | & T\langle M\rangle & \text{Cadre} \\
& | & \mathtt{grant}\ T\ \mathtt{in}\ M & \text{Terme privilégié} \\
& | & \mathtt{check}\ T\ \mathtt{for}\ M & \text{Test} \\
& | & \mathtt{fail} & \text{Echec}
\end{array}
$$

On retrouve la syntaxe présentée précédemment avec un nouveau test qui revient à retourner un échec `fail` lorsque le test échoue. La principale nouveauté dans cette variante du λ_{sec}-calcul est constituée par les valeurs encadrées.

$$
\begin{array}{lrcl}
\text{Valeur encadrée} & W \in \mathbf{W} & ::= & \lambda x.M \mid T\langle W\rangle \mid \mathtt{grant}\ T\ \mathtt{in}\ W \\
\text{Contexte encadré} & C_w[\,] & ::= & [\,] \mid T\langle C_w[\,]\rangle \mid \mathtt{grant}\ T\ \mathtt{in}\ C_w[\,] \\
\text{Sortie} & O & ::= & W \mid \mathtt{fail}
\end{array}
$$

Une valeur encadrée peut être une abstraction, une valeur encadrée W dans un cadre ou une valeur encadrée dans un terme privilégié. Intuitivement, les valeurs encadrées permettent de conserver l'origine des valeurs, alors que dans λ_{sec}-calcul, les principaux d'une valeur sont oubliés. Un contexte encadré est un contexte constitué de cadres ou de termes privilégiés. Pour toute valeur encadrée W, il existe un unique contexte encadré $C_w[\,]$ et une unique abstraction $\lambda x.M$ tels que $W = C_w[\lambda x.M]$. Une sortie est une valeur encadrée ou un échec.

Les règles de réduction, mentionnées sur la figure 4.5, sont largement modifiées. On observe en particulier que les valeurs étendues permettent de conserver les cadres autour des valeurs. Ainsi, contrairement à ce qui se passe dans le λ_{sec}-calcul, les permissions accordées aux valeurs sont conservées. La réduction (Appl$_{\mathsf{secW}}$) est reprise du λ_{sec}-calcul, à ceci près qu'on attend, en argument, une valeur encadrée W au lieu d'une valeur V. De même, on réduit à droite d'une application par la règle (App Rand$_{\mathsf{secW}}$) dès qu'un valeur encadrée est atteinte à gauche. Les réductions (Frame Rator$_{\mathsf{secW}}$) et (Grant Rator$_{\mathsf{secW}}$) sont les principales nouveautés. Alors que dans

$$(\text{Appl}_{\mathsf{secW}}) \qquad (\lambda x.M)W \to^{S,D}_{\mathsf{secW}} M\{x\backslash W\}$$

$$(\text{App Rator}_{\mathsf{secW}}) \qquad \frac{M_1 \to^{S,D}_{\mathsf{secW}} M_1'}{M_1 M_2 \to^{S,D}_{\mathsf{secW}} M_1' M_2}$$

$$(\text{App Rand}_{\mathsf{secW}}) \qquad \frac{M \to^{S,D}_{\mathsf{secW}} M'}{W M \to^{S,D}_{\mathsf{secW}} W M'}$$

$$(\text{Frame Rator}_{\mathsf{secW}}) \qquad T\langle W_1\rangle\, W_2 \to^{S,D}_{\mathsf{secW}} T\langle W_1\, W_2\rangle$$

$$(\text{Grant Rator}_{\mathsf{secW}}) \qquad (\texttt{grant}\ T\ \texttt{in}\ W_1)W_2 \to^{S,D}_{\mathsf{secW}} \texttt{grant}\ T\ \texttt{in}\ (W_1\, W_2)$$

$$(\text{Check}_{\mathsf{secW}}) \qquad \frac{T \subseteq D}{\texttt{check}\ T\ \texttt{for}\ M \to^{S,D}_{\mathsf{secW}} M}$$

$$(\text{Check Fail}_{\mathsf{secW}}) \qquad \frac{T \not\subseteq D}{\texttt{check}\ T\ \texttt{for}\ M \to^{S,D}_{\mathsf{secW}} \texttt{fail}}$$

$$(\text{Ctx Frame}_{\mathsf{secW}}) \qquad \frac{M \to^{T,T\cap D}_{\mathsf{secW}} M'}{T\langle M\rangle \to^{S,D}_{\mathsf{secW}} T\langle M'\rangle}$$

$$(\text{Ctx Grant}_{\mathsf{secW}}) \qquad \frac{M \to^{S,S\cap(D\cup T)}_{\mathsf{secW}} M'}{\texttt{grant}\ T\ \texttt{in}\ M \to^{S,D}_{\mathsf{secW}} \texttt{grant}\ T\ \texttt{in}\ M'}$$

$$(\text{Fail Rator}_{\mathsf{secW}}) \qquad \texttt{fail}\, M \to^{S,D}_{\mathsf{secW}} \texttt{fail}$$

$$(\text{Fail Rand}_{\mathsf{secW}}) \qquad W\, \texttt{fail} \to^{S,D}_{\mathsf{secW}} \texttt{fail}$$

$$(\text{Fail Frame}_{\mathsf{secW}}) \qquad T\langle\texttt{fail}\rangle \to^{S,D}_{\mathsf{secW}} \texttt{fail}$$

$$(\text{Fail Grant}_{\mathsf{secW}}) \qquad \texttt{grant}\ T\ \texttt{in}\ \texttt{fail} \to^{S,D}_{\mathsf{secW}} \texttt{fail}$$

FIG. 4.5 – *Règles de réduction du* λ_{secW}*-calcul*

le λ_t-calcul, les cadres et les privilèges autour des valeurs sont oubliés, ces derniers sont conservés dans le λ_{ts}-calcul. Pour pouvoir effectuer la réduction d'une radical par ($\text{Appl}_{\mathsf{secW}}$), les cadres ou les privilèges autour de l'abstraction à gauche sont *levés*, à la manière de ce qu'avaient proposé Abadi, Lampson et Lévy dans [3] : les règles (Frame $\text{Rator}_{\mathsf{secW}}$) et (Grant $\text{Rator}_{\mathsf{secW}}$) font passer les cadres et les termes privilégiés du membre gauche de l'application vers le haut de celle-ci. Du fait de l'utilisation du terme $\texttt{check}\, T\, \texttt{for}\, M$, les règles ($\text{Check}_{\mathsf{secW}}$) et (Check $\text{Fail}_{\mathsf{secW}}$) remplacent la règle (Demand) du λ_t-calcul. Comme pour cette dernière, si les permissions de T sont incluses dans les permissions dynamiques, alors le test est un succès et la réduction se poursuit avec M. Dans le cas contraire, on obtient le terme d'échec $\texttt{fail}$. Les autres règles sont des adaptations des règles du λ_t-calcul à la nouvelle notion de valeur.

Nous comparons maintenant le λ_{secW}-calcul avec le λ_{ts}-calcul. Pour cela, nous proposons une traduction d'un terme du λ_{secW}-calcul dans le λ_{ts}-calcul. La traduction du terme M, qui est notée $(\!|M|\!)$ est définie de la façon suivante.

$$\begin{aligned}
(\!|x|\!) &= x^{\mathtt{e}} \\
(\!|\lambda x.M|\!) &= (\lambda x.(\!|M|\!))^{\mathtt{e}}_{\chi_\emptyset} \\
(\!|MN|\!) &= ((\!|M|\!)\,(\!|N|\!))^{\mathtt{e}} \\
(\!|T\langle M\rangle|\!) &= \mathtt{c}_T.(\!|M|\!) \\
(\!|\texttt{grant}\, T\, \texttt{in}\, M|\!) &= \mathtt{g}_T.(\!|M|\!) \\
(\!|\texttt{check}\, T\, \texttt{for}\, M|\!) &= ((\lambda u.(\!|M|\!))^{\mathtt{e}}_{\chi_T}(\lambda x.x^{\mathtt{e}})^{\mathtt{e}}_{\chi_\emptyset})^{\mathtt{e}} \quad \text{où } u \notin \mathtt{FV}(M) \\
(\!|\texttt{fail}|\!) &= \texttt{fail}
\end{aligned}$$

Les termes qui ne modifient pas les permissions statiques et dynamiques de leurs sous-termes sont étiquetés par la lettre-vide $\mathtt{e}$. Ainsi, les traductions des variables et des applications sont étiquetées par $\mathtt{e}$. La traduction d'une abstraction reçoit aussi l'étiquette $\mathtt{e}$ et est testé par $\chi_\emptyset$. Ce test, qui est toujours vérifié, rend bien compte du fait que cette abstraction pourra toujours être contractée. La traduction du cadre $T\langle M\rangle$, dont l'effet est de modifier les permissions statiques et dynamiques pour M, consiste à concaténer l'étiquette $\mathtt{c}_T$ à la traduction de M. De ce fait, les termes de $(\!|M|\!)$ sont réduits avec pour chemin, un chemin φ contenant $\mathtt{c}_T$. Les permissions $p(\varphi)$ associées à ce chemin sont donc modifiées de la même manière que dans le λ_{sec}-calcul. De même, la traduction de $\texttt{grant}\, T\, \texttt{in}\, M$ concatène $\mathtt{g}_T$ avec $(\!|M|\!)$ ce qui modifie en conséquence les permissions $p(\varphi)$ des chemins menant aux sous-termes de M. La traduction du test $\texttt{check}\, T\, \texttt{for}\, M$ introduit un (β_{ts})-radical dont l'abstraction porte la condition χ_T qui correspond bien à l'ensemble de permissions T testé dans le λ_{secW}-calcul. La variable liée par ce radical est choisie pour ne pas être présente dans $(\!|M|\!)$. La valeur $(\lambda x.x^{\mathtt{e}})^{\mathtt{e}}_{\chi_\emptyset}$ placée en argument est inutilisée et a pour seul but de permettre la contraction du radical. L'échec du λ_{secW}-calcul se traduit en un échec du λ_{ts}-calcul.

Le système d'étiquette du λ_{ts}-calcul, qui contient notamment des soulignés et des surlignés, est plus détaillé que celui du λ_{secW}-calcul. De plus, si l'absence de modification des permissions statiques et dynamiques est implicite dans λ_{secW}-calcul, elle est en revanche explicite, via l'étiquette $\mathtt{e}$ dans le λ_{ts}-calcul. Pour ces raisons, on introduit une relation d'équivalence modulo $\mathtt{e}$, notée $\simeq$, sur les étiquettes et les termes du λ_{ts}-calcul pour pouvoir se rapprocher du λ_{secW}-calcul.

$$\alpha \simeq \alpha \qquad \frac{\alpha \simeq \beta \qquad \beta \simeq \gamma}{\alpha \simeq \gamma} \qquad \frac{\alpha \simeq \beta}{\beta \simeq \alpha}$$

$$\lceil \alpha|\beta \rceil \simeq \alpha \qquad \frac{\alpha_1 \simeq \alpha_2 \qquad \beta_1 \simeq \beta_2}{\alpha_1\beta_1 \simeq \alpha_2\beta_2} \qquad \lfloor \alpha|\beta \rfloor \simeq \beta$$

$$\mathtt{e}\alpha \simeq \alpha\mathtt{e} \simeq \alpha$$

La relation $\simeq$ sur les étiquettes est réflexive, transitive, symétrique. Les lettres-vides sont ignorées par cette relation. Seule la première étiquette d'une étiquette surlignée $\lceil \alpha|\beta \rceil$ est prise en compte.

En effet, dans le λ_{secW}-calcul, les permissions des arguments n'interviennent pas. Pour des raisons symétriques, seule la deuxième étiquette d'une étiquette soulignée est prise en compte. La relation $\simeq$ est compatible, à gauche et à droite, avec l'opération de concaténation.

$$\frac{\alpha \simeq \beta}{x^\alpha \simeq x^\beta} \qquad \mathtt{fail} \simeq \mathtt{fail}$$

$$\frac{\alpha \simeq \beta \quad M \simeq M'}{(\lambda x.M)^\alpha_{\chi_T} \simeq (\lambda x.M')^\beta_{\chi_T}} \qquad \frac{\alpha \simeq \beta \quad M \simeq M' \quad N \simeq N'}{(MN)^\alpha \simeq (M'N')^\beta}$$

Deux termes sont $\simeq$-équivalents si leurs étiquettes sont $\simeq$-équivalentes et si leurs sous-termes sont également $\simeq$-équivalents. Cette relation d'équivalence vérifie les propriétés élémentaires suivantes.

Lemme 4.4 *1. Si $M \simeq M'$ et $\alpha \simeq \beta$, alors $\alpha \cdot M \simeq \beta \cdot M'$.*
2. Si $M \simeq M'$ et $\alpha \simeq \beta$, alors $\lceil \alpha | \gamma \rceil \cdot M \simeq \lceil \beta | \delta \rceil \cdot M'$.
3. Si $V \simeq V'$, alors $\lfloor \alpha | V \rfloor \simeq V'$.
4. Si $M \simeq M'$ et $V \simeq V'$, alors $M\{x\backslash V\} \simeq M'\{x\backslash V'\}$.

Preuve : Les trois premiers points sont élémentaires. On montre le troisième point par induction sur la structure de M. On pose $M_1 = M\{x\backslash V\}$ et $M_2 = M'\{x\backslash V'\}$.

1. Si $M = x^\alpha$, alors $M' = x^\beta$ où $\alpha \simeq \beta$. De là, on obtient, en utilisant le premier point $M_1 = \alpha \cdot V \simeq \beta \cdot V' = M_2$.
2. Si $M = (\lambda y.N)^\alpha_{\chi_T}$, alors $M' = (\lambda y.N')^\beta_{\chi_T}$ avec $\alpha \simeq \beta$ et $N \simeq N'$. On peut supposer, sans perte de généralité $y \notin \mathtt{FV}(V)$. De là, on a $y \notin \mathtt{FV}(V')$ et $M_1 = (\lambda y.N\{x\backslash V\})^\alpha_{\chi_T}$ et $M_2 = (\lambda y.N'\{x\backslash V'\})^\beta_{\chi_T}$. Par hypothèse d'induction, on a $N\{x\backslash V\} \simeq N'\{x\backslash V'\}$ ce qui implique bien $M_1 \simeq M_2$.
3. Le cas $M = (NP)^\alpha$ est similaire au cas précédent. Les cas $M = y^\alpha$ où $y \neq x$ et $M = \mathtt{fail}$ sont élémentaires. □

Le premier point énonce la compatibilité, à gauche et à droite, de la relation $\simeq$ avec la fonction de concaténation "$\cdot$". Les deux points suivants montrent que les étiquettes à droite d'un surligné et à gauche d'un souligné n'interviennent pas dans la relation $\simeq$. Le dernier point est la compatibilité de $\simeq$ avec la substitution. Après avoir examiné les propriétés élémentaires de $\simeq$, on s'intéresse maintenant à la réduction $\xrightarrow{\kappa}_{ts}$.

Lemme 4.5 *Si $M \xrightarrow{\kappa}_{ts} M'$ et $p(\kappa) = p(\kappa')$, alors $M \xrightarrow{\kappa'}_{ts} M'$.*

Ce résultat prouve que le contexte n'intervient dans la réduction d'un terme qu'au travers de l'interprétation par p du chemin-contexte menant à la racine. En d'autres termes, seules les permissions issues du contexte interviennent dans la réduction d'un terme. La démonstration élémentaire de ce lemme s'appuie sur le fait que, dans le λ_{ts}-calcul, les chemins-contextes qui annotent la réduction n'interviennent que dans les tests où ils sont utilisés au travers de la fonction p. On examine maintenant les relations entre l'équivalence modulo $\mathtt{e}$ et la réduction $\xrightarrow{\kappa}_{ts}$.

Lemme 4.6 *Si $M \simeq N$, $p(\kappa_1) = p(\kappa_2)$ et $M \xrightarrow{\kappa_1}_{ts} M'$, alors il existe un terme N' tel que $N \xrightarrow{\kappa_2}_{ts} N'$ et $N \simeq N'$.*

Preuve : On procède par induction sur la réduction de M.

1. Si $M = ((\lambda x.P)^\alpha_{\chi_T} V)^\beta$ et $p(\kappa_1\beta) = (S,D)$, alors $N = ((\lambda x.Q)^\gamma_{\chi_T} V')^\delta$ avec $P \simeq Q$, $V \simeq V'$, $\alpha \simeq \gamma$ et $\beta \simeq \delta$. De là, on obtient $p(\kappa_2\delta) = (S,D)$.
   (a) Si on a $T \subseteq D$ et donc $M \xrightarrow{\kappa_1}_{ts} \beta \cdot \lceil \alpha | \tau(V) \rceil \cdot P\{x\backslash\lfloor \alpha | V \rfloor\}$, alors on obtient la réduction $N \xrightarrow{\kappa_2}_{ts} \delta \cdot \lceil \gamma | \tau(V') \rceil . Q\{x\backslash\lfloor \gamma | V' \rfloor\}$. L'utilisation du lemme 4.4 permet de conclure.
   (b) Si on a $T \not\subseteq D$ et $M \xrightarrow{\kappa_1}_{ts} \mathtt{fail}$, alors on a $N \xrightarrow{\kappa_2}_{ts} \mathtt{fail}$.
2. Si $M = (M_1M_2)^\alpha$ avec $M_1 \xrightarrow{\kappa_1\alpha@_2}_{ts} M_1'$ et $M' = (M_1'M_2)^\alpha$, alors $N = (N_1N_2)^\beta$ où $N_1 \simeq M_1$, $N_2 \simeq M_2$ et $\alpha \simeq \beta$. Ceci implique $p(\kappa_1\alpha@_2) = p(\kappa_2\beta@_2)$. L'application de l'hypothèse d'induction permet de conclure.

3. Le cas $M = (VM)^\alpha \xrightarrow{\kappa_1}_{ts} (VM')^\alpha$ est similaire au cas précédent.
4. Les cas $M = (\texttt{fail}\ P)^\alpha \xrightarrow{\kappa_1}_{ts} \texttt{fail}$ et $M = (V\ \texttt{fail})^\alpha \xrightarrow{\kappa_1}_{ts} \texttt{fail}$ sont élémentaires. □

Ce résultat, qui est illustré sur la figure 4.6, énonce le fait que des termes $\simeq$-équivalents placés dans des contextes qui donnent les mêmes permissions, se réduisent vers des termes $\simeq$-équivalents. On examine maintenant de quelle façon la réduction $\xrightarrow{\kappa}_{ts}$ interagit avec la concaténation "$\cdot$".

Lemme 4.7 *Si $M \xrightarrow{\kappa}_{ts} M'$ et si $p(\kappa\tau(M)) = p(\kappa'\tau(\alpha \cdot M))$, alors $\alpha \cdot M \xrightarrow{\kappa'}_{ts} \alpha \cdot M'$.*

Preuve : On montre cette propriété par induction sur la réduction de M à M'.

1. Si $M = ((\lambda x.N)^\beta_{\chi_T} V)^\gamma \xrightarrow{\kappa}_{ts} \gamma \cdot \lceil\beta|\tau(V)\rceil \cdot N\{x\backslash\lfloor\beta|V\rfloor\}$ avec $T \subseteq p(\kappa\gamma)$, alors comme par hypothèse on a $T \subseteq p(\kappa'\alpha\gamma)$, on obtient directement $\alpha\cdot M \xrightarrow{\kappa'}_{ts} \alpha\cdot\gamma\cdot\lceil\beta|\tau(V)\rceil\cdot N\{x\backslash\lfloor\beta|V\rfloor\}$.
2. Si $M = (M_1M_2)^\beta \xrightarrow{\kappa}_{ts} (M_1'M_2)^\beta$ avec $M_1 \xrightarrow{\kappa\beta@_1}_{ts} M_1'$, alors comme $p(\kappa\beta@_1) = p(\kappa'\alpha\beta@_1)$, en utilisant le lemme 4.5, on obtient $M_1 \xrightarrow{\kappa'\alpha\beta@_1}_{ts} M_1'$, ce qui implique $\alpha \cdot M \xrightarrow{\kappa'}_{ts} \alpha \cdot M'$.
3. Les autres cas sont similaires aux cas précédents. □

Si $\kappa\tau(M)$ et $\kappa'\alpha\tau(M)$ sont des chemins dont les permissions associées, $p(\kappa\tau(M))$ et $p(\kappa'\alpha\tau(M))$ sont égales, et si M se réduit par $\xrightarrow{\kappa}_{ts}$ vers M', alors $\alpha \cdot M$ se réduit par $\xrightarrow{\kappa'}_{ts}$ vers $\alpha \cdot M'$. Intuitivement, ce résultat permet de "casser" l'étiquette de tête d'un terme pour la placer en concaténation devant ce terme. Ceci se révélera utile au moment de considérer la traduction $\mathsf{c}_T.(\!|M|\!)$ d'un cadre $T\langle M\rangle$ dans la preuve de la correction de la traduction. On considère maintenant le résultat suivant qui occupe une place centrale pour le cas de la (β_{ts})-réduction dans la preuve de correction de la traduction.

Lemme 4.8 $(\!|M|\!)\{x\backslash(\!|W|\!)\} \simeq (\!|M\{x\backslash W\}|\!)$

Preuve : On pose $M_1 = (\!|M|\!)\{x\backslash(\!|W|\!)\}$ et $M_2 = (\!|M\{x\backslash W\}|\!)$. On procède par induction sur M.

1. Si $M = x$, alors on a $(\!|M|\!) = x^{\mathsf{e}}$. On obtient donc $M_1 = \mathsf{e}.(\!|W|\!)$ et $M_2 = (\!|W|\!)$. On obtient donc bien $M_1 \simeq M_2$.
2. Si $M = T\langle N\rangle$, on a $M_2 = (\!|T\langle N\{x\backslash W\}\rangle|\!) = \mathsf{c}_T.(\!|N\{x\backslash W\}|\!)$ et $M_1 = \mathsf{c}_T.(\!|N|\!)\{x\backslash(\!|W|\!)\}$. L'hypothèse d'induction permet de conclure.
3. Les cas $M =\texttt{grant}\ T\ \texttt{in}\ N$, $M =\texttt{check}\ T\ \texttt{for}\ N$, $M = N_1N_2$ et $M = \lambda y.N$ sont similaires au cas précédent. Les cas $M = y$ où $y \neq x$ et $M = \texttt{fail}$ sont élémentaires. □

La traduction commute avec la substitution modulo $\simeq$. L'utilisation de l'équivalence modulo e est fondamentale du fait de la présence de l'étiquette e sur les variables traduites. Ce résultat interviendra plus tard de façon cruciale dans le théorème 4.2 pour montrer qu'une (β_{ts})-réduction dans le λ_{ts}-calcul correspond exactement à une (β_{secW})-réduction du λ_{secW}-calcul. Le lemme 4.8 interviendra dans cette preuve sous la forme du corollaire suivant où il est combiné avec le lemme 4.4.

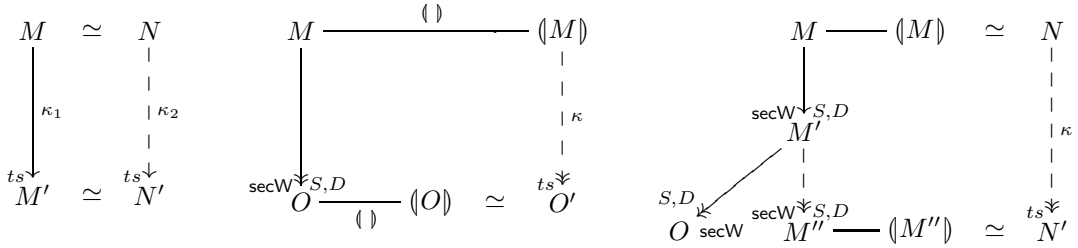
Corollaire 4.1 *Si $N \simeq (\!|M|\!)$ et $V \simeq (\!|W|\!)$, alors $N\{x\backslash V\} \simeq (\!|M\{x\backslash W\}|\!)$.*

Ce corollaire permet d'obtenir le résultat central de ce chapitre, à savoir la correction de la traduction $(\!|\ |\!)$.

Théorème 4.2 (Correction) *Si $M \twoheadrightarrow^{S,D}_{\mathsf{secW}} O$ et $p(\kappa) = (S,D)$, alors $(\!|M|\!) \xrightarrow{\kappa}\!\!\!\!\twoheadrightarrow_{ts} O'$ où $O' \simeq (\!|O|\!)$.*

Si un terme du λ_{secW}-calcul se réduit vers une sortie O, alors la traduction de ce terme dans le λ_{ts}-calcul se réduit également vers une sortie qui est $\simeq$-équivalente à la traduction de O. Ce résultat est illustré sur la figure 4.6. Le théorème établit une correspondance entre la réduction dans le λ_{secW}-calcul et le λ_{ts}-calcul. De façon plus générale, ce résultat fait le lien entre d'une part le λ_{secW}-calcul, qui est un langage issu de l'inspection de pile, c'est-à-dire d'un mécanisme de sécurité empirique, et d'autre part le λ_{ts}-calcul qui trouve ses sources dans un langage théorique, le λ-calcul étiqueté. Pour prouver ce résultat, on montre la propriété plus générale suivante, qui est illustrée sur la figure 4.6

Lemme 4.9 *Si $M \rightarrow^{S,D}_{\mathsf{secW}} M' \twoheadrightarrow^{S,D}_{\mathsf{secW}} O$ et si $N \simeq (\!|M|\!)$ et $p(\kappa) = (S,D)$, alors il existe deux termes M'' et N' tels que $M' \twoheadrightarrow^{S,D}_{\mathsf{secW}} M''$ et $N \xrightarrow{\kappa}\!\!\!\!\twoheadrightarrow_{ts} N'$ avec $N' \simeq (\!|M''|\!)$.*



Lemme 4.6 Théorème 4.2 Lemme 4.9

FIG. 4.6 – *Résultats de correction de* $(\!|\ |\!)$

Preuve : On montre cette propriété par induction sur la réduction du terme.

1. Si $M = (\lambda x.P)W \to^{S,D}_{\mathsf{secW}} P\{x\backslash W\} = M'$, alors on a $(\!|M|\!) = ((\lambda x.(\!|P|\!))^{\mathtt{e}}_{\chi_\emptyset}(\!|W|\!))^{\mathtt{e}}$. Ceci implique $N = ((\lambda x.Q)^{\alpha}_{\chi_\emptyset}V)^{\beta}$ où on a les relations $Q \simeq (\!|P|\!)$, $V \simeq (\!|W|\!)$, $\alpha \simeq \mathtt{e}$ et $\beta \simeq \mathtt{e}$. On obtient $N \xrightarrow{\kappa}_{ts} \beta\cdot\lceil\alpha|\tau(V)\rceil\cdot Q\{x\backslash\lfloor\alpha|V\rfloor\}$. Avec le lemme 4.4, on obtient $\beta\cdot\lceil\alpha|\tau(V)\rceil\cdot Q \simeq Q \simeq (\!|P|\!)$ et $\lfloor\alpha|V\rfloor \simeq V \simeq (\!|W|\!)$. Le corollaire 4.1 permet d'obtenir $\beta\cdot\lceil\alpha|\tau(V)\rceil\cdot Q\{x\backslash\lfloor\alpha|V\rfloor\} \simeq (\!|P\{x\backslash W\}|\!)$.
2. Si $M = T\langle W_1\rangle\ W_2 \to^{S,D}_{\mathsf{secW}} T\langle W_1W_2\rangle = M' \twoheadrightarrow^{S,D}_{\mathsf{secW}} O$, on a $W_1 = C_w[\lambda y.P]$. En appliquant successivement les règles (Frame $\mathrm{Rator}_{\mathsf{secW}}$) et (Grant $\mathrm{Rator}_{\mathsf{secW}}$), on obtient la réduction $M' \twoheadrightarrow^{S,D}_{\mathsf{secW}} T\langle C_w[(\lambda y.P)W_2]\rangle \to^{S,D}_{\mathsf{secW}} T\langle C_w[P\{y\backslash W_2\}]\rangle = T\langle C_w[P]\rangle\{y\backslash W_2\} = M''$. On définit récursivement l'étiquette α_{C_w} de la façon suivante.
   (a) Si $C_w[\,] = [\,]$ alors $\alpha_{C_w} = \mathtt{e}$.
   (b) Si $C_w[\,] = \mathtt{grant}\,T'\,\mathtt{in}\,C'_w[\,]$ alors $\alpha_{C_w} = \mathtt{g}_{T'}\alpha_{C'_w}$.
   (c) Si $C_w[\,] = T'\langle C'_w[\,]\rangle$ alors $\alpha_{C_w} = \mathtt{c}_{T'}\alpha_{C'_w}$.

   Dans ces conditions, on a $(\!|M''|\!) = (\!|T\langle C_w[P]\rangle\{y\backslash W_2\}|\!)$. Par ailleurs, la définition de la traduction donne $(\!|M|\!) = (\mathtt{c}_T\cdot\alpha_{C_w}\cdot(\lambda y.(\!|P|\!))^{\mathtt{e}}_{\chi_\emptyset}(\!|W|\!))^{\mathtt{e}}$ et donc $N = ((\lambda y.Q)^{\alpha}_{\chi_\emptyset}V)^{\beta}$ où $Q \simeq (\!|P|\!)$, $V \simeq (\!|W|\!)$, $\alpha \simeq \mathtt{c}_T\alpha_{C_w}$ et $\beta \simeq \mathtt{e}$. De là, on a $N \xrightarrow{\kappa}_{ts} \beta\cdot\lceil\alpha|\tau(V)\rceil\cdot Q\{y\backslash\lfloor\alpha|V\rfloor\} = N'$. En utilisant le lemme 4.4, on obtient d'une part $\beta\cdot\lceil\alpha|\tau(V)\rceil\cdot Q \simeq \mathtt{c}_T\cdot\alpha_{C_w}\cdot Q \simeq (\!|T\langle C_w[P]\rangle|\!)$ et d'autre part $\lfloor\alpha|V\rfloor \simeq V \simeq (\!|W|\!)$. De là, en utilisant le corollaire 4.1, on obtient $N' \simeq (\!|M''|\!)$.
3. Le cas $M = (\mathtt{grant}\ T\ \mathtt{in}\ W_1)W_2 \to^{S,D}_{\mathsf{secW}} \mathtt{grant}\,T\,\mathtt{in}\,(W_1W_2)$ est similaire au cas précédent.
4. Si $M = \mathtt{check}\,T\,\mathtt{for}\,P$, alors on a $(\!|M|\!) = ((\lambda u.(\!|P|\!))^{\mathtt{e}}_{\chi_T}(\lambda z.z^{\mathtt{e}})^{\mathtt{e}}_{\chi_\emptyset})^{\mathtt{e}}$. De là, N est de la forme $((\lambda u.Q)^{\alpha}_{\chi_T}(\lambda z.z^{\gamma})^{\delta}_{\chi_\emptyset})^{\beta}$ avec $Q \simeq (\!|P|\!)$ et $\alpha \simeq \beta \simeq \gamma \simeq \delta \simeq \mathtt{e}$.
   (a) Si $M \to^{S,D}_{\mathsf{secW}} P = M''$ avec $T \subseteq D$, alors comme $p(\kappa\beta) = p(\kappa) = (S,D)$, on obtient la réduction $N \xrightarrow{\kappa}_{ts} \beta\cdot\lceil\alpha|\delta\rceil\cdot Q = N'$. Avec le lemme 4.4, on a $\beta\cdot\lceil\alpha|\delta\rceil\cdot Q \simeq Q \simeq (\!|P|\!)$.
   (b) Si $M \to^{S,D}_{\mathsf{secW}} \mathtt{fail}$ avec $T \not\subseteq D$, alors comme $p(\kappa\beta) = (S,D)$, on obtient $N \xrightarrow{\kappa}_{ts} \mathtt{fail}$ et $\mathtt{fail} \simeq (\!|\mathtt{fail}|\!)$.
5. Si $M = M_1M_2 \to^{S,D}_{\mathsf{secW}} M'_1M_2 = M'$ avec $M_1 \to^{S,D}_{\mathsf{secW}} M'_1$, alors $(\!|M|\!) = ((\!|M_1|\!)(\!|M_2|\!))^{\mathtt{e}}$ et $N = (N_1N_2)^{\alpha}$ où $N_1 \simeq (\!|M_1|\!)$, $N_2 \simeq (\!|M_2|\!)$ et $\alpha \simeq \mathtt{e}$. On a, en particulier, $p(\kappa\alpha@_1) = (S,D)$. De là, par hypothèse d'induction, il existe deux termes M''_1 et N'_1 qui vérifient $M'_1 \twoheadrightarrow^{S,D}_{\mathsf{secW}} M''_1$, $N_1 \xrightarrow{\kappa\alpha@_1}\!\!\!\!\twoheadrightarrow_{ts} N'_1$ et $N'_1 \simeq (\!|M''_1|\!)$. Par conséquent, on obtient $M' \twoheadrightarrow^{S,D}_{\mathsf{secW}} M''_1M_2 = M''$ et $N \xrightarrow{\kappa}\!\!\!\!\twoheadrightarrow_{ts} (N'_1N_2)^{\alpha} = N'$ avec $N' \simeq (\!|M''|\!)$.
6. Le cas $M = WN \to^{S,D}_{\mathsf{secW}} WN' = M'$ est similaire au cas précédent.
7. Si $M = T\langle P\rangle \to^{S,D}_{\mathsf{secW}} T\langle P'\rangle = M'$ avec $P \to^{T,T\cap D}_{\mathsf{secW}} P'$, alors $(\!|M|\!) = \mathtt{c}_T\cdot(\!|P|\!)$ et $N \simeq \mathtt{c}_T\cdot(\!|P|\!)$. Par hypothèse d'induction, il existe P'' et Q' tels qu'on a $P \to^{T,T\cap D}_{\mathsf{secW}} P' \twoheadrightarrow^{T,T\cap D}_{\mathsf{secW}} P''$, $(\!|P|\!) \xrightarrow{\kappa'}\!\!\!\!\twoheadrightarrow_{ts} Q'$ et $Q' \simeq (\!|P''|\!)$ pour tout κ' tel que $p(\kappa'\tau((\!|P|\!))) = (T,T\cap D) = p(\kappa\tau(\mathtt{c}_T\cdot(\!|P|\!)))$. Par conséquent, en utilisant le lemme 4.7, on a $\mathtt{c}_T\cdot(\!|P|\!) \xrightarrow{\kappa}\!\!\!\!\twoheadrightarrow_{ts} \mathtt{c}_T\cdot Q'$. Avec le lemme 4.6, on

obtient qu'il existe un terme N' tel que $N \xrightarrow{\kappa}\!\!\!\!\rightarrow_{ts} N'$ et $N' \simeq \mathtt{c}_T \cdot Q' \simeq \mathtt{c}_T \cdot (\!|P''|\!) = (\!|M''|\!)$ où $M'' = T\langle P''\rangle$.

8. Le cas $M = \mathtt{grant}\ T\ \mathtt{in}\ P \rightarrow^{S,D}_{\mathsf{sec}} \mathtt{grant}\ T\ \mathtt{in}\ P' = M'$ est similaire au cas précédent.
9. Si $M = \mathtt{fail}\ P \rightarrow^{S,D}_{\mathsf{secW}} \mathtt{fail}$, alors on a $(\!|M|\!) = (\mathtt{fail}\ (\!|P|\!))^{\mathtt{e}}$ et $N = (\mathtt{fail}\ Q)^{\alpha}$ avec $\alpha \simeq \mathtt{e}$ et $Q \simeq (\!|P|\!)$. De là, $N \xrightarrow{\kappa}_{ts} \mathtt{fail}$ et $\mathtt{fail} \simeq (\!|\mathtt{fail}|\!)$.
10. Le cas $M = W\ \mathtt{fail}$ est similaire au cas précédent. □

Ce résultat permet de faire le lien entre une réduction d'un terme M du λ_{secW}-calcul et une réduction d'un terme N $\simeq$-équivalent à la traduction de M. Une réduction élémentaire de M peut être prolongée pour aboutir à un terme M'' dont la traduction est $\simeq$-équivalente à un terme N' issu d'une réduction de N dans le λ_{ts}-calcul. Ce résultat entraîne de façon élémentaire le théorème 4.2. Pour conclure ce chapitre, on illustre le théorème 4.2 en reprenant l'exemple de la réduction $\mathcal{R}_2$ mentionné en introduction. Pour ce faire, on adapte la définition du terme M_{Del} à la syntaxe du λ_{secW}-calcul, en remplaçant le `demand` par un `check`. De là, la réduction dans le λ_{secW}-calcul, du terme $M_{\mathsf{Nav}}N_{\mathsf{Plug}}$ est donnée par $\mathcal{R}_3$.

$$M_{\mathsf{Del}} = \lambda x.S\langle \mathtt{check}\ \{\mathbf{FileIO}\}\ \mathtt{for}\ (M_{\mathsf{pDel}}x)\rangle$$

$$\begin{aligned}
\mathcal{R}_3 : M_{\mathsf{Nav}}N_{\mathsf{Plug}} &\rightarrow_{\mathsf{secW}} S\langle (M_{\mathsf{Get}}\ N_{\mathsf{Plug}})\ M_{\mathsf{Del}}\rangle \\
&\rightarrow_{\mathsf{secW}} S\langle S\langle N_{\mathsf{Plug}}\ V\rangle\ M_{\mathsf{Del}}\rangle \\
&\rightarrow_{\mathsf{secW}} S\langle S\langle U\langle \lambda y.y\ V_{\mathsf{sys}}\rangle\ M_{\mathsf{Del}}\rangle\rangle \\
&\rightarrow\!\!\!\!\rightarrow_{\mathsf{secW}} S\langle S\langle U\langle (\lambda y.y\ V_{\mathsf{sys}})\ M_{\mathsf{Del}}\rangle\rangle\rangle \\
&\rightarrow_{\mathsf{secW}} S\langle S\langle U\langle M_{\mathsf{Del}}\ V_{\mathsf{sys}}\rangle\rangle\rangle \\
&\rightarrow_{\mathsf{secW}} S\langle S\langle U\langle S\langle \mathtt{check}\ \{\mathbf{FileIO}\}\ \mathtt{for}\ (M_{\mathsf{pDel}}\ V_{\mathsf{sys}})\rangle\rangle\rangle\rangle \\
&\rightarrow_{\mathsf{secW}} S\langle S\langle U\langle S\langle \mathtt{fail}\rangle\rangle\rangle\rangle \\
&\rightarrow\!\!\!\!\rightarrow_{\mathsf{secW}} \mathtt{fail}
\end{aligned}$$

La réduction correspondante dans le λ_{ts}-calcul est donnée sur la figure 4.7. Sur cette dernière, dans un souci de lisibilité, les étiquettes $\simeq$-équivalentes à $\mathtt{e}$ sont omises, et les termes traduits portent le même nom que les termes originaux du λ_{sec}-calcul. Les étiquettes du λ_{ts}-calcul donnent l'origine des valeurs ce qui permet de détecter l'intervention du Plug-In dans l'effacement du fichier système : le test de la permission dynamique **FileIO** aboutit à un échec car $\mathbf{FileIO} \notin U$. De là, le fichier n'est pas effacé, contrairement au cas de la réduction $\mathcal{R}_2$ et on obtient finalement un terme d'échec.

Dans ce chapitre, nous avons examiné le mécanisme de sécurité de l'inspection de pile au travers du prisme du λ-calcul. Fournet et Gordon avait déjà souligné dans [15] les limites de cette politique de sécurité. Ici, nous nous sommes inspiré de l'inspection de pile pour adapter un mécanisme similaire dans le λ-calcul. Synthétiquement, l'inspection de pile peut être vu comme un mécanisme de contrôle qui fonde ses décisions sur (1) une information locale (les permissions statiques) et (2) une information globale (les permissions dynamiques). Dans le λ_t-calcul, les étiquettes représentent l'information locale, alors que le chemin étiqueté menant à un sous-terme constitue son information globale. Chaque radical R porte une condition (qui dépend de l'étiquette et du chemin du radical) qui autorise ou non la contraction de R. Nous avons examiné les conditions de confluence du λ_t-calcul. Nous avons aussi introduit le λ_{ts}-calcul, qui est une instance du λ_t-calcul pour laquelle l'ordre d'évaluation est fixé et les conditions sont contraintes pour se rapprocher du mécanisme d'inspection de pile original. Nous avons défini une traduction correcte d'une variante de l'inspection de pile proposée par Fournet et Gordon vers le λ_{ts}-calcul. Cette traduction constitue le résultat central de ce chapitre ; elle établit un lien entre, d'une part l'inspection de pile c'est-à-dire un mécanisme de sécurité ad hoc, et d'autre part le λ_{ts}-calcul, un langage fondé sur le langage de programmation théorique qu'est le λ-calcul.

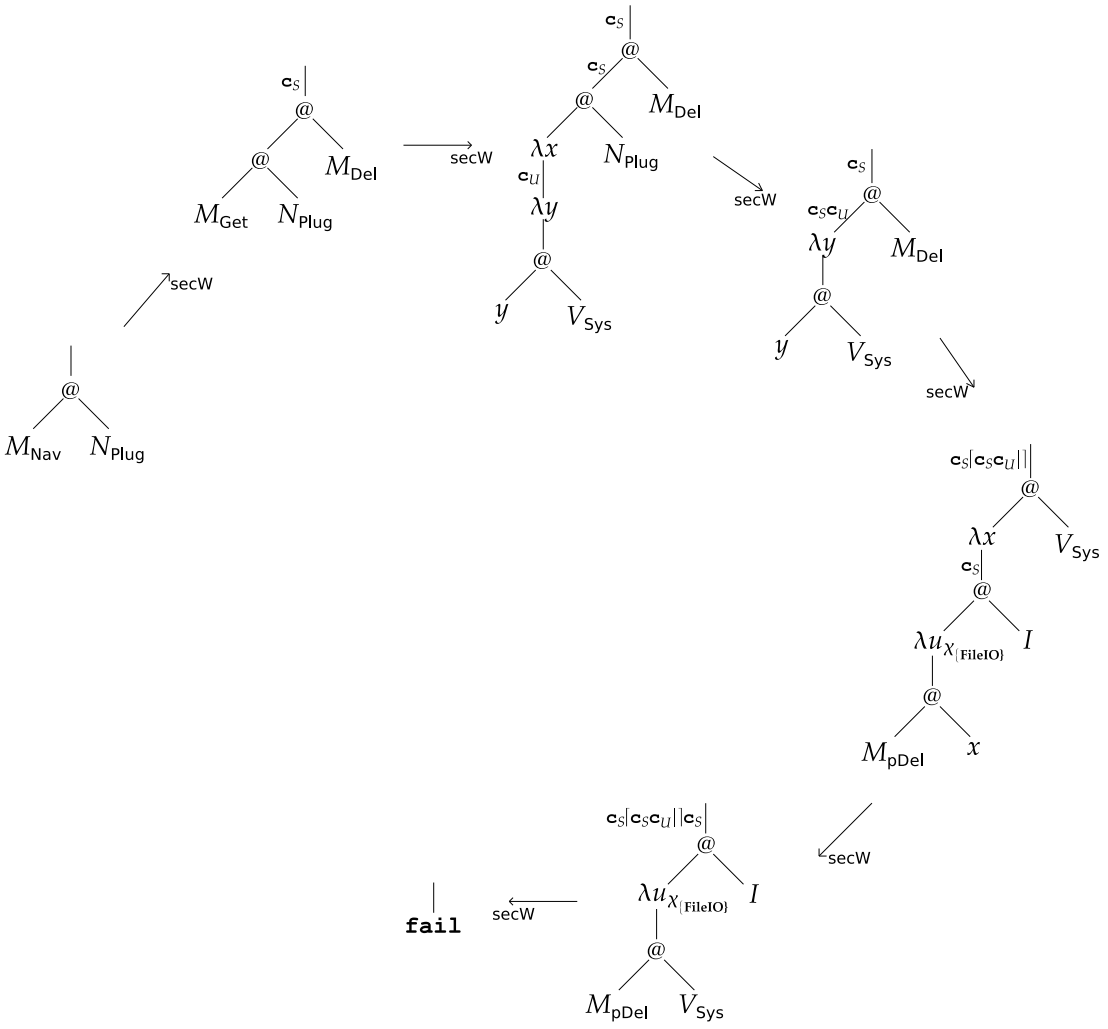


FIG. 4.7 – *Réduction dans le λ_{ts}-calcul de $M_{\mathsf{Nav}}N_{\mathsf{Plug}}$*

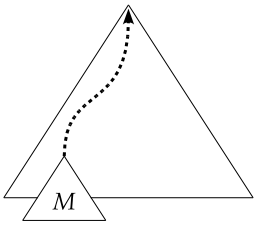
Chapitre 5

Non-interférence

La propriété de non-interférence a été introduite en 1982 par Goguen et Meseguer [16]. En intégrant le concept plus récent de principal, leur définition de cette notion est la suivante : un principal, qui dispose d'un certain ensemble de commandes, n'interfère pas avec un autre principal si les actions du premier n'ont aucun effet visible pour le second.

La non-interférence est une propriété qui s'est révélée particulièrement adaptée aux analyses statiques de flot d'information initiées par Denning et Denning [14, 13]. Intuitivement, ces analyses classent les données en fonction de leur niveau de sécurité. On peut, par exemple, considérer qu'une donnée est soit secrète, soit publique. Un programme peut recevoir en entrée et fournir en sortie des données de différents niveaux. Le but de l'analyse de flot d'information est de déterminer si, en observant les sorties publiques, on peut obtenir une information sur les entrées secrètes. Si tel est le cas, on considère que le programme n'est pas sûr puisqu'il révèle des informations secrètes. Plus concrètement, ces analyses déterminent habituellement si le résultat de l'exécution d'un programme (intuitivement public) révèle des informations sur les données secrètes présentes dans ce programme. On note que la terminaison de l'exécution ou les canaux dits cachés (par exemple le temps d'exécution ou la consommation électrique au cours de l'exécution) ne sont pas pris en compte. Il s'avère que la propriété de sécurité vérifiée par une telle analyse se ramène à la définition de la non-interférence de Goguen-Meseguer. En effet, si on suppose qu'un premier principal peut modifier les entrées secrètes du programme et qu'un deuxième principal n'a accès qu'aux sorties publiques, prouver que le programme est sûr au sens de l'analyse de flot d'information revient à vérifier une propriété de non-interférence. Volpano, Smith et Irvine [42] ont été les premiers à faire ce lien entre les analyses statiques de flot d'information et la non-interférence : le résultat de l'exécution d'un terme bien typé ne révèle pas d'information secrète. La correction de ce système de type est une propriété de non-interférence. Ces résultats ont engendré de nombreux travaux sur les analyses de flot d'information visant à obtenir des systèmes de type pour lesquels tout terme bien typé vérifie la propriété de non-interférence. Volpano et Smith [41] ont poursuivi leurs travaux sur les langages impératifs. Heintze, Riecke, Abadi et Banerjee [22, 1] ont étudié ces analyses dans le cadre de langages fonctionnels dérivés du λ-calcul pur. Simonet et Pottier [37, 39] ont poursuivi l'effort pour traiter Objective Caml, un langage fonctionnel complet, alors que Myers, Nystrom, Zdancewic et Zheng [32] se sont concentrés sur Java. Tous ces travaux portent sur des systèmes de type qui permettent de vérifier statiquement la propriété de non-interférence.

Une autre approche a été initiée par Abadi, Lampson et Lévy [3]. Ces derniers ont proposé une analyse de dépendances dans le contexte du λ-calcul. Les étiquettes du λ-calcul permettent de déterminer si un sous-terme M du terme initial de la réduction contribue au résultat de cette dernière. Cette situation est illustrée par la figure 5.1. Le résultat de cette réduction étant destiné à être public, il ne doit pas dépendre des sous-termes secrets du terme initial. Dans ce cas, les



Est-ce que le résultat de la réduction du terme dépend du sous-terme M ?

FIG. 5.1 – *Non-interférence*

données secrètes n'ont pas interféré dans l'obtention du résultat public de la réduction. Alors que les systèmes de type fournissent une analyse de flot statique, qui nécessite certaines approximations, les étiquettes attachées à un terme indiquent dynamiquement l'information portée par ce dernier. Les travaux de Conchon et Pottier [36] s'inspirent de cette approche. Ils utilisent un langage fonctionnel étiqueté dont les étiquettes, calculées dynamiquement, fournissent une analyse de flot dynamique. En exploitant ce calcul étiqueté, ils proposent un système de type permettant d'assurer statiquement une propriété de non-interférence. Dans ce cas, l'utilisation du calcul étiqueté permet d'obtenir de façon très commode la propriété de non-interférence via une propriété de stabilité. Cependant, en présence de références, cette approche devient plus délicate à cause des effets de bord. L'exemple $\texttt{ifz}\ x\ \texttt{then}\ ()\ \texttt{else}\ y{:=}0$ montre que si, après la réduction de ce terme, la valeur associée à y est non nulle, alors on peut en déduire $x = 0$. En d'autres termes, la non-réduction du sous-terme $y{:=}0$ donne une information sur x.

Dans ce chapitre, l'objectif est d'adapter l'approche d'Abadi et al. à un langage faisant intervenir des références. Dans un premier temps, dans la section 5.1, on étudie la propriété de non-interférence dans le λ-calcul et λ-calcul par valeur. Les valeurs de ces langages ne sont pas des constantes (e.g. des entiers). La définition de la propriété de non-interférence doit être soigneusement adaptée aux valeurs fonctionnelles. Dans le cas du λ-calcul, on établit une relation entre, d'une part la propriété de non-interférence et, d'autre part, les notions de stabilité et de sous-terme critique. Plus précisément, on montre que dans le λ-calcul, un sous-terme interfère si et seulement s'il est critique. Cette équivalence n'est toutefois pas automatique. En particulier, elle n'est pas vraie dans le λ-calcul par valeur. Après avoir étudié la propriété de non-interférence dans le cadre de langages purement fonctionnels, on examine les changements impliqués par la présence d'effets de bord. Dans la section 5.2, on introduit le λ_m-calcul, qui est un λ-calcul muni de δ-règles arithmétiques et conditionnelles et de traits impératifs tels que l'affectation. Ce langage nous permet d'examiner quelques exemples qui permettent de mettre en lumière les difficultés engendrées par la présence d'effets de bord pour la propriété de non-interférence. En plus de l'interférence fonctionnelle déjà observée dans le λ-calcul ou le λ-calcul par valeur, on constate qu'il existe une interférence de mémoire, c'est-à-dire liée à la mémoire. Ainsi, les adresses mémoire peuvent interférer sur le résultat d'une réduction. Et cette interférence s'exerce pendant un certain *intervalle* de temps : entre une écriture et une lecture en mémoire. Dans la section 5.3, on définit le λ_m-calcul étiqueté. Les étiquettes de ce langage visent à exprimer les interférences fonctionnelles et de mémoire. On exploite la propriété d'irréversibilité des chemins pour nommer les adresses avec un nom structurel. Comme pour la propriété de stabilité 1.15 du λ-calcul étiqueté, si un terme se réduit vers une valeur, les étiquettes du λ_m-calcul permettent de déterminer les sous-termes du terme initial qui ont contribué à cette valeur. De plus, les étiquettes permettent aussi d'identifier les intervalles de temps pendant lesquels des adresses de la mémoire ont contribué à cette valeur. Dans la section 5.3.2, on prouve la correction des étiquettes vis-à-vis des intervalles en définissant une réduction contrainte par

un ensemble d'intervalles. Dans la section 5.3.3, on montre que les étiquettes expriment bien une propriété de non-interférence.

5.1 Non-interférence dans le λ-calcul et le λ-calcul par valeur

Si M est un terme dont les différents sous-termes peuvent être publics ou secrets et si M se réduit vers une valeur V, l'enjeu de la non-interférence est de savoir si l'*observation* de cette valeur V, intuitivement publique, donne une information sur un sous-terme secret de M. Dans le cadre du λ-calcul, nous souhaitons examiner cette propriété intuitive en s'inspirant de l'approche des analyses de flot d'information telle que celle développée par Simonet et Pottier [37, 39]. Ces derniers garantissent une propriété de non-interférence en utilisant un typage dans lequel les types ont deux composantes : (1) un type à la ML "classique" (par exemple, le type entier `int`) et (2) un niveau de sécurité qui permet de distinguer les termes secrets des termes publics. La propriété de non-interférence s'énonce informellement de la façon suivante : si le terme M est du type entier public, si x est une variable libre de M dont le type est de niveau secret, si V et V' sont des valeurs de même type que x et si $M\{x\backslash V\}$ et $M\{x\backslash V'\}$ se réduisent vers des entiers n et n', alors ces entiers sont égaux. Ainsi, si la réduction de $M\{x\backslash V\}$ aboutit à une valeur n, cette valeur n ne dépend pas de V et ne fournit donc aucune information sur V. Pour revenir à l'intuition originale de Goguen et Meseguer, un changement d'une valeur secrète V en V' n'a pas d'effet visible du point de vue des valeurs publiques obtenues à l'issue des réductions de $M\{x\backslash V\}$ et $M\{x\backslash V'\}$. Il est important de noter ici que le *canal caché* constitué par l'information selon laquelle la réduction de $M\{x\backslash V\}$ aboutit ou non à une valeur, n'est pas pris en compte. Dans cette section, on s'inspire de la démarche de Simonet et Pottier pour définir une propriété de non-interférence dans le cadre du λ-calcul non typé.

Dans le cadre des travaux de Simonet et Pottier, prouver la non-interférence d'un sous-terme V de M avec le résultat n de la réduction de M, consiste à montrer que si en remplaçant V par une valeur (de même type que V) quelconque V', on obtient une valeur n', alors les valeurs n et n' sont *les mêmes* : on a $n = n'$. On souhaite adapter cette approche de la non-interférence au λ-calcul : si $(C[\,],N)$ est un sous-terme de M et si $M \twoheadrightarrow V$, montrer la non-interférence de N au cours de la réduction menant à V consiste, informellement, à montrer qu'en remplaçant N par N', si $C[N'] \twoheadrightarrow V'$, alors les valeurs V et V' sont *les mêmes*. Le λ-calcul et le λ-calcul par valeur ne contiennent pas de constantes telles que les entiers. Les seules valeurs sont les abstractions. Si la notion de "même valeur" est évidente pour les entiers, cette notion est moins claire dans le cas des abstractions. Pour illustrer ce sujet plus concrètement, on considère l'exemple des réductions des termes $M = (\lambda x.I\lambda y.x)V$ et $M' = (\lambda x.I\lambda y.x)V'$ où $I = V = \lambda z.z$ et $V' = \lambda z.u$.

$$\begin{aligned} \mathcal{R} &: (\lambda x.I\lambda y.x)V \;\to\; I\lambda y.V \;\to\; \lambda y.V \\ \mathcal{R}' &: (\lambda x.I\lambda y.x)V' \to I\lambda y.V' \to \lambda y.V' \end{aligned}$$

Le terme M' est le terme M où la valeur V a été changé en V'. Les valeurs finales des réductions $\mathcal{R}$ et $\mathcal{R}'$ ne sont pas égales. Pourtant, les valeurs V et V' n'ont participé à aucune réduction. Intuitivement, les valeurs $\lambda y.V$ et $\lambda y.V'$ sont identiques à un sous-terme (strict) près. Ces valeurs ont été obtenues de la même façon. Pour distinguer ces valeurs plus clairement, on pourrait utiliser le contexte $C_0[\,] = ([\,]I)I$. On a $C_0[\lambda y.V] \twoheadrightarrow I$ et $C_0[\lambda y.V'] \twoheadrightarrow u$. Ce contexte permet de distinguer ces résultats puisque dans le premier cas, on obtient une valeur, au contraire du deuxième cas. Cependant cette distinction est le fruit d'une interaction entre le contexte $C[\,]$ et les sous-termes V et V'. En d'autres mots, le contexte $C[\,]$ a *interféré* avec V et V'. En revanche, les sous-termes V et V' de M et M' n'interfèrent pas dans les réductions $\mathcal{R}$ et $\mathcal{R}'$. Ceci nous amène à ignorer les

sous-termes stricts des valeurs obtenues et à définir dans ce but la notion d'**observable**.

$$\mathcal{O}(\lambda x.M) = \lambda x.\Omega$$

L'observable d'une abstraction est l'abstraction $\lambda x.\Omega$. Avec cette définition, on peut donner une première tentative de définition de la propriété d'interférence dans le λ-calcul, qui s'inspire de la définition informelle donnée précédemment.

Enoncé 5.1 (Non-interférence) *Le sous-terme $(C[\,],N)$ de M n'interfère pas dans la réduction $M = C[N] \twoheadrightarrow V$ si et seulement si, pour tout N', la réduction $C[N'] \twoheadrightarrow V'$ implique la relation $\mathcal{O}(V) = \mathcal{O}(V')$.*

Un sous-terme n'interfère pas dans une réduction aboutissant à une valeur si, une modification de ce sous-terme ne permet pas d'obtenir une valeur dont l'observable est différent. Comme dans le cas des analyses de flot d'information, on ne tient pas compte ici du canal caché que constitue l'information d'aboutissement vers une valeur. Plus concrètement, on ignore les cas où, en remplaçant N par N' dans M, la réduction n'aboutit pas à une valeur. Ceci se traduit dans la définition de l'interférence par le fait que la convergence de $C[N']$ vers une valeur est une hypothèse. La définition de l'observable d'une valeur et l'énoncé précédent soulèvent toutefois une difficulté : toutes les valeurs, c'est-à-dire toutes les abstractions ont le même observable. Si on adoptait ces définitions, dans l'exemple de la réduction de $M = (\lambda x.I\lambda y.x)V$, on obtiendrait que M n'interfère pas dans la réduction $M \twoheadrightarrow \lambda y.V$. En effet, en remplaçant le sous-terme M par un terme N quelconque et en supposant $N \twoheadrightarrow W$, on a $\mathcal{O}(\lambda y.V) = \mathcal{O}(W) = \lambda y.\Omega$. Intuitivement, les valeurs $\lambda y.V$ et W ne sont pourtant pas les mêmes. La notion d'observable utilisée ici ne capture pas cette intuition. On peut rapprocher cette imprécision des *coïncidences syntaxiques* mentionnées dans la partie 1.3. Dans ce dernier cas, les étiquettes du λ-calcul permettent de distinguer des termes accidentellement identiques. Ainsi, le terme $I(Ix)$ peut se réduire de deux façons différentes vers Ix. Ces termes Ix sont bien syntaxiquement égaux mais ne sont pas intuitivement les mêmes. Dans le cas présent, on exploite aussi les étiquettes du λ-calcul pour pouvoir identifier les valeurs et plus particulièrement leur origine. On se place donc dans le λ-calcul étiqueté et on définit l'observable d'une valeur étiquetée de la façon suivante.

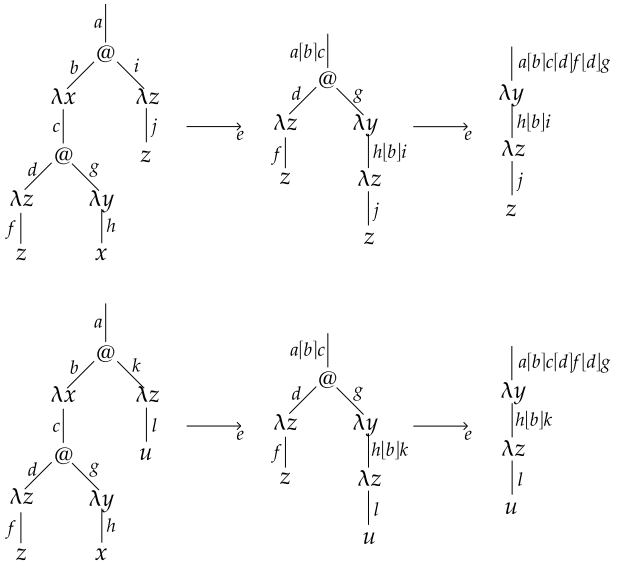
$$\mathcal{O}((\lambda x.M)^\alpha) = (\lambda x.\Omega)^\alpha$$

L'observable d'une abstraction est essentiellement caractérisé par son étiquette de tête. Avec cette définition de l'observable, on définit la propriété de non-interférence dans le λ-calcul étiqueté.

Définition 5.1 (Non-interférence) *Le sous-terme $(C[\,],N)$ du terme M n'interfère pas dans la réduction $M \twoheadrightarrow_e V$ si et seulement si, pour tout terme N', la réduction $C[N'] \twoheadrightarrow_e V'$ implique $\mathcal{O}(V) = \mathcal{O}(V')$.*

Cet énoncé est une adaptation de l'énoncé 5.1. A l'aide de ces nouvelles définitions, on réexamine les exemples mentionnés précédemment. Les réductions de $M = ((\lambda x.((\lambda z.z^f)^d(\lambda y.x^h)^g)^c)^bV)^a$ (où $V = (\lambda z.z^j)^i$) et de $M' = ((\lambda x.((\lambda z.z^f)^d(\lambda y.x^h)^g)^c)^bV')^a$ (où $V' = (\lambda z.u^j)^k$) sont représentées sur la figure 5.2. Conformément à l'intuition, on note que les observables des deux valeurs finales sont bien égaux à $(\lambda y.\Omega)^{a\lceil b\rceil c\lceil d\rceil f\lfloor d\rfloor g}$. Plus généralement, en remplaçant le sous-terme V par un sous-terme quelconque N', on peut montrer que l'observable d'une valeur obtenue est toujours $(\lambda y.\Omega)^{a\lceil b\rceil c\lceil d\rceil f\lfloor d\rfloor g}$. Ceci prouve que le sous-terme V de M n'interfère pas. La nouvelle définition de l'observable capture donc bien la notion de *même valeur* mentionnée dans la définition informelle de la non-interférence que nous avons donnée précédemment.

Dans [36], Conchon et Pottier font le lien entre la propriété de stabilité et la non-interférence en utilisant également un calcul étiqueté inspiré du λ-calcul étiqueté. Pour illustrer ce lien dans le cadre du λ-calcul, il est utile de faire le parallèle entre la définition, étroitement liée à la stabilité,



$M = ((\lambda x.((\lambda z.z^f)^d(\lambda y.x^h)^g)^c)^b V)^a$ où $V = (\lambda z.z^j)^i$
$M' = ((\lambda x.((\lambda z.z^f)^d(\lambda y.x^h)^g)^c)^b V')^a$ où $V' = (\lambda z.u^j)^k$

FIG. 5.2 – *Réductions comparées de M et M'*

de sous-terme *non-critique* et celle de la non-interférence. On considère la réduction $\mathcal{R} : M \twoheadrightarrow_e V$ où $M = C[N]$.

($C[\,]$,N) n'est pas critique ssi, pour tout N', **on a** $C[N'] \twoheadrightarrow_e V'$ **et** $\mathcal{O}(V) = \mathcal{O}(V')$
($C[\,]$,N) n'interfère pas ssi, pour tout N', **si** $C[N'] \twoheadrightarrow_e V'$ **alors** $\mathcal{O}(V) = \mathcal{O}(V')$

Ces définitions ne permettent pas de faire un lien immédiat entre sous-terme critique et sous-terme qui interfère. En revanche, dans le cadre du λ-calcul étiqueté, sous réserve que le terme M initial vérifie l'invariant `INIT`, le résultat 1.16 montre que les termes critiques de M sont ceux dont l'étiquette appartient à l'étiquette de tête de la valeur obtenue. On prouve un résultat similaire, qui permet de caractériser les sous-termes qui interfèrent à l'aide des étiquettes du λ-calcul étiqueté.

Théorème 5.1 (Non-interférence) *On suppose* `INIT`(M) *et* $\mathcal{R} : M \twoheadrightarrow_e V$. *Le sous-terme* ($C[\,]$,$N$) *de M interfère dans $\mathcal{R}$ si et seulement si* $\tau(N) \in |\tau(V)|$.

Preuve : Soit ($C[\,]$,N) un sous-terme de M. Du fait du résultat de stabilité 1.15, on obtient que le préfixe $\mathcal{P}_S(M) = [\![M]\!]_{|\tau(V)|}$ de M vérifie : il existe une valeur V_0 qui est atteinte par la réduction $\mathcal{P}_S(M) \twoheadrightarrow_e V_0$. Du fait du résultat de monotonie 1.13, on obtient $\mathcal{O}(V_0) = \mathcal{O}(V)$. Si $\tau(N) \notin |\tau(V)|$, on a $\mathcal{P}_S(M) = [\![M]\!]_{|\tau(V)|} \preccurlyeq C[\Omega]$. Soit N' un terme quelconque qui vérifie $C[N'] \twoheadrightarrow_e V'$. On obtient $\mathcal{P}_S(M) \preccurlyeq C[\Omega] \preccurlyeq C[N']$ et donc, par monotonie, la réduction $C[N'] \twoheadrightarrow_e V''$ et $V_0 \preccurlyeq V''$. La valeur V'' vérifie bien sûr $\mathcal{O}(V') = \mathcal{O}(V'')$ et $\mathcal{O}(V_0) = \mathcal{O}(V'')$ ce qui prouve que le terme N n'interfère pas dans $\mathcal{R}$. Réciproquement, on suppose que ($C[\,]$,N) n'interfère pas dans $\mathcal{R}$. Soit a une étiquette distincte des étiquettes présentes dans M. Soit N' le terme obtenu à partir de N en changeant l'étiquette de tête en a. Il est clair que $C[N']$ se réduit vers une valeur V'. Par définition de la non-interférence, on obtient $\mathcal{O}(V) = \mathcal{O}(V')$ puis $\tau(V) = \tau(V')$. Comme l'étiquette $\tau(N)$ est absente de $C[N']$, cette étiquette n'appartient pas à $\tau(V')$. On en déduit donc $\tau(N) \notin |\tau(V)|$. □

Si les étiquettes de M sont des lettres distinctes, un sous-terme N de M interfère dans la réduction $M \twoheadrightarrow_e V$ si et seulement si son étiquette est une lettre présente dans l'étiquette de tête de V. On en déduit, en utilisant le résultat 1.16, qu'un sous-terme interfère si et seulement s'il est critique. Pour revenir à la notion de stabilité, on peut adopter un point de vue plus global en définissant le

préfixe d'interférence $\mathcal{P}_I(M)$ d'un terme M qui vérifie $\mathcal{R} : M \twoheadrightarrow_e V$.

$$
\begin{array}{rcll}
\mathcal{P}_I(M) & = & \mathcal{P}_I(M,[\,]) & \\
\mathcal{P}_I(x^\alpha,C[\,]) & = & x^\alpha & \text{si } (C[\,],x^\alpha) \text{ interfère dans } \mathcal{R} \\
\mathcal{P}_I((\lambda x.M)^\alpha,C[\,]) & = & (\lambda x.\mathcal{P}_I(M,C[(\lambda x.[\,])^\alpha]))^\alpha & \text{si } (C[\,],(\lambda x.M)^\alpha) \text{ interfère dans } \mathcal{R} \\
\mathcal{P}_I((MN)^\alpha,C[\,]) & = & (\mathcal{P}_I(M,C[([\,]N)^\alpha])\,\mathcal{P}_I(N,C[(M[\,])^\alpha]))^\alpha & \text{si } (C[\,],(MN)^\alpha) \text{ interfère dans } \mathcal{R} \\
\mathcal{P}_I(M,C[\,]) & = & \Omega & \text{si } (C[\,],M) \text{ n'interfère pas dans } \mathcal{R}
\end{array}
$$

Intuitivement, le préfixe d'interférence de M est un préfixe de M où seuls les sous-termes qui interfèrent sont conservés. On note que cette définition ne dépend pas de la réduction (vers une valeur) considérée. L'introduction du préfixe d'interférence, en conjonction avec le théorème 5.1, permet d'aboutir au résultat suivant.

Théorème 5.2 *Si M vérifie* $\texttt{INIT}(M)$ *et si $M \twoheadrightarrow_e V$, alors les préfixes d'interférence et de stabilité de M coïncident : on a $\mathcal{P}_I(M) = \mathcal{P}_S(M)$.*

Preuve : Ce résultat est une conséquence directe du théorème 5.1 et du résultat 1.16. □

Si les étiquettes de M sont des lettres distinctes, les préfixes d'interférence et de stabilité de M coïncident. Cette propriété permet de faire le lien entre stabilité et non-interférence. Ce lien a été implicitement utilisé dans [36] pour prouver une propriété de non-interférence. Nous verrons dans le paragraphe suivant que cette coïncidence n'est pas systématique.

On examine maintenant la propriété de non-interférence dans le λ-calcul par valeur. De même que précédemment, on se place dans le λ-calcul étiqueté pour avoir une définition de l'observable qui permette d'identifier les valeurs. Dans le λ-calcul étiqueté, une réduction qui contracte des radicaux de la forme $((\lambda x.N)^\alpha V)^\beta$ est une réduction par valeur. Ici, nous n'utilisons pas les étiquettes introduites pour le λ-calcul par valeur dans le chapitre 2. En effet, ces étiquettes ont pour objet de capturer la propriété de stabilité du calcul par valeur. Ici, nous visons la propriété de non-interférence suivante.

Définition 5.2 (Non-interférence) *Le sous-terme $(C[\,],N)$ du terme M n'interfère pas dans la réduction par valeur $M \twoheadrightarrow_e V$ si et seulement si pour tout terme N', le fait que $C[N'] \twoheadrightarrow_e V'$ est une réduction par valeur implique $\mathcal{O}(V) = \mathcal{O}(V')$.*

Cette définition est une adaptation directe de la définition 5.1 dans laquelle on ne considère que des réductions par valeur. Il s'avère que le théorème de non-interférence qui fait le lien entre les étiquettes et la propriété de non-interférence est inchangé.

Théorème 5.3 (Non-interférence) *Soit M un terme tel que* $\texttt{INIT}(M)$ *et $\mathcal{R} : M \twoheadrightarrow_e V$ où $\mathcal{R}$ est une réduction par valeur. Le sous-terme $(C[\,],N)$ de M interfère dans la réduction par valeur $\mathcal{R}$ si et seulement si $\tau(N) \in |\tau(V)|$.*

Preuve : Soit $(C[\,],N)$ un sous-terme de M. Du fait du résultat de stabilité 1.15, on obtient que le préfixe $\mathcal{P}_S(M) = [\![M]\!]_{|\tau(V)|}$ de M vérifie : il existe une valeur V_0 telle que $\mathcal{P}_S(M) \twoheadrightarrow_e V_0$, où, comme nous l'avons vu dans le chapitre 2, cette réduction n'est pas nécessairement par valeur. Comme la réduction menant de M à V peut-être vue comme une réduction du λ-calcul étiqueté, en utilisant le résultat de monotonie 1.13, on obtient $\mathcal{O}(V_0) = \mathcal{O}(V)$. Si $\tau(N) \notin |\tau(V)|$, on a $\mathcal{P}_S(M) = [\![M]\!]_{|\tau(V)|} \preccurlyeq C[\Omega]$. Soit N' un terme quelconque qui vérifie $\mathcal{R} : C[N'] \twoheadrightarrow_e V'$ où $\mathcal{R}$ est une réduction par valeur. On obtient $\mathcal{P}_S(M) \preccurlyeq C[\Omega] \preccurlyeq C[N']$ et donc, par monotonie, la réduction $\mathcal{R}' : C[N'] \twoheadrightarrow_e V''$ avec $V_0 \preccurlyeq V''$. La réduction $\mathcal{R}'$ n'est pas nécessairement une réduction par valeur. En revanche, la valeur V'' vérifie $\mathcal{O}(V') = \mathcal{O}(V'')$ et $\mathcal{O}(V_0) = \mathcal{O}(V'')$ ce qui prouve que le terme N n'interfère pas dans $\mathcal{R}$. Réciproquement, on suppose que $(C[\,],N)$ n'interfère pas dans $\mathcal{R}$. Soit a une étiquette distincte des étiquettes présentes dans M. Soit N' le terme obtenu à partir de N en changeant l'étiquette de tête en a. Il est clair que $C[N']$ se réduit par valeur vers une valeur

V'. Par définition de la non-interférence, on obtient $\mathcal{O}(V) = \mathcal{O}(V')$ puis $\tau(V) = \tau(V')$. Comme l'étiquette $\tau(N)$ est absente de $C[N']$, cette étiquette n'appartient pas à $\tau(V')$. On en déduit donc $\tau(N) \notin |\tau(V)|$. □

Comme dans le cas du λ-calcul, si les étiquettes d'un terme M sont des lettres distinctes et si $M \twoheadrightarrow_e V$, un sous-terme interfère si son étiquette est une lettre de $\tau(V)$. De ce fait, les préfixes d'interférence pour le λ-calcul par valeur et pour le λ-calcul classique coïncident. Par conséquent, le préfixe d'interférence pour le λ-calcul par valeur coïncide avec le préfixe de stabilité du λ-calcul. On illustre ces propriétés en prenant l'exemple du terme $((\lambda x.(\lambda y.y^d)^c)^b(\lambda z.z^g)^f)^a$. Ce terme se réduit vers la valeur $V_0 = (\lambda y.y^d)^{a\lceil b\rceil c}$. En vertu du théorème 5.3, le préfixe d'interférence est donc $\mathcal{P}_I(M) = ((\lambda x.(\lambda y.\Omega)^c)^b\Omega)^a$. Dans le cadre du λ-calcul classique, le préfixe de stabilité est le même. Mais en se plaçant dans le λ-calcul par valeur, comme nous l'avons vu dans le chapitre 2, le préfixe de stabilité est en fait $\mathcal{P}_S(M) = ((\lambda x.(\lambda y.\Omega)^c)^b(\lambda z.\Omega)^f)^a$. Dans le λ-calcul par valeur, les préfixes de stabilité et de non-interférence ne coïncident plus ; l'équivalence entre terme critique et terme qui interfère n'est plus vraie. Cette non-coïncidence s'explique simplement en reprenant le parallèle entre les définitions de sous-terme non critique et sous-terme non-interférant mentionné sur la page 111. La non-coïncidence des préfixes de non-interférence et de stabilité pour les réductions par valeur s'explique par l'hypothèse d'obtention d'une valeur à partir du terme $C[N']$. Cette obtention de valeur est garantie dans la définition de sous-terme critique, alors qu'il s'agit d'une hypothèse d'implication dans le cas de la non-interférence. La propriété de stabilité est donc plus forte. Par conséquent, de façon générale, le préfixe de stabilité majore le préfixe de non-interférence.

Dans cette partie, nous avons mis en lumière la relation entre les propriétés de stabilité et d'interférence. Nous avons montré, en particulier, que ces notions coïncident dans le cas du λ-calcul. En revanche, dans le cas du λ-calcul par valeur, la propriété de stabilité est plus forte que la propriété d'interférence car en modifiant un terme non-critique, on obtient bien une valeur alors que si on modifie un terme qui n'interfère pas, l'obtention d'une valeur n'est pas garantie.

5.2 Le λ_m-calcul

Dans cette section, on examine informellement la propriété de non-interférence pour un langage inspiré de Core ML [37, 39] : le λ_m-calcul. Ce langage, fondé sur le λ-calcul présenté dans la section 1.1, dispose en plus de δ-règles pour les opérations arithmétiques et conditionnelles, et de traits impératifs dans le langage afin de pouvoir effectuer des affectations ou des lectures en mémoire. Nous aurions pu traiter dans une section intermédiaire le cas d'un λ-calcul augmenté seulement des δ-règles arithmétiques et conditionnelles. Cependant, comme l'ajout de ces traits fonctionnels ne change pas fondamentalement la situation par rapport au λ-calcul par valeur étudié dans la section 5.1, nous avons préféré ajouter ces δ-règles en même temps que les traits impératifs. Comme nous le verrons par la suite, la présence d'effets de bord change radicalement la nature de la question de la non-interférence.

La syntaxe du langage étudié dans cette section, le λ_m-calcul, est décrite sur la figure 5.3. Pour alléger au maximum les notations de ce chapitre, on se permet de réutiliser certaines notations du λ-calcul ($\mathbf{\Lambda}$,$\mathbf{V}$,...), dans la mesure où le contexte ne laisse aucune ambiguïté. En plus des variables, abstractions et applications du λ-calcul, on ajoute les entiers n, l'addition $M + N$ et le test `ifz` M `then` N `else` P qui porte sur la nullité d'un entier. On ajoute également des traits impératifs. Dans le λ_m-calcul, on dispose d'un ensemble dénombrable d'adresses $\mathbf{M}$. Ces adresses, notées m, représentent les emplacements de la mémoire. Formellement, une mémoire est une application μ définie sur un sous-ensemble fini de $\mathbf{M}$ et qui associe une valeur à chaque adresse de l'ensemble de définition $dom(\mu)$ de μ. Pour une bonne lisibilité, une mémoire μ sera notée

$\mu = \{m_i \mapsto V_i\}_{i \in I}$. Si la mémoire μ étend la mémoire μ' sur l'adresse m, on notera le produit tensoriel correspondant $\mu = (\mu'; m \mapsto V)$. La mémoire vide sera notée $\emptyset$. Pour manipuler la mémoire, quatre nouveaux types de termes sont ajoutés dans la syntaxe du λ_m-calcul. Les *adresses* m sont intuitivement absentes des termes initiaux. Elles sont créées au cours de la réduction par les *références* $\texttt{ref}(M)$. L'*affectation* $M{:=}N$ permet de modifier la valeur associée à une adresse. La *déréférence* $!M$ permet d'accéder à la valeur associée dans la mémoire à une adresse. Le terme *Unit* est ajouté pour les réductions qui ne produisent pas de résultat, par exemple l'affectation. Dans ce calcul, les valeurs sont les abstractions, les entiers, les adresses ou *Unit*.

La réduction $\rightarrow$ du langage est définie sur la figure 5.4. Cette réduction met en relation deux configurations constituées chacune d'un terme et d'une mémoire. La règle (β_m) est reprise du λ-calcul par valeur. L'opération de substitution du λ-calcul est étendue sur les nouveaux termes par la définition de la figure 5.5. Dans le λ_m-calcul, du fait de la définition de la (β_m)-réduction, on ne substitue aux variables que des valeurs. On note que l'ordre d'évaluation est fixé par la règle de contexte (Ctx). Les contextes d'évaluation, définis sur la figure 5.6, imposent une évaluation de gauche à droite sur les applications et les affectations. L'utilisation d'un ordre d'évaluation est dictée par la présence des effets de bord qui rendent le langage non confluent. Par exemple, la réduction de la configuration $((\lambda y.\lambda x.!m)(m{:=}2))(m{:=}0))/(m \mapsto 1)$ peut aboutir à 0, 1 ou 2 selon l'ordre d'évaluation choisi. La δ-règle (Plus) effectue l'addition de deux entiers. Si n est l'entier 0, la réduction de $\texttt{ifz}\ n\ \texttt{then}\ M\ \texttt{else}\ N$ par la δ-règle (Ifz-true) aboutit au terme M. Si le test de nullité échoue, le terme N est obtenu par la règle (Ifz-false). Les règles décrites jusqu'à présent ne modifient pas la mémoire.

La mémoire est modifiée par trois règles. La réduction de la configuration $\texttt{ref}(V)/\mu$ par la règle (Ref) ajoute une nouvelle adresse m à la mémoire μ et retourne cette adresse. Cette adresse est *fraîche* : elle n'appartient pas au domaine de μ. Hormis cette contrainte, le choix de m n'est pas spécifié. La valeur V lui est associée dans la mémoire $(\mu; m \mapsto V)$ obtenue. La réduction d'une configuration $m{:=}V/(\mu; m \mapsto V')$ par la règle (Assign) change la valeur associée par la mémoire à l'adresse m. La nouvelle mémoire associe V à m. Le résultat de la réduction est *Unit*. La réduction d'une déréférence $!m/(\mu; m \mapsto V)$ par la règle (Deref) permet d'accéder à la valeur associée à l'adresse m par la mémoire.

On illustre le langage et les difficultés engendrées pas les effets de bord en étudiant la réduction du terme $M = (\lambda y.(\lambda\_.!y)\ 1)\ \texttt{ref}(0)$ qui est illustrée sur la figure 5.7. Pour représenter la mémoire de façon intuitive, on utilise une bande horizontale pour chaque adresse mémoire. Les opérations d'écriture sont matérialisées par une flèche annotée par E alors que les opérations de lecture sont matérialisées par une flèche annotée par L. La première réduction de M réduit la référence $\texttt{ref}(0)$ ce qui ajoute à la mémoire une nouvelle adresse m_1. Cette dernière est associée en mémoire à la valeur 0. La deuxième réduction est une (β_m)-réduction. L'adresse m_1 est substituée sous la déréférence. La troisième réduction est également une (β_m)-réduction. Comme le corps de l'abstraction contractée ne contient pas la variable liée, l'argument 1 disparaît. On en déduit intuitivement que ce sous-terme n'intervient pas dans le résultat final. La dernière réduction est une déréférence de l'adresse m_1. La valeur associée à m_1 dans la mémoire est lue et donne la valeur finale de la réduction. Deux observations sont retenues de cet exemple : (1) le sous-terme 1 de M n'a pas contribué à la valeur finale. (2) L'adresse m_1 a participé à l'obtention de la valeur finale. Comme on l'a vu dans la section précédente, pour exprimer une propriété de non-interférence sur une réduction, il faut identifier les sous-termes du terme initial qui influencent la valeur finale. Si un sous-terme n'influence pas cette valeur, c'est-à-dire si la modification de ce sous-terme ne change pas l'observable de la valeur finale, alors ce sous-terme n'interfère dans cette valeur. Intuitivement, l'observation (1) nous pousse à émettre l'hypothèse suivante : le sous-terme 1 n'interfère pas dans la réduction $\mathcal{R}$.

Termes	$M,N,P \in \mathbf{\Lambda}$	$::= x$	Variable
		$\mid \lambda x.M$	Abstraction
		$\mid MN$	Application
		$\mid n$	Entier
		$\mid M + N$	Addition
		$\mid \texttt{ifz}\ M\ \texttt{then}\ N\ \texttt{else}\ P$	Branchement
		$\mid \texttt{ref}(M)$	Référence
		$\mid M{:=}N$	Affectation
		$\mid\ !M$	Déréférence
		$\mid m$	Adresse
		$\mid ()$	*Unit*
Valeurs	$V,W \in \mathbf{V}$	$::= \lambda x.M \mid n \mid m \mid ()$	
Mémoire	$\mu \in \mathcal{M}$	$\in\ \mathbf{M} \to \mathbf{V}$	

FIG. 5.3 – *Syntaxe du λ_m-calcul*

(β_m)	$(\lambda x.M)V/\mu \to M\{x\backslash V\}/\mu$
(Plus)	$\dfrac{\mathbf{n} + \mathbf{n'} = \mathbf{n''}}{n + n'/\mu \to n''/\mu}$
(Ifz-true)	$\dfrac{\mathbf{n} = 0}{\texttt{ifz}\ n\ \texttt{then}\ M\ \texttt{else}\ N/\mu \to M/\mu}$
(Ifz-false)	$\dfrac{\mathbf{n} \neq 0}{\texttt{ifz}\ n\ \texttt{then}\ M\ \texttt{else}\ N/\mu \to N/\mu}$
(Ref)	$\dfrac{m \notin dom(\mu)}{\texttt{ref}(V)/\mu \to m/(\mu; m \mapsto V)}$
(Assign)	$m{:=}V/(\mu; m \mapsto V') \to ()/(\mu; m \mapsto V)$
(Deref)	$!m/(\mu; m \mapsto V) \to V/(\mu; m \mapsto V)$
(Ctx)	$\dfrac{R/\mu \to R'/\mu'}{E[R]/\mu \to E[R']/\mu'}$

FIG. 5.4 – *Réduction* $\to$

$$
\begin{aligned}
x\{x\backslash V\} &= V \\
y\{x\backslash V\} &= y \quad \text{si } x \neq y \\
(MN)\{x\backslash V\} &= M\{x\backslash V\}N\{x\backslash V\} \\
(\lambda x.M)\{x\backslash V\} &= \lambda x.M \\
(\lambda y.M)\{x\backslash V\} &= \lambda z.(M\{y \leftarrow z\}\{x\backslash V\}) \quad \text{où } z = \mathsf{Conv}_\alpha(x,y,M,V) \\
\texttt{ref}(M)\{x\backslash V\} &= \texttt{ref}(M\{x\backslash V\}) \\
(!M)\{x\backslash V\} &= !(M\{x\backslash V\}) \\
(M_1{:=}M_2)\{x\backslash V\} &= M_1\{x\backslash V\}{:=}M_2\{x\backslash V\} \\
m\{x\backslash V\} &= m \\
()\{x\backslash V\} &= () \\
\texttt{ifz } M \texttt{ then } N \texttt{ else } P\{x\backslash V\} &= \texttt{ifz } M\{x\backslash V\} \texttt{ then } N\{x\backslash V\} \texttt{ else } P\{x\backslash V\}
\end{aligned}
$$

FIG. 5.5 – *Substitution dans le λ_m-calcul*

$$E[\,] ::= [\,] \mid E[\,]\,N \mid V\,E[\,] \mid E[\,]{:=}N \mid V{:=}E[\,] \mid\, !E[\,] \mid \texttt{ifz } E[\,] \texttt{ then } M \texttt{ else } N$$

FIG. 5.6 – *Les contextes d'évaluation du λ_m-calcul*

On éprouve notre hypothèse en considérant la réduction de $M' = (\lambda y.(\lambda\_.!y)\,\texttt{ref}(1))\,\texttt{ref}(0)$ qui est obtenu à partir du terme M en remplaçant le sous-terme 1 par $\texttt{ref}(1)$. Cette réduction est représentée sur la figure 5.8. La première étape de réduction est similaire à l'exemple précédent. Une adresse m_2 est ajoutée à la mémoire. Le fait que le choix du nom de l'adresse n'est pas spécifié fait que ce nom m_2 est potentiellement différent de m_1. Ensuite, le terme est réduit par (β_m)-réduction qui correspond à la deuxième étape de réduction de $\mathcal{R}$. La troisième réduction ajoute une nouvelle adresse m_3 à la mémoire. Enfin, les deux réductions finales correspondent aux dernières réductions de $\mathcal{R}$. On obtient la même valeur finale 0 ce qui tend à confirmer la non-interférence des sous-termes 1 de M et $\texttt{ref}(1)$ de M', On observe que la réduction $\mathcal{R}'$ ressemble à la réduction $\mathcal{R}$. Seul un pas de réduction causé par le sous-terme $\texttt{ref}(1)$ a été inséré dans $\mathcal{R}'$. Il est clair que les adresses m_1 et m_2 jouent le même rôle dans les deux réductions. Pourtant ces adresses ne portent pas, a priori, le même nom. Ceci illustre bien le fait que le choix du nom d'une adresse n'a pas d'importance, de la même manière que pour le nom de la variable liée par une abstraction. Par une opération similaire à l'α-conversion, on pourrait changer l'adresse m_2 en m_1 dans la réduction $\mathcal{R}$. Nous verrons dans la section suivante qu'on opte pour une solution plus commode : le nom des adresses est choisi

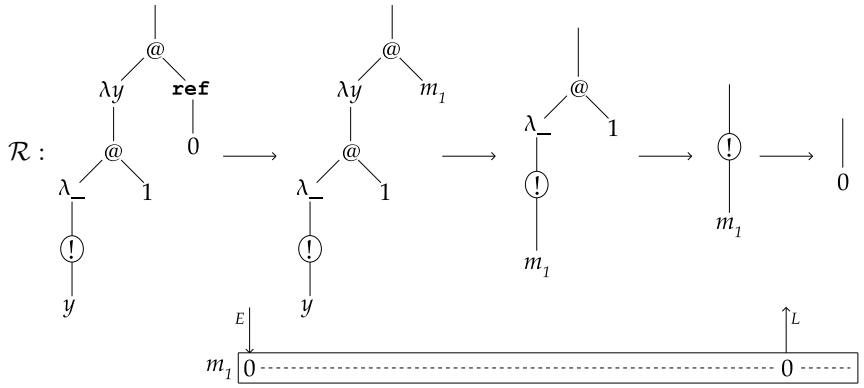


FIG. 5.7 – *Réduction de $M = (\lambda y.(\lambda\_.!y)\,1)\,\texttt{ref}(0)$*

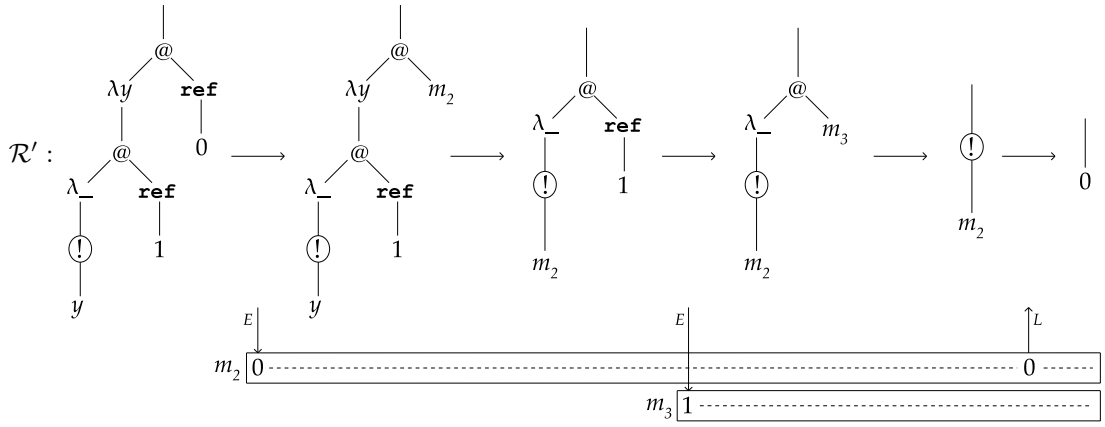


FIG. 5.8 – *Réduction de* $M' = (\lambda y.(\lambda\_.!y)\ \texttt{ref}(1))\ \texttt{ref}(0)$

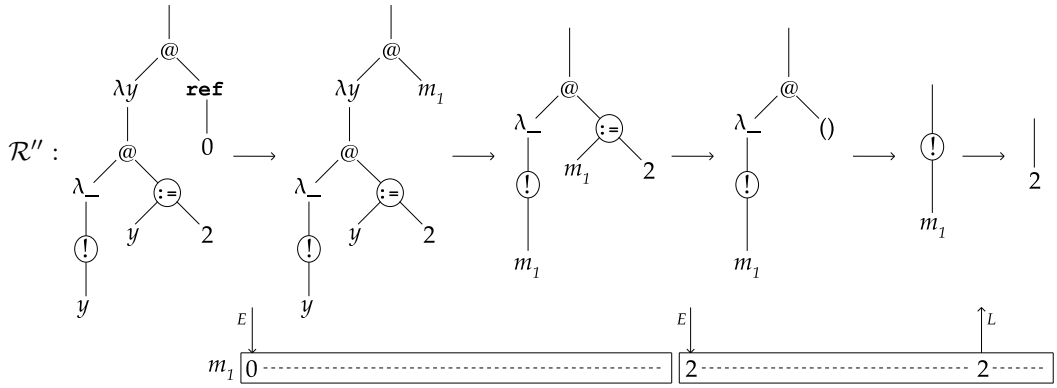
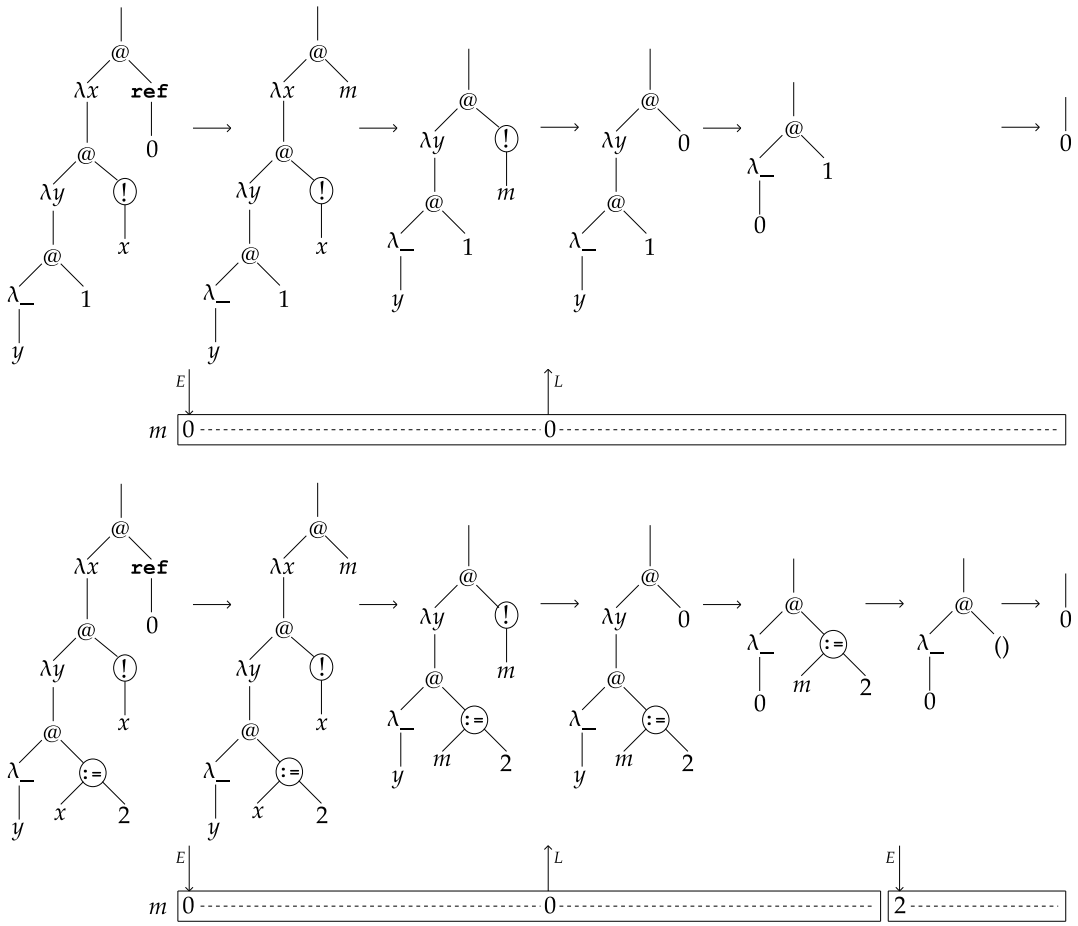


FIG. 5.9 – *Réduction de* $M'' = (\lambda y.(\lambda\_.!y)\ y{:=}2)\ \texttt{ref}(0)$

en fonction de la structure du terme au moment de la réduction (Ref). Comme les termes M et M' sont similaires (en particulier, les chemins qui mènent à la référence réduite $\texttt{ref}(0)$ sont identiques) l'adresse choisie sera la même dans les deux cas. Ceci simplifiera la comparaison entre deux réductions.

A la lumière des conclusions de la section 5.1, on pourrait penser qu'on peut obtenir un préfixe de M dans lequel seuls les sous-termes qui interfèrent, c'est-à-dire participent à l'obtention de la valeur finale, seraient conservés. Tout terme minoré par ce préfixe et qui se réduit vers une valeur, se réduirait, comme M, vers 0. Il s'avère que cette hypothèse est fausse comme le montre la réduction du terme $M'' = (\lambda y.(\lambda\_.!y)\ y{:=}2)\texttt{ref}(0)$ que l'on a obtenu à partir de M en remplaçant le sous-terme 1, supposé ne pas interférer, par $y{:=}2$. Cette réduction est représentée sur la figure 5.9. On observe que les deux premières réductions de $\mathcal{R}''$ sont similaires aux premières réductions de $\mathcal{R}$. La troisième réduction est une affectation $m_1{:=}2$ qui modifie la valeur associée à m_1 en mémoire. Bien entendu, le sous-terme () n'influence pas le résultat puisqu'il disparaît lors de la réduction suivante. En revanche, cette affectation modifie la mémoire ; c'est par ce biais que le résultat final est modifié. A partir de cet instant, les valeurs associées à m_1 dans les mémoires des réductions $\mathcal{R}$ et $\mathcal{R}''$ diffèrent. Au moment de la réduction de la déréférence $!m_1$, cette différence de mémoire entraîne des valeurs finales différentes. Le fait d'avoir remplacé le sous-terme 1 de M par $y{:=}2$ dans M' a donc modifié la valeur finale, sans que le sous-terme () ne soit intervenu. En comparant l'utilisation de m_1 dans $\mathcal{R}$ et $\mathcal{R}''$, on observe que l'écriture et la lecture effectuées au cours de $\mathcal{R}$ se retrouvent bien dans $\mathcal{R}''$. Le problème réside, en réalité, dans le fait qu'une écriture supplémentaire



$$M_1 \;=\; (\lambda y.(\lambda x.(\lambda\_.x)1)!y)\,\texttt{ref}(0)$$
$$M_2 \;=\; (\lambda y.(\lambda x.(\lambda\_.x)(y{:=}2))!y)\,\texttt{ref}(0)$$

FIG. 5.10 – *Réductions comparées de M_1 et M_2*

a lieu entre ces deux instants dans $\mathcal{R}''$. La valeur lue au moment de la déréférence a été modifiée par cette deuxième écriture, ce qui explique les valeurs finales différentes. Intuitivement, l'interférence introduite dans cet exemple n'est pas de la même nature que l'interférence "fonctionnelle" présentée dans la section 5.1 : ici l'interférence n'est pas due à un sous-terme mais à un effet sur une adresse.

On utilise un dernier exemple pour préciser la notion d'interférence sur la mémoire introduite dans l'exemple précédent. Dans cette optique, on compare pas à pas les réductions des termes $M_1 = (\lambda y.(\lambda x.(\lambda\_.x)1)!y)\texttt{ref}(0)$ et $M_2 = (\lambda y.(\lambda x.(\lambda\_.x)(y{:=}2))!y)\texttt{ref}(0)$. Le terme M_2 est le terme M_1 où l'on a remplacé le sous-terme 1 par $y{:=}2$. Ces réductions sont représentées sur la figure 5.10. Les premières réductions de M_1 et M_2 sont similaires. Dans un premier temps, l'adresse m est ajoutée à la mémoire. La valeur 0 lui est associée. Puis une (β_m)-réduction est effectuée : l'adresse m (en argument) est substituée à x. Cette adresse intervient donc dans la déréférence $!m$ et dans l'affectation $m{:=}2$ (dans la réduction de M_2). L'étape suivante consiste à réduire cette déréférence $!m$. La valeur associée à m, à savoir 0 dans les deux cas, est lue. La réduction suivante est, dans les deux cas, une (β_m)-réduction. La cinquième réduction constitue le moment où les deux réductions divergent. Dans la réduction de M_2, l'affectation $m{:=}2$ est réduite, ce qui modifie la valeur associée à m en mémoire. Puis, dans les deux réductions, une (β_m)-réduction permet d'obtenir la valeur finale 0 qui est issue de la lecture de l'adresse m.

La figure 5.10 montre que l'utilisation de l'adresse m est différente dans les deux réductions. Mais ces utilisations différentes n'engendrent pas de différences dans la valeur finale. En effet, la valeur finale 0 est issue de la réduction de la déréférence $!m$. Au moment de cette lecture en mémoire, la valeur associée à m est la même dans les deux réductions. La modification ultérieure, dans la réduction de M_2 ne modifie donc pas le résultat final. Seule la valeur associée à m entre l'écriture initiale et la lecture contribue au résultat. Dans la réduction de M_2, la deuxième écriture a lieu après cette lecture, eq qui ne perturbe donc pas le résultat. On est donc amené à considérer des *intervalles de temps* pour l'utilisation de la mémoire. Ces intervalles sont délimités par une écriture et une lecture en mémoire. Si on reprend l'exemple de la réduction $\mathcal{R}$, on observe que l'intervalle de m_1 qui contribue à la valeur finale est délimité par l'écriture initiale et la lecture. Au cours de la réduction $\mathcal{R}''$, une écriture a lieu dans cet intervalle, ce qui explique qu'on n'obtienne pas la même valeur finale. Cette notion d'intervalle est donc fondamentale : intuitivement, une adresse n'influence le résultat final que pendant un certain intervalle de temps : entre l'écriture et la lecture d'une valeur qui contribue au résultat final.

Dans le cadre du λ-calcul ou du λ-calcul par valeur, la démarche pour obtenir la propriété de non-interférence pour un terme M donné, consiste à obtenir un préfixe de ce terme dont tous les sous-termes participent au résultat. A contrario, les sous-termes de M qui n'apparaissent pas dans ce préfixe n'influencent pas la valeur finale : ils *n'interfèrent pas*. Dans le cadre présent, ce préfixe ne suffit pas. En plus de l'interférence fonctionnelle déjà étudiée dans la section précédente, il existe, du fait de la présence des effets de bord, une deuxième façon d'interférer : l'interférence sur la mémoire. Comme le montre l'exemple de la réduction $\mathcal{R}''$, une écriture qui intervient au cours d'un intervalle participant à l'obtention de la valeur finale, entraîne une modification de cette dernière. Pour obtenir la non-interférence dans le cadre du λ_m-calcul, il faut donc déterminer, en plus du préfixe mentionné précédemment, les intervalles des adresses qui contribuent au résultat.

5.3 Le λ_m-calcul étiqueté

Dans cette section, nous introduisons le λ_m-calcul étiqueté pour exprimer une propriété de non-interférence. Comme annoncé dans la section précédente, si un terme M se réduit vers une valeur V, les étiquettes du λ_m-calcul doivent permettre de déterminer (1) un préfixe de M qui contient les sous-termes de M qui interfèrent, c'est-à-dire contribuent à l'obtention de V, et (2) l'ensemble des intervalles des adresses qui interfèrent. La syntaxe des termes et des valeurs du λ_m-calcul étiqueté est décrite sur la figure 5.11 : les termes du λ_m-calcul sont munis d'une étiquette. En plus des variables, abstractions et applications du λ-calcul étiqueté, on ajoute les entiers N^α, une addition $(M+N)^\alpha$ et un test à zéro $(\texttt{ifz}\ M\ \texttt{then}\ N\ \texttt{else}\ P)^\alpha$. On retrouve les termes associés à la mémoire. Les adresses m représentent les emplacements de la mémoire, la référence $(\texttt{ref}(M))^\alpha$ permet d'ajouter une nouvelle adresse à la mémoire, l'affectation $(M{:=}N)^\alpha$ permet de modifier la valeur associée à une adresse et la déréférence $(!M)^\alpha$ donne accès à la valeur associée à une adresse. Le terme *Unit* est utilisé pour retourner un résultat vide après une affectation. La syntaxe est également enrichie d'un terme Ω qui représente intuitivement la limite d'un préfixe. Comme dans la section précédente, les valeurs sont les abstractions, les entiers, les adresses ou *Unit*.

Contrairement aux chapitres 1 et 2, dans le λ_m-calcul, le rôle des étiquettes n'est pas d'obtenir une propriété de stabilité. Ici, la propriété visée est la non-interférence. Comme on l'a vu dans la section précédente, dans un langage en appel par valeur comme le λ-calcul par valeur ou le λ_m-calcul, la propriété de non-interférence ne coïncide pas avec la propriété de stabilité. Plus précisément, dans le cas du λ-calcul par valeur, la propriété de non-interférence coïncide avec la propriété de stabilité du λ-calcul général. C'est pourquoi, bien que le λ_m-calcul est un langage en

Termes	$M,N,P \in \mathbf{\Lambda}$	$::= x^\alpha$	Variable
		$\mid (\lambda x.M)^\alpha$	Abstraction
		$\mid (MN)^\alpha$	Application
		$\mid n^\alpha$	Entier
		$\mid (M+N)^\alpha$	Addition
		$\mid (\texttt{ifz}\ M\ \texttt{then}\ N\ \texttt{else}\ P)^\alpha$	Branchement
		$\mid (\texttt{ref}(M))^\alpha$	Référence
		$\mid (M{:=}N)^\alpha$	Affectation
		$\mid (!M)^\alpha$	Déréférence
		$\mid m^\alpha$	Adresse
		$\mid ()^\alpha$	Unit
		$\mid \Omega$	Préfixe
Valeurs	$V,W \in \mathbf{V}$	$::= (\lambda x.M)^\alpha \mid n^\alpha \mid m^\alpha \mid ()^\alpha$	

FIG. 5.11 – *Syntaxe des termes et des valeurs du λ_m-calcul étiqueté*

appel par valeur, les étiquettes du λ_m-calcul, dont la syntaxe est décrite sur la figure 5.12, sont plus proches des étiquettes du λ-calcul que des étiquettes du λ-calcul par valeur. Comme dans les sections précédentes, une étiquette peut être une lettre a ou une concaténation $\alpha\beta$ de deux étiquettes α et β. Une étiquette peut aussi être une étiquette surlignée $\lceil\alpha\rceil^{\mathtt{x}}$ ou soulignée $\lfloor\alpha\rfloor^{\mathtt{x}}$. Le surlignement (respectivement le soulignement) de α signifie intuitivement que cette étiquette a été créée par le haut (resp. le bas) par la contraction d'un radical de nom α. Cette création est décrite par la *précision* portée en exposant par le surligné ou le souligné : l'étiquette peut être créée par une β-réduction (`b`), par une addition (`p`), par un conditionnel (`i`), par une référence (`c`) ou par une affectation (`w`). Au moment de la contraction d'un terme de la forme $(n^\alpha + n'^\beta)^\gamma$, les origines des termes gauche et droit sont conservées de façon indépendante, dans l'étiquette juxtaposée $\alpha|\beta$. Un *intervalle* $[m,\varphi,\varphi']$ enregistre l'utilisation d'une adresse. Il s'agit d'un triplet constitué de l'adresse impliquée et de deux chemins. Nous montrerons par la suite, dans le théorème 5.4, que la propriété d'irréversibilité des chemins est valable dans le λ_m-calcul étiqueté. Cette propriété implique que certains chemins peuvent être interprétés comme des dates. Ainsi, les chemins φ et φ' correspondent aux dates d'écriture et de lecture de l'adresse m. Un ensemble d'intervalles est appelé *utilisation-mémoire*. Une utilisation-mémoire contient typiquement l'ensemble des intervalles des adresses qui ont contribué au résultat d'une réduction. On utilise la notation $C(B)$ pour obtenir l'ensemble des chemins présents dans une utilisation-mémoire B.

$$C(\{[m_i,\varphi_i,\varphi'_i]\}_{i\in\{1\ldots n\}}) = \bigcup_{i=1}^{n}\{\varphi_i,\varphi'_i\}$$

Les *nœuds* permettent de caractériser la traversée des nœuds sur un parcours d'un chemin issu de la racine. La traversée d'une abstraction est notée avec le nœud λ ; la traversée vers le membre gauche (respectivement droit) d'une application est notée avec le nœud $@_1$ (resp. $@_2$). De même, les nœuds $+_i$ et $:=_i$ sont utilisés pour les chemins qui traversent une addition ou une affectation ($+_1$ et $:=_1$ pour le membre gauche, $+_2$ et $:=_2$ pour le membre droit). Les nœuds `ref` et ! interviendront dans les chemins qui traversent, respectivement, une référence et une déréférence. Dans le cas du branchement, $\texttt{ifz}_1$ est le nœud associé au terme à tester ; $\texttt{ifz}_2$ et $\texttt{ifz}_3$ sont respectivement les nœuds associés au terme de succès et au terme d'échec. Comme précédemment, un chemin est intuitivement la succession des étiquettes et des nœuds rencontrés pendant le parcours depuis la racine vers un nœud de l'arbre de syntaxe associé à un terme. Deux types de chemins sont utilisés. Un chemin-contexte peut être un chemin vide, noté $\bot$, ou une suite alternée d'étiquettes et de

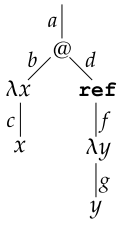
Etiquettes	$\alpha,\beta \in \mathbf{E}$	$::= a$	Lettre
		$\mid \alpha\beta$	Concaténation
		$\mid \lceil\alpha\rceil^{\mathtt{x}}$	Surlignement
		$\mid \lfloor\alpha\rfloor^{\mathtt{x}}$	Soulignement
		$\mid \alpha\vert\beta$	Juxtaposition
		$\mid [m,\varphi,\varphi']$	Intervalle
Précision	$\mathtt{x}$	$::= \mathtt{b} \mid \mathtt{p} \mid \mathtt{i} \mid \mathtt{c} \mid \mathtt{w}$	
Utilisation-mémoire	B	$::= \{[m_i,\varphi_i,\varphi'_i]\}_{i=1\ldots n}$	
Nœud	$\theta \in \mathbf{N}$	$::= \lambda \mid @_i \mid +_i \mid \mathtt{ref} \mid :=_i \mid ! \mid \mathtt{ifz}_j$	$i \in \{1,2\}$ et $j \in \{1,2,3\}$
Chemin-contexte	$\kappa \in \mathbf{K}$	$::= \alpha_1\theta_1\alpha_2\theta_2\ldots\alpha_n\theta_n$	$n \in \mathbb{N}$
Chemin	$m,\varphi,\psi \in \mathbf{\Phi}$	$::= \alpha\theta_1\alpha_1\theta_2\alpha_2\ldots\theta_n\alpha_n$	$n \in \mathbb{N}$

FIG. 5.12 – *Syntaxe des étiquettes et des chemins du λ_m-calcul*

nœuds commençant par une étiquette et finissant par un nœud. Un chemin est une suite alternée commençant et finissant par une étiquette. Comme précédemment, on utilise librement la notation de concaténation pour mettre bout à bout deux suites alternées. Un chemin-contexte correspond à la succession d'étiquettes et de nœuds rencontrés sur le parcours de l'arbre de syntaxe d'un contexte $C[\,]$ menant de la racine au trou de $C[\,]$. Cette correspondance est formalisée par la fonction $\sigma(C[\,])$ qui retourne le chemin-contexte correspondant à $C[\,]$.

$$
\begin{aligned}
\sigma([\,]) &= \bot & \qquad \sigma((C[\,]{:=}N)^\alpha) &= \alpha{:=}_1\sigma(C[\,])\\
\sigma((\lambda x.C[\,])^\alpha) &= \alpha\lambda\sigma(C[\,]) & \qquad \sigma((M{:=}C[\,])^\alpha) &= \alpha{:=}_2\sigma(C[\,])\\
\sigma((C[\,]N)^\alpha) &= \alpha@_1\sigma(C[\,]) & \qquad \sigma((\mathtt{ifz}\ C[\,]\ \mathtt{then}\ N\ \mathtt{else}\ P)^\alpha) &= \alpha\mathtt{ifz}_1\sigma(C[\,])\\
\sigma((MC[\,])^\alpha) &= \alpha@_2\sigma(C[\,]) & \qquad \sigma((\mathtt{ifz}\ M\ \mathtt{then}\ C[\,]\ \mathtt{else}\ P)^\alpha) &= \alpha\mathtt{ifz}_2\sigma(C[\,])\\
\sigma((\mathtt{ref}(C[\,]))^\alpha) &= \alpha\mathtt{ref}\sigma(C[\,]) & \qquad \sigma((\mathtt{ifz}\ M\ \mathtt{then}\ N\ \mathtt{else}\ C[\,])^\alpha) &= \alpha\mathtt{ifz}_3\sigma(C[\,])\\
\sigma((!C[\,])^\alpha) &= \alpha!\sigma(C[\,])
\end{aligned}
$$

Par contraste, un chemin relie la racine à l'étiquette d'un nœud sans contenir le nœud. Nous verrons par la suite qu'une adresse m est, en réalité, un chemin. Certains ensembles sont distingués. On note $\mathbf{\Phi}_\theta$ l'ensemble des chemins qui arrivent sur un nœud θ. Plus formellement, on pose $\mathbf{\Phi}_\theta = \{\varphi \in \mathbf{\Phi} \mid \varphi = \psi\theta\alpha\}$. On définit le prédécesseur $\mathsf{Pre}(\varphi)$ de chemin φ. C'est le plus grand chemin strictement contenu dans φ. Plus formellement, si $\varphi = \varphi'\theta\alpha$, alors $\mathsf{Pre}(\varphi) = \varphi'$. Pour illustrer la notion de chemin, on considère le terme $M = ((\lambda x.x^c)^b(\mathtt{ref}((\lambda y.y^g)^f))^d)^a$ dont le contexte d'évaluation est $E[\,] = ((\lambda x.x^c)^b[\,])^a$.



Le chemin-contexte associé à ce contexte d'évaluation est $\sigma(E[\,]) = a@_2$. Le chemin menant au radical $R = (\mathtt{ref}((\lambda y.y^g)^f))^d$ de M est $\varphi = a@_2d$. Le contexte $C[\,] = (\mathtt{ref}((\lambda y.[\,])^f))^d$ est un contexte de R. Le chemin-contexte associé est $\sigma(C[\,]) = d\mathtt{ref}f\lambda$. De la même manière que les

$$
\begin{array}{rcl@{\qquad}rcl}
\alpha \cdot x^{\beta} & = & x^{\alpha\beta} & \alpha \cdot (\lambda x.M)^{\beta} & = & (\lambda x.M)^{\alpha\beta} \\
\alpha \cdot (MN)^{\beta} & = & (MN)^{\alpha\beta} & \alpha \cdot (N+M)^{\beta} & = & (N+M)^{\alpha\beta} \\
\alpha \cdot n^{\beta} & = & n^{\alpha\beta} & \alpha \cdot (\texttt{ifz}\ M\ \texttt{then}\ N\ \texttt{else}\ P)^{\beta} & = & (\texttt{ifz}\ M\ \texttt{then}\ N\ \texttt{else}\ P)^{\alpha\beta} \\
\alpha \cdot (\texttt{ref}(M))^{\beta} & = & (\texttt{ref}(M))^{\alpha\beta} & \alpha \cdot (M{:=}N)^{\beta} & = & (M{:=}N)^{\alpha\beta} \\
\alpha \cdot (!M)^{\beta} & = & (!M)^{\alpha\beta} & \alpha \cdot ()^{\beta} & = & ()^{\alpha\beta} \\
\alpha \cdot m^{\beta} & = & m^{\alpha\beta} & \alpha \cdot \Omega & = & \Omega
\end{array}
$$

FIG. 5.13 – *Définition de la fonction de concaténation* “ $\cdot$ ”

contextes peuvent s'imbriquer, on utilise la concaténation pour composer les chemins-contextes. Le chemin-contexte associé à $E[C[\,]]$ est $\sigma(E[C[\,]]) = \sigma(E[\,])\sigma(C[\,]) = a@_2 d\texttt{ref} f \lambda$. Au moment de la définition des réductions, nous verrons que nous associerons à M *le chemin* $\varphi_0 = a@_2 d\texttt{ref} f$ *menant au cœur du radical* R. On remarque en particulier $\mathsf{Pre}(\varphi_0) = \varphi$ et $\psi \in \mathbf{\Phi}_{\texttt{ref}}$. La relation de préfixe sur les chemins qui a été introduite dans la partie 1.3 s'étend de façon élémentaire aux chemins considérés ici.

$$
\begin{array}{l}
\kappa \preceq \kappa' \iff \exists \kappa'' \in \mathbf{K} \,.\, \kappa\kappa'' = \kappa' \\
\kappa \prec \varphi \iff \exists \varphi' \in \mathbf{\Phi} \,.\, \kappa\varphi' = \varphi \\
\varphi \prec \kappa \iff \exists (\kappa', \theta) \in \mathbf{K} \times \mathbf{N} \,.\, \varphi\theta\kappa' = \kappa \\
\varphi \preceq \varphi' \iff \varphi = \varphi' \text{ ou } \exists (\varphi'', \theta) \in \mathbf{\Phi} \times \mathbf{N} \,.\, \varphi\theta\varphi'' = \varphi'
\end{array}
$$

La relation $\preceq$ est un ordre bien fondé sur $\mathbf{K} \cup \mathbf{\Phi}$ dont le plus petit élément est $\bot$.

Dans le λ_m-calcul étiqueté, on adapte la définition de la mémoire, afin de déterminer les dates d'écriture et de lecture des adresses.

$$\text{Mémoire} \qquad \mu \in \Phi \rightarrow \mathbf{\Phi} \times \mathbf{V} \qquad \Phi \subseteq \mathbf{\Phi} \text{ et } \Phi \text{ fini}$$

Comme annoncé précédemment, les adresses sont, dans le λ_m-calcul étiqueté, des chemins. Une mémoire est une fonction finie qui associe à une adresse $m \in \Phi$, un couple (φ, V) constitué d'un chemin φ et d'une valeur V. Cette dernière est bien entendu la valeur associée en mémoire à m. Le chemin φ correspond à la date d'écriture de V en mémoire. Comme pour le λ_m-calcul, pour une bonne lisibilité, une mémoire μ peut s'écrire $\mu = \{m_i \mapsto (\varphi_i, V_i)\}_{i \in I}$ ou bien $\mu = \{m_i \overset{\varphi_i}{\longmapsto} V_i\}_{i \in I}$. Si μ est définie sur $\Phi \cup \{m\}$ avec $\mu(m) = (\varphi, V)$ et si sa restriction sur Φ est μ_0, alors on pourra écrire μ sous la forme du produit tensoriel $\mu = (\mu_0; m \overset{\varphi}{\longmapsto} V)$. La mémoire vide sera notée $\emptyset$. Une *configuration* est un couple constitué d'un terme M et d'une mémoire μ. Intuitivement, cette dernière contient les valeurs associées aux adresses présentes dans M. Cette configuration est notée M/μ.

La réduction étiquetée du λ_m-calcul est décrite sur la figure 5.14. Pour alléger les notations, nous utilisons le même symbole que pour la réduction sans étiquettes. La réduction $\rightarrow$ est définie à l'aide de la réduction $\overset{\kappa}{\rightarrow}$ pour laquelle κ est le chemin-contexte associé au contexte du radical. Les définitions de ces réductions s'appuient sur les définitions de la fonction “ $\cdot$ ” de concaténation (sur la figure 5.16), de l'étiquette de tête τ (sur la figure 5.13) et de la substitution (sur la figure 5.15). La fonction $|\alpha|$ qui fournit les lettres présentes dans l'étiquette α est adaptée simplement du λ-calcul étiqueté ; sa définition est donnée sur la figure 5.17.

Comme annoncé précédemment, la réduction étiquetée du λ_m-calcul vise la propriété de non-interférence et non la stabilité. De ce fait, bien que la stratégie d'évaluation choisie pour le λ_m-calcul est en appel par valeur, la réduction $\rightarrow$ s'inspire davantage de la réduction étiquetée du λ-calcul

$$(\beta_{me}) \qquad ((\lambda x.M)^\alpha V)^\beta/\mu \xrightarrow{\kappa} \beta \cdot \lceil\alpha\rceil^{\mathtt{b}} \cdot M\{x\backslash\lfloor\alpha\rfloor^{\mathtt{b}} \cdot V\}/\mu$$

$$(\mathrm{Plus}_e) \qquad \frac{\mathbf{n_1} + \mathbf{n_1} = \mathbf{n}}{(n_1^\alpha + n_2^\beta)^\gamma/\mu \xrightarrow{\kappa} n^{\gamma\lceil\alpha|\beta\rceil^{\mathtt{p}}}/\mu}$$

$$(\text{Ifz-true}_e) \qquad \frac{\mathbf{n} = 0}{(\mathtt{ifz}\ n^\alpha\ \mathtt{then}\ M\ \mathtt{else}\ N)^\beta/\mu \xrightarrow{\kappa} \beta \cdot \lceil\alpha\rceil^{\mathtt{i}} \cdot M/\mu}$$

$$(\text{Ifz-false}_e) \qquad \frac{\mathbf{n} \neq 0}{(\mathtt{ifz}\ n^\alpha\ \mathtt{then}\ M\ \mathtt{else}\ N)^\beta/\mu \xrightarrow{\kappa} \beta \cdot \lceil\alpha\rceil^{\mathtt{i}} \cdot N/\mu}$$

$$(\mathrm{Ref}_e) \qquad \frac{m = \kappa\beta \qquad \alpha = \tau(V) \qquad \varphi = \kappa\beta\mathtt{ref}\alpha}{(\mathtt{ref}(V))^\beta/\mu \xrightarrow{\kappa} m^{\lceil\beta\rceil^{\mathtt{c}}}/(\mu; m \overset{\varphi}{\longmapsto} V)}$$

$$(\mathrm{Assign}_e) \qquad \frac{\varphi' = \kappa\beta{:=_1}\alpha}{(m^\alpha{:=}V)^\beta/(\mu; m \overset{\varphi}{\longmapsto} V') \xrightarrow{\kappa} ()^{\lceil\beta\rceil^{\mathtt{w}}}/(\mu; m \overset{\varphi'}{\longmapsto} V)}$$

$$(\mathrm{Deref}_e) \qquad \frac{\varphi' = \kappa\beta!\alpha}{(!m^\alpha)^\beta/(\mu; m \overset{\varphi}{\longmapsto} V) \xrightarrow{\kappa} \beta \cdot [m,\varphi,\varphi'] \cdot V/(\mu; m \overset{\varphi}{\longmapsto} V)}$$

$$(\mathrm{Ctx}_e) \qquad \frac{R/\mu \xrightarrow{\sigma(E[\,])} R'/\mu'}{E[R]/\mu \to E[R']/\mu'}$$

FIG. 5.14 – *Réductions* $\to$ *et* $\xrightarrow{\kappa}$ $(\kappa \in \mathbf{K})$

$$\begin{aligned}
x^\alpha\{x\backslash V\} &= \alpha \cdot V \\
y^\alpha\{x\backslash V\} &= y^\alpha \\
n^\alpha\{x\backslash V\} &= n^\alpha \\
(\lambda x.M)^\alpha\{x\backslash V\} &= (\lambda x.M)^\alpha \\
(\lambda y.M)^\beta\{x\backslash V\} &= (\lambda z.M\{y \leftarrow z\}\{x\backslash V\})^\beta \text{ où } z = \mathsf{Conv}_\alpha(x,y,M,V) \\
(MN)^\alpha\{x\backslash V\} &= (M\{x\backslash V\}N\{x\backslash V\})^\alpha \\
(\mathtt{ref}(M))^\alpha\{x\backslash V\} &= (\mathtt{ref}(M\{x\backslash V\}))^\alpha \\
!(M)^\alpha\{x\backslash V\} &= (M\{x\backslash N\})^\alpha \\
(M_1{:=}M_2)^\alpha\{x\backslash V\} &= (M_1\{x\backslash V\}M_2\{x\backslash V\})^\alpha \\
m^\alpha\{x\backslash V\} &= m^\alpha \\
()^\alpha\{x\backslash V\} &= ()^\alpha \\
(\mathtt{ifz}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P)^\alpha\{x\backslash V\} &= (\mathtt{ifz}\ M\{x\backslash V\}\ \mathtt{then}\ N\{x\backslash V\}\ \mathtt{else}\ P\{x\backslash V\})^\alpha
\end{aligned}$$

FIG. 5.15 – *Définition de la substitution par une valeur*

$$\begin{aligned}
\tau(x^\alpha) &= \alpha & \tau((\lambda x.M)^\alpha) &= \alpha \\
\tau((MN)^\alpha) &= \alpha & \tau((N+M)^\alpha) &= \alpha \\
\tau(n^\alpha) &= \alpha & \tau((\mathtt{ifz}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P)^\alpha) &= \alpha \\
\tau((\mathtt{ref}(M))^\alpha) &= \alpha & \tau((M{:=}N)^\alpha) &= \alpha \\
\tau((!M)^\alpha) &= \alpha & \tau(()^\alpha) &= \alpha \\
\tau(m^\alpha) &= \alpha
\end{aligned}$$

FIG. 5.16 – *Définition de la fonction* τ *d'étiquette de tête*

$$\begin{aligned}
|a| &= \{a\} & |\lfloor\alpha\rfloor^{\mathtt{x}}| &= |\alpha| & |\lceil\alpha\rceil^{\mathtt{x}}| &= |\alpha| \\
|\alpha\beta| &= |\alpha| \cup |\beta| & |[m,\varphi,\varphi']| &= |m| \cup |\varphi| \cup |\varphi'| & |\alpha|\beta| &= |\alpha| \cup |\beta| \\
& & |\varphi\theta\alpha| &= |\varphi| \cup |\alpha|
\end{aligned}$$

FIG. 5.17 – *Lettres présentes dans une étiquette ou un chemin*

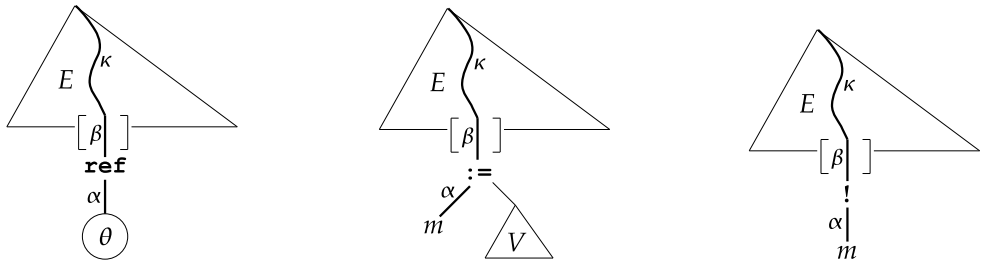
$$E[\,] := [\,] \mid (E[\,]N)^{\alpha} \mid (V\,E[\,])^{\alpha} \mid (E[\,]{:=}N)^{\alpha} \mid (V{:=}E[\,])^{\alpha} \mid (!E[\,])^{\alpha} \mid (\texttt{ifz}\ E[\,]\ \texttt{then}\ M\ \texttt{else}\ N)^{\alpha}$$

FIG. 5.18 – *Contexte d'évaluation du λ_m-calcul étiqueté*

que celle du λ-calcul par valeur. De ce fait, l'étiquette de tête de la valeur intervenant dans la β_{me}-réduction n'intervient pas dans cette réduction. Cette étiquette sera visible dans le résultat final uniquement si la valeur V est effectivement utilisée. Dans le cas de la règle (Plus_e), les deux entiers interviennent effectivement dans le résultat. C'est pourquoi les étiquettes, qui représentent les histoires de ces entiers, sont *juxtaposées* dans le résultat. Pour les règles (Ifz-true_e) et (Ifz-false_e), une étiquette $\lceil\alpha\rceil^{\mathtt{i}}$ est créée pour enregistrer l'histoire de l'entier qui a commandé la δ-règle conditionnelle. La règle (Ctx_e) permet d'imposer l'ordre d'évaluation qui correspond aux contextes d'évaluation décrits sur la figure 5.18. Parmi les chemins des termes, certains seront plus particulièrement utilisés dans la suite de la section. Si $M = E[R]$ et $M/\mu \to M'/\mu'$, le **chemin menant au radical contracté** entre M et M' est défini par $\varphi_r = \sigma(E[\,])\tau(R)$.

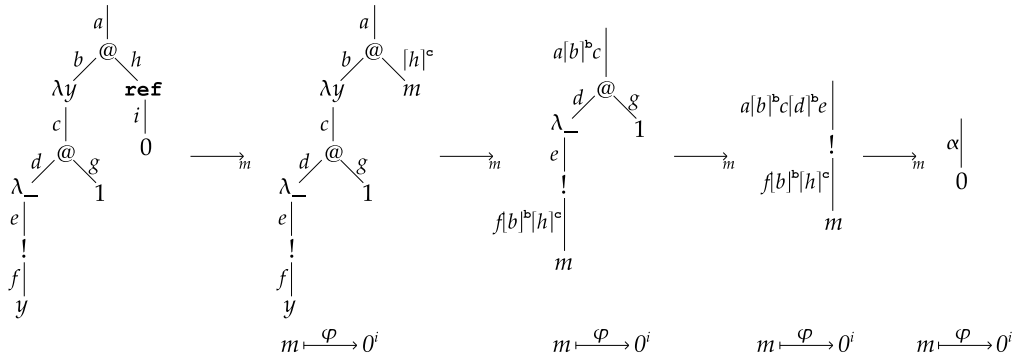
Dans les règles de réductions qui manipulent la mémoire, en plus d'identifier les sous-termes qui participent au résultat, les étiquettes enregistrent les intervalles associés aux adresses. La règle (Ref_e) réduit le terme $(\texttt{ref}(V))^{\alpha}$ et ajoute une nouvelle adresse m à la mémoire μ. L'adresse choisie $m = \kappa\beta$ correspond au chemin menant au radical. Ce choix est *structurel* : il ne dépend que du terme à réduire. Ce choix est aussi *correct* : comme on le montrera plus tard dans le théorème 5.4, cette adresse n'appartient pas au domaine de la mémoire (sous certaines hypothèses raisonnables). L'adresse m retournée est étiquetée par $\lceil\beta\rceil^{\mathtt{c}}$. En effet, le sous-terme m ne dépend pas de la mémoire, ni de la valeur V. Les sous-termes qui ont contribué à obtenir cette adresse sont donc les mêmes que ceux qui ont contribué à obtenir le radical $(\texttt{ref}(V))^{\beta}$. La valeur V et le chemin $\varphi = \kappa\beta\texttt{ref}\alpha$ sont associés à m en mémoire. En ce qui concerne l'interférence de la mémoire, deux informations doivent être enregistrées dans le cas de cette réduction : (1) l'étiquette α qui permet d'identifier l'ensemble des sous-termes ayant contribué à créer le radical qui effectue l'écriture (interférence fonctionnelle) et (2) la date de l'écriture de V dans l'adresse m (interférence de la mémoire). Comme on l'a vu dans la section 1.3, le chemin $\kappa\beta$ menant au radical peut être considéré comme une date, puisque, du fait de la propriété d'irréversibilité des chemins, ce chemin ne peut réapparaître dans la suite de la réduction. Par conséquent, le chemin φ peut aussi être considéré comme une date. Ce chemin est une façon concise de garder les deux informations à enregistrer. Ce chemin donne accès à l'étiquette α et au chemin menant au radical par $\mathsf{Pre}(\varphi) = \kappa\beta$. Dans le cas de cette réduction, l'étiquette α enregistre à la fois les sous-termes ayant contribué à obtenir la valeur V mais aussi les sous-termes ayant contribué à l'obtention du radical. En conséquence, cette étiquette est gardée à la fois dans le chemin φ associé à m en mémoire et dans l'étiquette de tête de V. On pourrait sans doute se contenter du chemin $\kappa\beta$ à la place de φ. Nous avons préféré utiliser φ par souci d'uniformité avec les réductions (Assign_e) et (Deref_e) présentées ci-dessous.

La règle (Assign_e) réduit l'affectation $(m^{\alpha}{:=}V)^{\beta}$. Le terme *Unit* retourné est étiqueté par $\lceil\beta\rceil^{\mathtt{w}}$. En effet, le résultat de l'affectation ne dépend ni de m ni de V. Les sous-termes qui ont contribué à obtenir () sont bien les mêmes que ceux qui ont contribué à obtenir le radical $(m^{\alpha}{:=}V)^{\beta}$. La valeur V et le chemin $\varphi = \kappa\beta{:=}_1\alpha$ sont associés en mémoire à l'adresse m. Comme pour la règle (Ref_e), deux informations concernant l'interférence mémoire doivent être enregistrées dans le cas de cette réduction : (1) l'étiquette α qui permet d'identifier l'ensemble des sous-termes ayant contribué à créer le radical qui effectue l'écriture (interférence fonctionnelle) et (2) la date de l'écriture de V dans l'adresse m (interférence de la mémoire). Comme précédemment, le chemin φ, qui peut être considéré comme une date, est une façon concise de garder les deux informations à enregistrer. Ce



Cas (Ref) : $\varphi = \kappa\beta\mathtt{ref}\alpha$ Cas (Assign) : $\varphi = \kappa\beta{:=}_1\alpha$ Cas (Deref) : $\varphi = \kappa\beta!\alpha$

FIG. 5.19 – *Chemin menant au cœur du radical pour* $\kappa = \sigma(E[\,])$



$$
\begin{aligned}
m &= a@_2h\\
\varphi &= a@_2h\mathtt{ref}i\\
\varphi' &= a\lceil b\rceil^{\mathtt{b}}c\lceil d\rceil^{\mathtt{b}}e!f\lfloor b\rfloor^{\mathtt{b}}\lceil h\rceil^{\mathtt{c}}\\
\alpha &= a\lceil b\rceil^{\mathtt{b}}c\lceil d\rceil^{\mathtt{b}}e[m,\varphi,\varphi']i
\end{aligned}
$$

FIG. 5.20 – *Réduction de* $M = ((\lambda y.((\lambda\_.(!y^f)^e)^d)^c\ 1^g)^b\ (\mathtt{ref}(0^i))^h)^a$

chemin donne accès à l'étiquette α et au chemin menant au radical par $\mathsf{Pre}(\varphi) = \kappa\beta$.

La règle (Deref$_e$) réduit la déréférence $(!m^\alpha)^\beta$ et lit la valeur associée à l'adresse m en mémoire. Cette valeur dépend de l'utilisation de l'adresse m pendant l'intervalle entre la dernière écriture et cette lecture. L'étiquette intervalle $[m,\varphi,\varphi']$ est donc concaténée en tête de la valeur retournée. Le chemin φ est la dernière date d'écriture enregistrée en mémoire. Comme précédemment, le chemin φ' enregistre une double information : (1) l'étiquette α qui permet d'identifier l'ensemble des sous-termes ayant contribué à créer le radical qui effectue l'écriture (interférence fonctionnelle) et (2) la date de lecture de l'adresse m (interférence de la mémoire). Les chemins introduits dans les règles qui manipulent la mémoire sont appelés *chemins menant au cœur du radical* et sont illustrés sur la figure 5.19.

On illustre plus concrètement ces règles de réduction en reprenant l'exemple du terme suivant.

$$M = ((\lambda y.((\lambda\_.(!y^f)^e)^d)^c\ 1^g)^b\ (\mathtt{ref}(0^i))^h)^a$$

Comme dans les sections précédentes, l'étiquette de tête permet d'obtenir un ensemble de lettres $A = \{a,b,c,d,e,f,h,i\}$. De cet ensemble, on déduit le préfixe P des sous-termes de M qui interfèrent dans la valeur : on a $P = ((\lambda y.((\lambda\_.(!y^f)^e)^d)^c\ \Omega)^b\ (\mathtt{ref}(0^i))^h)^a$. Mais cette étiquette de tête donne aussi l'utilisation-mémoire de la réduction : $B = \{[m,\varphi,\varphi']\}$. Dans la suite de cette partie, on montre

que tout terme préfixé par P qui se réduit *en respectant* B vers une valeur se réduit nécessairement vers 0^α.

5.3.1 Propriétés de la réduction étiquetée

De la définition des règles de réduction, on tire une première remarque sur les chemins associés aux adresses dans la mémoire.

Remarque 5.1 *On considère la réduction $M_0/\mu_0 \to M_1/\mu_1 \to \ldots \to M_n/\mu_n$. Pour i tel que $1 \leq i \leq n$, le chemin menant au radical contracté entre M_{i-1} et M_i est noté φ_r^i. Si $m \in dom(\mu_n)$ et $\mu_n(m) = (\varphi,V)$, deux cas sont possibles.*

1. *$m \in dom(\mu_0)$ et pour tout i tel que $0 \leq i \leq n$, on a $\mu_i(m) = (\varphi,V)$.*
2. *Il existe un indice j tel que $\varphi_r^j = \mathsf{Pre}(\varphi)$, φ est un chemin de M_{j-1} et pour tout i tel que $j \leq i \leq n$, on a $\mu_i(m) = (\varphi,V)$.*

Un chemin est associé en mémoire à une adresse au moment de l'ajout ou d'une affectation d'une adresse. De ce fait, si une adresse m appartient au domaine d'une mémoire issue d'une réduction, alors les scenarii pouvant aboutir à cette situation se classent en deux catégories. (1) L'adresse est présente dans la mémoire initiale et elle n'a pas été modifiée au cours du calcul. (2) L'adresse a été créée et/ou modifiée au cours de la réduction. Dans ce cas, si $\mu(m) = (\varphi,V)$, φ est le chemin menant au cœur du radical de (Ref) ou (Assign) impliqué dans la dernière modification. Et l'adresse m n'est plus modifiée jusqu'à la configuration finale.

Les intervalles jouent un rôle particulier parmi les étiquettes. On utilise les notations $T(M/\mu)$, $T(M)$, $T(\mu)$, $T(\alpha)$ et $T(\varphi)$ pour obtenir l'ensemble des intervalles, respectivement, d'une configuration, d'un terme, d'une mémoire, d'une étiquette et d'un chemin. Cette notation est formellement définie de la façon suivante.

$$
\begin{array}{rcl@{\qquad}rcl}
T(x^\alpha) & = & T(\alpha) & T((\lambda x.M)^\alpha) & = & T(M) \cup T(\alpha) \\
T((MN)^\alpha) & = & T(M) \cup T(N) \cup T(\alpha) & T(n^\alpha) & = & T(\alpha) \\
T((M+N)^\alpha) & = & T(M) \cup T(N) \cup T(\alpha) & T(()^\alpha) & = & T(\alpha) \\
T((\mathtt{ref}(M))^\alpha) & = & T(M) \cup T(\alpha) & T((!M)^\alpha) & = & T(M) \cup T(\alpha) \\
T((M{:=}N)^\alpha) & = & T(M) \cup T(N) \cup T(\alpha) & T(m^\alpha) & = & T(m) \cup T(\alpha) \\
T((\mathtt{ifz}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P)^\alpha) & = & T(M) \cup T(N) \cup T(P) \cup T(\alpha) & T(\Omega) & = & \emptyset \\
\multicolumn{3}{r@{\qquad}}{} & & & \\
T(\{m_i \overset{\varphi_i}{\longmapsto} V_i\}_{i \in I}) & = & \bigcup_{i \in I}(T(\varphi_i) \cup T(V_i)) & T(M/\mu) & = & T(M) \cup T(\mu) \\
\multicolumn{3}{r@{\qquad}}{} & & & \\
T(\mathtt{a}) & = & \emptyset & T(\alpha\beta) & = & T(\alpha) \cup T(\beta) \\
T(\lceil\alpha\rceil^{\mathtt{x}}) & = & T(\alpha) & T(\lfloor\alpha\rfloor^{\mathtt{x}}) & = & T(\alpha) \\
T([m,\varphi,\varphi']) & = & \{[m,\varphi,\varphi']\} \cup T(m) \cup T(\varphi) \cup T(\varphi') & T(\alpha|\beta) & = & T(\alpha) \cup T(\beta) \\
\multicolumn{3}{r@{\qquad}}{} & & & \\
T(\varphi\theta\alpha) & = & T(\varphi) \cup T(\alpha) & & &
\end{array}
$$

Cette notation permet d'introduire une deuxième remarque élémentaire portant sur les intervalles.

Remarque 5.2 *On suppose $M_0/\mu_0 \to M_1/\mu_1 \to \ldots \to M_n/\mu_n$. Pour i tel que $1 \leq i \leq n$, le chemin menant au radical contracté entre M_{i-1} et M_i est noté φ_r^i. Si $[m,\varphi,\varphi'] \in T(M_n/\mu_n)$, trois cas sont possibles :*

1. *$[m,\varphi,\varphi'] \in T(M_0/\mu_0)$*
2. *L'adresse m vérifie $m \in dom(\mu_0)$ et $\mu_0(m) = (\varphi,V)$ et il existe un indice $i \in \{1 \ldots n\}$ tel que $\varphi_r^i = \mathsf{Pre}(\varphi')$, φ' est un chemin de M_{i-1}, $\varphi' \in \mathbf{\Phi}_!$ et pour tout j tel que $0 \leq j \leq i$, on a $\mu_j(m) = (\varphi,V)$.*
3. *Il existe deux indices i et i' et une valeur V tels que :*

(a) Le chemin φ est un chemin de M_{i-1} qui appartient à $\mathbf{\Phi}_{\mathtt{ref}} \cup \mathbf{\Phi}_{:=}$ et vérifie $\varphi_r^i = \mathsf{Pre}(\varphi)$. De plus, si $\varphi \in \mathbf{\Phi}_{\mathtt{ref}}$, alors $\varphi_r^i = m$. Et si $\varphi \in \mathbf{\Phi}_{:=}$, alors le chemin φ mène, dans M_{i-1}, à l'adresse m.

(b) Le chemin φ' est un chemin de $M_{i'-1}$ qui appartient à $\mathbf{\Phi}_!$ et qui vérifie $\varphi_r^{i'} = \mathsf{Pre}(\varphi')$. Dans $M_{i'-1}$, le chemin φ' mène à m.

(c) Si j vérifie $i \leq j \leq i'$, alors on a $\mu_j(m) = (\varphi,V)$.

Cette remarque, tirée d'une inspection des règles de réduction, énonce les trois origines possibles d'un intervalle présent dans une configuration issue d'une réduction. (1) Il peut provenir de la configuration initiale. (2) Il peut être créé par une déréférence d'une adresse déjà présente dans la configuration initiale. Dans ce cas, le chemin φ' est le chemin menant au cœur d'un radical contracté par la règle (Deref) au cours de la réduction. On en déduit $\varphi' \in \mathbf{\Phi}_!$. (3) L'intervalle peut être créé par une déréférence d'une adresse qui a été modifiée au cours de la réduction. Dans ce cas, φ est le chemin menant au cœur du radical contracté au moment de l'écriture de m. Deux règles de réduction peuvent modifier une référence : (i) si m est modifiée par (Ref), alors $\varphi \in \mathbf{\Phi}_{\mathtt{ref}}$; (ii) si m est modifiée par (Assign), alors $\varphi \in \mathbf{\Phi}_{:=}$ est le chemin qui mène à l'adresse m. Comme dans le cas précédent, φ' est un chemin menant au cœur d'un radical contracté par la règle (Deref) au cours de la réduction. On en déduit $\varphi' \in \mathbf{\Phi}_!$. Comme nous le remarquions plus tôt, l'adresse m n'est pas modifiée entre la date d'écriture φ et la date de lecture φ'.

Comme annoncé précédemment, le langage que nous avons introduit exploite la propriété d'irréversibilité évoquée dans la section 1.3. Cette propriété est conservée dans le cadre présent.

Théorème 5.4 (Irréversibilité) *On suppose $M_0/\mu_0 \to M_1/\mu_1 \to \ldots \to M_n/\mu_n$. Pour i tel que $1 \leq i \leq n$, le chemin menant au radical contracté entre M_{i-1} et M_i est noté φ_r^i. Si $i \leq j$, le chemin φ_r^i ne préfixe aucun chemin de M_j.*

Preuve : On prouve ce résultat de la même façon que pour le théorème 1.2. On utilise en particulier le lemme 5.1 mentionné ci-dessous et qui correspond au lemme 1.2. □

A chaque étape de réduction, le chemin menant au radical disparaît, et ne peut pas réapparaître dans un terme de la suite de la réduction. Cette propriété, qui est bien entendu fausse en l'absence d'étiquettes, exploite la relation $\preceq$ sur les étiquettes que l'on a introduite dans la partie 1.3 et qu'on adapte ci-dessous à la syntaxe des étiquettes du λ_m-calcul.

$\alpha \preceq \alpha$	
$\alpha \preceq \beta$	si $\alpha \preceq \gamma$ et $\gamma \preceq \beta$
$\alpha \preceq \alpha_1\alpha_2$	si $\alpha \preceq \alpha_i$ pour $i \in \{1,2\}$
$\alpha \preceq \lceil\alpha\rceil^{\mathtt{x}}$	pour $\mathtt{x} \in \{\mathtt{b},\mathtt{p},\mathtt{i},\mathtt{c},\mathtt{w}\}$
$\alpha \preceq \lfloor\alpha\rfloor^{\mathtt{x}}$	pour $\mathtt{x} \in \{\mathtt{b},\mathtt{p},\mathtt{i},\mathtt{c},\mathtt{w}\}$
$\alpha \preceq \alpha_1\vert\alpha_2$	si $\alpha \preceq \alpha_i$ pour $i \in \{1,2\}$
$\alpha \preceq [m,\varphi,\varphi']$	si $\alpha \preceq m$ ou $\alpha \preceq \varphi$ ou $\alpha \preceq \varphi'$
$\alpha \preceq \alpha_0\theta_1\alpha_1\cdots\theta_n\alpha_n$	si $\alpha \preceq \alpha_i$ pour $i \in \{0,n\}$

La relation $\preceq$ est un ordre bien fondé sur les étiquettes et les chemins. La relation d'ordre stricte associée $\prec$ est exploitée dans le lemme suivant qui est la propriété essentielle sur laquelle repose le théorème d'irréversibilité.

Lemme 5.1 *1. Si $R/\mu \xrightarrow{\kappa} R'/\mu'$, alors on a $\tau(R) \prec \tau(R')$.*
2. Si $M/\mu \xrightarrow{\kappa} M'/\mu'$, alors on a $\tau(M) \preceq \tau(M')$.

Preuve : Ce résultat se prouve en inspectant les différentes règles de réduction. □

L'étiquette de tête d'un terme réduit contient l'étiquette du terme avant réduction. Si le terme considéré est le radical, alors l'étiquette de tête du contractum contient strictement l'étiquette de tête du radical.

Un corollaire du théorème d'irréversibilité est la correction du choix des adresses. Pour prouver et exploiter ce résultat par la suite, on définit l'invariant suivant.

Invariant 5.1 *La configuration M/μ vérifie l'invariant $\mathcal{L}$, ce que l'on note $\mathcal{L}(M,\mu)$, si et seulement si pour toute réduction $M/\mu \twoheadrightarrow E[(\mathtt{ref}(V))^\alpha]/\mu'$ on a $\sigma(E[\,])\alpha \notin dom(\mu')$.*

Cet invariant signifie qu'au moment de la création d'une adresse, l'adresse choisie est bien *fraîche*, c'est-à-dire qu'elle n'appartient pas au domaine de la mémoire.

Corollaire 5.1 (Correction du choix des adresses) *Pour tout terme M, on a $\mathcal{L}(M,\emptyset)$.*

En partant d'une mémoire initiale vide, à chaque application de la règle (Ref), l'adresse choisie est bien *fraîche* puisqu'elle n'appartient pas au domaine de la mémoire. L'adresse choisie ne dépend que du chemin menant à la racine. Cette technique est intéressante pour deux raisons. D'une part, elle résout le problème de non-reproductibilité des réductions que nous avons évoqué précédemment. D'autre part, cette technique est aussi utile pour comparer les réductions de deux termes M_1 et M_2 qui ne diffèrent que sur un ensemble de sous-termes. Pour surmonter ce problème, Pottier et Simonet ajoutent au langage une construction spécifique $\langle N_1|N_2\rangle$. Avec cette construction, M_1 et M_2 sont codés dans un seul terme et les sous-termes communs de M_1 et M_2 sont factorisés. De ce fait, les adresses créées en dehors des crochets sont partagées par les deux termes. Dans le λ_m-calcul, le choix de l'adresse utilisé offre une solution, de notre point de vue, plus simple : si un radical $(\mathtt{ref}(V))^\beta$ est contracté dans une partie commune à M_1 et M_2, le chemin menant à ce radical est le même dans les deux cas, ce qui signifie que dans les deux réductions, l'adresse créée aura le même nom. On peut donc comparer directement les deux réductions.

L'utilisation du théorème d'irréversibilité et de la remarque 5.2 fournit la propriété suivante sur les intervalles créés au cours d'une réduction.

Lemme 5.2 *On considère la réduction $\mathcal{R} : M_0/\emptyset \to M_1/\mu_1 \to \ldots \to M_n/\mu_n$ où $T(M_0/\emptyset) = \emptyset$. Pour $1 \leq i \leq n$, on appelle φ_r^i le chemin menant au radical contracté entre M_{i-1} et M_i. On pose $\Phi_n = C(T(M_n/\mu_n))$.*

1. *Si φ est un chemin de Φ_n, alors il existe un unique indice i tel que $\varphi_r^i = \mathsf{Pre}(\varphi)$.*
2. *Pour i tel que $1 \leq i \leq n$, il existe au plus un chemin φ de Φ_n qui vérifie $\varphi_r^i = \mathsf{Pre}(\varphi)$.*

Preuve : On montre successivement les deux points.

1. Si $\varphi \in \Phi_n$, la remarque 5.2 permet d'obtenir l'existence d'un indice i tel que $\varphi_r^i = \mathsf{Pre}(\varphi)$. S'il existe un deuxième indice j tel que $1 \leq j \leq n$ et $\varphi_r^j = \mathsf{Pre}(\varphi)$, alors, en utilisant le théorème 5.4, on obtient $i = j$. D'où l'unicité de l'indice.
2. Soient φ et φ' deux chemins de Φ_n qui vérifient $\varphi_r^i = \mathsf{Pre}(\varphi)$ et $\varphi_r^i = \mathsf{Pre}(\varphi')$. En utilisant la remarque 5.2, on obtient un indice j tel que $1 \leq j \leq n$ tel que $\varphi_r^j = \mathsf{Pre}(\varphi)$ et φ est un chemin de M_{j-1}. De même, on obtient un indice j' tel que $1 \leq j' \leq n$, $\varphi_r^{j'} = \mathsf{Pre}(\varphi')$ et φ' est un chemin de $M_{j'-1}$. En utilisant le théorème 5.4, on obtient $j = j' = i$. Les chemins φ et φ' sont donc des chemins d'un même terme qui vérifient $\varphi_r^i = \mathsf{Pre}(\varphi) = \mathsf{Pre}(\varphi')$. On en déduit donc $\varphi = \varphi'$. □

Au cours d'une réduction $\mathcal{R}$ partant d'une configuration sans intervalle et de mémoire vide, pour tout chemin φ présent dans un intervalle de la configuration finale, il correspond un unique chemin menant à un radical R contracté au cours de $\mathcal{R}$; plus précisément φ mène au cœur de R. Inversement, un chemin menant à un radical φ_r^i ne peut être associé par Pre qu'à au plus un chemin de $C(T(M_n/\mu_n))$. Ce résultat prouve que les chemins menant au cœur d'un radical sont liés de façon injective avec les chemins menant aux radicaux. Ceci signifie que, comme ces derniers, on peut les utiliser comme des *dates* de la réduction.

$$(\beta_{\mathrm{B}}) \qquad ((\lambda x.M)^{\alpha}V)^{\beta}/\mu/\Phi \xrightarrow{\kappa}_B \beta \cdot \lceil\alpha\rceil^{\mathtt{b}} \cdot M\{x \backslash \lfloor\alpha\rfloor^{\mathtt{b}} \cdot V\}/\mu/\Phi$$

$$(\mathrm{Plus}_{\mathrm{B}}) \qquad \frac{\mathbf{n_1} + \mathbf{n_2} = \mathbf{n}}{(n_1^{\alpha} + n_2^{\beta})^{\gamma}/\mu/\Phi \xrightarrow{\kappa}_B n^{\gamma\lceil\alpha|\beta\rceil^{\mathtt{p}}}/\mu/\Phi}$$

$$(\text{Ifz-true}_{\mathrm{B}}) \qquad \frac{\mathbf{n} = 0}{(\mathtt{ifz}\ n^{\alpha}\ \mathtt{then}\ M\ \mathtt{else}\ N)^{\beta}/\mu/\Phi \xrightarrow{\kappa}_B \beta \cdot \lceil\alpha\rceil^{\mathtt{i}} \cdot M/\mu/\Phi}$$

$$(\text{Ifz-false}_{\mathrm{B}}) \qquad \frac{\mathbf{n} \neq 0}{(\mathtt{ifz}\ n^{\alpha}\ \mathtt{then}\ M\ \mathtt{else}\ N)^{\beta}/\mu/\Phi \xrightarrow{\kappa}_B \beta \cdot \lceil\alpha\rceil^{\mathtt{i}} \cdot N/\mu/\Phi}$$

$$(\mathrm{Ref}^{o}_{\mathrm{B}}) \qquad \frac{\varphi = \kappa\beta\mathtt{ref}\alpha \qquad B(m,\varphi) = \emptyset \qquad m = \kappa\beta \qquad \alpha = \tau(V)}{(\mathtt{ref}(V))^{\beta}/\mu/\Phi \xrightarrow{\kappa}_B m^{\lceil\beta\rceil^{\mathtt{c}}}/(\mu; m \overset{\varphi}{\longmapsto} V)/\Phi}$$

$$(\mathrm{Ref}^{i}_{\mathrm{B}}) \qquad \frac{\varphi = \kappa\beta\mathtt{ref}\alpha \qquad B(m,\varphi) \neq \emptyset \qquad m = \kappa\beta \qquad \alpha = \tau(V)}{(\mathtt{ref}(V))^{\beta}/\mu/\Phi \xrightarrow{\kappa}_B m^{\lceil\beta\rceil^{\mathtt{c}}}/(\mu; m \overset{\varphi}{\longmapsto} V)/\Phi \cup \{\varphi\}}$$

$$(\mathrm{Deref}^{o}_{\mathrm{B}}) \qquad \frac{\varphi' = \kappa\beta!\alpha \qquad [m,\varphi,\varphi'] \notin B}{(!m^{\alpha})^{\beta}/(\mu; m \overset{\varphi}{\longmapsto} V)/\Phi \xrightarrow{\kappa}_B \beta \cdot [m,\varphi,\varphi'] \cdot V/(\mu; m \overset{\varphi}{\longmapsto} V)/\Phi}$$

$$(\mathrm{Deref}^{i}_{\mathrm{B}}) \qquad \frac{\varphi' = \kappa\beta!\alpha \qquad [m,\varphi,\varphi'] \text{ est } (B,\Phi)\text{-actif}}{(!m^{\alpha})^{\beta}/(\mu; m \overset{\varphi}{\longmapsto} V)/\Phi \xrightarrow{\kappa}_B \beta \cdot [m,\varphi,\varphi'] \cdot V/(\mu; m \overset{\varphi}{\longmapsto} V)/\Phi \cup \{\varphi'\}}$$

$$(\mathrm{Assign}^{o}_{\mathrm{B}}) \qquad \frac{\begin{array}{c}\forall t \in B(m,\varphi)\ .\ t \text{ est } (B,\Phi)\text{-inactif} \\ \varphi' = \kappa\beta{:=_1}\alpha \qquad B(m,\varphi') = \emptyset\end{array}}{(m^{\alpha}{:=}V)^{\beta}/(\mu; m \overset{\varphi}{\longmapsto} V')/\Phi \xrightarrow{\kappa}_B ()^{\lceil\beta\rceil^{\mathtt{w}}}/(\mu; m \overset{\varphi'}{\longmapsto} V)/\Phi}$$

$$(\mathrm{Assign}^{i}_{\mathrm{B}}) \qquad \frac{\begin{array}{c}\forall t \in B(m,\varphi)\ .\ t \text{ est } (B,\Phi)\text{-inactif} \\ \varphi' = \kappa\beta{:=_1}\alpha \qquad B(m,\varphi') \neq \emptyset\end{array}}{(m^{\alpha}{:=}V)^{\beta}/(\mu; m \overset{\varphi}{\longmapsto} V')/\Phi \xrightarrow{\kappa}_B ()^{\lceil\beta\rceil^{\mathtt{w}}}/(\mu; m \overset{\varphi'}{\longmapsto} V)/\Phi \cup \{\varphi'\}}$$

$$(\mathrm{Ctx}_{\mathrm{B}}) \qquad \frac{M/\mu/\Phi \xrightarrow{\sigma([\,])}_B M'/\mu'/\Phi'}{E[M]/\mu/\Phi \rightarrow_B E[M']/\mu'/\Phi'}$$

FIG. 5.21 – *Réduction* $\rightarrow_B$ *et* $\xrightarrow{\kappa}_B$ $(\kappa \in \mathbf{K})$

5.3.2 Réduction respectant une utilisation-mémoire

On montre dans cette section que les étiquettes employées dans le λ_m-calcul donnent une information correcte de l'utilisation de la mémoire. Pour ce faire, on définit une nouvelle réduction $\rightarrow_B$ paramétrée par une utilisation-mémoire B. Cette réduction impose la conformité de l'utilisation des adresses vis-à-vis des intervalles présents dans B. Intuitivement, si $[m,\varphi,\varphi'] \in B$ et si la date φ est passée, alors l'adresse m appartient au domaine de la mémoire. Et toute modification de la valeur associée à m est interdite tant que la date φ' n'est pas passée.

Plus concrètement, la réduction $\rightarrow_B$ porte sur des configurations étendues $M/\mu/\Phi$ constituées d'un terme M, d'une mémoire μ et d'un ensemble de chemin Φ appelé *passé*. Cet ensemble Φ contient les chemins de B qui sont intervenus précédemment dans la réduction. En reprenant l'analogie temporelle, cet ensemble contient effectivement les dates d'écriture et de lecture d'adresses qui sont déjà passées. Les chemins de B qui n'apparaissent pas dans Φ peuvent être interprétés comme des dates futures. On définit la notion d'**intervalle** (B,Φ)***-actif*** qui sera utilisée par la suite : $[m,\varphi,\varphi']$ est (B,Φ)-actif si et seulement si $[m,\varphi,\varphi'] \in B$, $\varphi \in \Phi$ et $\varphi' \notin \Phi$. Si un intervalle n'est pas (B,Φ)-actif, il est (B,Φ)-inactif. Par souci de concision, on utilise dans les règles de réduction la notation suivante : $B(m,\varphi) = \{[m,\varphi,\varphi'] \in B\}$. Les règles de réductions de $\rightarrow_B$ et $\xrightarrow{\kappa}_B$ sont données par la figure 5.21. Les réductions (β_{B}), $(\mathrm{Plus}_{\mathrm{B}})$, $(\mathrm{Ifz}_{\mathrm{B}})$ et $(\mathrm{Ctx}_{\mathrm{B}})$ sont inchangées : elles ne modi-

fient pas le passé Φ. L'ensemble B contient les intervalles qui décrivent les utilisations de certaines adresses. La réduction $\xrightarrow{\kappa}_B$ contraint au respect de ces utilisations. Plus concrètement, en reprenant l'analogie temporelle évoquée précédemment, si $[m,\varphi,\varphi'] \in B$, l'adresse m est protégée entre les dates φ et φ'. Si $[m,\varphi,\varphi']$ est (B,Φ)-actif, c'est-à-dire si $\varphi \in \Phi$ et $\varphi' \notin \Phi$, l'adresse m ne peut subir une affectation. Cette protection est assurée par la condition "$\forall t \in B(m,\varphi).t$ est (B,Φ)-inactif" des règles ($\text{Assign}^i_{\text{B}}$) et ($\text{Assign}^o_{\text{B}}$). Par ailleurs, pour chacun des termes faisant intervenir la mémoire, il existe deux règles de réduction qui traitent séparément les cas $B(m,\varphi) = \emptyset$ et $B(m,\varphi) \neq \emptyset$. Plaçons-nous dans le cas d'une référence $(\texttt{ref}(V))^\beta$. Si $B(m,\varphi) \neq \emptyset$, alors le chemin φ qui mène au cœur du radical intervient dans un intervalle $t = [m,\varphi,\varphi']$ de B. La règle (Ref^i_{B}) s'applique. Ce chemin est ajouté au passé Φ. La date de début de protection de m est donc passée. En d'autres termes, l'intervalle t devient (B,Φ)-actif et l'adresse m ne peut donc plus être modifiée avant que le chemin φ' n'apparaisse. Si $B(m,\varphi) = \emptyset$, aucune adresse n'est protégée à partir de la date φ. Cette date n'est donc pas une date d'intérêt : elle n'est pas ajoutée à Φ. Le cas de l'affectation est similaire au cas de la référence. Considérons le cas de la déréférence. Comme précédemment, si $[m,\varphi,\varphi'] \notin B$, alors la date φ' ne libère aucune adresse. Le passage de cette date n'a donc pas d'effet. La règle ($\text{Deref}^o_{\text{B}}$) n'ajoute donc pas φ' à Φ. Si $[m,\varphi,\varphi'] \in B$, le chemin φ' correspond à une date de fin de protection de l'adresse m. Cette date est donc ajoutée par la règle ($\text{Deref}^i_{\text{B}}$) à Φ et l'intervalle $[m,\varphi,\varphi']$ devient inactif. On dira que M/μ se réduit vers N/ν en respectant B si et seulement si $M/\mu/\emptyset \xrightarrow{\kappa}\!\!\!\!\rightarrow_B N/\nu/\Phi$.

Intuitivement, la définition des règles de calcul qui font intervenir la mémoire est telle que seuls les chemins présents dans B sont ajoutés au passé. Cette *spécialisation* des règles vis-à-vis de B a pour conséquence de doubler les règles pour la référence, l'affectation et la déréférence. On aurait pu éviter cette complexité en ajoutant systématiquement les chemins au passé, en ne faisant pas la distinction de l'appartenance du chemin dans B. La raison de ce choix, a priori plus complexe, est que nous souhaitons comparer les réductions de deux termes : si on reprend l'exemple utilisé précédemment, nous voulons comparer les réductions $\mathcal{R}$ et $\mathcal{R}'$. Cette dernière réduction produit un effet de bord supplémentaire par rapport à $\mathcal{R}$, en créant une adresse m_2. Si nous avions décidé d'ajouter systématiquement tous les chemins associés aux opérations sur la mémoire au passé, les passés associés aux réductions $\mathcal{R}$ et $\mathcal{R}'$ auraient divergé. Ceci aurait rendu la comparaison des réductions moins aisée. Par contraste, nous avons choisi de n'ajouter au passé que des chemins significatifs, c'est-à-dire des chemins correspondant aux intervalles présents dans B. Ce choix simplifie la synchronisation entre les réductions $\mathcal{R}$ et $\mathcal{R}'$ car les passés de ces réductions sont comparables.

On utilise un invariant pour formaliser certaines intuitions sur les intervalles que nous avons évoquées précédemment.

Invariant 5.2 *L'ensemble B vérifie l'invariant $\mathcal{B}$, ce que l'on note $\mathcal{B}(B)$, si et seulement si les propriétés suivantes sont vérifiées :*

1. *Si $[m,\varphi,\varphi'] \in B$, alors $\varphi \in \mathbf{\Phi}_{\texttt{ref}} \cup \mathbf{\Phi}_{:=}$ et $\varphi' \in \mathbf{\Phi}_!$.*
2. *Si $[m,\varphi,\psi] \in B$ et $[m',\varphi,\psi'] \in B$, alors $m = m'$.*
3. *Si $[m,\varphi,\psi] \in B$ et $[m',\varphi',\psi] \in B$, alors $m = m'$ et $\varphi = \varphi'$.*

Le premier point de l'invariant indique qu'une date d'écriture d'un intervalle est nécessairement un chemin dont le dernier nœud est une référence ou une affectation. De même, la date de lecture est nécessairement un chemin dont le dernier nœud est une déréférence. Ce point exprime bien l'interprétation des intervalles donnée précédemment : φ (respectivement φ') est un chemin menant au cœur d'un radical de référence ou d'affectation (resp. déréférence). Le deuxième point indique que le premier chemin d'un intervalle est lié de façon injective avec l'adresse de l'intervalle. Le troisième point indique que le deuxième chemin est lié de façon injective avec le premier chemin et

l'adresse de l'intervalle. Ces deux derniers points formalisent l'intuition selon laquelle un chemin menant au cœur d'un radical correspond de façon univoque à une date du calcul. Comme une date ne peut pas survenir deux fois, un chemin ne peut pas être lié à deux adresses différentes.

Bien entendu, cet invariant n'est pas vrai en général puisque les éléments de B peuvent être arbitraires. Cependant, cet invariant est vrai si les éléments de B ont été créés par un calcul, comme le montre le résultat suivant.

Lemme 5.3 *Soit M un terme tel que $T(M) = \emptyset$. Si $M/\emptyset \twoheadrightarrow V/\nu$ et $B = T(V/\nu)$ alors $\mathcal{B}(B)$.*

Preuve : Le premier point de $\mathcal{B}$ est obtenu directement à l'aide de la remarque 5.2. La réduction de M à M' peut s'écrire $M/\emptyset = M_0/\emptyset \to M_1/\mu_1 \to \ldots \to M_n/\mu_n = M'/\mu'$. Pour $i \in \{1 \ldots n\}$, on note φ_r^i le chemin menant au radical contracté entre M_{i-1} et M_i. On suppose $[m,\varphi,\psi] \in B$. Comme $T(M) = \emptyset$, on obtient, en utilisant la remarque 5.2, deux indices i et j tels que $\varphi = \mathsf{Pre}(\varphi_r^i)$, φ est un chemin de M_{i-1}, $\psi = \mathsf{Pre}(\varphi_r^j)$, ψ est un chemin de M_{j-1} et pour tout $l \in \{i \ldots j\}$, on a $\mu_l(m) = (\varphi,V)$. Si $[m',\varphi,\psi'] \in B$, on obtient de même un indice i' tel que $\varphi = \mathsf{Pre}(\varphi_r^{i'})$ et φ est un chemin de $M_{i'-1}$. En utilisant le lemme 5.4, on obtient $i = i'$. Deux cas sont possibles.

- Si $\varphi \in \mathbf{\Phi}_{\mathtt{ref}}$, la remarque 5.2 implique que $\varphi_r^i = m = m'$.
- Si $\varphi_r^i \in \mathbf{\Phi}_{:=}$, la remarque 5.2 implique que φ est un chemin de M_{i-1} qui mène à la fois à m et m'. De là, on obtient $m = m'$.

Le deuxième point de l'invariant est donc prouvé. Si $[m',\varphi',\psi] \in B$, on obtient, en utilisant la remarque 5.2, deux indices i' et j' tels que $\varphi' = \mathsf{Pre}(\varphi_r^{i'})$, φ' est un chemin de $M_{i'-1}$, $\psi = \mathsf{Pre}(\varphi_r^{j'})$, ψ est un chemin de $M_{j'-1}$ et pour tout $l \in \{i' \ldots j'\}$, on a $\mu_l(m) = (\varphi',V')$. En utilisant le lemme 5.4, on obtient $j = j'$. De là, on obtient $\varphi = \varphi'$. En utilisant le point précédent, on en déduit $m = m'$. □

Si la mémoire de la configuration initiale est vide et si le terme initial ne contient pas d'intervalle, alors les intervalles présents dans le terme final ont été calculés au cours de la réduction. Cette utilisation-mémoire vérifie l'invariant $\mathcal{B}$.

On montre maintenant le résultat central de cette section : les étiquettes du λ_m-calcul donnent une information correcte de l'utilisation de la mémoire. Ce résultat s'énonce formellement de la façon suivante.

Lemme 5.4 *Soit M un terme tel que $T(M) = \emptyset$. Si $M/\emptyset \twoheadrightarrow V/\nu$ et $B = T(\tau(V))$, alors $M/\emptyset/\emptyset \twoheadrightarrow_B V/\nu/C(B)$.*

Preuve : La réduction de M à V peut s'écrire : $M/\emptyset = M_0/\emptyset \to M_1/\mu_1 \to \ldots \to M_n/\mu_n = V/\nu$. L'utilisation du lemme 5.3 prouve $\mathcal{B}(B)$. Pour $i \in \{1 \ldots n\}$, on note φ_r^i le chemin menant au radical contracté entre M_{i-1} et M_i. On définit récursivement les ensembles $\{L_i\}_{i\in\{0\ldots n\}}$ de la façon suivante.

- $L_0 = \emptyset$
- Pour $i \in \{1 \ldots n\}$, $L_i = L_{i-1} \cup \{\varphi \in C(B) \mid \varphi_r^i = \mathsf{Pre}(\varphi)\}$.

La propriété $\mathcal{P}(i)$ est définie de la façon suivante : la propriété $\mathcal{P}(i)$ est vraie si et seulement si $M_0/\emptyset/\emptyset \to_B M_1/\mu_1/\Phi_1 \to_B \ldots \to_B M_i/\mu_i/\Phi_i$ et $\forall j \leq i.\Phi_j = L_j$. On montre par récurrence cette propriété pour $i \in \{0 \ldots n\}$. Cette propriété est trivialement vraie pour $i = 0$. On suppose $\mathcal{P}(i)$. On procède par cas sur la réduction entre M_i et M_{i+1}.

1. Si $M_i = E[(\mathtt{ref}(W))^\beta]$, la réduction considérée est $M_i/\mu_i \to E[\lceil\beta\rceil^{\mathtt{b}}.m]/\mu_{i+1}$ où μ_{i+1} vérifie $\mu_{i+1} = (\mu_i; m \overset{\varphi_0}{\longmapsto} W)$ avec $\alpha = \tau(W)$, $\varphi_0 = \varphi_r^{i+1}\mathtt{ref}\alpha$, et $m = \varphi_r^{i+1}$. Deux cas sont à considérer.
   (a) Si $B(m,\varphi_0) = \emptyset$. Soit ψ un chemin de $C(B)$ qui vérifie $\varphi_r^{i+1} = \mathsf{Pre}(\psi)$. En utilisant le lemme 5.2, on obtient $\psi = \varphi_0$, ce qui contredit $B(m,\varphi_0) = \emptyset$. On en déduit $L_{i+1} = L_i$. En utilisant $\mathcal{P}(i)$, on obtient donc $M_i/\mu_i/\Phi_i \to_B M_{i+1}/\mu_{i+1}/\Phi_{i+1}$ avec $L_{i+1} = \Phi_{i+1}$.

(b) Si $B(m,\varphi_0) \neq \emptyset$, alors on a $\Phi_{i+1} = \Phi_i \cup \{\varphi_0\}$. Soit ψ un chemin de $C(B)$ tel que $\varphi_r^{i+1} = \mathsf{Pre}(\psi)$. De la même façon que précédemment, on montre $\psi = \varphi_0$. De là, on a $L_{i+1} = L_i \cup \{\varphi_0\}$. En utilisant $\mathcal{P}(i)$, on obtient donc $M_i/\mu_i/\Phi_i \rightarrow_B M_{i+1}/\mu_{i+1}/\Phi_{i+1}$ avec $L_{i+1} = \Phi_{i+1}$.

2. Si $M_i = E[(m^\alpha{:=}W)^\beta]$, la réduction considérée est $M_i/\mu_i \rightarrow E[()^{\lceil\beta\rceil^\varpi}]/\mu_{i+1}$ où on utilise les notations $\varphi_1 = \varphi_r^{i+1}{:=}_1\alpha$, $\mu_i = (\nu; m \overset{\varphi_0}{\longmapsto} W_0)$ et $\mu_{i+1} = (\nu; m \overset{\varphi_1}{\longmapsto} W)$. Soit $[m,\varphi_0,\psi]$ un intervalle de $B(m,\varphi_0)$. On veut montrer que $[m,\varphi_0,\psi]$ est (B,Φ_i)-inactif. Comme la mémoire initiale est vide et $T(M_0/0) = \emptyset$, en utilisant la remarque 5.2, on obtient deux indices i_1 et i_2 tels que $\varphi_r^{i_1} = \mathsf{Pre}(\varphi_0)$, $\varphi_r^{i_2} = \mathsf{Pre}(\psi)$ et pour tout $j \in \{i_1,\ldots,i_2\}$, on a $\mu_j(m) = (\varphi_0,W_0')$. De même, comme la mémoire initiale est vide, en utilisant la remarque 5.1, on obtient un indice i_0 tel que $\varphi_r^{i_0} = \mathsf{Pre}(\varphi_0)$ et pour tout $j \in \{i_0,\ldots,i\}$, on a $\mu_j(m) = (\varphi_0,W_0)$. L'égalité $\varphi_r^{i_0} = \varphi_r^{i_1}$ implique, d'après le théorème 5.4, l'égalité $i_0 = i_1$ et donc $W_0' = W_0$. Comme $\mu_{i+1}(m) \neq (\varphi_0,W_0)$, on en déduit $i_2 \leq i$. Comme $\varphi_r^{i_2} = \mathsf{Pre}(\psi)$, par définition de L_{i_2}, on a $\psi \in L_{i_2} \subseteq L_i$. Par hypothèse de récurrence $\mathcal{P}(i)$, on obtient $\psi \in \Phi_i$. L'intervalle $[m,\varphi_0,\psi]$ est donc (B,Φ_i)-inactif. Deux cas sont maintenant à considérer.

(a) Si $B(m,\varphi_1) = \emptyset$, alors $\Phi_{i+1} = \Phi_i$. Soit ψ un chemin de $C(B)$ qui vérifie $\varphi_r^{i+1} = \mathsf{Pre}(\psi)$. En utilisant le lemme 5.2, on obtient $\psi = \varphi_1$, ce qui contredit $B(m,\varphi_1) = \emptyset$. On en déduit $L_{i+1} = L_i$. En utilisant $\mathcal{P}(i)$, on obtient donc $M_i/\mu_i/\Phi_i \rightarrow_B M_{i+1}/\mu_{i+1}/\Phi_{i+1}$ avec $L_{i+1} = \Phi_{i+1}$.

(b) Si $B(m,\varphi_1) \neq \emptyset$, alors $\Phi_{i+1} = \Phi_i \cup \{\varphi_1\}$. Soit $\psi \in C(B)$ tel que $\varphi_r^{i+1} = \mathsf{Pre}(\psi)$. De la même façon que précédemment, on montre $\psi = \varphi_1$. De là, on a $L_{i+1} = L_i \cup \{\varphi_1\}$. En utilisant $\mathcal{P}(i)$, on obtient donc $M_i/\mu_i/\Phi_i \rightarrow_B M_{i+1}/\mu_{i+1}/\Phi_{i+1}$ avec $L_{i+1} = \Phi_{i+1}$.

3. Si $M_i = E[(!m^\alpha)^\beta]$, la réduction considérée est $M_i/\mu_i \rightarrow_B E[\beta \cdot [m,\varphi_0,\varphi_1] \cdot V_0]/\mu_{i+1}$ avec $\varphi_1 = \varphi_r^{i+1}\mathtt{ref}\alpha$, $\mu_i = (\nu; m \overset{\varphi_0}{\longmapsto} W_0)$ et $\mu_{i+1} = \mu_i$. Deux cas sont à considérer.

(a) Si $[m,\varphi_0,\varphi_1] \notin B$, alors $\Phi_{i+1} = \Phi_i$. Soit ψ un chemin de $C(B)$ qui vérifie $\varphi_r^{i+1} = \mathsf{Pre}(\psi)$. En utilisant le lemme 5.2, on obtient $\psi = \varphi_1$. Comme $\varphi_1 \in \mathbf{\Phi}_!$, φ_1 est présent dans B dans un intervalle de la forme $[m',\psi',\varphi_1]$. L'invariant $\mathcal{B}(B)$ donne alors $m' = m$ et $\psi' = \varphi_0$ ce qui apporte une contradiction à l'hypothèse $[m,\varphi_0,\varphi_1] \notin B$. On en déduit $L_{i+1} = L_1$. En utilisant $\mathcal{P}(i)$, on obtient donc $M_i/\mu_i/\Phi_i \rightarrow_B M_{i+1}/\mu_{i+1}/\Phi_{i+1}$ avec $L_{i+1} = \Phi_{i+1}$.

(b) Si $[m,\varphi_0,\varphi_1] \in B$, alors $\Phi_{i+1} = \Phi_i \cup \{\varphi_1\}$. Soit ψ un chemin de $C(B)$ qui vérifie $\varphi_r^{i+1} = \mathsf{Pre}(\psi)$. Comme dans le cas précédent, on obtient $\psi = \varphi_1$. De là, on obtient $L_{i+1} = L_i \cup \{\varphi_1\}$. En utilisant $\mathcal{P}(i)$, on obtient donc $M_i/\mu_i/\Phi_i \rightarrow_B M_{i+1}/\mu_{i+1}/\Phi_{i+1}$ avec $L_{i+1} = \Phi_{i+1}$.

4. Les autres cas sont élémentaires. □

Si une réduction aboutit à une valeur V dont tous les intervalles ont été calculés, alors la même configuration initiale se réduit vers la même configuration finale en respectant les intervalles présents dans l'étiquette de tête de V. On en déduit que les étiquettes donnent une information suffisante sur l'utilisation de la mémoire. Ce résultat de cohérence justifie la dénomination de "réduction respectant une utilisation-mémoire".

5.3.3 Non-interférence

Nous nous penchons dans cette section sur la dernière phase du raisonnement qui aboutit au théorème de non-interférence. On considère ici une configuration $M/\emptyset$ qui se réduit vers une valeur V. Comme dans les parties précédentes consacrées à la propriété de stabilité, on obtient, dans un premier temps, un préfixe P de M à partir de l'étiquette de tête de V. Cette étiquette fournit aussi une utilisation-mémoire $B = T(\tau(V))$. Dans un second temps, on considère un terme N préfixé

par P. On montre que si N se réduit vers V' en *respectant l'utilisation-mémoire* B, alors V et V' admettent un préfixe commun, non réduit à Ω. Ceci implique que les observables de V et V' sont égaux. La définition de l'observable d'une valeur étend la définition introduite dans le section précédente.

$$\begin{array}{rlrl} \mathcal{O}((\lambda x.M)^\alpha) &= (\lambda x.\Omega)^\alpha & \mathcal{O}(n^\alpha) &= n^\alpha \\ \mathcal{O}(m^\alpha) &= m^\alpha & \mathcal{O}(()^\alpha) &= ()^\alpha \end{array}$$

On note que les observables d'une valeur qui n'est pas une abstraction est égale à la valeur. On s'inspire de la définition de la non-interférence introduite dans le cadre du λ-calcul pour adapter cette définition au λ_m-calcul.

Définition 5.3 (Non-interférence) *On suppose $\mathcal{R} : M/\emptyset \twoheadrightarrow V/\mu$ et $B = T(\tau(V))$. Le sous-terme $(C[\,],N)$ du terme M n'interfère pas dans $\mathcal{R}$ si et seulement si, pour tout terme N', la réduction $C[N']/\emptyset/\emptyset \twoheadrightarrow_B V'/\mu/\Phi$ implique la relation $\mathcal{O}(V) = \mathcal{O}(V')$.*

Comme nous l'avons vu au moment de l'étude des exemples de la section 5.2, dans un calcul muni d'effet de bord, les interférences liées à l'utilisation de la mémoire s'ajoutent aux interférences fonctionnelles déjà présentes dans le λ-calcul ou le λ-calcul par valeur. La définition de la non-interférence tient compte de ces deux composantes. Comme dans le λ-calcul, la non-interférence fonctionnelle est capturée par le relation $\mathcal{O}(V) = \mathcal{O}(V')$. Le fait de ne considérer que des réductions qui respectent B permet de prendre en ligne de compte la non-interférence sur la mémoire.

Dans les sections précédentes, on obtenait le préfixe de stabilité de M en effaçant dans M les sous-termes dont l'étiquette de tête contenait des lettres absentes de $A = |\tau(V)|$. Ici, en plus de considérer cet ensemble de lettres, on utilise aussi les intervalles de $B = T(\tau(V))$. Le préfixe de M est obtenu par la fonction $[\![\cdot]\!]_B^A$ qui est définie de la façon suivante.

$$\begin{array}{rcll} [\![\Omega]\!]_B^A &=& \Omega & \\ [\![M]\!]_B^A &=& \Omega \quad \text{où } \tau(M) = \alpha & \text{si } |\alpha| \not\subseteq A \text{ ou } T(\alpha) \not\subseteq B \\ \left.\begin{array}{r} [\![x^\alpha]\!]_B^A \\ [\![(\lambda x.M)^\alpha]\!]_B^A \\ [\![(MN)^\alpha]\!]_B^A \\ [\![()^\alpha]\!]_B^A \\ [\![n^\alpha]\!]_B^A \\ [\![(M+N)^\alpha]\!]_B^A \\ [\![(\mathtt{ref}(M))^\alpha]\!]_B^A \\ [\![(M{:=}N)^\alpha]\!]_B^A \\ [\![(!M)^\alpha]\!]_B^A \\ [\![m^\alpha]\!]_B^A \\ [\![(\mathtt{ifz}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P)^\alpha]\!]_B^A \end{array}\right. & \left.\begin{array}{l} = x^\alpha \\ = (\lambda x.[\![M]\!]_B^A)^\alpha \\ = ([\![M]\!]_B^A\ [\![N]\!]_B^A)^\alpha \\ = () \\ = n^\alpha \\ = ([\![M]\!]_B^A + [\![N]\!]_B^A)^\alpha \\ = (\mathtt{ref}([\![M]\!]_B^A))^\alpha \\ = ([\![M]\!]_B^A{:=}[\![N]\!]_B^A)^\alpha \\ = (!M)^\alpha \\ = m^\alpha \\ = (\mathtt{ifz}\ [\![M]\!]_B^A\ \mathtt{then}\ [\![N]\!]_B^A\ \mathtt{else}\ [\![P]\!]_B^A)^\alpha \end{array}\right. & \text{si } |\alpha| \subseteq A \text{ et } T(\alpha) \subseteq B \end{array}$$

L'effacement d'un sous-terme M dépend de son étiquette de tête α. M est effacé non seulement si α contient des lettres qui sont absentes de A, mais aussi si α contient un triplet absent de B. Par la suite, une étiquette α qui vérifie $|\alpha| \subseteq A$ et $T(\alpha) \subseteq B$ sera dite (A,B)**-compatible**. Pour comparer deux termes, on étend de façon naturelle la relation de préfixe utilisée dans les sections

précédentes.

$$
\begin{array}{rcll}
M & \preccurlyeq & M' & \text{si } M \equiv M' \\
M & \preccurlyeq & M' & \text{si } M \preccurlyeq M'' \text{ et } M'' \preccurlyeq M' \\
\Omega & \preccurlyeq & M & \\
(\lambda x.M)^\alpha & \preccurlyeq & (\lambda x.M')^\alpha & \text{si } M \preccurlyeq M' \\
(MN)^\alpha & \preccurlyeq & (M'N')^\alpha & \text{si } M \preccurlyeq M' \text{ et } N \preccurlyeq N' \\
(M+N)^\alpha & \preccurlyeq & (M'+N')^\alpha & \text{si } M \preccurlyeq M' \text{ et } N \preccurlyeq N' \\
(\mathtt{ref}(M))^\alpha & \preccurlyeq & (\mathtt{ref}(M'))^\alpha & \text{si } M \preccurlyeq M' \\
(M{:=}N)^\alpha & \preccurlyeq & (M'{:=}N')^\alpha & \text{si } M \preccurlyeq M' \text{ et } N \preccurlyeq N' \\
(!M)^\alpha & \preccurlyeq & (!M')^\alpha & \text{si } M \preccurlyeq M' \\
(\mathtt{ifz}\ M\ \mathtt{then}\ N\ \mathtt{else}\ P)^\alpha & \preccurlyeq & (\mathtt{ifz}\ M'\ \mathtt{then}\ N'\ \mathtt{else}\ P')^\alpha & \text{si } M \preccurlyeq M' \text{ et } N \preccurlyeq N' \text{ et } P \preccurlyeq P'
\end{array}
$$

L'opération $[\![\cdot]\!]^A_B$ vérifie des propriétés syntaxiques similaires à celles vérifiées par les fonctions d'effacement considérées dans les sections précédentes.

Lemme 5.5 *1. Si $M \preccurlyeq N$, alors on a $\alpha \cdot M \preccurlyeq \alpha \cdot N$.*
2. Si $M \preccurlyeq N$ et $V \preccurlyeq W$, alors $M\{x\backslash V\} \preccurlyeq N\{x\backslash W\}$.

Preuve : On montre successivement ces propriétés.

1. Ce point se montre de façon élémentaire par cas sur M.
2. On procède par induction sur la structure de M.
   (a) Si $M = \Omega$, alors on a $M\{x\backslash V\} = \Omega$ et le résultat est élémentaire.
   (b) Si $M = x^\alpha$, alors on a $N = x^\alpha$. De là, on obtient les relations $M\{x\backslash V\} = \alpha \cdot V$ et $N\{x\backslash W\} = \alpha \cdot W$. On conclut par le point précédent.
   (c) Si $M = (\lambda y.M')^\alpha$, alors on a $N = (\lambda y.N')^\alpha$ avec $M' \preccurlyeq N'$. De là, on obtient les relations $M\{x\backslash V\} = (\lambda y.M'\{x\backslash V\})^\alpha$ et $N\{x\backslash W\} = (\lambda y.N'\{x\backslash W\})^\alpha$. On conclut par hypothèse d'induction.
   (d) Les autres cas sont similaires aux cas précédents. □

La relation de préfixe est compatible avec les opérations de concaténation et de substitution (à gauche et à droite). On examine ensuite les propriétés élémentaires de la fonction d'effacement.

Lemme 5.6 *1. Si $|\alpha| \subseteq A$, alors $\alpha \cdot [\![M]\!]^A_B = [\![\alpha \cdot M]\!]^A_B$*
2. $[\![M]\!]^A_B\{x\backslash[\![N]\!]^A_B\} = [\![M\{x\backslash N\}]\!]^A_B$

Preuve : On montre successivement ces propriétés.

1. Ce point se montre de façon élémentaire par cas sur M.
2. On procède par induction sur la structure de M.
   (a) Si $M = \Omega$, l'égalité est triviale.
   (b) Si $\tau(M)$ n'est pas (A,B)-compatible, alors $[\![M]\!]^A_B = \Omega$ et l'étiquette $\tau(M\{x\backslash N\})$ n'est pas (A,B)-compatible. On a donc $[\![M\{x\backslash N\}]\!]^A_B = \Omega$.
   (c) Si $\tau(M)$ est (A,B)-compatible, on procède par cas.
      i. Si $M = x^\alpha$, alors $[\![M]\!]^A_B = x^\alpha$ et $[\![M]\!]^A_B\{x\backslash[\![N]\!]^A_B\} = \alpha \cdot [\![N]\!]^A_B$. Par ailleurs, on a $[\![M\{x\backslash N\}]\!]^A_B = [\![\alpha \cdot N]\!]^A_B$. On conclut par le lemme 5.5.
      ii. Si $M = (\lambda y.M')^\alpha$, alors on obtient la relation $[\![M]\!]^A_B = (\lambda y.[\![M']\!]^A_B)^\alpha$. De là, on obtient $[\![M\{x\backslash N\}]\!]^A_B = (\lambda y.[\![M']\!]^A_B\{x\backslash N\})^\alpha$. On conclut par hypothèse d'induction.
      iii. Les autres cas sont similaires aux précédents. □

Le premier point indique que si α est une étiquette (A,B)-compatible, la fonction de concaténation avec α commute avec la fonction préfixe $[\![\cdot]\!]^A_B$. Le deuxième point indique que la fonction préfixe commute avec la fonction de substitution.

Nous nous intéressons maintenant aux deux interprétations que nous avons données d'un intervalle (B,Φ)-actif dans une configuration $M/\mu/\Phi$. (1) L'adresse d'un intervalle (B,Φ)-actif appartient bien au domaine de la mémoire. (2) La valeur associée en mémoire à une adresse d'un intervalle (B,Φ)-actif ne peut être modifiée. Nous énonçons formellement la première interprétation sous la forme de l'invariant suivant.

Invariant 5.3 *Le triplet (μ,Φ,B) vérifie l'invariant $\mathcal{J}$, ce que l'on note $\mathcal{J}(\mu,\Phi,B)$, si et seulement si pour tout triplet $[m,\varphi,\varphi']$ qui est (B,Φ)-actif, on a $m \in dom(\mu)$.*

Cet invariant assure la cohérence entre la mémoire μ, le passé Φ et l'utilisation-mémoire B. Dans le résultat suivant, on montre à la fois que si B vérifie la propriété de régularité $\mathcal{B}$, alors l'invariant $\mathcal{J}$ est préservé et la deuxième interprétation d'un intervalle (B,Φ)-actif est vérifiée.

Lemme 5.7 *Soient B une utilisation-mémoire et M/μ une configuration qui vérifient $\mathcal{B}(B)$ et $\mathcal{L}(M,\mu)$. Si $M/\mu/\Phi \to_B M'/\mu'/\Phi'$ et $\mathcal{J}(\mu,\Phi,B)$, alors $\mathcal{J}(\mu',\Phi',B)$ et si $[m,\varphi,\varphi']$ est (B,Φ)-actif alors $\mu(m) = \mu'(m)$.*

Preuve : Dans la suite de cette preuve, l'hypothèse $\mathcal{J}(\mu,\Phi,B)$ est notée $\mathcal{J}_0$. Le terme M est de la forme $M = E[R]$. On note $\kappa = \sigma(E[\,])$ et φ_r est le chemin menant au radical contracté $\varphi_r = \kappa\tau(R)$. On procède par cas sur la réduction.

1. Si $M = E[(\mathtt{ref}(V))^\beta]$, on a $M/\mu/\Phi \to_B E[m_0^{\lceil\beta\rceil^c}]/(\mu; m_0 \overset{\psi_0}{\longmapsto} V)/\Phi'$ avec $\varphi_r = \kappa\beta$, $m_0 = \varphi_r$ et $\psi_0 = \varphi_r\mathtt{ref}\tau(V)$. Par $\mathcal{L}(M,\mu)$, on obtient $m_0 \notin dom(\mu)$. Deux cas sont à considérer.
   (a) Si $B(m_0,\psi_0) = \emptyset$, alors $\Phi = \Phi'$. Soit $[m,\varphi,\varphi']$ un intervalle (B,Φ)-actif : on a $\varphi \in \Phi$ et $\varphi' \notin \Phi$. De là, on obtient, par $\mathcal{J}_0$, $m \in dom(\mu)$. Ceci implique, d'une part $m \in dom(\mu')$, et d'autre part $m \neq m_0$ et $\mu'(m_0) = \mu(m)$.
   (b) Si $B(m_0,\psi_0) \neq \emptyset$, on a $\Phi' = \Phi \cup \{\psi_0\}$. Soit $[m,\varphi,\varphi']$ un intervalle (B,Φ')-actif : on a $\varphi \in \Phi'$ et $\varphi' \notin \Phi'$. Deux cas sont à considérer :
      – Si $\varphi \in \Phi$, alors $[m,\varphi,\varphi']$ est (B,Φ)-actif. De là, par $\mathcal{J}_0$, on obtient $m \in dom(\mu)$. Ceci implique, d'une part $m \in dom(\mu')$, et d'autre part $m \neq m_0$ et $\mu(m) = \mu'(m)$.
      – Si $\varphi = \psi_0$, comme $B(m_0,\psi_0) \neq \emptyset$, en utilisant l'hypothèse $\mathcal{B}(B)$, on obtient $m = m_0$. De là, $m \in dom(\mu')$.
2. Si $M = E[(m_0^\alpha{:=}V)^\beta]$, on a la réduction $M/\mu/\Phi \to_B E[()^{\lceil\beta\rceil^{\mathtt{w}}}]/\mu'/\Phi'$ avec les notations $\mu = (\mu_0; m_0 \overset{\psi_0}{\longmapsto} V_0)$, $\mu' = (\mu_0; m_0 \overset{\psi_1}{\longmapsto} V)$ et $\psi_1 = \varphi_r{:=}_1\alpha$. Soit $[m,\varphi,\varphi']$ un intervalle qui est (B,Φ')-actif. Deux cas sont à considérer.
   (a) Si $B(m_0,\psi_1) = \emptyset$, alors $\Phi' = \Phi$. De ce fait, $[m,\varphi,\varphi']$ est (B,Φ)-actif. Du fait de la réduction de l'affectation, tout intervalle de $B(m_0,\psi_0)$ est (B,Φ)-inactif. En utilisant $\mathcal{J}_0$, on obtient $m \in dom(\mu) - \{m_0\}$. Ceci implique $m \in dom(\mu')$ et $\mu'(m) = \mu(m)$.
   (b) Si $B(m_0,\psi_1) \neq \emptyset$, on a $\Phi' = \Phi \cup \{\psi_1\}$. Deux cas sont à considérer.
      – Si $\varphi \in \Phi$, alors $[m,\varphi,\varphi']$ est (B,Φ)-actif. De même que précédemment, on obtient $m \in dom(\mu) - \{m_0\}$, ce qui implique $m \in dom(\mu')$ et $\mu'(m) = \mu(m)$.
      – Si $\varphi = \psi_1$, en utilisant l'hypothèse $\mathcal{B}(B)$, on obtient $m = m_0$. De là, on a bien $m_0 \in dom(\mu')$.
3. Si $M = E[(!m_0^\alpha)^\beta]$, on a $M/\mu/\Phi \to_B E[\beta \cdot [m_0,\psi_0,\psi_1] \cdot V]/\mu/\Phi'$ avec $\mu = (\mu_0; m_0 \overset{\psi_0}{\longmapsto} V)$ et $\psi_1 = \varphi_r!\alpha$. En utilisant l'hypothèse $\mathcal{B}(B)$, on montre que dans tous les cas, un intervalle (B,Φ')-actif est nécessairement (B,Φ)-actif, ce qui permet de conclure comme dans les cas précédents.
4. Dans les autres cas, on a $\mu' = \mu$ et $\Phi = \Phi'$, ce qui permet de conclure directement. □

Si m est une adresse impliquée dans un intervalle $[m,\varphi,\varphi']$ (B,Φ)-actif, alors le contenu associé en mémoire à m ne peut être modifié tant que l'intervalle demeure (B,Φ)-actif, c'est-à-dire qu'un

radical de chemin φ' n'a pas été contracté. Si cette contraction a lieu, φ' appartient au passé et $[m,\varphi,\varphi']$ devient inactif. Une adresse active devient donc inactive avant d'être modifiée.

Les chemins présents dans un passé Φ sont, a priori, arbitraires. Mais du fait de la spécialisation vis-à-vis de B des règles de réduction, on observe que les chemins ajoutés à Φ au cours d'une réduction ne peuvent être que des chemins présents dans B. Le résultat suivant exploite cette remarque.

Lemme 5.8 *On suppose $M/\mu/\Phi \to_B M'/\mu'/\Phi'$. Soit φ_r le chemin menant au radical contracté entre M et M'. Si φ_r ne préfixe aucun chemin de $C(B) - \Phi$, alors $\Phi = \Phi'$.*

Preuve : Par l'absurde, on suppose $\Phi \neq \Phi'$. Dans tous les cas de figure, on a $\Phi' = \Phi \cup \{\varphi\}$ où $\varphi \in C(B) - \Phi$ et $\varphi_r = \mathsf{Pre}(\varphi) \prec \varphi =$. Ceci contredit l'hypothèse du lemme. □

Intuitivement, les chemins de B sont les étapes importantes de l'utilisation de la mémoire ; ce sont les étapes qui participent à l'obtention de la valeur finale. Le passé enregistre les étapes importantes de B qui sont survenues. Si pour une réduction, le chemin menant au radical contracté ne préfixe pas un chemin de B qui n'est pas déjà passé, cela signifie que cette réduction n'est pas une de ces étapes importantes. Le passé est donc inchangé.

Pour montrer la propriété de non-interférence, on veut comparer les réductions de deux termes qui aboutissent à deux valeurs et montrer que ces valeurs sont liées l'une à l'autre. Pour cela, on introduit l'invariant suivant dont on montrera la conservation par $\to_B$.

Invariant 5.4 *Soit B une utilisation-mémoire. Un quintuplet (M,μ,Φ,N,ν) vérifie l'invariant $\mathcal{I}_B$, ce que l'on note $\mathcal{I}_B(M,\mu,\Phi,N,\nu)$, si et seulement si*

(I1) $M/\mu/\Phi \twoheadrightarrow_B V/\mu'/\Phi'$ *avec* $A = |\tau(V)|$ *et* $B = T(\tau(V))$.

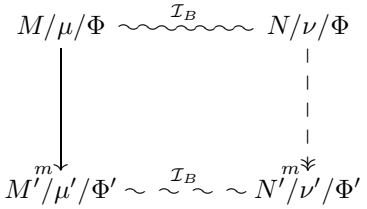
(I2) $N/\nu/\Phi \twoheadrightarrow_B W/\nu'/\Phi''$

(I3) $[\![M]\!]_B^A \preccurlyeq N$

(I4) Si $[m,\varphi,\varphi']$ *est* (B,Φ)*-actif, alors on a (a)* $m \in dom(\mu)$ *et* $m \in dom(\nu)$,
(b) $\mu(m) = (\varphi,V_1)$ *et* $\nu(m) = (\varphi,W_1)$,
(c) $[\![V_1]\!]_B^A \preccurlyeq W_1$.

Cet invariant porte sur deux configurations synchronisées (i.e. dont les passés sont égaux) $M/\mu/\Phi$ et $N/\nu/\Phi$ et se décompose en quatre points. Le point (I1) indique que la configuration étendue $M/\mu/\Phi$ se réduit vers une valeur. De cette étiquette de tête, on tire l'ensemble de lettres A et l'utilisation-mémoire B. Le point (I2) indique, de même, que la configuration étendue $N/\nu/\Phi$ se réduit aussi vers une valeur. Le point (I3) relie les termes M et N : le préfixe de M obtenu à partir de A et B est également un préfixe de N. De façon duale, le point (I4) relie les mémoires μ et ν. Toutes les adresses impliquées dans un intervalle (B,Φ)-actif appartiennent au domaine de μ et ν et les valeurs qui leur sont associées sont reliées de la même manière que pour le point (I3). Cet invariant est conservé par réduction.

Lemme 5.9 *Soit B un ensemble tel que $\mathcal{B}(B)$. On considère la réduction $M/\mu/\Phi \to_B M'/\mu'/\Phi'$ où le chemin menant au radical contracté est nommé φ_r. Si on a $\mathcal{I}_B(M,\mu,\Phi,N,\nu)$ et si φ_r ne préfixe aucun élément de $C(B) - \Phi'$, alors il existe une réduction $N/\nu/\Phi \twoheadrightarrow_B N'/\nu'/\Phi'$ qui vérifie $\mathcal{I}_B(M',\mu',\Phi',N',\nu')$.*



Si la configuration étendue $M/\mu/\Phi$ se réduit en une étape vers $M'/\mu'/\Phi'$, alors il existe une configuration étendue issue de $N/\nu/\Phi$ qui est *synchronisée* avec $M'/\mu'/\Phi'$ et qui vérifie l'invariant $\mathcal{I}_B$ avec cette dernière. On note que des conditions supplémentaires sur B sur le chemin menant au radical sont ajoutées aux hypothèses.

Preuve : Dans la suite de cette preuve, l'invariant $\mathcal{I}_B(M,\mu,\Phi,N,\nu)$ sera noté $\mathcal{I}_0$; l'hypothèse selon laquelle φ_r ne préfixe aucun élément de $C(B) - \Phi'$ est notée $\mathcal{K}_0$. On remarque que $\mathcal{I}_0$ implique $\mathcal{J}(\mu,\Phi,B)$ et $\mathcal{J}(\nu,\Phi,B)$; ces propriétés sont respectivement notées $\mathcal{J}_0$ et $\mathcal{J}'_0$. Soient R le radical contracté entre M et M' et $E[\,]$ son contexte d'évaluation dans M. On a $M = E[R]$. En posant $\kappa = \sigma(E[\,])$, on a $R/\mu/\Phi \xrightarrow{\kappa}_B R'/\mu'/\Phi'$ et $M' = E[R']$. On note $\beta = \tau(R)$; le chemin menant au radical contracté est donc $\varphi_r = \kappa\beta$. On note que, par définition de B, si $\varphi \in C(B)$, alors $|\varphi| \subseteq A$. On procède par cas.

1. Si $|\varphi_r| \not\subseteq A$, on a d'une part $[\![M]\!]^A_B = [\![M']\!]^A_B \preccurlyeq N$ et d'autre part, le chemin φ_r menant au radical R ne préfixe aucun chemin de $C(B)$. Par conséquent, en utilisant le lemme 5.8, on obtient $\Phi' = \Phi$. On veut montrer $\mathcal{I}_B(M',\mu',\Phi,N,\nu)$. Les points (I1), (I2) et (I3) sont vérifiés. Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ)-actif. Par $\mathcal{I}_0$, on a $m_0 \in dom(\nu)$, $m_0 \in dom(\mu)$, $\mu(m_0) = (\varphi,V_0)$ et et $\nu(m_0) = (\varphi,W_0)$ avec $[\![V_0]\!]^A_B \preccurlyeq W_0$. En utilisant le lemme 5.7 avec $\mathcal{J}_0$, on obtient $m_0 \in dom(\mu')$ et $\mu'(m_0) = \mu(m_0)$. Le point (I4) est donc aussi vérifié.
2. Si $|\varphi_r| \subseteq A$, alors $[\![E[\,]]\!]^A_B$ est un contexte. On pose $[\![E[\,]]\!]^A_B = E_0[\,]$. On a $[\![M]\!]^A_B = E_0[[\![R]\!]^A_B]$ et $[\![M']\!]^A_B = E_0[[\![R']\!]^A_B]$. Comme, par $\mathcal{I}_0$, on a $[\![M]\!]^A_B \preccurlyeq N$, il existe un contexte $E_1[\,]$ tel que $E_0[\,] \preccurlyeq E_1[\,]$, $N = E_1[N_1]$ et $[\![R]\!]^A_B \preccurlyeq N_1$. On procède par cas sur la réduction.
   (a) Si $R = ((\lambda x.M_1)^\alpha V_1)^\beta$, on a $R/\mu/\Phi \xrightarrow{\kappa}_B \beta\cdot\lceil\alpha\rceil^{\mathtt{b}} \cdot M_1\{x\backslash\lfloor\alpha\rfloor^{\mathtt{b}} \cdot V_1\}/\mu/\Phi$. On considère les cas suivants.
      i. Si $|\alpha| \not\subseteq A$, on a $[\![M']\!]^A_B = E_0[\Omega] \preccurlyeq E_0[[\![R]\!]^A_B] = [\![M]\!]^A_B \preccurlyeq N$. De là, on obtient $\mathcal{I}_B(M',\mu,\Phi,N,\nu)$.
      ii. Si $|\alpha| \subseteq A$, on a $[\![M]\!]^A_B = E_0[((\lambda x.[\![M_1]\!]^A_B)^\alpha [\![V_1]\!]^A_B)^\beta]$ et $N = E_1[((\lambda x.N_2)^\alpha N_3)^\beta]$ avec $[\![M_1]\!]^A_B \preccurlyeq N_2$ et $[\![V_1]\!]^A_B \preccurlyeq N_3$. Comme N se réduit vers une valeur, on a nécessairement $N/\nu/\Phi \twoheadrightarrow_B N'/\nu'/\Phi_1$ où $N' = E_1[((\lambda x.N_2)^\alpha W_1)^\beta]$. Les chemins menant aux radicaux contractés entre N et N' sont préfixés par φ_r. Par hypothèse, le chemin φ_r ne préfixe aucun chemin de $C(B) - \Phi$. Par conséquent, en utilisant itérativement le lemme 5.8, on obtient la relation $\Phi_1 = \Phi$. On a donc la réduction $N/\nu/\Phi \twoheadrightarrow_B N'/\nu'/\Phi \to_B N''/\nu'/\Phi$ où $N'' = E_1[\beta\cdot\lceil\alpha\rceil^{\mathtt{b}}\cdot N_2\{x\backslash\lfloor\alpha\rfloor^{\mathtt{b}}.W_1\}]$. On veut montrer $\mathcal{I}_B(M',\mu,\Phi,N'',\nu')$. Les points (I1) et (I2) sont vérifiés. Pour le point (I3), deux cas sont à envisager.
         – Si $|\tau(V_1)| \not\subseteq A$, alors $[\![V_1]\!]^A_B = \Omega \preccurlyeq W_1$. En utilisant le lemme 5.6, on obtient $[\![\beta\cdot\lceil\alpha\rceil^{\mathtt{b}}.M_1\{x\backslash\lfloor\alpha\rfloor^{\mathtt{b}}\cdot V_1\}]\!]^A_B \preccurlyeq \beta\cdot\lceil\alpha\rceil^{\mathtt{b}}\cdot N_2\{x\backslash\lfloor\alpha\rfloor^{\mathtt{b}}.W_1\}$ ce qui permet de conclure $[\![M']\!]^A_B \preccurlyeq N''$.
         – Si $|\tau(V_1)| = |\gamma| \subseteq A$, alors $[\![V_1]\!]^A_B$ est une valeur qui vérifie $[\![V_1]\!]^A_B \preccurlyeq N_3$. Par conséquent, N_3 est une valeur. On en déduit donc $N_3 = W_1$ et $[\![V_1]\!]^A_B \preccurlyeq W_1$. En utilisant le lemme 5.6, on obtient bien $[\![M']\!]^A_B \preccurlyeq N''$.

      Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ)-actif. En utilisant le lemme 5.7 avec $\mathcal{J}_0$ et $\mathcal{J}'_0$, on obtient $m_0 \in dom(\mu')$ et $m_0 \in dom(\nu')$ avec $\mu'(m_0) = \mu(m_0)$ et $\nu'(m_0) = \nu(m_0)$. Par $\mathcal{I}_0$, on a $\mu(m_0) = (\varphi,V_0)$ et $\nu(m_0) = (\varphi,W_0)$ avec $[\![V_0]\!]^A_B \preccurlyeq W_0$. Le point (I4) est donc aussi vérifié.
   (b) Si $R = (\mathtt{ref}(V_1))^\beta$, on pose $\alpha = \tau(V_1)$, $m = \varphi_r = \kappa\beta$ et $\varphi_0 = \varphi_r\mathtt{ref}\alpha$. La contraction considérée est $R/\mu/\Phi \xrightarrow{\kappa}_B m^{\lceil\beta\rceil^{\mathtt{c}}}/\mu'/\Phi'$ où $\mu' = (\mu; m \xmapsto{\varphi_0} V_1)$. On considère les cas suivants.

i. Si $B(m,\varphi_0) = \emptyset$, alors on a $\Phi' = \Phi$. De là, on obtient $[\![M]\!]^A_B = E_0[(\mathtt{ref}([\![V_1]\!]^A_B))^\beta]$ et $N = E_1[(\mathtt{ref}(N_2))^\beta]$ avec $[\![V_1]\!]^A_B \preccurlyeq N_2$. Comme N se réduit vers une valeur, on a nécessairement $N/\nu/\Phi \twoheadrightarrow_B N'/\nu'/\Phi_1$ où $N' = E_1[m^{\lceil\beta\rceil^c}]$. Les chemins menant aux radicaux contractés entre N et N' sont préfixés par φ_r. Par hypothèse, φ_r ne préfixe aucun chemin de $C(B) - \Phi' = C(B) - \Phi$. Par conséquent, en utilisant itérativement le lemme 5.8, on obtient $\Phi_1 = \Phi$. On veut montrer $\mathcal{I}_B(M',\mu',\Phi,N',\nu')$. Les points (I1), (I2) et (I3) sont vérifiés. Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ)-actif. En utilisant le lemme 5.7 avec $\mathcal{J}_0$ et $\mathcal{J}'_0$, on obtient $m_0 \in dom(\mu')$ et $m_0 \in dom(\nu')$ avec $\mu'(m_0) = \mu(m_0)$ et $\nu'(m_0) = \nu(m_0)$. Par $\mathcal{I}_0$, on obtient $\mu(m_0) = (\varphi,V_0)$ et $\nu(m_0) = (\varphi,W_0)$ avec $[\![V_0]\!]^A_B \preccurlyeq W_0$. Le point (I4) est donc aussi vérifié.

ii. Si $B(m,\varphi_0) \neq \emptyset$, alors on a $\Phi' = \Phi \cup \{\varphi_0\}$. Comme $\varphi_0 = \kappa\beta\mathtt{ref}\alpha \in C(B)$, on en déduit $|\alpha| \subseteq A$. De là, on a $[\![M]\!]^A_B = E_0[(\mathtt{ref}([\![V_1]\!]^A_B))^\beta]$ et $N = E_1[(\mathtt{ref}(W_1))^\beta]$ où W_1 est une valeur qui vérifie $[\![V_1]\!]^A_B \preccurlyeq W_1$ et $\tau(W_1) = \alpha$. On a donc la réduction $N/\nu/\Phi \to_B N'/\nu'/\Phi'$ où $N' = E_1[m^{\lceil\beta\rceil^c}]$ et $\nu' = (\nu; m \overset{\varphi_0}{\longmapsto} W_1)$. On veut montrer $\mathcal{I}_B(M',\mu',\Phi,N',\nu')$. Les points (I1), (I2) et (I3) sont vérifiés. Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ')-actif : on a $\varphi \in \Phi' = \Phi \cup \{\varphi_0\}$ et $\varphi' \notin \Phi'$. Deux cas sont à considérer.

– Si $\varphi \in \Phi$, alors l'intervalle $[m_0,\varphi,\varphi']$ est (B,Φ)-actif. Par conséquent, en utilisant le lemme 5.7 avec $\mathcal{J}_0$ et $\mathcal{J}'_0$, on obtient $m_0 \in dom(\mu')$ et $m_0 \in dom(\nu')$ avec $\mu(m_0) = \mu'(m_0)$ et $\nu(m_0) = \nu'(m_0)$. Par $\mathcal{I}_0$, on a $\mu(m_0) = (\varphi,V_2) = \mu'(m_0)$ et $\nu(m_0) = (\varphi,W_2) = \nu'(m_0)$ avec $[\![V_2]\!]^A_B \preccurlyeq W_2$.

– Si $\varphi = \varphi_0$, alors, du fait de l'invariant $\mathcal{B}$, on a $m_0 = m$. On obtient donc $\mu'(m) = (\varphi_0,V_1)$ et $\nu'(m) = (\varphi_0,W_1)$.

Le point (I4) est donc aussi vérifié.

(c) Si $R = (m^\alpha{:=}V_1)^\beta$, alors on pose $\varphi_1 = \varphi_r{:=}_1\alpha$. La contraction considérée dans ce cas est $R/\mu/\Phi \xrightarrow{\kappa}_B ()^{\lceil\beta\rceil^u}/\mu'/\Phi'$ où on pose $\mu = (\mu_0; m \overset{\varphi_0}{\longmapsto} V_0)$ et $\mu' = (\mu_0; m \overset{\varphi_1}{\longmapsto} V_1)$. On examine les cas suivants.

i. Si $B(m,\varphi_1) = \emptyset$, alors on a $\Phi' = \Phi$. De là, on obtient $[\![M]\!]^A_B = E_0[([\![m^\alpha]\!]^A_B{:=}[\![V_1]\!]^A_B)^\beta]$ et $N = E_1[(N_1{:=}N_2)^\beta]$ avec $[\![m^\alpha]\!]^A_B \preccurlyeq N_1$ et $[\![V_1]\!]^A_B \preccurlyeq N_2$. Comme N se réduit vers une valeur, on a $N/\nu/\Phi \xrightarrow{\kappa}\!\!\!\!\twoheadrightarrow_B N'/\nu'/\Phi_1$ avec $N' = E_1[()^{\lceil\beta\rceil^u}]$. Les chemins menant aux radicaux contractés entre N et N' sont préfixés par φ_r. Par hypothèse, ce dernier ne préfixe aucun chemin de $C(B) - \Phi' = C(B) - \Phi$. Par conséquent, en utilisant itérativement le lemme 5.8, on obtient $\Phi = \Phi_1$. On veut montrer $\mathcal{I}_B(M',\mu',\Phi,N',\nu')$. Les points (I1), (I2) et (I3) sont vérifiés. Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ)-actif. Par $\mathcal{I}_0$, on a $\mu(m_0) = (\varphi,V_2)$ et $\nu(m_0) = (\varphi,W_2)$ avec $[\![V_2]\!]^A_B \preccurlyeq W_2$. En utilisant le lemme 5.7 avec $\mathcal{J}_0$ et $\mathcal{J}'_0$, on obtient $m_0 \in dom(\mu')$ et $m_0 \in dom(\nu')$ avec $\mu'(m_0) = \mu(m_0)$ et $\nu'(m_0) = \nu(m_0)$. Le point (I4) est donc aussi vérifié.

ii. Si $B(m,\varphi_1) \neq \emptyset$, alors on a $\Phi' = \Phi \cup \{\varphi_1\}$. Comme $\varphi_1 = \varphi_r{:=}_1\alpha \in C(B)$, on obtient $|\alpha| \subseteq A$. De là, on a $[\![M]\!]^A_B = E_0[(m^\alpha{:=}[\![V_1]\!]^A_B)^\beta]$ et $N = E_1[(m^\alpha{:=}N_2)^\beta]$ avec $[\![V_1]\!]^A_B \preccurlyeq N_2$. Comme N se réduit vers une valeur, on a $N/\nu/\Phi \xrightarrow{\kappa}\!\!\!\!\twoheadrightarrow_B N'/\nu'/\Phi_1$ avec $N' = E_1[(m^\alpha{:=}W_1)^\beta]$. Soit ψ un chemin menant à un radical contracté entre N et N'. Ce chemin est nécessairement de la forme $\psi = \varphi_r{:=}_2\psi'$. Ce chemin est préfixé par φ_r. Comme, par hypothèse, φ_r ne préfixe aucun chemin de $C(B) - \Phi'$, de même ψ ne préfixe aucun chemin de $C(B) - \Phi'$. Comme $C(B) - \Phi = (C(B) - \Phi') \cup \{\varphi_1\}$ et $\varphi_1 = \varphi_r{:=}_1\alpha$, on en déduit que ψ ne préfixe aucun chemin de $C(B) - \Phi$. Par conséquent, en utilisant itérativement le lemme 5.8, on obtient $\Phi_1 = \Phi$. De là, on obtient la réduction : $E_1[(m^\alpha{:=}W_1)^\beta]/\nu'/\Phi \xrightarrow{\kappa}_B N'/\nu''/\Phi'$ avec $N' = E_1[()^{\lceil\beta\rceil^u}]$.

On veut montrer $\mathcal{I}_B(M',\mu',\Phi',N',\nu'')$. Les points (I1), (I2) et (I3) sont vérifiés. Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ)-actif. Deux cas sont à considérer.

- Si $\varphi \in \Phi$, alors $[m_0,\varphi,\varphi']$ est (B,Φ)-actif. Par $\mathcal{I}_0$, on a $\mu(m_0) = (\varphi,V_2)$ et $\nu(m_0) = (\varphi,W_2)$ avec $[\![V_2]\!]_B^A \preccurlyeq W_2$. En utilisant le lemme 5.7 avec les hypothèses $\mathcal{J}_0$ et $\mathcal{J}_0'$, on obtient $m_0 \in dom(\mu')$ et $m_0 \in dom(\nu'')$ avec $\mu(m_0) = \mu'(m_0)$ et $\nu(m_0) = \nu''(m_0)$.
- Si $\varphi = \varphi_1$, alors, du fait de l'invariant $\mathcal{B}$, on a $m_0 = m$. On obtient donc $\mu'(m) = (\varphi_1,V_1)$ et $\nu''(m) = (\varphi_1,W_1)$.

Le point (I4) est donc aussi vérifié.

(d) Si $R = (!m^\alpha)^\beta$, alors on pose $\varphi_1 = \varphi_r!\alpha$. La contraction considérée dans ce cas est $R/\mu/\Phi \xrightarrow{\kappa}_B \beta\cdot[m,\varphi_0,\varphi_1]\cdot V_0/\mu/\Phi'$ où $\mu = (\mu_0; m \overset{\varphi_0}{\longmapsto} V_0)$. On considère les cas suivants.

i. Si $[m,\varphi_0,\varphi_1] \notin B$, on a $\Phi' = \Phi$ et $[\![M']\!]_B^A = E_0[\Omega] \preccurlyeq [\![M]\!]_B^A \preccurlyeq N$. On obtient donc immédiatement $\mathcal{I}_B(M',\mu,\Phi,N,\nu)$.

ii. Si $[m,\varphi_0,\varphi_1] \in B$, alors cet intervalle est (B,Φ)-actif et $\Phi' = \Phi\cup\{\varphi_1\}$ et $|\alpha| \subseteq A$. De là, on a $[\![M]\!]_B^A = E_0[(!m^\alpha)^\beta]$ et $N = E_1[(!m^\alpha)^\beta]$. Comme l'intervalle $[m,\varphi_0,\varphi_1]$ est (B,Φ)-actif, en utilisant $\mathcal{I}_0$, on a $m \in dom(\nu)$ avec $\nu(m) = (\varphi_0,W_0)$ et $[\![V_0]\!]_B^A \preccurlyeq W_0$. On obtient donc la réduction $N/\nu/\Phi \xrightarrow{\kappa}_B N'/\nu/\Phi'$ où $N' = E_1[\beta\cdot[m,\varphi_0,\varphi_1].W_0]$. On veut montrer $\mathcal{I}_B(M',\mu,\Phi',N',\nu)$. Les points (I1) et (I2) sont vérifiés. Comme $[\![V_0]\!]_B^A \preccurlyeq W_0$, on obtient le point (I3) en utilisant le lemme 5.6. Soit $[m_0,\varphi,\varphi']$ un intervalle (B,Φ')-actif : on a $\varphi \in \Phi'$ et $\varphi' \notin \Phi'$. Deux cas sont à considérer.

- Si $\varphi \in \Phi$, alors $[m_0,\varphi,\varphi']$ est (B,Φ)-actif. Par $\mathcal{I}_0$, on a $\mu(m_0) = (\varphi,V_1) = \mu'(m_0)$ et $\nu(m_0) = (\varphi,W_1) = \nu'(m_0)$ avec $[\![V_1]\!]_B^A \preccurlyeq W_1$. En utilisant le lemme 5.7 avec $\mathcal{J}_0$ et $\mathcal{J}_0'$, on obtient $m_0 \in dom(\mu')$ et $m_0 \in dom(\nu')$ avec $\mu(m_0) = \mu'(m_0)$ et $\nu(m_0) = \nu'(m_0)$.
- Le cas $\varphi = \varphi_1$ est exclu par l'invariant $\mathcal{B}$ car $[m_0,\varphi,\varphi'] \in B$ et $[m,\varphi_0,\varphi_1] \in B$.

Le point (I4) est donc aussi vérifié.

(e) Si $R = (\texttt{ifz}\ 0^\alpha\ \texttt{then}\ M_1\ \texttt{else}\ M_2)^\beta$, alors la contraction considérée dans ce cas est $R/\mu/\Phi \xrightarrow{\kappa}_B \beta\cdot\lceil\alpha\rceil^{\mathtt{i}}\cdot M_1/\mu/\Phi$. On examine les cas suivants.

i. Si $|\alpha| \not\subseteq A$, on a $[\![M']\!]_B^A = E_0[\Omega] \preccurlyeq N$. On obtient donc $\mathcal{I}_B(M',\mu,\Phi,N,\nu)$.

ii. Si $|\alpha| \subseteq A$, alors on a $[\![M]\!]_B^A = E_0[(\texttt{ifz}\ 0^\alpha\ \texttt{then}\ [\![M_1]\!]_B^A\ \texttt{else}\ [\![M_2]\!]_B^A)^\beta]$. De là, on obtient $N = E_1[(\texttt{ifz}\ 0^\alpha\ \texttt{then}\ N_1\ \texttt{else}\ N_2)^\beta]$ où $[\![M_1]\!]_B^A \preccurlyeq N_1$. Par conséquent, on a $N/\nu/\Phi \xrightarrow{\kappa}_B N'/\nu/\Phi$ avec $N' = E_1[\beta\cdot\lceil\alpha\rceil^{\mathtt{i}}\cdot N_1]$. On veut montrer $\mathcal{I}_B(M',\mu,\Phi,N',\nu)$. Les points (I1), (I2) et (I4) sont vérifiés. On obtient le point (I3) en utilisant le lemme 5.6.

(f) Les autres cas sont similaires aux cas précédents. □

Ce résultat permet d'obtenir le lemme suivant qui constitue le résultat de base du théorème de non-interférence.

Lemme 5.10 *Soit M un terme tel que $T(M) = \emptyset$. On suppose $M/\emptyset \twoheadrightarrow V/\mu$. Soient $A = |\tau(V)|$ et $B = T(\tau(V))$. Si le terme N vérifie $[\![M]\!]_B^A \preccurlyeq N$ et $N/\emptyset/\emptyset \twoheadrightarrow_B W/\nu/\Phi$, alors $[\![V]\!]_B^A \preccurlyeq W$.*

Preuve : En utilisant le lemme 5.3 on obtient $\mathcal{B}(B)$. Avec le lemme 5.4, on obtient la réduction $M/\emptyset/\emptyset \twoheadrightarrow_B V/\mu/C(B)$. Cette réduction peut s'écrire de la façon suivante

$$M/\emptyset/\emptyset = M_0/\mu_0/\Phi_0 \to_B M_1/\mu_1/\Phi_1 \to_B \ldots \to_B M_n/\mu_n/\Phi_n = V/\mu/C(B)$$

Pour $i \in \{1\ldots n\}$, on nomme φ^i le chemin menant au radical contracté entre M_{i-1} et M_i. Soit $j \in \{1\ldots n-1\}$; on veut montrer que φ_j ne préfixe aucun chemin de $C(B) - \Phi_j$. Par l'absurde,

soit $\psi \in C(B) - \Phi_j$ tel que $\varphi_j \preceq \psi$. Comme ψ appartient au passé final, il existe un indice k tel que $j < k \leq n$ tel que $\varphi^k = \mathsf{Pre}(\psi)$. Par conséquent, ceci implique $\varphi^j \preceq \varphi^k$ ce qui contredit le théorème 5.4. Par conséquent, φ_j ne préfixe aucun chemin de $C(B) - \Phi_j$. De là, en utilisant itérativement le lemme 5.9, on obtient $\mathcal{I}_B(V,\mu,C(B),W,\nu)$ et donc $[\![V]\!]_B^A \preccurlyeq W$. □

De la valeur obtenue de la réduction de M, on obtient un préfixe P de M. Si un terme est préfixé par P et se réduit vers une valeur W, en respectant l'utilisation-mémoire de V, alors V et W admettent un préfixe commun non réduit à Ω. Si on se place dans le cas particulier où V est un entier n^α, on obtient $W = n^\alpha = V$. Dans ce cas, l'observation du résultat ne donne aucune information sur les sous-termes de M qui sont effacés dans P. On exploite ce résultat dans le théorème de non-interférence.

Théorème 5.5 (Non-interférence) *Soit M un terme tel que $\mathtt{INIT}(M)$ et $\mathcal{R} : M/\emptyset \twoheadrightarrow V/\mu$. Le sous-terme $(C[\,],N)$ de M interfère dans $\mathcal{R}$ si et seulement si $\tau(N) \in |\tau(V)|$.*

Preuve : En utilisant le lemme 5.10, on obtient que si $\tau(N) \notin |\tau(\alpha)|$, alors N n'interfère pas. Réciproquement, si N n'interfère pas, on peut le remplacer par N' qui est le terme N où l'on a changé l'étiquette de tête pour une lettre qui n'intervient pas dans M. Il est clair que $C[N']/\emptyset/\emptyset$ se réduit vers une valeur V' en respectant $T(\tau(V))$. Par définition de la non-interférence, on a $\tau(V) = \tau(V')$ ce qui implique $\tau(N) \notin |\tau(V)|$. □

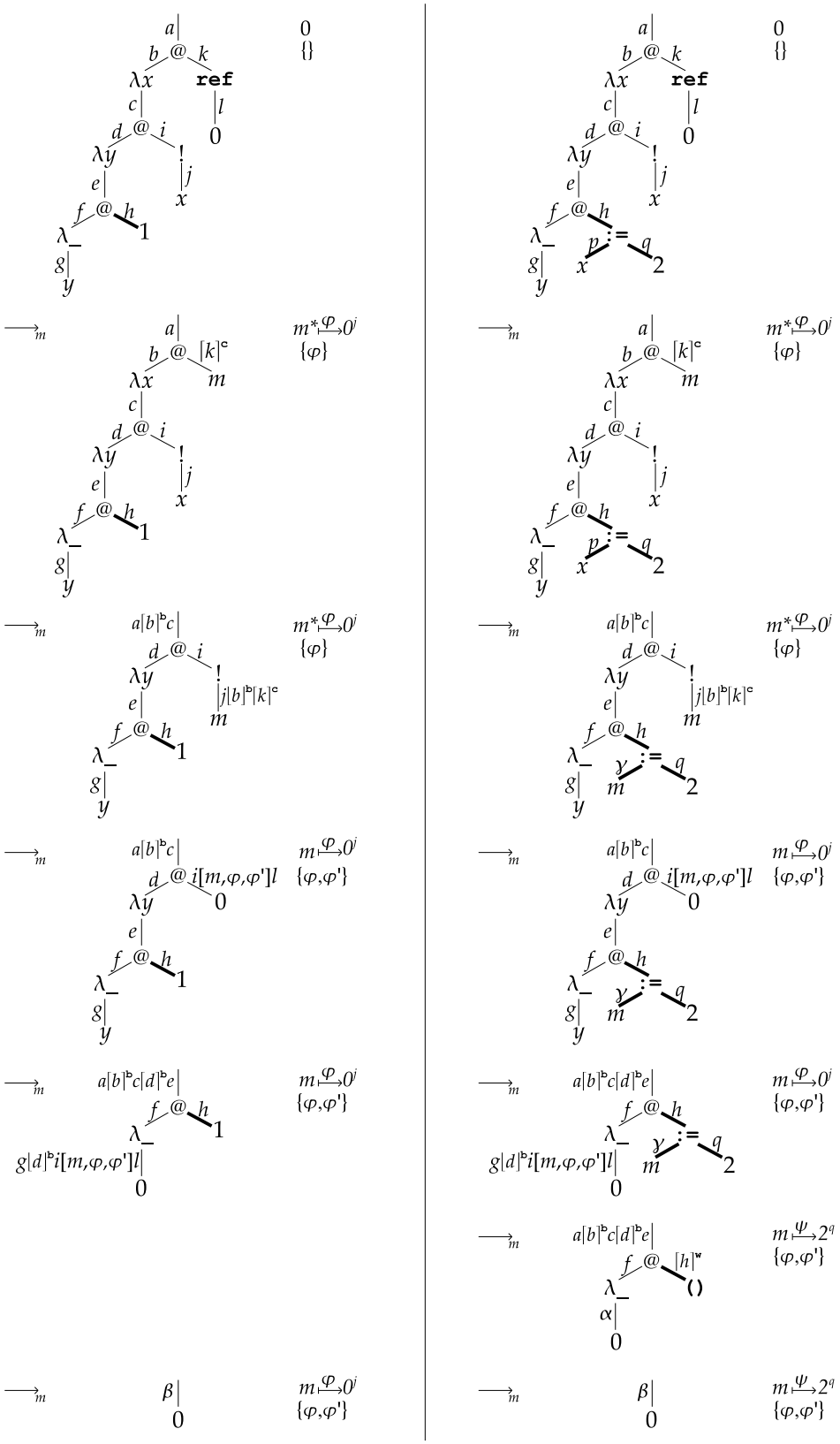
Si les étiquettes de M sont des lettres distinctes, un sous-terme N de M interfère, au sens de la définition 5.3, dans la réduction $M/\emptyset \twoheadrightarrow_e V/\mu$ si et seulement si son étiquette est une lettre présente dans l'étiquette de tête de V. Pour illustrer cette propriété de non-interférence, on reprend les réductions $\mathcal{R}_1$ et $\mathcal{R}_2$ citées en exemple dans la section 5.3. Ces termes, une fois étiquetés, sont :

$$M_1 = ((\lambda x.((\lambda y.((\lambda\_.y^g)^f 1^h)^e)^d(!x^j)^i)^c)^b(\mathtt{ref}(0^l))^k)^a$$
$$M_2 = ((\lambda x.((\lambda y.((\lambda\_.y^g)^f(x^p{:=}2^q)^h)^e)^d(!x^j)^i)^c)^b(\mathtt{ref}(0^l))^k)^a$$

Les réductions de ces termes sont illustrées sur la figure 5.22. Cette figure est composée de deux colonnes. Dans la première colonne, on trouve la réduction de M_1. Dans la colonne de droite, on trouve la réduction de M_2. Les parties non communes aux deux colonnes sont signalées en gras. Le préfixe obtenu avec $[\![\,]\!]_B^A$ à partir du terme de la colonne de gauche correspond à la partie non grasse, c'est-à-dire à la partie commune entre les termes des deux réduction. Ceci est bien conforme au lemme 5.9. De même, lorsque l'adresse m est active (ce qui est mentionné par une étoile), la valeur associée à m est égale dans les réductions de M_1 et M_2.

Si on applique l'analyse statique de Pottier et Simonet [37, 39] sur M_2, en supposant que 2^q est secret, alors on obtient que le résultat de la réduction est secret. En effet, cette analyse statique attribue à chaque adresse un niveau de sécurité qui correspond au niveau de sécurité des valeurs qui sont associées à l'adresse au cours de la réduction. Comme 2^q est affecté à m, alors le niveau de sécurité de m est secret. En réalité, deux valeurs sont successivement associées à m dans la mémoire. Tout d'abord, l'adresse est initialisée avec la valeur publique 0^l. Puis, la valeur secrète 2^q est affecté à m. Mais seule la première valeur contribue au résultat, qui devrait donc être considéré comme public. Bien entendu, une analyse statique nécessite certaines approximations pour être décidable. Si le système présenté ici ne fournit pas une analyse statique, il offre, en revanche, une base théorique pour raisonner sur une telle analyse.

Dans ce chapitre nous avons successivement examiné la propriété de non-interférence dans le λ-calcul, dans le λ-calcul par valeur et dans un λ-calcul muni de traits impératifs. Pour aborder ce problème, on s'inspire de l'approche des analyses statiques de flot d'information : si la réduction du terme M aboutit sur une valeur V, on souhaite savoir si l'observation de la valeur V (intuitivement publique) permet d'obtenir une information sur certains sous-termes (intuitivement secrets) de M. Les étiquettes du λ-calcul s'avèrent être un outil précieux puisque ces étiquettes fournissent



$$\begin{aligned}
\beta &= a\lceil a\rceil^{\mathtt{b}} c\lceil d\rceil^{\mathtt{b}} e\lceil f\rceil^{\mathtt{b}} g\lfloor d\rfloor^{\mathtt{b}} i[m,\varphi,\varphi']l & \gamma &= p\lfloor b\rfloor^{\mathtt{b}}\lceil k\rceil^{\mathtt{c}} \\
m &= a@_2 k & \psi &= a\lceil b\rceil^{\mathtt{b}} c\lceil d\rceil^{\mathtt{b}} e@_2 h{:=_1}\gamma \\
\varphi &= a@_2 k\,\mathtt{ref}\,l & \varphi' &= a\lceil b\rceil^{\mathtt{b}} c@_2 i!j\lfloor b\rfloor^{\mathtt{b}}\lceil k\rceil^{\mathtt{c}}
\end{aligned}$$

FIG. 5.22 – *Réduction des termes M_1 et M_2*

intuitivement une analyse dynamique de dépendance des termes vis-à-vis des sous-termes du terme initial. Dans le cas du λ-calcul et du λ-calcul par valeur, l'étiquette de tête de V permet d'obtenir le préfixe X des sous-termes de M dont V dépend. Ces sous-termes interfèrent dans V. A contrario, les sous-termes qui sont effacés dans X n'interfèrent pas dans V puisque cette valeur ne donne aucune information sur ces sous-termes. Nous avons remarqué au passage que les préfixes d'interférence et de stabilité coïncident dans le cas du λ-calcul. En revanche, dans le λ-calcul par valeur, le préfixe d'interférence est inclus dans le préfixe de stabilité. Nous avons expliqué cette inclusion par les différences entre les définitions de sous-terme non-critique et de sous-terme qui n'interfère pas. Si M se réduit vers une valeur, un sous-terme $(C[\,],N)$ de M n'est pas critique si en changeant N pour N', on obtient une valeur de même observable. Par contraste, ce sous-terme n'interfère pas si, en changeant N pour N' *et* dans le cas où $C[N']$ se réduit vers une valeur, l'observable de cette valeur est inchangé.

Les étiquettes du λ-calcul permettent d'obtenir simplement la propriété de non-interférence, dans le cas de ces langages fonctionnels. En présence de traits impératifs tels que l'affectation ou la déréférence, un nouveau type d'interférence vient se combiner à l'interférence fonctionnelle présente dans le λ-calcul et le λ-calcul par valeur. Pour étudier ce phénomène, on introduit le λ_m-calcul et le λ_m-calcul étiqueté. Au cours de la réduction $M/\emptyset \twoheadrightarrow V/\mu$, des écritures et des lectures en mémoire ont lieu. Certaines adresses de la mémoire peuvent interférer dans V. Plus précisément, une adresse peut contribuer à V pendant certains intervalles de temps : entre une écriture et une lecture. Si un effet de bord vient interférer entre cette écriture et cette lecture, la valeur lue est modifiée, ce qui entraîne la modification de la valeur finale. A l'interférence "fonctionnelle" vient s'ajouter une interférence sur la mémoire. Les étiquettes du λ_m-calcul étiqueté permettent d'identifier à la fois les sous-termes qui interfèrent (interférence fonctionnelle) et les intervalles actifs des adresses qui interfèrent (interférence de la mémoire). Les étiquettes du λ_m-calcul permettent d'étendre l'approche initiée par Abadi et al. dans [3]. Une comparaison formelle entre notre approche très théorique et l'approche pragmatique des analyses de flot d'information telles que Flow Caml pourrait être très fructueuse.

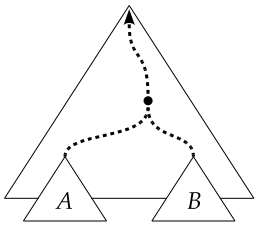
Chapitre 6

Indépendance

Dans ce chapitre, nous nous intéressons aux politiques de sécurité dynamiques comme la Muraille de Chine. On montre comment mettre en œuvre cette dernière dans le λ_n-calcul étiqueté qui est un langage inspiré du λ-calcul étiqueté. On montre que la Muraille de Chine garantit la propriété de sécurité d'indépendance.

Nous avons étudié, dans le chapitre 4, le mécanisme d'inspection de pile. Ce dispositif technique est utilisé dans les machines virtuelles de Java (JVM) et de la plate-forme .NET (CLR). L'inspection de pile vise à contrôler dynamiquement l'exécution d'un programme constitué de sous-programmes d'origines diverses, parfois douteuses, afin que ce programme ne fasse pas *quelque chose de mal*. Comme l'ont montré les travaux de Fournet et Gordon [15], l'apport de ce mécanisme de sécurité n'est pas clairement défini. En revanche, cet exemple illustre de façon claire les définitions de politique de sécurité et de propriété de sécurité. Une **politique de sécurité** est un mécanisme, un protocole ou plus généralement un *moyen* d'atteindre un *objectif* en terme de sécurité. Cet objectif est appelé **propriété de sécurité**. Dans l'exemple présent, la politique de sécurité est le mécanisme d'inspection de pile ; la propriété de sécurité n'est pas définie. L'exemple de l'analyse de flot d'information est plus probant. Si on prend les travaux de Pottier et Simonet en référence, le but est clairement identifié : la propriété de sécurité visée est la non-interférence. Pour parvenir à ce but, ils utilisent une analyse statique de flot d'information fondée sur un système de types. Leur politique de sécurité consiste à se restreindre à n'exécuter que des termes bien typés.

La propriété de non-interférence, définie dans [16] par Goguen et Meseguer, a donné lieu à de nombreux travaux dans le domaine de l'analyse de flot d'information [3, 36, 14, 13, 22, 37, 39, 42]. Cependant, il s'avère que cette propriété peut ne pas correspondre à la propriété de sécurité recherchée. Pour illustrer cette affirmation, on peut considérer le problème pratique des enchères secrètes, mentionné par Chong et Myers dans [12]. Dans cet exemple, un certain nombre d'enchérisseurs sont en compétition pour acheter un objet. Chaque enchérisseur soumet son offre dans une enveloppe qu'il scelle. Une fois l'enveloppe scellée, l'enchérisseur ne peut plus modifier son offre. Aucune enveloppe scellée ne peut être ouverte avant le scellement de toutes les enveloppes. Lorsque toutes les enveloppes sont scellées, elles sont toutes descellées pour déterminer le vainqueur des enchères. Cette *politique de sécurité*, ici définie informellement, sera appelée par la suite les Enchères scellées. Cette dernière vise à assurer une *propriété de sécurité* d'indépendance des enchères. Informellement, le processus de décision d'un enchérisseur ne doit pas dépendre des autres enchérisseurs. Ceci se traduit par le fait que l'ordre des prises de décision de chaque enchérisseur ne change pas les enchères de chacun d'entre eux. Pour obtenir cette propriété, le protocole des Enchères scellées assure le secret des enchères définitivement soumises avant l'événement que constitue le scellement de toutes les enveloppes. Après cet événement, les enchères secrètes deviennent



Le résultat de la réduction dépend-t-il de l'interaction entre A et B ?

FIG. 6.1 – *Indépendance*

publiques. Ce passage du niveau secret au niveau public, appelé *déclassification*, pose problème dans le cadre d'une analyse de flot d'information. Chong et Myers intègrent un mécanisme de déclassification explicite dans leur système de type et ils adaptent la propriété de non-interférence en conséquence. Dans ce chapitre, plutôt que d'essayer d'adapter la non-interférence pour traiter les cas de déclassification, nous souhaitons mettre en lumière la véritable propriété de sécurité recherchée : le but de la politique de sécurité des Enchères scellées est d'assurer *l'indépendance* des enchères.

Une autre politique de sécurité peut être rapprochée des Enchères scellées : la *Muraille de Chine*. Cette politique s'inspire des conflits d'intérêts économiques. Elle a été introduite et formalisée par Brewer et Nash [10] en 1989. Cette politique de sécurité agit dans la situation concrète suivante. On suppose qu'Alice et Bob sont concurrents. Charlie est un partenaire économique potentiel pour Alice et Bob : par exemple Charlie pourrait être un conseiller qui pourrait aider Alice et Bob à fixer le prix des produits vendus par Alice et Bob. Dans ce contexte, si Charlie travaillait avec Alice puis Bob, ce dernier pourrait, pour fixer son prix, tirer profit des connaissances acquises par Charlie sur le prix de revient du produit d'Alice. C'est ce que la *politique de sécurité* de la Muraille de Chine doit empêcher. Pour ce faire, l'application de la Muraille de Chine consiste à contrôler les interactions entre Alice, Bob et Charlie. Plus précisément, une fois que Charlie a interagi avec Alice (respectivement Bob), Charlie n'a plus le droit d'interagir avec Bob (resp. Alice). La propriété de sécurité visée par la Muraille de Chine est de conserver l'indépendance des actions d'Alice et Bob, malgré l'intervention de Charlie, qui peut potentiellement interagir avec Alice ou Bob. Le caractère dynamique de cette propriété de sécurité fait qu'il est difficile de la mettre en place en s'appuyant sur une analyse statique de flot d'information. Cette politique de sécurité se rapproche de l'exemple des Enchères scellées dans le sens où on souhaite assurer que les actions des protagonistes (Alice et Bob ou les enchérisseurs) sont *indépendantes* les unes des autres. Pour aboutir à cette propriété de sécurité d'indépendance, que nous définirons formellement dans le cadre du λ-calcul, ces politiques de sécurité se fondent sur l'histoire des événements passés (interaction passée entre Alice et Charlie ou scellement de toutes les enveloppes) pour autoriser ou interdire certaines actions (interaction entre Bob et Charlie ou ouverture d'une enveloppe). Les étiquettes du λ-calcul étiqueté, qui contiennent intuitivement l'histoire de la réduction, permettront de définir formellement la politique de la Muraille de Chine dans le λ-calcul étiqueté.

Dans la section 6.1, nous présentons le λ_n-calcul : il s'agit d'une variante du λ-calcul qui fait intervenir explicitement la notion de principal. Cette notion permet notamment de distinguer les sous-termes qui appartiennent à Alice des sous-termes de Bob. Par contraste avec les étiquettes du λ-calcul, un principal utilisé dans le λ_n-calcul est irrémédiablement attaché à un sous-terme : il disparaît si ce sous-terme disparaît. Si les étiquettes du λ-calcul sont dynamiques (elles se composent au cours du calcul), les principaux sont statiques. Dans le λ_n-calcul, on définit la notion

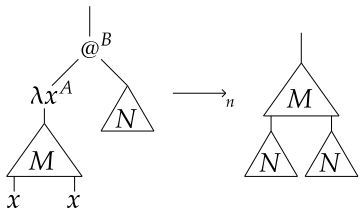


FIG. 6.2 – *β_n-réduction*

de **réduction indépendante de l'interaction entre deux principaux** A **et** B. Cette propriété est illustrée sur la figure 6.1. Cette définition reprend l'idée intuitive selon laquelle l'indépendance des actions d'Alice et Bob se traduit par le fait que l'ordre de ces actions est indifférent : si les pas de réduction dans lesquels interviennent A et B peuvent être permutés, cette réduction est indépendante de l'interaction entre A et B. Dans la section 6.2, nous introduisons les étiquettes du λ-calcul dans le λ_n-calcul. Comme on l'a déjà vu dans les parties précédentes, l'étiquette d'un radical R contient les noms des radicaux qu'il a fallu contracter pour créer R. Les étiquettes donnent donc l'information sur les dépendances entre réductions, ce qui correspond bien à la chronologie des événements qui est utilisée dans la Muraille de Chine et dans les Enchères scellées. En imposant des contraintes sur les noms des radicaux contractés, on définit formellement la politique de sécurité de la Muraille de Chine dans le λ_n-calcul étiqueté. On prouve que cette *politique de sécurité* assure la *propriété de sécurité* d'indépendance. Plus généralement, on prouve que les étiquettes du λ_n-calcul expriment simplement la notion de réduction indépendante de l'interaction entre A et B.

6.1 Le λ_n-calcul

Dans cette section, nous introduisons le λ_n-calcul. Ce langage est un λ-calcul dans lequel les sous-termes sont annotés de façon statique par des **principaux**. Comme mentionné dans la section 4, un principal peut être un individu, tel que Alice, Bob ou Charlie, ou une organisation, un programme ou n'importe quelle entité dont l'identité est utilisée. Alors que les étiquettes du λ-calcul, expriment l'histoire des réductions et des dépendances, les principaux utilisés dans le λ_n-calcul permettent simplement d'identifier l'origine des sous-termes au cours du calcul. Ces principaux sont notés A, $B, \ldots$ et l'ensemble des principaux est dénombrable. Il est noté $\mathbf{P}$. Ils sont simplement attachées irrémédiablement à un sous-terme. Si le sous-terme est dupliqué, son principal est dupliqué ; si le sous-terme disparaît, alors le principal disparaît aussi. La syntaxe des termes et des valeurs du λ_n-calcul est définie ci-dessous.

$$
\begin{array}{rcll}
M, N & ::= & x & \text{Variable} \\
 & | & (\lambda x.N)^A & \text{Abstraction} \\
 & | & (MN)^A & \text{Application} \\
V, W & ::= & (\lambda x.N)^A & \text{Valeurs}
\end{array}
$$

La syntaxe est reprise du λ-calcul classique. La seule différence est qu'un principal est attaché à chaque abstraction et chaque application. Le principal d'une abstraction ou d'une application est le principal A attaché à ce terme. On dira que les termes $(\lambda x.N)^A$ et $(MN)^A$ *appartiennent* au principal A. La syntaxe peut sembler proche du λ-calcul étiqueté. Cependant, si on représente les termes par des arbres, les principaux employés dans le λ_n-calcul sont intuitivement attachés aux nœuds de l'arbre alors que les étiquettes du λ-calcul sont intuitivement attachées au arêtes. La figure 6.3, qui représente le terme $M = (((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$, illustre cette intuition. Comme le

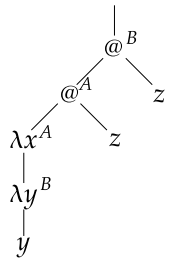


FIG. 6.3 – *Terme* $M = (((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$

montrent les règles de réduction, le système d'étiquette est statique : les principaux sont attachés irrémédiablement à un sous-terme.

$$(\beta_n) \qquad ((\lambda x.M)^A N)^B \to_n M\{x \backslash N\}$$

$$(\nu_n)\ \frac{M \to_n M'}{(MN)^A \to_n (M'N)^A} \qquad (\mu_n)\ \frac{N \to_n N'}{(MN)^A \to_n (MN')^A} \qquad (\xi_n)\ \frac{M \to_n M'}{(\lambda x.M)^A \to_n (\lambda x.M')^A}$$

La règle de réduction de base est la β_n-réduction. Elle est illustrée sur la figure 6.2. La contraction du β_n-radical $((\lambda x.M)^A N)^B$ fait disparaître l'application et l'abstraction qui le constituent. Les principaux attachés à ces sous-termes disparaissent également alors que dans le λ-calcul étiqueté, le nom du radical perdure sous une forme surlignée et soulignée : ce mécanisme de création de nouveaux principaux n'existe pas ici. Ceci justifie l'appellation d'annotation *statique*. Les règles (ν_n), (μ_n) et (ξ_n) sont de simples adaptations des règles de contexte du λ-calcul. La notion de radical et de résidu est adaptée de façon élémentaire à partir des définitions correspondantes dans le λ-calcul. Le λ_n-calcul, qui n'est qu'une variante syntaxique du λ-calcul avec des annotations statiques, vérifie les mêmes propriétés fondamentales que le λ-calcul.

Théorème 6.1 (Confluence) *Si $M \twoheadrightarrow_n M_1$ et $M \twoheadrightarrow_n M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_n N$ et $M_2 \twoheadrightarrow_n N$.*

Théorème 6.2 (Développements finis) *Soit $\mathcal{F}$ un ensemble de β_n-radicaux de M.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Tous les développements de $\mathcal{F}$ finissent sur un même terme N.*
3. *L'ensemble des résidus d'un β_n-radical R de M dans N est indépendant du développement considéré.*

Théorème 6.3 (Standardisation) *Si $M \twoheadrightarrow_n N$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_n N$ telle que $\mathcal{R}$ est standard.*

Preuve : En considérant les principaux comme des lettres, les termes du λ_n-calcul peuvent être vus comme des termes du λ-calcul étiqueté. Dans la suite de cette preuve, on suppose que les termes initiaux considérés sont étiquetés par des lettres, c'est-à-dire des principaux. Les théorèmes 6.1, 6.2 et 6.3 sont prouvés en utilisant les fonctions Φ et Ψ suivantes.

$$\Phi((\lambda x.M)^\alpha) = (\lambda x.\Phi(M))^{\Psi(\alpha)} \qquad \Phi(x^\alpha) = x^{\Psi(\alpha)} \qquad \Phi(MN)^\alpha = (\Phi(M)\Phi(N))^{\Psi(\alpha)}$$

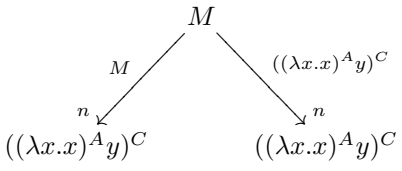
La fonction Φ modifie les étiquettes d'un terme du λ-calcul par l'intermédiaire de la fonction Ψ. On définit Ψ en remarquant que, si les étiquettes initiales d'un terme sont des principaux, toutes les étiquettes intervenant au cours du calcul sont de la forme $\alpha = A_1\beta_1 A_2\beta_2 \ldots A_2\beta_2 A_n$ où les

étiquettes β_i vérifient $\beta_i = \lceil \alpha_i \rceil$ ou $\beta_i = \lfloor \alpha_i \rfloor$ avec α_i de la même forme que α. Cette remarque est exploitée dans la définition de Ψ.

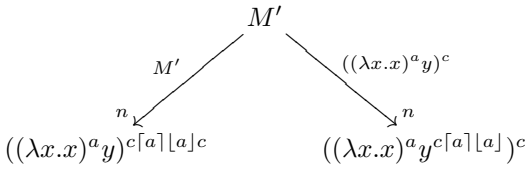
$$\Psi(A_1\beta_1 A_2\beta_2 \dots A_2\beta_2 A_n) = A_n$$

La fonction Ψ ne conserve que la dernière lettre des étiquettes. Synthétiquement, le terme $\Phi(M)$ est le terme M où toutes les étiquettes α sont remplacées par $\Psi(\alpha)$. De ce fait, $\Phi(M)$ est un terme du λ_n-calcul. La fonction Φ est un homomorphisme : si $M \to_e N$ (où les étiquettes de M sont des principaux), alors $\Phi(M) \to_n \Phi(N)$. Et si $M \to_n N$, alors il existe un terme N' tel que $M \to_e N'$ et $\Phi(N') = N$. Ces fonctions permettent de montrer que la confluence et les théorèmes des développements finis et de standardisation du λ-calcul étiqueté impliquent la confluence et les théorèmes des développements finis et de standardisation du λ_n-calcul. □

Le λ_n-calcul est confluent et vérifie les théorèmes des développements finis et de standardisation. On note que, du fait du caractère statique des étiquettes, le λ_n-calcul ne vérifie pas la propriété d'irréversibilité du λ-calcul étiqueté, comme le montrent les réductions comparées de $M = ((\lambda x.x)^A((\lambda x.x)^A y)^C)^C$ et $M' = ((\lambda x.x)^a((\lambda x.x)^a y)^c)^c$.



Réduction du λ_n-calcul



Réduction du λ-calcul étiqueté

Contrairement au λ-calcul étiqueté, des coïncidences syntaxiques peuvent survenir dans le λ_n-calcul. La contraction des deux radicaux de M aboutit au même terme, alors que ces situations sont intuitivement différentes comme le montre la notion de résidu qui n'est pas la même dans les deux cas. De cet exemple, on conclut que, contrairement aux apparences, le λ_n-calcul est plus proche du λ-calcul que du λ-calcul étiqueté.

On introduit désormais la notion d'*indépendance*. Le but recherché est d'exprimer le fait que A et B n'interagissent pas, directement ou indirectement, au cours d'une réduction. Ceci nous conduit à définir la notion de réduction ignorant le principal A.

Définition 6.1 (Réduction ignorant un principal) *La réduction* $\mathcal{R} : M \xrightarrow{((\lambda x.N)^B P)^C}_n M'$ *ignore le principal* A*, ce qu'on note* $M \xrightarrow{\neg A}_n M'$*, si et seulement si* $A \notin \{B,C\}$.

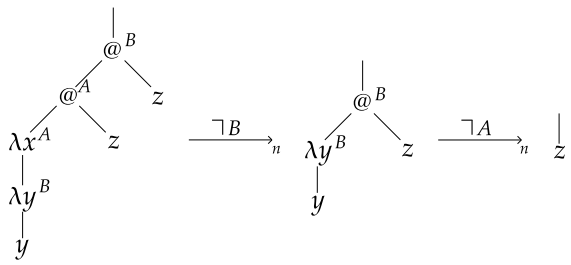


FIG. 6.4 – *Réduction de* $(((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$

Une réduction ignore le principal A si les principaux de l'abstraction et de l'application du radical contracté ne sont ni l'un ni l'autre A. En première approche, on pourrait penser qu'une succession de réductions qui ignorent A ou B serait une réduction qui laisse A et B indépendants. La réduction

de la figure 6.4 montre les limites de cette approche. Cette réduction aboutit au terme z après une réduction ignorant B puis une réduction ignorant A. Cependant, intuitivement, si les actions du principal A ne dépendent pas des actions du principal B, alors les actions de A devraient pouvoir être effectuées avant les actions de B. Ce qui n'est pas le cas ici. En effet, le radical $((\lambda y.y)^B z)^A$ est créé par la réduction ignorant B. Cet exemple souligne que, si A et B sont indépendants, alors les réductions dans lesquelles A et B interviennent doivent être interchangeables. On traduit plus formellement cette intuition dans la définition suivante.

Définition 6.2 (Indépendance) *Une réduction $\mathcal{R} : M \twoheadrightarrow_n N$ est indépendante de l'interaction entre A et B si et seulement s'il existe deux réductions $\mathcal{R}_A : M \xrightarrow{\neg A}\!\!\!\!\twoheadrightarrow_n M_A$ et $\mathcal{R}_B : M \xrightarrow{\neg B}\!\!\!\!\twoheadrightarrow_n M_B$ qui vérifient $\mathcal{R} \leq \mathcal{R}' = \mathcal{R}_A \sqcup \mathcal{R}_B$.*

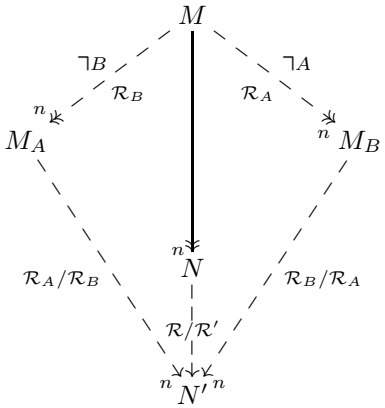


FIG. 6.5 – *Indépendance*

Cette définition est illustrée par la figure 6.5. Intuitivement, on peut prolonger la réduction $\mathcal{R}$ (par la réduction $\mathcal{R}/\mathcal{R}'$) pour obtenir une réduction qui peut se décomposer en deux réductions interchangeables : une réduction ignorant A et une réduction ignorant B. Avec ce critère, la réduction mentionnée en figure 6.4 n'est pas indépendante de l'interaction entre A et B. Dans la suite de cette partie, nous montrerons que cette propriété de sécurité est garantie par la politique de sécurité de la Muraille de Chine.

6.2 Le λ_n-calcul étiqueté

Les politiques de sécurité de la Muraille de Chine et des Enchères scellées utilisent l'histoire des interactions passées. Pour illustrer ce fait, on se replace dans le cadre que nous avions mentionné en introduction de ce chapitre : Alice et Bob sont des concurrents et Charlie est un partenaire économique susceptible de collaborer avec Alice ou Bob. Avant toute interaction, Charlie peut choisir de collaborer avec Alice ou Bob. Mais la politique de sécurité de la Muraille de Chine stipule qu'une fois que Charlie et Alice ont collaboré, alors Charlie n'a plus le droit d'interagir avec Bob : intuitivement les informations collectées au cours de la collaboration avec Alice ne doivent pas pouvoir servir à Bob, directement ou indirectement. L'interaction passée entre Alice et Charlie est donc utilisée pour autoriser ou interdire une interaction entre Bob et Charlie. L'histoire est aussi utilisée dans la politique des Enchères scellées. Comme mentionné en introduction, des enchères secrètes sont organisées. Pour assurer le secret (et l'indépendance) de ces enchères, la politique de sécurité des Enchères scellées stipule que chaque enchérisseur doit soumettre son enchère dans une enveloppe qui est ensuite scellée. Une fois l'enveloppe scellée, l'enchérisseur ne peut plus modifier son offre. Ces enveloppes ne peuvent être ouvertes que si tous les enchérisseurs

ont scellé leur enveloppe. Ici aussi, l'autorisation ou l'interdiction de l'ouverture des enveloppes dépend d'interactions passées : les scellements des enveloppes.

Pour exprimer des politiques de sécurité qui utilisent l'histoire des interactions passées, on souhaite pouvoir accéder à cette histoire directement dans le calcul. Cette histoire peut-être fournie par les étiquettes du λ-calcul. Ces étiquettes expriment les dépendances vis-à-vis des interactions passées. Dans cette section, on étend le λ_n-calcul avec les étiquettes *dynamiques* du λ-calcul. On peut ainsi définir formellement la politique de sécurité de la Muraille de Chine dans le cadre du λ_n-calcul étiqueté. On montre ensuite que les étiquettes du λ_n-calcul permettent d'exprimer simplement la notion d'indépendance qui a été définie dans la section précédente, en dehors du formalisme des étiquettes du λ-calcul. Enfin, on prouve que la *correction* de la Muraille de Chine. Cette *politique* de sécurité garantit la *propriété* de sécurité d'indépendance.

Pour introduire les étiquettes dans le λ_n-calcul, on s'inspire du λ-calcul faible étiqueté. On utilise une notion d'étiquette atomique, de séquence d'étiquettes et d'étiquette composée. Ces étiquettes interviennent dans la syntaxe des termes via une nouvelle construction $\alpha : M$.

$$
\begin{array}{rcll}
M, N & ::= & x & \text{Variable} \\
 & | & (\lambda x.N)^A & \text{Abstraction} \\
 & | & (MN)^A & \text{Application} \\
 & | & \alpha : M & \text{Intercalaire} \\
V, W & ::= & (\lambda x.N)^A \mid \alpha : V & \text{Valeurs} \\
\alpha, \beta & ::= & \lceil \alpha' \rceil \mid \lfloor \alpha' \rfloor & \text{Etiquettes atomiques} \\
\overrightarrow{\alpha} & ::= & \alpha_1 \alpha_2 \cdots \alpha_n \quad n \geq 0 & \text{Séquences d'étiquettes} \\
\alpha', \beta' & ::= & A \overrightarrow{\alpha} B & \text{Etiquettes composées}
\end{array}
$$

La syntaxe du λ_n-calcul étiqueté étend celle du λ_n-calcul avec les intercalaires qui sont des termes étiquetés par une étiquette atomique. Le principal A est le principal de l'abstraction $(\lambda x.N)^A$ et de l'application $(MN)^A$. On note qu'un terme du λ_n-calcul est un terme du λ_n-calcul étiqueté. Une étiquette atomique peut être une étiquette composée surlignée $\lceil \alpha' \rceil$ ou soulignée $\lfloor \alpha' \rfloor$. Les séquences d'étiquettes sont utilisées pour simplifier les notations quand des étiquettes atomiques s'empilent. On a $\overrightarrow{\alpha} \circ M = \alpha_1 : \alpha_2 : \ldots : \alpha_n : M$ avec $\overrightarrow{\alpha} = \alpha_1 \alpha_2 \cdots \alpha_n$. La séquence vide est notée $\perp$. On utilise la notation de concaténation pour mettre bout à bout deux séquences : si $\overrightarrow{\alpha} = \alpha_1 \cdots \alpha_n$ et $\overrightarrow{\beta} = \beta_1 \cdots \beta_m$, alors $\overrightarrow{\alpha}\,\overrightarrow{\beta} = \alpha_1 \cdots \alpha_n \beta_1 \cdots \beta_m$. Une étiquette composée est une séquence d'étiquettes atomiques encadrée par deux principaux. On les utilise pour nommer les radicaux. On utilise l'opérateur τ, défini ci-dessous, pour obtenir la séquence d'étiquettes en tête d'un terme.

$$
\begin{array}{rcrcl}
\tau(x) & = & \perp & \qquad & \tau((\lambda x.M)^A) = \perp \\
\tau((MN)^A) & = & \perp & & \tau(\alpha : M) = \alpha \tau(M)
\end{array}
$$

Les principaux portés par les termes du λ_n-calcul étiqueté n'interviennent pas dans l'étiquette de tête. Seules les étiquettes atomiques des intercalaires sont retenues. On adapte la définition de la fonction $|\cdot|$. Dans le cadre présent, cette fonction associe à une séquence d'étiquettes ou à une étiquette atomique ou composée, l'ensemble des principaux qui la constituent.

$$
\begin{array}{rcrcl}
|\lceil \alpha' \rceil| & = & |\alpha'| & \qquad & |\lfloor \alpha' \rfloor| = |\alpha'| \\
|A \overrightarrow{\alpha} B| & = & \{A, B\} \cup |\overrightarrow{\alpha}| & & |\alpha_1 \ldots \alpha_n| = \bigcup_{i=1}^{n} |\alpha_i|
\end{array}
$$

Intuitivement, les principaux sont des marqueurs *statiques* qui donnent l'origine des sous-termes. Ces marqueurs ne participent pas au calcul. Par contraste, les étiquettes sont *dynamiques* ; elles contiennent l'histoire du calcul. Cette histoire est enrichie à chaque étape de réduction comme le montre la règle de réduction de base : la β_{ne}-réduction.

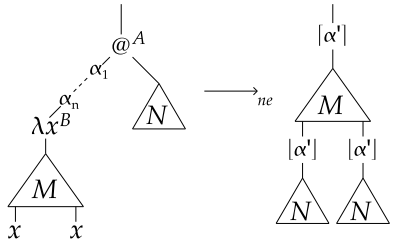


FIG. 6.6 – *β_{ne}-réduction*

$$(\beta_{ne}) \qquad (\overrightarrow{\alpha} \circ (\lambda x.M)^B N)^A \ \rightarrow_{ne} \ \lceil \alpha' \rceil : M\{x \backslash \lfloor \alpha' \rfloor : N\}$$
$$\text{où } \alpha' = A\overrightarrow{\alpha}B$$
$$\text{et } \mathsf{nom}((\overrightarrow{\alpha} \circ (\lambda x.M)^B N)^A) = \alpha'$$

Comme le montre la figure 6.6, le corps de la fonction est encadré par le nom du radical qui est surligné en tête du terme et souligné sur les arêtes qui menaient aux variables. De cette façon, le nom d'un radical créé par le haut (respectivement par le bas) par la contraction du radical de nom α' contient l'étiquette atomique $\lceil \alpha' \rceil$ (resp. $\lfloor \alpha' \rfloor$). On observe que, comme dans le λ-calcul, si R' est un résidu de R, ces radicaux ont le même nom : $\mathsf{nom}(R') = \mathsf{nom}(R)$. Les réductions de contexte (ν_{ne}), (μ_{ne}) et (ξ_{ne}) sont de simples adaptations du λ_n-calcul.

$$(\nu_{ne}) \ \frac{M \rightarrow_{ne} M'}{(MN)^A \rightarrow_{ne} (M'N)^A} \qquad (\mu_{ne}) \ \frac{N \rightarrow_{ne} N'}{(MN)^A \rightarrow_{ne} (MN')^A} \qquad (\xi_{ne}) \ \frac{M \rightarrow_{ne} M'}{(\lambda x.M)^A \rightarrow_{ne} (\lambda x.M')^A}$$

Dans le λ-calcul étiqueté, les étiquettes contiennent l'histoire des interactions. Par définition de la règle (β_{ne}), si le radical R_0 crée le radical R, alors le nom de R contient le nom de R_0. Plus généralement, le nom d'un radical R contient les noms des radicaux dont la contraction passée a contribué à créer R. Les étiquettes rendent donc compte des dépendances entre radicaux. Pour exprimer formellement cette notion de dépendance, on introduit la relation $\preceq$ sur les séquences d'étiquettes atomiques, donc sur les noms des radicaux.

$$\begin{array}{ll} \alpha' \preceq \alpha' & \\ \alpha' \preceq A\alpha_1 \ldots \alpha_n B & \text{si } \exists i \in \{1 \ldots n\} \,.\, \alpha_i = \lceil \beta' \rceil \text{ et } \alpha' \preceq \beta' \\ \alpha' \preceq A\alpha_1 \ldots \alpha_n B & \text{si } \exists i \in \{1 \ldots n\} \,.\, \alpha_i = \lfloor \beta' \rfloor \text{ et } \alpha' \preceq \beta' \end{array}$$

Lemme 6.1 *La relation $\preceq$ est un ordre bien fondé.*

Preuve : La relation $\preceq$ est réflexive et anti-symétrique. On montre que cette relation est transitive. On suppose $\alpha' \preceq \beta'$ et $\beta' \preceq \gamma'$. On procède par récurrence sur la profondeur maximale d'imbrication de soulignement ou de surlignement dans $\gamma' = C\gamma_1 \ldots \gamma_n D$. Cette profondeur est notée p. Le cas de base est élémentaire : on a $\alpha' = \beta' = \gamma' = CD$. Si $p > 0$, deux cas sont à envisager. (1) Si $\alpha' = \beta'$ ou $\beta' = \gamma'$, alors le résultat est élémentaire. (2) Sinon, on a $\beta' = A\alpha_1 \ldots \alpha_m B$ et il existe deux étiquettes composées δ' et η' et deux indices i et j tels que :

1. $\alpha' \preceq \delta'$
2. $\beta_i = \lceil \delta' \rceil$ ou $\beta_i = \lfloor \delta' \rfloor$
3. $\beta' \preceq \eta'$
4. $\gamma_j = \lceil \eta' \rceil$ ou $\gamma_j = \lfloor \eta' \rfloor$

On a aussi $\delta' \preceq \beta'$. Comme la profondeur de η' est strictement inférieure à celle de γ', on obtient, par récurrence, $\delta' \preceq \eta'$ puis $\alpha' \preceq \eta'$. Comme $\gamma_j = \lceil \eta' \rceil$ ou $\gamma_j = \lfloor \eta' \rfloor$, on obtient bien $\alpha' \preceq \gamma'$. Cette démonstration montre que $\preceq$ est donc un ordre bien fondé. □

L'ordre bien fondé $\preceq$ est la relation d'imbrication d'étiquettes composées. La relation $\alpha' \prec \beta'$ signifie intuitivement que le radical de nom β' a été créé (directement ou indirectement) par la

contraction d'un radical de nom α'. Cet ordre rend donc bien compte des dépendances d'un radical vis-à-vis des contractions passées. On en déduit que les noms des radicaux fournissent un accès direct à l'histoire du calcul, ce qui permettra de définir formellement la politique de la Muraille de Chine.

Comme la syntaxe du λ_n-calcul étiqueté est une extension de la syntaxe du λ_n-calcul, la notion de réduction indépendante des interactions entre les principaux A et B est conservée telle quelle. Le λ_n-calcul étiqueté, qui n'est qu'une variante syntaxique du λ-calcul étiqueté, vérifie les propriétés fondamentales du λ-calcul étiqueté.

Théorème 6.4 (Confluence) *Si $M \twoheadrightarrow_{ne} M_1$ et $M \twoheadrightarrow_{ne} M_2$, alors il existe un terme N tel que $M_1 \twoheadrightarrow_{ne} N$ et $M_2 \twoheadrightarrow_{ne} N$.*

Théorème 6.5 (Développements finis) *Soit $\mathcal{F}$ un ensemble de β_{ne}-radicaux de M.*

1. *Les réductions relatives à $\mathcal{F}$ sont de longueur finie.*
2. *Tous les développements de $\mathcal{F}$ finissent sur un même terme N.*
3. *L'ensemble des résidus d'un β_{ne}-radical R de M dans N est indépendant du développement considéré.*

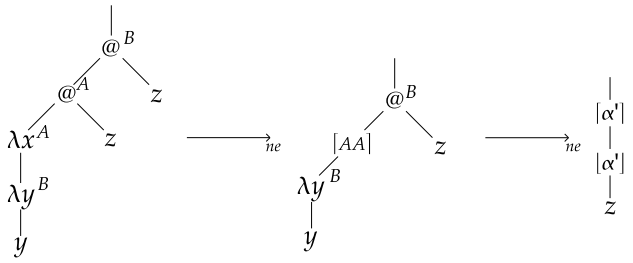
Théorème 6.6 (Standardisation)

1. *Si $M \twoheadrightarrow_{ne} N$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_{ne} N$ telle que $\mathcal{R}$ est standard.*
2. *Si $M \twoheadrightarrow_{ne} V$, il existe une réduction $\mathcal{R} : M \twoheadrightarrow_{ne} W$ telle que $\mathcal{R}$ est une réduction de tête.*

Preuve : On adapte à la syntaxe du λ_n-calcul étiqueté les preuves correspondantes du λ-calcul étiqueté [29]. □

Le λ_n-calcul étiqueté est un calcul confluent, qui vérifie les théorèmes des développements finis et de standardisation. Le théorème des développements finis nous permet d'introduire une nouvelle notation. La réduction *complète* $M \overset{\alpha'}{\Rightarrow}_{ne} M'$ est un développement des radicaux de M dont le nom est α'. Du fait du théorème des développement finis, cette notation est correcte puisque le terme M' final ne dépend pas du développement choisi.

Les étiquettes du λ_n-calcul expriment les dépendances vis-à-vis des réductions passées. On peut les utiliser pour formaliser des politiques de sécurité qui font appel à l'histoire des événements passés, comme la Muraille de Chine ou les Enchères scellées. Dans le cadre du λ_n-calcul étiqueté, une **politique de sécurité** consiste à autoriser ou interdire la contraction des radicaux en fonction de leur nom, c'est-à-dire leur histoire. Comme illustration, nous examinons plus particulièrement le cas de la Muraille de Chine dans le cadre simplifié que nous avons présenté en préambule de cette partie (p.144). Dans ce cas, deux principaux jouent un rôle central A (Alice) et B (Bob). Ces principaux représentent des concurrents. Les autres principaux, par exemple C (Charlie), sont des partenaires économiques potentiels pour Alice et Bob. Le but de la Muraille de Chine est d'interdire les interactions directes ou indirectes entre Alice et Bob. Pour ce faire, cette politique de sécurité consiste à créer un mur entre les principaux ayant interagi avec Alice et ceux ayant interagi avec Bob : ceci justifie le nom de cette politique de sécurité. Plus concrètement, les interactions directes entre Alice et Bob sont interdites : on autorise seulement la contraction de radicaux qui ignorent soit A soit B. Mais la Muraille de Chine, telle qu'elle est définie par Brewer et Nash [10] interdit aussi l'interaction en Alice et Charlie si ce dernier a déjà interagi avec Bob, et inversement : intuitivement, on veut pas que des informations confidentielles passent d'Alice à Bob par l'intermédiaire d'un consultant. Dans le cadre du λ_n-calcul étiqueté, le fait qu'Alice et Charlie ont interagi s'exprime par le fait que les principaux A et C sont présents dans une étiquette d'un terme et situés dans un même souligné ou un même surligné. Par conséquent, si cette situation se



$$\alpha' = B\lceil AA\rceil B$$

FIG. 6.7 – *Réduction de* $(((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$ *dans le* λ_n*-calcul étiqueté*

produit, on pourrait vouloir interdire les interactions ultérieures entre Bob et Charlie, c'est-à-dire la contraction des radicaux dont les noms contiennent à la fois B et C. En effet, la contraction d'un tel radical créerait un surligné sous lequel B et C seraient présents. Cette mesure est en réalité trop restrictive. En effet, on ne veut pas interdire les interactions entre Bob et Charlie mais interdire les interactions entre Bob et *le Charlie qui a interagi avec Alice*. En effet, si le principal C est présent dans plusieurs sous-termes, ces derniers sont indépendants du point de vue du calcul : informellement, il n'y a pas de communication entre les sous-termes dont les étiquettes contiennent le même principal. Dans cette optique, on interdit simplement les interactions entre Alice et Charlie si *ce* Charlie a interagi dans le passé avec Bob : cette interaction passée entre Bob et Charlie se traduit dans un nom de radical par une étiquette soulignée ou surlignée. Par conséquent, on interdit la contraction des radicaux dont le nom contient A et B. Plus formellement, on définit la politique de sécurité de la Muraille de Chine en définissant les étiquettes $\mathcal{M}(A,B)$-compatibles, c'est-à-dire les étiquettes des radicaux dont on autorise la contraction.

Définition 6.3 *L'étiquette composée* α' *est* $\mathcal{M}(A,B)$*-compatible si et seulement si* $\{A, B\} \not\subseteq |\alpha'|$.

Les étiquettes $\mathcal{M}(A,B)$-compatibles ne contiennent pas à la fois le principal A et le principal B. Ceci permet de définir ce qu'est une réduction qui vérifie la politique de la Muraille de Chine $\mathcal{M}(A,B)$.

Définition 6.4 (Muraille de Chine $\mathcal{M}(A,B)$**)** *Une réduction* $\mathcal{R}$ *est conforme à la Muraille de Chine pour* A *et* B*, ce que l'on note* $\mathcal{M}(A,B)$*, si et seulement si tous les radicaux réduits au cours de la réduction sont* $\mathcal{M}(A,B)$*-compatibles.*

Pour illustrer cette définition, on reprend, sur la figure 6.7, l'exemple de la figure 6.4 dans le cadre du λ_n-calcul étiqueté. La première réduction est $\mathcal{M}(A,B)$-conforme puisque le nom du radical contracté est AA. En revanche, la deuxième réduction n'est pas $\mathcal{M}(A,B)$-conforme car l'étiquette $B\lceil AA\rceil B$ n'est pas $\mathcal{M}(A,B)$-compatible.

On souhaite relier les étiquettes, qui expriment les dépendances vis-à-vis des réductions passées, avec la notion de réduction indépendante des interactions entre deux principaux A et B. Comme le montre le diagramme 6.5 (p.148), la propriété d'indépendance suppose de pouvoir effectuer en premier les réductions ignorant A ou, indifféremment, les réductions ignorant B. Ceci exprime concrètement le fait que les réductions qui ignorent A ne dépendent pas des réductions qui ignorent B. Dans le cadre du λ_n-calcul étiqueté, si la contraction d'un radical R dépend de la contraction du radical R_0, c'est-à-dire si cette dernière crée R, alors on a la relation $\mathsf{nom}(R_0) \prec \mathsf{nom}(R)$. En appliquant cette remarque sur une réduction indépendante de l'interaction entre A et B, on déduit que les étiquettes atomiques créées par la réduction qui ignore A sont intuitivement indépendantes de celles créées par la réduction qui ignore B. Les étiquettes créées au cours de la réduction sont

donc constituées d'étiquettes atomiques créées séparément par ces deux réductions. Pour formaliser cette intuition, on introduit la notion suivante de séparation de principaux.

Définition 6.5 (Séparation de principaux) *La séquence d'étiquettes $\alpha_1 \dots \alpha_n$ sépare les principaux A et B si et seulement si pour $1 \leq i \leq n$, on a $A \notin |\alpha_i|$ ou $B \notin |\alpha_i|$.*

De façon plus intuitive, une séquence d'étiquettes $\overrightarrow{\alpha}$ *sépare les principaux A et B* si et seulement si les principaux A et B ne sont pas présents dans $\overrightarrow{\alpha}$ dans un même souligné ou surligné. En revanche, ces principaux peuvent être présents dans des soulignés ou des surlignés séparés. Plus concrètement, la séquence d'étiquettes $\overrightarrow{\alpha} = \lceil AB \rceil \lfloor CD \rfloor$ ne sépare pas les principaux A et B. En revanche, cette séquence sépare bien A et C. Ceci illustre bien les origines séparées et indépendantes des étiquettes issues de la réduction qui ignore A et celles issues de la réduction qui ignore B.

On souhaite montrer, dans un premier temps, que (sous certaines hypothèses) si une réduction indépendante de l'interaction entre A et B aboutit à une valeur V, alors l'étiquette de tête de V sépare A et B. Cette démonstration fait appel à deux invariants : $\mathcal{H}_A$ et $\mathcal{K}_{AB}$.

Invariant 6.1 *Un terme M vérifie l'invariant $\mathcal{H}_A$, ce que l'on note $\mathcal{H}_A(M)$, si et seulement si tous les sous-termes de M de la forme $\alpha : N$ vérifient $A \notin |\alpha|$.*

Cet invariant interdit au principal A d'apparaître dans une étiquette atomique, bien qu'il puisse apparaître comme principal d'un sous-terme. Intuitivement, les étiquettes atomiques ont été créées par les réductions passées. Le fait que le principal A n'intervienne pas dans une étiquette atomique signifie que ce principal n'a pas joué de rôle dans le passé de M. Si $\mathcal{H}_A(M)$, on dira que **le principal A n'est pas intervenu dans l'histoire de M**. On remarque que les termes du λ_n-calcul qui ne contiennent pas d'intercalaire, vérifient l'invariant $\mathcal{H}_A$. Le lemme suivant formalise cette interprétation intuitive de l'invariant $\mathcal{H}_A$.

Lemme 6.2 (Préservation de $\mathcal{H}_A$ par $\xrightarrow{\neg A}_{ne}$) *Si $\mathcal{H}_A(M)$ et $M \xrightarrow{\neg A}_{ne} M'$, alors $\mathcal{H}_A(M')$ et le nom du radical contracté ne contient pas A.*

Preuve : Soit R le radical contracté entre M et M'. Soit $\alpha : N'$ un sous-terme de M'. Deux cas sont possibles : (1) Il existe un sous-terme de $\alpha : N$ dans M. On conclut alors par $\mathcal{H}_A(M)$. (2) L'étiquette α est créée par la contraction de R. Par conséquent $\alpha = \lceil \mathsf{nom}(R) \rceil$ ou $\alpha = \lfloor \mathsf{nom}(R) \rfloor$. L'hypothèse $A \notin \mathsf{nom}(R)$ permet de conclure. □

L'invariant $\mathcal{H}_A$ est préservé par une réduction ignorant A. Intuitivement, ce lemme indique que si A n'est pas intervenu dans l'histoire de M et que la réduction entre M et M' ignore A, alors A n'est pas intervenu dans l'histoire de M'. Ceci justifie bien l'interprétation de $\mathcal{H}_A$. On introduit le deuxième invariant $\mathcal{K}_{AB}$.

Invariant 6.2 *Un terme M vérifie l'invariant $\mathcal{K}_{AB}$, ce que l'on note $\mathcal{K}_{AB}(M)$, si et seulement si toutes les étiquettes α, qui sont présentes dans un sous-terme de la forme $\alpha : N$, séparent les principaux A et B.*

Cet invariant impose que toutes les étiquettes du terme séparent A et B, bien que ces principaux puissent apparaître en tant que principal d'un sous-terme. Intuitivement, ceci signifie que les principaux A et B ne sont pas intervenus ensemble dans une réduction passée. Si $\mathcal{K}_{AB}(M)$, on dira que **A et B n'ont pas collaboré dans l'histoire de M**. Cette interprétation est justifiée par le lemme suivant.

Lemme 6.3 (Préservation de $\mathcal{K}_{AB}$) *Si on a $\mathcal{K}_{AB}(M)$ et $M \xrightarrow{R}_{ne} M'$ et si $A \notin |\mathsf{nom}(R)|$ ou $B \notin |\mathsf{nom}(R)|$, alors on a $\mathcal{K}_{AB}(M')$.*

Preuve : On pose $R = (\overrightarrow{\alpha} \circ (\lambda x.N)^D N')^C$. On suppose $A \notin |\mathsf{nom}(R)| = |C \overrightarrow{\alpha} D|$ où $\overrightarrow{\alpha} = \alpha_1 \dots \alpha_n$ (le cas $B \notin |\mathsf{nom}(R)|$ est similaire). Soit $\alpha : P'$ un sous-terme de M'. Deux cas sont possibles. (1) Il existe un sous-terme $\alpha : P$ dans M. On conclut alors par $\mathcal{K}_{AB}(M)$. (2) L'étiquette α est créée par la contraction de R. Par conséquent, on a $\alpha = \lceil \mathsf{nom}(R) \rceil$ ou $\alpha = \lfloor \mathsf{nom}(R) \rfloor$. Par hypothèse on a

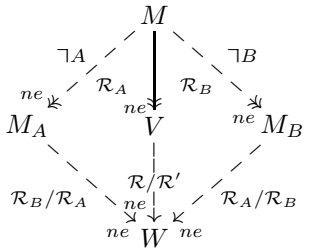
$A \notin |\mathsf{nom}(R)|$, ce qui implique : α sépare A et B. □

L'invariant $\mathcal{K}_{AB}$ est préservé par la réduction qui contracte un radical dont le nom ne contient pas simultanément A et B. Intuitivement, dans cette réduction, A et B ne collaborent pas. Comme ces principaux n'ont pas collaboré dans l'histoire de M, alors A ou B n'ont pas collaboré dans l'histoire de M'.

Ces lemmes élémentaires aboutissent à la propriété recherchée : sous certaines hypothèses initiales, si une réduction indépendante de l'interaction entre les principaux A et B aboutit à une valeur, alors l'étiquette de tête de cette valeur sépare A et B.

Théorème 6.7 (Séparation) *Si $H_A(M)$ et $H_B(M)$ et s'il existe une réduction $M \twoheadrightarrow_{ne} V$ indépendante de l'interaction entre A et B, alors l'étiquette $\tau(V)$ sépare A et B.*

Preuve : On pose $\mathcal{R} : M \twoheadrightarrow_{ne} V$. Par définition d'une réduction indépendante de l'interaction entre A et B, il existe deux réductions $\mathcal{R}_A : M \xrightarrow{\neg A}\!\!\!\!\!\twoheadrightarrow_{ne} M_A$, $\mathcal{R}_B : M \xrightarrow{\neg B}\!\!\!\!\!\twoheadrightarrow_{ne} M_B$ et une valeur W qui vérifient $M_A \xrightarrow{\mathcal{R}_B/\mathcal{R}_A}_{ne} W$, $M_B \xrightarrow{\mathcal{R}_A/\mathcal{R}_B}_{ne} W$ et $\mathcal{R} \leq \mathcal{R}' = \mathcal{R}_A \sqcup \mathcal{R}_B$.



Soient $\alpha_1, \ldots, \alpha_n$ (respectivement $\beta_1, \ldots, \beta_m$) les noms des radicaux contractés au cours de la réduction $\mathcal{R}_A$ (resp. $\mathcal{R}_B$). En utilisant le lemme 6.2, on obtient $\mathcal{H}_A(M_A)$ et, pour $1 \leq i \leq n$, on a $A \notin |\alpha_i|$. Symétriquement, on obtient $\mathcal{H}_B(M_B)$ et, pour $1 \leq j \leq m$, on a $B \notin |\beta_j|$. Les noms des résidus contractés au cours de la réduction $\mathcal{R}_B/\mathcal{R}_A$ forment un sous-ensemble de $\{\beta_j \mid 1 \leq j \leq m\}$. En particulier, ces noms ne contiennent pas le principal B. Comme $\mathcal{H}_A$ implique $\mathcal{K}_{AB}$, en utilisant itérativement le lemme 6.3, on obtient $\mathcal{K}_{AB}(W)$. Du fait de la confluence du langage, toutes les valeurs issues de M ont la même étiquette de tête, ce qui donne $\tau(V) = \tau(W)$. On a donc $\tau(V)$ sépare A et B. □

Les hypothèses initiales sont $\mathcal{H}_A$ et $\mathcal{H}_B$. En d'autres termes, les principaux A et B ne doivent pas être intervenus dans l'histoire du terme initial. Ce résultat est un premier lien entre la notion de réduction indépendante de l'interaction entre A et B et la notion d'étiquette séparant A et B.

On examine désormais la propriété réciproque. On souhaite montrer que si un terme M se réduit vers une valeur dont l'étiquette de tête sépare A et B, alors il existe une réduction aboutissant à une valeur qui est indépendante de l'interaction entre A et B. Comme on souhaite introduire une réduction aboutissant à une valeur, il est naturel de considérer la réduction *minimale* issue de M qui aboutit à une valeur. Pour cela, on fait appel à un corollaire du théorème de standardisation qui correspond au lemme 2.16 et dont la preuve s'adapte aisément au cadre présent.

Lemme 6.4 *Si $\mathcal{R} : M \twoheadrightarrow_{ne} V$ est une réduction standard, cette réduction peut se décomposer en $\mathcal{R} : M \twoheadrightarrow_{ne} W \twoheadrightarrow_{ne} V$ où $M \twoheadrightarrow_{ne} W$ est une réduction de tête.*

Preuve : On adapte la preuve du lemme 2.16 au λ_n-calcul étiqueté. □

Une réduction standard qui aboutit à une valeur peut s'écrire comme la composition d'une réduction de tête et d'une réduction standard. En combinant ce résultat au fait que le nom d'un radical créé contient le nom du radical qui l'a créé, on obtient le lemme suivant.

Lemme 6.5 *Si $\mathcal{R} : M \xrightarrow{R_1}_{ne} M_1 \xrightarrow{R_2}_{ne} \ldots \xrightarrow{R_{n-1}}_{ne} M_{n-1} \xrightarrow{R_n}_{ne} V$ est une réduction en tête où M_{n-1} n'est pas une valeur, alors, pour i tel que $1 \leq i \leq n$, on a $\mathsf{nom}(R_i) \prec \tau(V)$.*

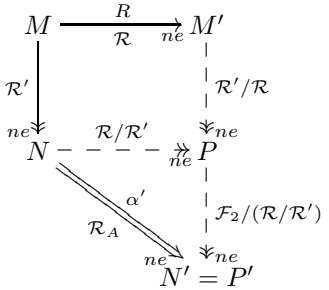
Preuve : On procède par récurrence sur la longueur n de la réduction $\mathcal{R}$. Si $n = 0$, le résultat est trivial. On suppose $n > 0$ et on procède par induction sur M.

1. Les cas $M = x$ et $M = (\lambda x.M')^A$ sont exclus. Le cas $M = \alpha : N$ se traite par une induction élémentaire.
2. Si $M = (M'M'')^A$, alors, comme la réduction $\mathcal{R}$ aboutit à une valeur, cette réduction de tête s'écrit nécessairement (avec $\alpha' = A\overrightarrow{\alpha}B$ et $1 \leq k \leq n-1$)
$$\mathcal{R} : M = (M'M'')^A \xrightarrow{R_1}_{ne} \ldots \xrightarrow{R_k}_{ne} (\overrightarrow{\alpha} \circ (\lambda x.N)^A M'')^B \xrightarrow{R_{k+1}}_{ne} \lceil \alpha' \rceil : N\{x \backslash \lfloor \alpha' \rfloor : M''\} \xrightarrow{R_{k+2}}_{ne} \ldots \xrightarrow{R_n}_{ne} V$$
où la réduction $\mathcal{R}' : M' \xrightarrow{R_1}_{ne} \ldots \xrightarrow{R_k}_{ne} \overrightarrow{\alpha} \cdot (\lambda x.N)^A$ est une réduction de tête dont les termes intermédiaires ne sont pas des valeurs. Par hypothèse de récurrence sur $\mathcal{R}'$, on obtient $\mathsf{nom}(R_i) \prec \overrightarrow{\alpha}$ pour i tel que $1 \leq i \leq k$. On applique à nouveau l'hypothèse de récurrence sur la réduction $M_{k+1} = \lceil \alpha' \rceil : N\{x \backslash \lfloor \alpha' \rfloor : M''\} \xrightarrow{R_{k+2}}_{ne} \ldots \xrightarrow{R_n}_{ne} V$. Si i vérifie $k+2 \leq i \leq n$, on obtient $\mathsf{nom}(R_i) \prec \tau(V)$. L'étiquette de tête de V est de la forme $\tau(V) = \lceil \alpha' \rceil \overrightarrow{\beta}$. En résumé, on a obtenu les propriétés suivantes.
   (a) Si $1 \leq i \leq k$, alors on a $\mathsf{nom}(R_i) \prec \overrightarrow{\alpha} \prec \tau(V)$.
   (b) Si $i = k+1$, alors on a $\mathsf{nom}(R_{k+1}) = \alpha' \prec \tau(V)$.
   (c) Si $k+1 \leq i \leq n$, alors on a $\mathsf{nom}(R_i) \prec \tau(V)$. □

Tous les noms des radicaux qui interviennent dans la réduction (de tête) pour obtenir une valeur sont strictement contenus dans l'étiquette de tête de la valeur obtenue. En d'autres termes, l'étiquette de tête contient les noms des radicaux dont la contraction est nécessaire pour obtenir une valeur. Bien entendu, il peut ne pas être nécessaire de contracter tous les radicaux portant un nom mentionné dans l'étiquette de tête pour obtenir une valeur. Cependant, cette opération est suffisante pour obtenir une valeur, comme le prouve le résultat suivant.

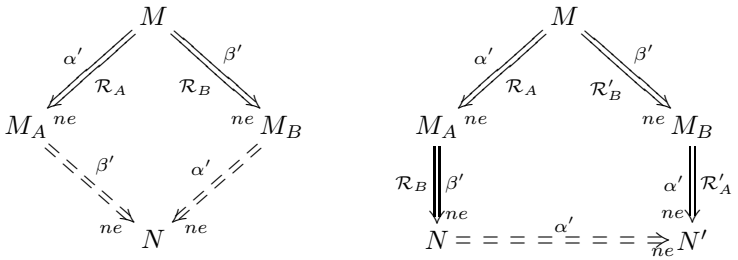
Lemme 6.6 *Si* $\mathcal{R} : M \xrightarrow{R_1}_{ne} M_1 \xrightarrow{R_2}_{ne} \ldots \xrightarrow{R_n}_{ne} M_n$ *et si, pour* $1 \leq i \leq n$, *on pose* $\alpha'_i = \mathsf{nom}(R_i)$, *alors on a* $\mathcal{R}' : M \overset{\alpha'_1}{\Longrightarrow}_{ne} M'_1 \overset{\alpha'_2}{\Longrightarrow}_{ne} \ldots \overset{\alpha'_n}{\Longrightarrow}_{ne} M'_n$ *et* $\mathcal{R} \leq \mathcal{R}'$.

Preuve : On montre la propriété intermédiaire suivante : si $\mathcal{R} : M \xrightarrow{R}_{ne} M'$ (où $\mathsf{nom}(R) = \alpha'$), si $\mathcal{R}' : M \twoheadrightarrow_{ne} N$ et si $\mathcal{R}_A : N \overset{\alpha'}{\Longrightarrow}_{ne} N'$, alors on a $\mathcal{R} \leq (\mathcal{R}'; \mathcal{R}_A)$.
Soit $\mathcal{F}$ l'ensemble des radicaux de N dont le nom est α'. On partitionne cet ensemble en deux ensembles disjoints $\mathcal{F}_1 = R/\mathcal{R}'$ et $\mathcal{F}_2 = \mathcal{F} - \mathcal{F}_1$. Soit P le terme tel que $(\mathcal{R}/\mathcal{R}') : N \twoheadrightarrow_{ne} P$ et $(\mathcal{R}'/\mathcal{R}) : M' \twoheadrightarrow_{ne} P$. La réduction $\mathcal{R}/\mathcal{R}'$ est un développement de $\mathcal{F}_1$. Soit P' le terme vérifiant $\mathcal{R}''' : P \xrightarrow{\mathcal{F}_2/(\mathcal{R}/\mathcal{R}')}_{ne} P'$. La réduction $(\mathcal{R}/\mathcal{R}'; \mathcal{R}''')$ est un développement de $\mathcal{F}$. On obtient donc $P' = N'$ et $\mathcal{R}_A \sim ((\mathcal{R}/\mathcal{R}'); \mathcal{R}''')$. Ceci implique $\mathcal{R}/(\mathcal{R}'; \mathcal{R}_A) = (\mathcal{R}/\mathcal{R}')/\mathcal{R}_A \sim \emptyset$ et donc $\mathcal{R} \leq (\mathcal{R}'; \mathcal{R}_A)$.



On considère la réduction $\mathcal{R}' : M \overset{\alpha'_1}{\Longrightarrow}_{ne} M'_1 \overset{\alpha'_2}{\Longrightarrow}_{ne} \ldots \overset{\alpha'_n}{\Longrightarrow}_{ne} M'_n$. En utilisant itérativement la propriété intermédiaire, on obtient $\mathcal{R} \leq \mathcal{R}'$. □

Si $\mathcal{R}$ est une réduction, en transformant chaque pas de réduction en réduction complète, la réduction



Lemme 6.7
$\alpha' \not\prec \beta'$ et $\beta' \not\prec \alpha'$

Lemme 6.8
$\beta' \prec \alpha'$

FIG. 6.8 – *Permutation de réductions complètes*

$\mathcal{R}'$ obtenue contient la réduction $\mathcal{R}$ au sens où $\mathcal{R} \leq \mathcal{R}'$. Intuitivement, en effectuant les réductions complètes des radicaux de nom α'_i dans le même ordre que les radicaux R_i, on s'assure à chaque étape complète que les radicaux résidus de R_i ont bien été créés. En effet, l'ordre de contraction des ensembles de radicaux α'_i est important. En effet, si le radical R_j est créé par la contraction du radical R_i où $i < j$, effectuer la réduction complète $\overset{\alpha'_j}{\Rightarrow}_{ne}$ avant d'avoir effectuer $\overset{\alpha'_i}{\Rightarrow}_{ne}$ n'a pas de sens puisque les radicaux de nom α'_j n'ont pas encore été créés par les radicaux nommés α'_i.

Le résultat précédent nous permet de travailler sur des séquences de réductions complètes. Ces dernières présentent l'avantage de vérifier des propriétés de permutation simples, comme le montrent les lemmes suivants.

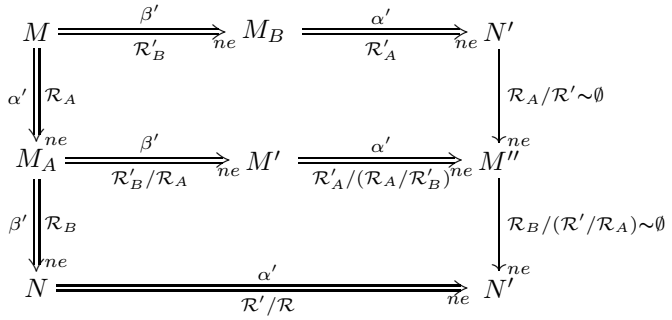
Lemme 6.7 *On suppose $\alpha' \not\prec \beta'$ et $\beta' \not\prec \alpha'$. On pose $\mathcal{R}_A : M \overset{\alpha'}{\Rightarrow}_{ne} M_A$ et $\mathcal{R}_B : M \overset{\beta'}{\Rightarrow}_{ne} M_B$. La réduction $\mathcal{R}_A/\mathcal{R}_B$ (respectivement $\mathcal{R}_B/\mathcal{R}_A$) est un développement complet des radicaux de nom α' (resp. β').*

Preuve : Soit $\mathcal{F}_A$ (respectivement $\mathcal{G}_A$) l'ensemble des radicaux de M (resp. M_B) de nom α'. Comme $\mathcal{R}_A$ est un développement de $\mathcal{F}_A$, la réduction $\mathcal{R}_A/\mathcal{R}_B$ est un développement de $\mathcal{F}'_A = \mathcal{F}_A/\mathcal{R}_B$. Les résidus de $\mathcal{F}_A$ portent le même nom. Par conséquent tous les radicaux de $\mathcal{F}'_A$ ont α' pour nom. Réciproquement, on considère un radical R de $\mathcal{G}_A$. Si R était créé au cours la réduction $\mathcal{R}_B$, le nom de R contiendrait strictement β' ce qui est exclu par l'hypothèse $\alpha' \not\prec \beta'$. R est donc un résidu d'un radical R_0 de M. R_0 a le même nom que son résidu R : on a $\mathsf{nom}(R_0) = \alpha'$. On en déduit $\mathcal{F}'_A = \mathcal{G}_A$ et donc $\mathcal{R}_A/\mathcal{R}_B$ est un développement complet des radicaux de M_A de nom α'. Par le même raisonnement, on obtient que $\mathcal{R}_B/\mathcal{R}_A$ est un développement complet des radicaux de M_B de nom β'. □

Si α' et β' sont incomparables par la relation $\preceq$, alors on peut permuter les réductions complètes $\overset{\alpha'}{\Rightarrow}_{ne}$ et $\overset{\beta'}{\Rightarrow}_{ne}$. Plus généralement, ce résultat s'inscrit dans le cadre de la théorie du treillis des réductions complètes développées dans [29]. On obtient un résultat similaire dans le cas où il existe une relation entre les deux noms α' et β'.

Lemme 6.8 *Si $\mathcal{R} : M \overset{\alpha'}{\Rightarrow}_{ne} M_A \overset{\beta'}{\Rightarrow}_{ne} N$ et $\beta' \prec \alpha'$, les réductions $\mathcal{R}' : M \overset{\beta'}{\Rightarrow}_{ne} M_B \overset{\alpha'}{\Rightarrow}_{ne} N'$ et $\mathcal{R}'' : N \overset{\alpha'}{\Rightarrow}_{ne} N''$ vérifient $\mathcal{R} \leq \mathcal{R}'$ et $(\mathcal{R}; \mathcal{R}'') \sim \mathcal{R}'$.*

Preuve : Comme le montre la figure 6.8, les réductions $M \overset{\alpha'}{\Rightarrow}_{ne} M_A$ et $M_A \overset{\beta'}{\Rightarrow}_{ne} N$ sont respectivement nommées $\mathcal{R}_A$ et $\mathcal{R}_B$. La réduction $\mathcal{R}$ est donc la composition de $\mathcal{R}_A$ et $\mathcal{R}_B$. Les réductions $M \overset{\beta'}{\Rightarrow}_{ne} M_B$ et $M_B \overset{\alpha'}{\Rightarrow}_{ne} N'$ sont respectivement nommées $\mathcal{R}'_B$ et $\mathcal{R}'_A$; la réduction $\mathcal{R}'$ est la composition de $\mathcal{R}'_B$ et $\mathcal{R}'_A$.



On considère la réduction-résidu $\mathcal{R}/\mathcal{R}'$. On a $\mathcal{R}/\mathcal{R}' = (\mathcal{R}_A;\mathcal{R}_B)/\mathcal{R}' = (\mathcal{R}_A/\mathcal{R}');(\mathcal{R}_B/(\mathcal{R}'/\mathcal{R}_A))$. La réduction $\mathcal{R}_A$ contracte un ensemble $\mathcal{F}_A$ de radicaux de nom α'. Comme la réduction $\mathcal{R}'_A$ contracte tous les radicaux de nom α', on a $\mathcal{F}_A/\mathcal{R}' = \emptyset$ et $\mathcal{R}_A/\mathcal{R}' \sim \emptyset$. Par ailleurs, on a la relation $\mathcal{R}'/\mathcal{R}_A = (\mathcal{R}'_B/\mathcal{R}_A);(\mathcal{R}'_A/(\mathcal{R}'_B/\mathcal{R}_A))$. Comme $\mathcal{R}_A$ ne crée pas de radical de nom β', la réduction $\mathcal{R}'_B/\mathcal{R}_A$ contracte tous les radicaux de M_A portant le nom β'. Cette réduction aboutit à un terme M'. La réduction $\mathcal{R}'_A$ contracte tous les radicaux de M_B portant le nom α'. Comme la réduction $\mathcal{R}_A/\mathcal{R}'_B$ ne crée pas de radicaux de nom α', la réduction-résidu $\mathcal{R}'_A/(\mathcal{R}_A/\mathcal{R}'_B)$ contracte tous les radicaux de M' portant le nom α'. On a donc $(\mathcal{R}'/\mathcal{R}_A) : M_A \stackrel{\beta'}{\Rightarrow}_{ne} M' \stackrel{\alpha'}{\Rightarrow}_{ne} M''$. Comme M' ne contient aucun radical de nom β' et comme la contraction de radicaux de nom α' ne crée pas de radical de nom β', on en déduit que M'' ne contient aucun radical de nom β'. Par conséquent, la réduction $\mathcal{R}_B/(\mathcal{R}'/\mathcal{R}_A)$ est vide et on a $\mathcal{R} \leq \mathcal{R}'$. Par un raisonnement similaire, on montre que la réduction $\mathcal{R}'/\mathcal{R}$ contracte tous les radicaux de N de nom α' ce qui prouve $(\mathcal{R};\mathcal{R}'') \sim \mathcal{R}'$. □

Si $\beta' \prec \alpha'$ et si dans M on réduit complètement les radicaux α' puis les radicaux β', on obtient en changeant l'ordre des réductions complètes une réduction plus grande que la première réduction (au sens de l'ordre $\leq$ sur les réductions). Ces deux réductions ne sont pas nécessairement équivalentes puisque, en plus des résidus des radicaux α' déjà présents dans M, la réduction complète de $M_B \stackrel{\alpha'}{\Rightarrow}_{ne} N'$ contracte aussi les radicaux α' créés par la réduction complète des radicaux β'. Ces propriétés de permutation de réductions complètes sont combinées pour obtenir le résultat suivant, qui raffine le lemme 6.6.

Lemme 6.9 *Si* $\mathcal{R} : M \xrightarrow{R_1}_{ne} M_1 \xrightarrow{R_2}_{ne} \ldots \xrightarrow{R_n}_{ne} M_n$ *et si, pour* $1 \leq i \leq n$, *on pose* $\alpha'_i = \mathsf{nom}(R_i)$, *alors il existe une réduction* $\mathcal{R}' : M \stackrel{\beta'_1}{\Rightarrow}_{ne} M'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_p}{\Rightarrow}_{ne} M'_p$ *qui vérifie les propriétés suivantes.*

1. *La suite* $\{\beta'_i\}_{1\leq i\leq p}$ *est constituée d'éléments distincts de la suite* $\{\alpha'_i\}_{1\leq i\leq n}$.
2. *Si* $1 \leq i < j \leq p$, *alors on a* $\beta'_j \not\prec \beta'_i$.
3. *On a* $\mathcal{R} \leq \mathcal{R}'$.

Preuve : Par le lemme 6.6, on obtient une réduction $\mathcal{R}_0 : M \stackrel{\alpha'_1}{\Rightarrow}_{ne} M'_1 \stackrel{\alpha'_2}{\Rightarrow}_{ne} \ldots \stackrel{\alpha'_n}{\Rightarrow}_{ne} M'_n$ telle que $\mathcal{R} \leq \mathcal{R}_0$. On montre par récurrence sur le nombre de réductions complètes n de $\mathcal{R}_0$ qu'il existe une réduction $\mathcal{R}' : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_{p-1}}{\Rightarrow}_{ne} N'_{p-1} \stackrel{\beta'_p}{\Rightarrow}_{ne} W$ telle que (1) la suite $\{\beta'_i\}_{1\leq i\leq p}$ est constituée d'éléments de $\{\alpha'_i\}_{1\leq i\leq n}$, (2) si $i < j$, alors on a $\beta'_i \not\prec \beta'_j$ et (3) $\mathcal{R} \leq \mathcal{R}'$. Si $n = 2$, le résultat est obtenu par le lemme 6.8. On suppose maintenant $n > 2$. On considère la réduction $\mathcal{R}_1 : M \stackrel{\alpha'_1}{\Rightarrow}_{ne} M'_1 \stackrel{\alpha'_2}{\Rightarrow}_{ne} \ldots \stackrel{\alpha'_{n-1}}{\Rightarrow}_{ne} M'_{n-1}$. Par hypothèse de récurrence, il existe une réduction $\mathcal{R}''_1 : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_p}{\Rightarrow}_{ne} N'_p$ qui vérifie les propriétés suivantes :

1. La suite $\{\beta'_i\}_{1\leq i\leq p}$ est constituée d'éléments distincts de $\{\alpha'_i\}_{1\leq i\leq n-1}$.
2. Si $i < j$, alors on a $\beta'_j \not\prec \beta'_i$.
3. $\mathcal{R}_1 \leq \mathcal{R}''_1$

On considère la réduction $\mathcal{R}'_1 : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_p}{\Rightarrow}_{ne} N'_p \stackrel{\alpha'_n}{\Rightarrow}_{ne} N'_{p+1}$. Cette réduction vérifie $\mathcal{R}_0 \leq \mathcal{R}'_1$. Si pour tout i, on a $\alpha'_n \not\prec \beta'_i$, la propriété est montrée. Sinon, on pose i le plus grand

indice tel que $\alpha'_n \prec \beta'_i$. Si $i < j \leq p$, on a $\alpha'_n \not\prec \beta'_j$ et $\beta'_j \not\prec \beta'_i$. Si $\beta'_i \prec \beta'_j$, on aurait $\alpha'_n \prec \beta'_j$ ce qui est contradictoire. Par conséquent, on obtient la relation $\beta'_i \not\prec \beta'_j$. Avec le lemme 6.7, on permute l'ordre des réductions complètes pour obtenir la réduction suivante.

$$\mathcal{R}''_2 : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_{i-1}}{\Longrightarrow}_{ne} N'_{i-1} \stackrel{\beta_{i+1}}{\Longrightarrow}_{ne} \ldots \stackrel{\beta_p}{\Rightarrow}_{ne} P_p \stackrel{\beta'_i}{\Rightarrow}_{ne} N'_p$$

Cette réduction vérifie $\mathcal{R}''_1 \sim \mathcal{R}''_2$. En utilisant le lemme 6.8, on obtient la réduction suivante.

$$\mathcal{R}'_2 : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_{i-1}}{\Longrightarrow}_{ne} N'_{i-1} \stackrel{\beta_{i+1}}{\Longrightarrow}_{ne} \ldots \stackrel{\beta_p}{\Rightarrow}_{ne} P_p \stackrel{\alpha'_n}{\Rightarrow}_{ne} Q \stackrel{\beta_i}{\Rightarrow}_{ne} Q'$$

Cette réduction vérifie $\mathcal{R}'_1 \leq \mathcal{R}'_2$. On extrait de $\mathcal{R}'_2$ la réduction suivante.

$$\mathcal{R}''_3 : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N'_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_{i-1}}{\Longrightarrow}_{ne} N'_{i-1} \stackrel{\beta_{i+1}}{\Longrightarrow}_{ne} \ldots \stackrel{\beta_p}{\Rightarrow}_{ne} P_p \stackrel{\alpha'_n}{\Rightarrow}_{ne} Q$$

En appliquant la récurrence, on obtient une réduction $\mathcal{R}''_4 : M \stackrel{\gamma'_1}{\Rightarrow}_{ne} N''_1 \stackrel{\gamma'_2}{\Rightarrow}_{ne} \ldots \stackrel{\gamma_q}{\Rightarrow}_{ne} N''_q$ qui vérifie les points suivants.

1. La suite $\{\gamma'_i\}_{1\leq i\leq q}$ est constituée d'éléments distincts de $\{\beta'_i\}_{1\leq i\leq p}$.
2. Si $i < j$, alors on a $\gamma'_j \not\prec \gamma'_i$.
3. $\mathcal{R}''_3 \leq \mathcal{R}''_4$

La réduction $\mathcal{R}' : M \stackrel{\gamma'_1}{\Rightarrow}_{ne} N''_1 \stackrel{\gamma'_2}{\Rightarrow}_{ne} \ldots \stackrel{\gamma_q}{\Rightarrow}_{ne} N''_q \stackrel{\beta'_i}{\Rightarrow}_{ne} N''_{q+1}$ vérifie les propriétés voulues. □

Si une réduction $\mathcal{R}$ aboutit à une valeur, on peut obtenir une réduction $\mathcal{R}'$ qui contient cette réduction (au sens où $\mathcal{R} \leq \mathcal{R}'$), en effectuant à chaque étape la réduction complète d'un nom d'un radical réduit au cours de $\mathcal{R}$, dans le respect des dépendances entre radicaux (condition 2). Ce résultat, combiné avec le lemme 6.6, est utilisé pour montrer que la notion de réduction indépendante entre les principaux A et B est exprimée par les étiquettes.

Théorème 6.8 (Indépendance) *Si $M \twoheadrightarrow_{ne} V$ où $\tau(V)$ sépare A et B, alors il existe une réduction $\mathcal{R} : M \twoheadrightarrow_{ne} V'$ telle que $\mathcal{R}$ est indépendante de l'interaction entre A et B.*

Preuve : En utilisant le théorème de standardisation et le lemme 6.4, on obtient qu'il existe une réduction $\mathcal{R} : M \twoheadrightarrow_{ne} V'$ qui est une réduction de tête. On peut supposer, sans perte de généralité, que V' est la première valeur atteinte au cours de la réduction $\mathcal{R}$. Par confluence, on a $\tau(V) = \tau(V')$. Les radicaux contractés au cours de la réduction $\mathcal{R}$ entre M et V' sont nommés $R_1, \ldots, R_n$ ce qui peut s'écrire : $\mathcal{R} : M \xrightarrow{R_1}_{ne} M_1 \xrightarrow{R_2}_{ne} \ldots \xrightarrow{R_{n-1}}_{ne} M_{n-1} \xrightarrow{R_n}_{ne} V'$. En utilisant le lemme 6.9, on obtient une réduction $\mathcal{R}' : M \stackrel{\alpha'_1}{\Rightarrow}_{ne} M'_1 \stackrel{\alpha'_2}{\Rightarrow}_{ne} M'_2 \stackrel{\alpha'_3}{\Rightarrow}_{ne} \ldots \stackrel{\alpha'_{n'}}{\Longrightarrow}_{ne} W$ telle que (1) pour tout $1 \leq i \leq n'$, il existe j tel que $1 \leq j \leq n$ et $\mathsf{nom}(R_j) = \alpha'_i$, (2) si $1 \leq i < j \leq n'$, alors $\alpha'_j \not\prec \alpha'_i$ et (3) on a $\mathcal{R} \leq \mathcal{R}'$. Ce dernier point implique bien que le terme final W de $\mathcal{R}'$ est une valeur qui vérifie $\tau(W) = \tau(V')$. En utilisant le lemme 6.5, on obtient $\alpha'_i \prec \tau(V')$. On peut extraire deux suites $\{\beta'_j\}_{1\leq j\leq m}$ et $\{\gamma'_k\}_{1\leq k\leq p}$ qui vérifient les propriétés suivantes.

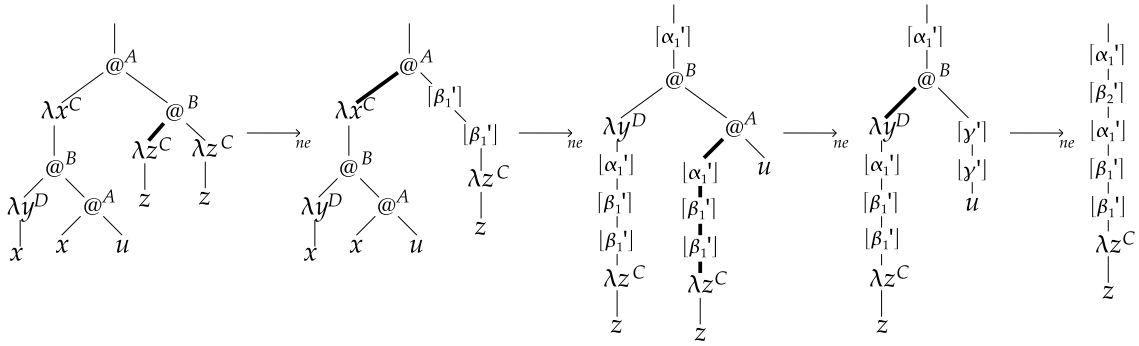
1. $\{\beta'_i\}_{1\leq i\leq m}$ est la suite des éléments α'_i de $\{\alpha'_i\}_{1\leq i\leq n'}$ qui vérifient $\{A,B\} \cap |\alpha'_i| = \emptyset$.
2. $\{\gamma'_i\}_{1\leq i\leq p}$ est la suite des éléments α'_i de $\{\alpha'_i\}_{1\leq i\leq n'}$ qui vérifient $\{A,B\} \cap |\alpha'_i| \neq \emptyset$.

Ces suites forment bien entendu une partition de la suite $\{\alpha'_i\}_{1\leq i\leq n'}$. Pour tout couple (i,j) tel que $1 \leq i \leq m$ et $1 \leq j \leq p$, on a $\gamma'_i \not\prec \beta'_j$. Il existe des indices k et k' distincts tels que $1 \leq k,k' \leq n'$ et $\beta'_i = \alpha'_k$ et $\gamma'_j = \alpha'_{k'}$. Si $k' < k$, on a $\beta'_i \not\prec \gamma'_j$. De là, en utilisant itérativement le lemme 6.7, on obtient la réduction $\mathcal{R}_0$ suivante.

$$\mathcal{R}_0 : M \stackrel{\beta'_1}{\Rightarrow}_{ne} N_1 \stackrel{\beta'_2}{\Rightarrow}_{ne} \ldots \stackrel{\beta'_m}{\Longrightarrow}_{ne} N_m \stackrel{\gamma'_1}{\Rightarrow}_{ne} P_1 \stackrel{\gamma'_2}{\Rightarrow}_{ne} \ldots \stackrel{\gamma'_p}{\Rightarrow}_{ne} W$$

Cette réduction vérifie $\mathcal{R}_0 \sim \mathcal{R}'$. Comme les éléments de $\{\beta'_i\}_{1\leq i\leq m}$ sont de strictes sous-étiquettes de $\tau(V)$ et que cette dernière sépare A et B, on peut partitionner la suite $\{\beta'_i\}_{1\leq i\leq m}$ en deux suites de la façon suivante.

1. $\{\delta'_j\}_{1\leq j\leq q}$ est la suite des éléments β'_i de $\{\beta'_i\}_{1\leq i\leq m}$ qui vérifient $A \in |\alpha'_i|$.
2. $\{\eta'_j\}_{1\leq j\leq q'}$ est la suite des éléments β'_i de $\{\beta'_i\}_{1\leq i\leq m}$ qui vérifient $B \in |\alpha'_i|$.



$$\beta'_1 = BC \qquad \alpha'_1 = AC \qquad \gamma' = A\lfloor\alpha'_1\rfloor\lceil\beta'_1\rceil\lfloor\beta'_1\rfloor C \qquad \beta'_2 = BD$$

FIG. 6.9 – *Réduction de* $M = ((\lambda x.((\lambda y.x)^D (x\ u)^A)^B)^C((\lambda z.z)^C(\lambda z.z)^C)^B)^A$

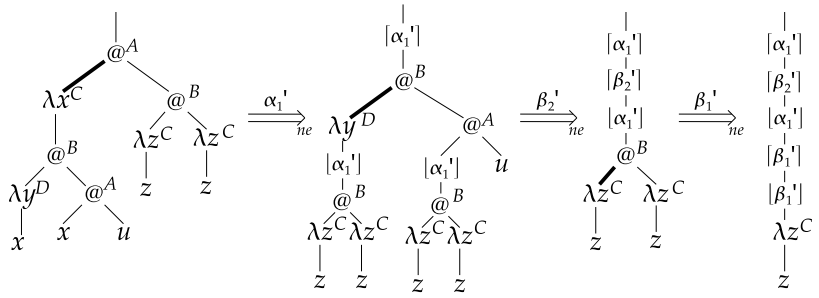


FIG. 6.10 – *Réduction de* M *par une séquence de réductions complètes*

.

Pour tout couple (i,j) tel que $1 \leq i \leq q$ et $1 \leq j \leq q'$, on a $\delta'_i \not\sim \eta'_j$ et $\eta'_j \not\sim \delta'_i$. De là, en utilisant itérativement le lemme 6.7, on obtient les réductions suivantes.

$$\mathcal{R}_1 : M \overset{\beta'_1}{\Rightarrow}_{ne} N_1 \overset{\beta'_2}{\Rightarrow}_{ne} \ldots \overset{\beta'_m}{\Rightarrow}_{ne} N_m \overset{\eta'_1}{\Rightarrow}_{ne} \ldots \overset{\eta'_q}{\Rightarrow}_{ne} Q \overset{\delta'_1}{\Rightarrow}_{ne} \ldots \overset{\delta'_{q'}}{\Rightarrow}_{ne} W$$
$$\mathcal{R}_2 : M \overset{\beta'_1}{\Rightarrow}_{ne} N_1 \overset{\beta'_2}{\Rightarrow}_{ne} \ldots \overset{\beta'_m}{\Rightarrow}_{ne} N_m \overset{\delta'_1}{\Rightarrow}_{ne} \ldots \overset{\delta'_{q'}}{\Rightarrow}_{ne} Q' \overset{\eta'_1}{\Rightarrow}_{ne} \ldots \overset{\eta'_q}{\Rightarrow}_{ne} W$$

Ces réductions vérifient les réductions d'équivalence $\mathcal{R}_1 \sim \mathcal{R}'$ et $\mathcal{R}_2 \sim \mathcal{R}'$. On considère les réductions suivantes.

$$\mathcal{R}_A : M \overset{\beta'_1}{\Rightarrow}_{ne} N_1 \overset{\beta'_2}{\Rightarrow}_{ne} \ldots \overset{\beta'_m}{\Rightarrow}_{ne} N_m \overset{\eta'_1}{\Rightarrow}_{ne} \ldots \overset{\eta'_q}{\Rightarrow}_{ne} Q$$
$$\mathcal{R}_B : M \overset{\beta'_1}{\Rightarrow}_{ne} N_1 \overset{\beta'_2}{\Rightarrow}_{ne} \ldots \overset{\beta'_m}{\Rightarrow}_{ne} N_m \overset{\delta'_1}{\Rightarrow}_{ne} \ldots \overset{\delta'_{q'}}{\Rightarrow}_{ne} Q'$$

La réduction $\mathcal{R}_A$ (respectivement $\mathcal{R}_B$) ignore le principal A (resp. B). De plus, la réduction-résidu $\mathcal{R}_A/\mathcal{R}_B$ (resp.$(\mathcal{R}_B/\mathcal{R}_A)$) est $Q' \overset{\eta'_1}{\Rightarrow}_{ne} \ldots \overset{\eta'_q}{\Rightarrow}_{ne} W$ (resp. $Q \overset{\delta'_1}{\Rightarrow}_{ne} \ldots \overset{\delta'_{q'}}{\Rightarrow}_{ne} W$). On en déduit que la réduction $\mathcal{R}$ est indépendante de l'interaction entre les principaux A et B. □

On illustre pas à pas la démonstration effectuée ici. On considère la réduction du terme M représentée sur la figure 6.9. Les noms des radicaux contractés, sont, dans l'ordre, β'_1, α'_1, γ', α'_2. L'étiquette de tête de la valeur obtenue est $\lceil\alpha'_1\rceil\lceil\beta'_2\rceil\lfloor\alpha'_1\rfloor\lceil\beta'_1\rceil\lfloor\beta'_1\rfloor$. En examinant, les noms des radicaux, on constate que cette séquence d'étiquettes atomiques sépare A et B car aucune de ces étiquettes atomiques ne contient à la fois A et B. En réduisant en tête M, on réduit les radicaux α'_1 puis β'_2 puis β'_1 pour obtenir la valeur finale. En utilisant le lemme 6.6, on obtient la séquence de réductions complètes représentée sur la figure 6.10. De là, on peut permuter ces réductions complètes, en utilisant le lemme 6.7, pour montrer que cette réduction est indépendante de l'interaction entre les principaux A et B. Ces permutations sont illustrées sur la figure 6.11. On

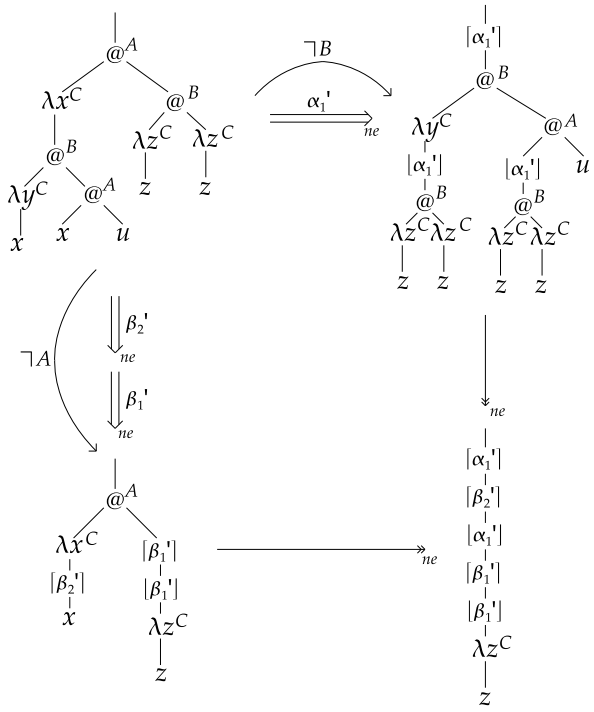


FIG. 6.11 – *La réduction de la figure 6.10 est indépendante de l'interaction entre A et B.*

a donc montré que si une réduction $\mathcal{R}$ de M aboutit à une valeur dont l'étiquette de tête sépare A et B, alors il existe une réduction $\mathcal{R}'$ issue de M indépendante de l'interaction entre A et B qui aboutit à une valeur. On note que $\mathcal{R}$ et $\mathcal{R}'$ peuvent être très différentes. La réduction initiale $\mathcal{R}$ de notre exemple contracte un radical γ' qui n'est pas contracté par $\mathcal{R}'$. A contrario, le fait de compléter la réduction de tête peut contracter des radicaux non contractés dans la réduction initiale.

On prouve maintenant que la politique de sécurité de la Muraille de Chine, telle qu'elle a été formalisée précédemment dans le λ_n-calcul étiqueté, garantit une propriété d'indépendance.

Théorème 6.9 (Correction de la Muraille de Chine) *Si la réduction $\mathcal{R} : M \twoheadrightarrow_{ne} N$ est $\mathcal{M}(A,B)$-compatible, alors cette réduction est indépendante de l'interaction entre A et B.*

Preuve : On procède de façon similaire à la preuve du théorème 6.8. La réduction $M \twoheadrightarrow_{ne} V$ peut s'écrire $\mathcal{R} : M \xrightarrow{R_1}_{ne} M_1 \xrightarrow{R_2}_{ne} \cdots \xrightarrow{R_n}_{ne} V$. En utilisant le lemme 6.9, on obtient une réduction $\mathcal{R}' : M \overset{\alpha'_1}{\Longrightarrow}_{ne} M'_1 \overset{\alpha'_2}{\Longrightarrow}_{ne} M'_2 \overset{\alpha'_3}{\Longrightarrow}_{ne} \ldots \overset{\alpha'_{n'}}{\Longrightarrow}_{ne} N'$ qui vérifie les propriétés suivantes.

1. Pour tout $1 \leq i \leq n'$, il existe j tel que $1 \leq j \leq n$ et $\mathsf{nom}(R_j) = \alpha'_i$.
2. Si $1 \leq i < j \leq n'$, alors $\alpha'_j \not\prec \alpha'_i$.
3. On a $\mathcal{R} \leq \mathcal{R}'$.

Soient $\{\beta'_j\}_{1\leq j\leq m}$ et $\{\gamma'_k\}_{1\leq k\leq p}$ les deux suites extraites de la suite $\{\alpha'_i\}_{1\leq i\leq n'}$ qui vérifient les propriétés suivantes.

1. $\{\beta'_i\}_{1\leq i\leq m}$ est la suite des éléments α'_i de $\{\alpha'_i\}_{1\leq i\leq n'}$ qui vérifient $\{A,B\} \cap |\alpha'_i| = \emptyset$.
2. $\{\gamma'_i\}_{1\leq i\leq p}$ est la suite des éléments α'_i de $\{\alpha'_i\}_{1\leq i\leq n'}$ qui vérifient $\{A,B\} \cap |\alpha'_i| \neq \emptyset$.

Ces suites forment bien entendu une partition de la suite $\{\alpha'_i\}_{1\leq i\leq n'}$. Pour tout couple (i,j) tel que $1 \leq i \leq m$ et $1 \leq j \leq p$, on a $\gamma'_i \not\prec \beta'_j$. Il existe des indices k et k' distincts tels que $1 \leq k,k' \leq n'$ et $\beta'_i = \alpha'_k$ et $\gamma'_j = \alpha'_{k'}$. Si $k' < k$, on a $\beta'_i \not\prec \gamma'_j$. De là, en utilisant itérativement le lemme 6.7, on obtient la réduction $\mathcal{R}_0$ suivante.

$$\mathcal{R}_0 : M \stackrel{\beta'_1}{\Longrightarrow}_{ne} N_1 \stackrel{\beta'_2}{\Longrightarrow}_{ne} \ldots \stackrel{\beta'_m}{\Longrightarrow}_{ne} N_m \stackrel{\gamma'_1}{\Longrightarrow}_{ne} P_1 \stackrel{\gamma'_2}{\Longrightarrow}_{ne} \ldots \stackrel{\gamma'_p}{\Longrightarrow}_{ne} N'$$

Cette réduction vérifie $\mathcal{R}_0 \sim \mathcal{R}'$. Comme $\mathcal{R}$ est $\mathcal{M}(A,B)$-compatible, on peut partitionner la suite $\{\beta'_i\}_{1\leq i\leq m}$ en deux suites de la façon suivante.

1. $\{\delta'_j\}_{1\leq j\leq q}$ est la suite des éléments β'_i de $\{\beta'_i\}_{1\leq i\leq m}$ qui vérifient $A \in |\alpha'_i|$.
2. $\{\eta'_j\}_{1\leq j\leq q'}$ est la suite des éléments β'_i de $\{\beta'_i\}_{1\leq i\leq m}$ qui vérifient $B \in |\alpha'_i|$.

Pour tout couple (i,j) tel que $1 \leq i \leq q$ et $1 \leq j \leq q'$, on a $\delta'_i \not\asymp \eta'_j$ et $\eta'_j \not\asymp \delta'_i$. De là, en utilisant itérativement le lemme 6.7, on obtient les réductions suivantes.

$$\mathcal{R}_1 : M \stackrel{\beta'_1}{\Longrightarrow}_{ne} N_1 \stackrel{\beta'_2}{\Longrightarrow}_{ne} \ldots \stackrel{\beta'_m}{\Longrightarrow}_{ne} N_m \stackrel{\eta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\eta'_q}{\Longrightarrow}_{ne} Q \stackrel{\delta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\delta'_{q'}}{\Longrightarrow}_{ne} N'$$
$$\mathcal{R}_2 : M \stackrel{\beta'_1}{\Longrightarrow}_{ne} N_1 \stackrel{\beta'_2}{\Longrightarrow}_{ne} \ldots \stackrel{\beta'_m}{\Longrightarrow}_{ne} N_m \stackrel{\delta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\delta'_{q'}}{\Longrightarrow}_{ne} Q' \stackrel{\eta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\eta'_q}{\Longrightarrow}_{ne} N'$$

Ces réductions vérifient les réductions d'équivalence $\mathcal{R}_1 \sim \mathcal{R}'$ et $\mathcal{R}_2 \sim \mathcal{R}'$. On considère les réductions suivantes.

$$\mathcal{R}_A : M \stackrel{\beta'_1}{\Longrightarrow}_{ne} N_1 \stackrel{\beta'_2}{\Longrightarrow}_{ne} \ldots \stackrel{\beta'_m}{\Longrightarrow}_{ne} N_m \stackrel{\eta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\eta'_q}{\Longrightarrow}_{ne} Q$$
$$\mathcal{R}_B : M \stackrel{\beta'_1}{\Longrightarrow}_{ne} N_1 \stackrel{\beta'_2}{\Longrightarrow}_{ne} \ldots \stackrel{\beta'_m}{\Longrightarrow}_{ne} N_m \stackrel{\delta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\delta'_{q'}}{\Longrightarrow}_{ne} Q'$$

La réduction $\mathcal{R}_A$ (respectivement $\mathcal{R}_B$) ignore le principal A (resp. B). De plus, la réduction-résidu $\mathcal{R}_A/\mathcal{R}_B$ (resp.$(\mathcal{R}_B/\mathcal{R}_A)$) est $Q' \stackrel{\eta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\eta'_q}{\Longrightarrow}_{ne} N'$ (resp. $Q \stackrel{\delta'_1}{\Longrightarrow}_{ne} \ldots \stackrel{\delta'_{q'}}{\Longrightarrow}_{ne} N'$). On en déduit que la réduction $\mathcal{R}$ est indépendante de l'interaction entre les principaux A et B. □

Si $\mathcal{R}$ est une réduction qui respecte la Muraille de Chine $\mathcal{M}(A,B)$, alors cette réduction est indépendante de l'interaction A et B. Les résultats d'indépendance obtenus pour la Muraille de Chine prouvent que cette politique de sécurité, dont l'effet est local, garantit une propriété de sécurité globale.

Dans cette partie, nous avons défini un calcul, le λ_n-calcul, permettant d'introduire la notion de principal dans le λ-calcul. Dans ce calcul, nous avons défini une propriété de sécurité, l'indépendance, qui formalise le fait que les actions dans lesquelles sont impliquées deux principaux sont indépendantes puisque l'ordre de ces actions n'a pas d'influence. En étendant ce calcul avec les étiquettes du λ-calcul, nous pouvons exprimer formellement une politique de sécurité qui fait appel à l'histoire de la réduction, comme la Muraille de Chine. Nous avons montré que la propriété d'indépendance est étroitement liée aux étiquettes. Et nous avons prouvé que la Muraille de Chine garantit une propriété d'indépendance. Si l'on se penche maintenant sur la politique de sécurité des Enchères scellées, on observe que cette dernière vise à assurer que les enchérisseurs déterminent leur enchère de façon indépendante les uns des autres. Le but de cette politique de sécurité locale est donc bien de garantir une propriété de sécurité globale : l'indépendance du calcul des enchères. Pour examiner cette politique de sécurité, il faudrait étendre le λ_n-calcul étiqueté avec des constructions permettant l'opération de scellement. Les étiquettes permettraient de définir formellement les Enchères scellées et seraient un outil puissant pour montrer que cette politique de sécurité garantit l'indépendance du calcul des enchères.

Conclusion

Avant de reprendre et de mettre en perspective les contributions apportées par cette dissertation, il est intéressant d'examiner quelle a été ma démarche au cours de ces travaux. Après avoir travaillé, au cours de mon stage de DEA, sur le sujet d'une analyse statique portant sur l'utilisation de l'inspection de pile dans la plate-forme logicielle .NET de Microsoft, je me suis intéressé à d'autres mécanismes permettant de sécuriser un programme par l'intermédiaire du langage de programmation. Si l'inspection de pile est un mécanisme de sécurité largement répandu via les langages Java et $C\#$, ce système ne garantit pas une propriété de sécurité claire. L'approche de l'analyse de flot d'information s'est révélée plus fructueuse sur le plan des fondations théoriques. Les systèmes de type de Pottier, Simonet, Myers, Heintze garantissent la propriété de non-interférence pour les termes bien typés. Sur le plan pratique, cette approche a abouti à des langages de programmation complets qui n'ont toutefois pas une diffusion comparable à Java ou $C\#$. Ces analyses montrent cependant des limites pour des exemples, comme le test de mot de passe, pourtant bien concrets. De plus, le caractère statique de l'analyse est contraignant : les analyses de flot d'information ne sont pas assez flexibles pour mettre en œuvre des politiques de sécurité dynamiques telles que la Muraille de Chine ou les Enchères Scellées. En partant de la remarque que ces différentes approches utilisent de façon fondamentale l'histoire, c'est-à-dire les événements passés ou les dépendances vis-à-vis du passé, j'ai voulu comparer ces approches en les formalisant dans un cadre commun, fondé sur le λ-calcul étiqueté. En effet, comme l'avaient remarqué Abadi et al. dans [3], les étiquettes du λ-calcul fournissent une analyse dynamique de dépendance qui peut tenir lieu d'histoire pour les approches considérées ici. Ces travaux qui sont décrits dans les trois derniers chapitres, ont occasionné certains développements sur le λ-calcul étiqueté : il est en effet bien difficile d'étudier des propriétés de sécurité exprimées dans le λ-calcul sans étudier ce même λ-calcul. L'examen de la propriété de non-interférence dans le λ_m-calcul, muni de références, a engendré deux développements théoriques portant sur le λ-calcul. D'une part, le fait que le λ_m-calcul est réduit en appel par valeur m'a conduit à formaliser le λ-calcul par valeur, à étudier ses propriétés fondamentales et à adapter le système d'étiquettes en conséquence. D'autre part, la nécessité de nommer les adresses mémoire du λ_m-calcul de façon structurelle a suscité la découverte de la propriété d'irréversibilité des contextes et des chemins. Bien que mon travail sur le λ-calcul et la formalisation dans le λ-calcul a pris la forme d'allers-retours incessants, la présentation synthétique de ces travaux adopte naturellement une structure bipolaire : les trois premiers chapitres sont consacrés aux développements sur le λ-calcul et certaines de ses variantes. Les trois derniers chapitres traitent des trois approches sur la sécurité considérées ici.

Le chapitre 1 apporte une première contribution : nous montrons que le λ-calcul étiqueté vérifie la propriété d'irréversibilité des contextes (p. 20). Ainsi, on a la réduction $C[M] \twoheadrightarrow_e C[M']$ si et seulement si on a $M \twoheadrightarrow_e M'$. Ceci signifie intuitivement, qu'un contexte qui contient un radical ou une partie de radical contracté, ne peut pas être reconstruit ensuite à l'identique. En d'autres mots, si $C[N] \rightarrow_e N'$ et si $C[\,]$ n'est pas un contexte de N', alors ce contexte disparaît de façon *irréversible* : si $N' \twoheadrightarrow_e N''$, alors $C[\,]$ n'est pas un contexte de N''. Ce résultat d'irréversibilité, qui

est faux dans le λ-calcul sans étiquettes, vient confirmer une intuition déjà mentionnée dans la thèse de Lévy [29] : les étiquettes du λ-calcul permettent d'éviter les coïncidences syntaxiques. Comme un chemin d'un terme peut être vu comme un contexte particulier, l'irréversibilité des contextes a pour corollaire l'irréversibilité des chemins (p. 24). En particulier, le chemin qui mène à un radical contracté disparaît de façon irréversible. Ce résultat permet d'établir une analogie entre, d'une part, les chemins menant aux radicaux contractés lors d'une réduction et, d'autre part, les dates d'une période temporelle : dans les deux cas, on ne peut avoir une répétition d'un chemin/d'une date. Ces chemins particuliers permettent de dater les pas de réduction : cette analogie est exploitée dans le chapitre 5 pour définir l'intervalle de temps pendant lequel une adresse de la mémoire contribue au résultat final d'une réduction.

Le chapitre 2 introduit une nouvelle variante du λ-calcul : le λ-calcul par valeur. Cette variante s'inspire des stratégies d'évaluation en appel par valeur employées dans plusieurs langages de programmation fonctionnels [28, 40]. En plus de définir formellement cette variante, nous adaptons les étiquettes du λ-calcul afin que la propriété de stabilité du langage puisse s'exprimer simplement à l'aide des étiquettes. La contribution majeure de ce chapitre réside dans la preuve élégante et intuitive du théorème des développements finis (p. 60). Cette preuve est fondée sur une notion d'imbrication étendue des radicaux de l'ensemble $\mathcal{F}$ considéré. Un radical R de $\mathcal{F}$ est imbriqué dans un radical S de $\mathcal{F}$ (ce qui est noté $S \Subset_{\mathcal{F}} R$) si S contient R ou bien si, après la réduction de radicaux de $\mathcal{F}$, un résidu de S contient un résidu de R. Cette relation permet de définir la notion de profondeur d'un radical de $\mathcal{F}$ qui est la longueur maximale des chaînes d'imbrication de radicaux de $\mathcal{F}$ menant à R. Le multi-ensemble des profondeurs des radicaux de $\mathcal{F}$ décroît strictement au cours d'un développement de $\mathcal{F}$. La démonstration ainsi obtenue s'adapte aisément au λ-calcul ou au λ-calcul faible avec ou sans étiquettes. Cette démonstration est plus concrète, plus intuitive et donc, de mon point de vue, plus satisfaisante que les preuves utilisées jusqu'ici dans la littérature [7, 24, 29].

Le chapitre 3 est consacré au λ-calcul faible étiqueté. Dans un calcul faible, les sous-termes des abstractions ne sont pas réductibles, contrairement au λ-calcul classique, dit fort. Le λ-calcul faible étiqueté est fondé sur une variante confluente du λ-calcul faible, qui a été introduite dans [30]. Nous prouvons que ce calcul étiqueté est confluent et qu'il vérifie les théorèmes des développements finis et de standardisation. La contribution de ce chapitre réside dans le théorème de partage (p. 87) : sous des hypothèses raisonnables, les sous-termes qui portent la même étiquette sont égaux et peuvent donc être partagés. Les étiquettes de ce langage s'inspirent largement des travaux de Wadsworth [43]. En effet, dans le calcul faible, on peut représenter les termes avec partage à l'aide de graphes acycliques orientés. Comme le montre le deuxième algorithme de Wadsworth, après contraction d'un radical, l'argument peut être partagé. Toutefois, si une abstraction partagée est impliquée dans un radical contracté, elle est dupliquée : plus précisément, ses sous-termes qui contiennent une variable liée sont dupliqués. Dans [38], Shivers et Wand ont mis en œuvre un algorithme similaire dans le cadre d'une implémentation plus réaliste. Ceci constitue une simplification notable par rapport au λ-calcul fort. Dans celui-ci, on ne peut pas se contenter de partager des sous-termes : il est nécessaire de partager des sous-contextes. De ce fait, dans l'algorithme de Lamping [25], les termes sont représentés par des graphes. Les sous-contextes sont partagés par des nœuds *fan-in*. Et les nœuds *fan-out* permettent de remplir les trous des sous-contextes. Le λ-calcul faible étiqueté permet de raisonner sur la notion de partage à l'aide de termes, comme dans le λ-calcul classique, ce qui permet d'éviter des preuves délicates sur des graphes.

Dans le chapitre 4, nous nous inspirons du mécanisme d'inspection de pile et nous introduisons le λ_t-calcul. L'inspection de pile peut être vue de façon synthétique comme un mécanisme de sécurité qui contrôle l'exécution d'un programme en prenant des décisions à partir d'informations locales et globales. Les permissions statiques, déterminées en fonction du principal de la fonction

courante constituent l'information locale. L'information globale est représentée par les permissions dynamiques qui sont calculées à partir de la chaîne d'appel aboutissant à la fonction courante. Dans le λ_t-calcul, on adapte ce mécanisme en attribuant aux étiquettes le rôle d'information locale et aux chemins le rôle d'information globale. Dans ce calcul, les contractions des radicaux sont conditionnées par l'étiquette du radical et le chemin menant au radical. Ce calcul est étudié à travers le prisme de la propriété de confluence locale. Une condition suffisante sur les conditions pour obtenir un langage localement confluent est énoncée. A partir de ce langage, on obtient une variante, le λ_{ts}-calcul, dans le but de se rapprocher du mécanisme d'inspection de pile. La principale contribution de ce chapitre consiste en une traduction correcte du λ_{secW}-calcul vers le λ_{ts}-calcul. Le λ_{secW}-calcul est une variante de l'inspection de pile proposée par Fournet et Gordon dans [15]. Cette traduction permet de faire le lien entre un calcul théorique fondé sur les étiquettes du λ-calcul et le mécanisme de sécurité empirique qu'est l'inspection de pile.

Nous étudions la propriété de non-interférence dans le chapitre 5. Le problème est abordé de la façon suivante : on considère une réduction $M \twoheadrightarrow V$ qui aboutit sur une valeur. On souhaite savoir si l'observation de la valeur V (intuitivement publique) permet d'obtenir une information sur certains sous-termes (intuitivement secrets) de M. Si on se place dans le cadre du λ-calcul, les étiquettes du λ-calcul fournissent une analyse dynamique de dépendance des termes vis-à-vis des sous-termes du terme initial. Ainsi, l'étiquette de tête de V permet d'obtenir le préfixe X des sous-termes de M dont V dépend. Ces sous-termes interfèrent dans V. A contrario, les sous-termes qui sont effacés dans X n'interfèrent pas dans V. Par conséquent, V ne donne aucune information sur ces sous-termes. Dans ce cas, l'ensemble des sous-termes qui interfèrent coïncide avec le préfixe minimum de la propriété de stabilité. Ainsi, en vertu du résultat 1.15, tout préfixe Y tel que $X \preccurlyeq Y$, vérifie $Y \twoheadrightarrow V'$ et $V \preccurlyeq V'$. Dans le cas du λ-calcul par valeur, on obtient de même, le préfixe des sous-termes qui interfèrent à partir de l'étiquette de tête du terme obtenu avec la réduction étiquetée du λ-calcul. Par conséquent, dans ce cas, l'ensemble des sous-termes qui interfèrent est simplement inclus dans le préfixe de stabilité du λ-calcul par valeur. Cette inclusion est logique puisque le préfixe de stabilité se réduit vers une valeur, alors que l'obtention d'une valeur est une hypothèse dans le cas de la non-interférence. Si les étiquettes du λ-calcul permettent d'obtenir simplement la propriété de non-interférence, l'ajout de traits impératifs tels que l'affectation change radicalement la situation. Pour étudier ce cas, on introduit le λ_m-calcul, un λ-calcul étendu avec des traits impératifs et des δ-règles.

En présence d'effets de bord, un nouveau type d'interférence vient s'ajouter à l'interférence "fonctionnelle" présente dans le λ-calcul calcul. Au cours de la réduction $M/\emptyset \twoheadrightarrow V/\mu$, des écritures et des lectures en mémoire ont lieu. Certaines adresses de la mémoire peuvent interférer dans V. Plus précisément, une adresse peut contribuer à V pendant certains intervalles de temps : entre une écriture et une lecture. Si un effet de bord vient interférer entre cette écriture et cette lecture, la valeur lue est modifiée, ce qui entraîne la modification de la valeur finale. A l'interférence "fonctionnelle" vient s'ajouter une interférence sur la mémoire. Il s'agit donc de déterminer, en plus de l'ensemble des sous-termes qui interfèrent, l'ensemble des intervalles des adresses qui interfèrent. La contribution de ce chapitre consiste à présenter un λ_m-calcul étiqueté qui permet d'énoncer une propriété de non-interférence dans un langage muni de traits impératifs, en dehors de toute analyse statique. Plus précisément, les étiquettes permettent de déterminer les sous-termes du terme initial ainsi que les intervalles des adresses qui interfèrent. Ce résultat repose fondamentalement sur le théorème d'irréversibilité des chemins (p. 24). Cette propriété permet de nommer les adresses de façon structurelle : dans le λ_m-calcul étiqueté, les adresses sont des chemins. De plus, l'analogie temporelle entre chemins et dates est exploitée pour définir les *dates* qui délimitent les intervalles des adresses. Le λ_m-calcul étiqueté fournit des fondations précises pour raisonner sur la propriété de non-interférence et obtenir une analyse statique qui la garantit. En conjonction d'une telle analyse,

des tests dynamiques portant sur les étiquettes ou une abstraction des étiquettes pourraient être utilisés pour raffiner l'analyse.

La politique de la Muraille de Chine est examinée dans le chapitre 6. En annotant les termes du λ-calcul avec des principaux, on obtient le λ_n-calcul. Ce langage permet d'exprimer une propriété de sécurité nouvelle : l'indépendance (p. 148). Une réduction est indépendante de l'interaction entre deux principaux A et B si les réductions élémentaires de cette réduction peut être réordonnées pour que les réductions élémentaires dans laquelle l'un des deux principaux n'intervient pas soient effectuées en premier. Le fait que l'ordre des actions impliquant A ou B n'a pas d'importance rend bien compte de la notion intuitive d'indépendance. En ajoutant au λ_n-calcul les étiquettes du λ-calcul, l'histoire des interactions passées devient simplement accessible. De ce fait, on peut définir formellement la politique de sécurité de la Muraille de Chine dans le λ_n-calcul étiqueté : cette politique impose une contrainte sur le nom des radicaux contractés. Outre la notion nouvelle que constitue la propriété d'indépendance, la contribution principale de ce chapitre est la preuve de correction de la politique de la Muraille de Chine vis-à-vis de l'indépendance (p. 160). Si A et B sont séparés par la politique de la Muraille de Chine, une réduction conforme à cette dernière est indépendante de l'interaction entre A et B. Plus généralement, on montre que si une réduction $M \twoheadrightarrow_{ne} V$ aboutit à une valeur, l'étiquette de cette valeur permet de caractériser l'existence d'une réduction indépendante issue de M et aboutissant à une valeur. Ce chapitre apporte un nouvel éclairage sur certaines politiques de sécurité dynamiques telles que la Muraille de Chine et les Enchères Scellées. On montre en particulier que la propriété de sécurité visée par ces politiques n'est pas la non-interférence mais bien l'indépendance.

Pour conclure cette dissertation, nous mentionnons quelques perspectives intéressantes engendrées par les travaux présentés ici. Bien que notre approche de la propriété de non-interférence a été inspirée par les travaux de Simonet et Pottier sur l'analyse de flot d'information, nous n'avons établi aucun lien formel entre les étiquettes employées pour le λ_m-calcul et le système de type utilisé pour Core ML. En plus d'un intérêt purement formel, une mise en relation de ces deux techniques pourrait aider à comprendre comment exploiter une information dynamique sur le niveau de sécurité des termes dans le cadre d'une analyse statique. Jusqu'à présent, les analyses de flot d'information sont purement statiques. En prenant pour exemple le système Flow Caml, on observe que toutes les informations relatives aux niveaux de sécurité des termes sont oubliées après l'opération de typage : elles ne sont pas présentes au moment de la réduction. Ceci a, bien entendu, des avantages en terme de performance. Cependant, il pourrait être intéressant de propager une information partielle sur les niveaux de sécurité des termes au cours de la réduction. Cette information pourrait être utilisée au moment de tests dynamiques. Ces tests impliqueraient un surcoût de calcul mais pourraient être exploités dans le typage pour obtenir une analyse plus fine.

Dans la section consacrée à la non-interférence dans le cadre du λ-calcul pur, nous avons observé que les étiquettes du λ-calcul permettent d'exprimer cette propriété. Plus précisément, si la réduction d'un terme M (qui vérifie $\texttt{INIT}(M)$) aboutit à une valeur V, les *lettres* présentes dans l'étiquette de tête de V permettent de déterminer les sous-termes de M qui interfèrent. Par conséquent, la propriété de non-interférence n'exploite pas la structure de soulignement ou de surlignement des étiquettes. Cette structuration des étiquettes rend compte des dépendances entre les contractions des radicaux. Intuitivement, le fait que le nom du radical R contienne l'étiquette α soulignée ou surlignée signifie que R a été créé par la contraction d'un radical S de nom α. Récursivement, si α contient l'étiquette β soulignée ou surlignée alors les radicaux de nom α ont été créés par la contraction de radicaux de nom β. Plus généralement, la structure des étiquettes indique les contraintes d'ordre entre les contractions des différents radicaux, créés ou non. Ceci explique intuitivement que la structure des étiquettes est intimement liée à la propriété d'indépendance, comme le formalise le théorème de séparation 6.7. Les étiquettes du λ-calcul expriment donc les

propriétés de non-interférence et d'indépendance de façons différentes. Une perspective d'étude consisterait à exprimer ces propriétés dans un cadre commun, qui ferait abstraction du système d'étiquette employé (étiquette du λ-calcul, du λ-calcul par valeur, du λ_m-calcul) et en particulier des différents types de soulignement et de surlignement. Ce cadre commun pourrait être une logique modale exprimant l'ensemble des étiquettes conformes à une propriété de sécurité. La modalité $\Box$ correspondrait à une dépendance vis-à-vis du passé. Informellement, la formule $\Box F$ doit être comprise : dans tous les passés, on a F. La modalité duale $\Diamond$ ($\Diamond F = \neg\Box\neg F$) doit être comprise : il existe un passé pour lequel on a F. Dans le cadre du λ-calcul, les passés d'une étiquette sont les sous-étiquettes soulignées ou surlignées. Avec cette interprétation, la formule $\Box(\neg A \vee \neg B)$ représente l'ensemble des étiquettes où les principaux A et B ne sont pas présents sous un même soulignement ou surlignement, c'est-à-dire l'ensemble des étiquettes qui séparent A et B (p. 153). En utilisant le théorème de séparation (p. 154), on obtient que cette formule exprime la propriété d'indépendance. En plus d'établir un lien entre la non-interférence et l'indépendance, cette démarche permettrait d'exprimer de nouvelles propriétés de sécurité de façon abstraite et indépendamment du cadre du λ-calcul. On pourrait par exemple étendre la propriété d'indépendance entre deux principaux à une indépendance entre deux principaux ayant une histoire spécifique. On peut illustrer cette extension en reprenant l'analogie économique dont est issue la politique de sécurité de la Muraille de Chine. On suppose qu'Alice, Bob et Charlie ne sont pas des concurrents. Cependant, du fait de la mise en commun de leurs compétences, l'alliance de Bob et Charlie concurrence Alice. Pour assurer l'indépendance entre ces concurrents, on appliquerait une politique de Muraille de Chine entre, d'une part, le principal Alice et, d'autre part, l'interaction entre Bob et Charlie. Dans la logique modale informellement introduite plus tôt, la présence d'une interaction entre B et C s'écrit $\Diamond(B \wedge C)$. De ce fait, la propriété d'indépendance étendue s'écrirait $\Box(\neg A \vee \neg\Diamond(B \wedge C))$. Cet exemple montre qu'une logique modale faciliterait l'expression synthétique des propriétés de sécurité ce qui pourrait permettre de les manipuler de façon abstraite et uniforme.

Table des figures

1.1 β-réduction . . . 7
1.2 Confluence . . . 8
1.3 Résidus d'un radical S après la contraction du radical R . . . 9
1.4 Monotonie . . . 12
1.5 Réduction dans le λ-calcul étiqueté . . . 16
1.6 Résidus d'un radical S après la contraction du radical S . . . 16
1.7 Confluence et monotonie dans le λ-calcul étiqueté . . . 18
1.8 Les étiquettes du λ-calcul éliminent les coïncidences syntaxiques. . . . 20
1.9 Les trois cas de figures du lemme 1.3 . . . 22

2.1 Le λ-calcul par valeur est localement confluent . . . 28
2.2 Propriétés de confluence de $\rightrightarrows_v$ et $\rightarrow_v$ . . . 29
2.3 Le λ-calcul par valeur vérifie le lemme des déplacements parallèles. . . . 30
2.4 Réduction par valeur du terme $M = \Delta((\lambda x.\lambda y.II)(KI))$ . . . 31
2.5 Réduction par valeur du terme $M = \Delta((\lambda x.\lambda y.II)(KI))$ . . . 32
2.6 Ordre $<_{st}$ entre deux radicaux d'un terme . . . 32
2.7 Diagramme de construction d'une réduction standard . . . 36
2.8 Monotonie dans le λ-calcul par valeur . . . 40
2.9 Réduction par valeur de $M = ((\lambda x.(\lambda y.y^d)^c)^b\ ((\lambda z.z^h)^g\ (\lambda u.u^j)^i)^f)^a$ . . . 44
2.10 Réduction dans le λ-calcul par valeur étiqueté . . . 46
2.11 Confluence locale du λ-calcul par valeur étiqueté . . . 50
2.12 Confluence locale forte de $\rightrightarrows_{ve}$ et confluence de $\rightarrow_{ve}$ . . . 52
2.13 Preuve de la confluence $\rightarrow_{ve}$ . . . 53
2.14 Développement de $\mathcal{F}$ dans $M = (\lambda x.(\lambda y.(\lambda z.yy)I)\lambda u.xx)R$ . . . 54
2.15 Relations d'imbrication sur les radicaux . . . 56
2.16 Permutation locale . . . 61
2.17 Ordre strict $<^e_{st}$ entre deux radicaux d'un terme . . . 63
2.18 Monotonie . . . 64
2.19 Réduction de $M = ((\lambda x.(x^d(\lambda y.((\lambda z.z^i)^h x^j)^g)^f)^c)^b(\lambda u.(\lambda v.v^j)^g)^f)^a$ . . . 66

3.1 Réductions de $M = (\lambda x.(xy)(xz))\lambda u.Iu$ par les deux algorithmes de Wadsworth . . 70
3.2 Réductions de $R = (\lambda x.\lambda y.x)(Iu)$ dans le λ-calcul faible . . . 71
3.3 Confluence locale du λ-calcul faible . . . 73
3.4 Confluence du λ-calcul faible . . . 75
3.5 Terme du λ-calcul faible étiqueté : $((\lambda x.(((x^f\ y^g)^d\ (x^i\ z^j)^h)^c)^b\ (\lambda u.((\lambda v.v^n)^m\ u^o)^l)^k)^a$ 78
3.6 β_{we}-réduction . . . 78
3.7 Réduction $(((\lambda x.(\lambda y.y^f)^d)^c\ v^g)^b\ u^h)^a \rightarrow_{we} (\lceil\alpha'\rceil : (\lambda y.y^f)^d\ u^h)^a$ où $\alpha' = bc$ . . . 79
3.8 Nom du radical créé . . . 80
3.9 Concluence locale du λ-calcul faible étiqueté . . . 81

3.10 Confluence locale forte de $\rightrightarrows_{we}$ et confluence de $\rightarrow_{we}$ 83
3.11 Réduction étiquetée de $M = ((\lambda x.((x^f\ y^g)^d\ (x^i\ z^j)^h)^b\ (\lambda u.((\lambda v.v^n)^m\ u^o)^l)^k)^a$ 87

4.1 Règles de réduction du λ_{sec}-calcul 91
4.2 Réduction de $M = ((\lambda x.((\lambda y.y^f)^e_{c_1}(\lambda t.x^h)^g)^d)^b(\lambda t.((\lambda z.z^m)^l_{c_2}(\lambda u.u^o)^n)^j)^i)^a$ 94
4.3 Condition confluente 95
4.4 Définition de la fonction $p_{\mathbf{E}}$ 97
4.5 Règles de réduction du λ_{secW}-calcul 99
4.6 Résultats de correction de $(\!|\ |\!)$ 103
4.7 Réduction dans le λ_{ts}-calcul de $M_{\mathsf{Nav}}N_{\mathsf{Plug}}$ 105

5.1 Non-interférence 108
5.2 Réductions comparées de M et M' 111
5.3 Syntaxe du λ_m-calcul 115
5.4 Réduction $\rightarrow$ 115
5.5 Substitution dans le λ_m-calcul 116
5.6 Les contextes d'évaluation du λ_m-calcul 116
5.7 Réduction de $M = (\lambda y.(\lambda\_.!y)\ 1)\ \mathtt{ref}(0)$ 116
5.8 Réduction de $M' = (\lambda y.(\lambda\_.!y)\ \mathtt{ref}(1))\ \mathtt{ref}(0)$ 117
5.9 Réduction de $M'' = (\lambda y.(\lambda\_.!y)\ y{:=}2)\ \mathtt{ref}(0)$ 117
5.10 Réductions comparées de M_1 et M_2 118
5.11 Syntaxe des termes et des valeurs du λ_m-calcul étiqueté 120
5.12 Syntaxe des étiquettes et des chemins du λ_m-calcul 121
5.13 Définition de la fonction de concaténation "$\cdot$" 122
5.14 Réductions $\rightarrow$ et $\xrightarrow{\kappa}$ $(\kappa \in \mathbf{K})$ 123
5.15 Définition de la substitution par une valeur 123
5.16 Définition de la fonction τ d'étiquette de tête 123
5.17 Lettres présentes dans une étiquette ou un chemin 123
5.18 Contexte d'évaluation du λ_m-calcul étiqueté 124
5.19 Chemin menant au cœur du radical pour $\kappa = \sigma(E[\,])$ 125
5.20 Réduction de $M = ((\lambda y.((\lambda\_.(!y^f)^e)^d)^c\ 1^g)^b\ (\mathtt{ref}(0^i))^h)^a$ 125
5.21 Réduction $\rightarrow_B$ et $\xrightarrow{\kappa}_B$ $(\kappa \in \mathbf{K})$ 129
5.22 Réduction des termes M_1 et M_2 141

6.1 Indépendance 144
6.2 β_n-réduction 145
6.3 Terme $M = (((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$ 146
6.4 Réduction de $(((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$ 147
6.5 Indépendance 148
6.6 β_{ne}-réduction 150
6.7 Réduction de $(((\lambda x.(\lambda y.y)^B)^A z)^A z)^B$ dans le λ_n-calcul étiqueté 152
6.8 Permutation de réductions complètes 156
6.9 Réduction de $M = ((\lambda x.((\lambda y.x)^D (x\ u)^A)^B)^C((\lambda z.z)^C(\lambda z.z)^C)^B)^A$ 159
6.10 Réduction de M par une séquence de réductions complètes 159
6.11 La réduction de la figure 6.10 est indépendante de l'interaction entre A et B. . . . 160

Bibliographie

[1] Martín Abadi, Anindya Banerjee, Nevin Heintze et Jon G. Riecke. A Core Calculus of Dependency. Dans *ACM Symposium on Principles of Programming Languages (POPL)*, pages 147–160, janvier 1999.

[2] Martín Abadi et Cédric Fournet. Access Control based on Execution History. Dans *Network and Distributed System Symposium (NDSS'03)*, pages 107–121. Internet Society, février 2003.

[3] Martín Abadi, Butler Lampson et Jean-Jacques Lévy. Analysis and Caching of Dependencies. Dans *ACM International Conference on Functional Programming (ICFP)*, pages 83–91, mai 1996.

[4] Ana Almeida Matos et Gérard Boudol. On Declassification and the Non-Disclosure Policy. Dans *Proceedings of the Computer Security Foundations Workshop (CSFW'05)*, juin 2005.

[5] Andrea Asperti, Paolo Coppola et Simone Martini. (Optimal) duplication is not elementary recursive. *Information and Computation*, 193/1:21–56, 2004.

[6] Andrea Asperti et Stefano Guerrini. *The Optimal Implementation of Functional Programming Languages*. Cambridge University Press, 1999.

[7] Henk P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics*. North-Holland, 1981.

[8] Frédéric Besson, Thomas de Grenier de Latour et Thomas Jensen. Secure Calling Contexts for Stack Inspection. Dans *4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, pages 76–87, 2002.

[9] Don Box. *Essential .NET Volume I: The Common Language Runtime*. Addison Wesley, 2002.

[10] David F. C. Brewer et Michael J. Nash. The Chinese Wall Security Policy. Dans *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pages 206–214, 1989.

[11] Naim Çağman et J. Roger Hindley. Combinatory Weak Weduction in Lambda-Calculus. *Theoretical Computer Science*, 198:239–249, 1998.

[12] Stephen Chong et Andrew C. Myers. Security policies for Downgrading. Dans *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)*, pages 189–209, octobre 2004.

[13] Dorothy E. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

[14] Dorothy E. Denning et Peter J. Denning. Certification of Programs for Secure Information Flow. *Communications of the ACM*, 20(7):504–513, juillet 1977.

[15] Cédric Fournet et Andrew D. Gordon. Stack Inspection: Theory and Variants. Rapport Technique MSR–TR–2001–103, Microsoft Research, 2001.

[16] Joseph Goguen et José Meseguer. Security Policies and Security Models. Dans *Proceedings of the 3rd Symposium on Security ans Privacy*, pages 11–20. IEEE Computer Society Press, avril 1982.

[17] Li Gong. *Inside Java™ 2 Platform Security*. Addison Wesley, 1999.

[18] Georges Gonthier, Martín Abadi et Jean-Jacques Lévy. The Geometry of Optimal Lambda Reduction. Dans *Proc. of the 19th Conference on Principles of Programming Languages*, volume 8. ACM Press, 1992.

[19] Georges Gonthier, Jean-Jacques Lévy et Paul-André Melliès. An Abstract Standardisation Theorem. Dans *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 72–81, juin 1992.

[20] Paul Graham. *On Lisp: Advanced Techniques for common Lisp.* Prentice Hall, 1993.

[21] Norm Hardy. The Confused Deputy. *ACM Operating Systems Review*, 22(4):36–38, octobre 1988.

[22] Nevin Heintze et Jon G. Riecke. The SLam Calculus: Programming with Secrecy and Integrity. Dans *ACM Symposium on Principles of Programming Languages (POPL)*, pages 365–377, janvier 1998.

[23] Thomas Jensen, Daniel Le Métayer et Tommy Thorn. Verification of control flow based security properties. Dans *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 89–103. IEEE Computer Society Press, 1999.

[24] Jan Willem Klop. *Combinatory Reduction Systems.* PhD thesis, Rijksuniversiteit Utrecht, 1980.

[25] John Lamping. An Algorithm for Optimal Lambda-Calculus Reduction. Dans *Proceedings of the 17th Annual ACM Symposium on Principles of Programming Languages*, pages 16–30, 1990.

[26] Sebastian Lange, Brian LaMacchia, Matthew Lyons, Rudi Martin, Brian Pratt et Greg Singleton. *.NET Framework Security.* Addison Wesley, 2002.

[27] Julia L. Lawall et Harry G. Mairson. On the Global Dynamics of Optimal Graph Reduction. Dans *ACM International Conference on Functional Programming*, 1997.

[28] Xavier Leroy, Damien Doligez, Jacques Garrigue, Didier Rémy et Jérôme Vouillon. The Objective Caml manual. http://caml.inria.fr/.

[29] Jean-Jacques Lévy. *Réductions correctes et optimales dans le lambda-calcul.* Thèse de doctorat, Université Paris 7, 1978.

[30] Jean-Jacques Lévy et Luc Maranget. Explicit substitutions and programming languages. Dans *Proc. 19th Conference on the Foundations of Software Technology and Theoretical Computer Science Electronic Notes in Theoretical Computer Science*, pages 181–200, 1999.

[31] Tim Lindholm et Frank Yellin. *The Java Virtual Machine Specification.* Addison Wesley, 1996.

[32] Andrew C. Myers, Nathaniel Nystrom, Steve Zdancewic et Lantian Zheng. Jif: Java + information flow, 2001. http://www.cs.cornell.edu/jif/.

[33] M. H. A. Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics*, 43(2):223–243, 1942.

[34] Simon Peyton Jones, éditeur. *Haskell 98 Language and Libraries : The Revised Report.* Cambridge University Press, avril 2003.

[35] Fran cois Pottier, Christian Skalka et Scott Smith. A Systematic Approach to Access Control. Dans *Programming Languages and Systems (ESOP 2001)*, volume 2028 de *LNCS*, pages 30–45. Springer-Verlag, 2001.

[36] François Pottier et Sylvain Conchon. Information Flow Inference for Free. Dans *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP'00)*, pages 46–57, Montréal, Canada, septembre 2000.

[37] François Pottier et Vincent Simonet. Information Flow Inference for ML. *ACM Transactions on Programming Languages and Systems*, 25(1):117–158, janvier 2003.

[38] Olin Shivers et Mitchell Wand. Bottom-up beta-substitution: uplinks and lambda-DAGs. Rapport Technique, BRICS RS-04-38, DAIMI, Department of Computer Science, University of Århus, Århus, Denmark, 2004.

[39] Vincent Simonet. *Inférence de flots d'information pour ML: formalisation et implantation.* Thèse de doctorat, Université Paris 7 - Denis Diderot, 2004.

[40] Standard ML of New Jersey. http://www.smlnj.org/.

[41] Dennis Volpano et Geoffrey Smith. A Type-Based Approach to Program Security. *Lecture Notes in Computer Science*, 1214:607–621, avril 1997.

[42] Dennis Volpano, Geoffrey Smith et Cynthia Irvine. A Sound Type System for Secure Flow Analysis. *Journal of Computer Security*, 4(3):167–187, 1996.

[43] Christopher P. Wadsworth. *Semantics and pragmatics of the lambda-calculus.* PhD thesis, Oxford University, 1971.

[44] Dan S. Wallach, Andrew W. Appel et Edward W. Felten. Safkasi: A Security Mechanism for Language-based Systems. *ACM Transactions on Software Engineering and Methodology*, 9(4):341–378, 2000.